

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Využití umělé inteligence v oblasti rozšířené reality

Disertační práce

Autor: Ing. Milan Košťák
Studijní program: P1802 Aplikovaná informatika
Studijní obor: 1802V001 Aplikovaná informatika

Školitel: prof. RNDr. Antonín Slabý, CSc.
Katedra školitele: Katedra informatiky a kvantitativních metod

PROHLÁŠENÍ

Prohlašuji, že jsem disertační práci zpracoval samostatně a s použitím uvedené literatury a pramenů.

V Hradci Králové dne 14. dubna 2023

Milan Košťák

PODĚKOVÁNÍ

Rád bych poděkoval mému školiteli prof. RNDr. PhDr. Antonínu Slabému, CSc. za pomoc, cenné rady, podněty a psychickou podporu nejen při zpracování mé disertační práce ale také během celého doktorského studia. Dále bych rád poděkoval Ing. Bruno Ježkovi, Ph.D., který mi během celého mého studia pomáhal nesčítným množstvím odborných rad a konzultací, bez kterých by tato práce vznikala mnohem obtížněji.

ANOTACE

Práce se zabývá možnostmi využití strojového učení v oblasti rozšířené reality. Soustředuje se zejména na aplikaci umělých konvolučních neuronových sítí, které nacházejí bohaté využití v počítačovém vidění.

Text disertační práce se ve své první polovině zabývá návrhem designu markeru, který je specificky vhodný pro použití v aplikacích rozšířené reality. V takových aplikacích se typicky do reálné scény doplňují virtuální objekty. Markery představují jeden ze způsobů, jak získat potřebné informace pro korektní a přesné doplnění virtuálních objektů. Při návrhu markeru je kladen důraz na jeho jednoduchost a snadné splnutí s okolím. Dále z něj musí být možné získat informace o poloze, orientaci a velikosti a autorem navržený marker splňuje všechna zmíněná kritéria.

Práce se dále ve své druhé polovině věnuje návrhu konkrétních algoritmů a postupů, které navržený marker umí detekovat. Zde nachází využití jak umělé neuronové sítě, tak klasické algoritmy, a oba přístupy jsou podrobně zkoumány a testovány. Na základě provedeného testování jsou doporučena konkrétní použití pro konkrétní situace. Ve svém závěru se práce zabývá způsobem získání všech jmenovaných informací z detekovaného markeru a navržený způsob je pro obě detekční metody kvalitativně porovnán.

Klíčová slova:

umělé neuronové sítě, rozšířená realita, počítačové vidění, vizuální markery

ANNOTATION

Use of artificial intelligence in augmented reality

The thesis explores the possibilities of using machine learning in the field of augmented reality. It focuses mainly on the application of artificial convolutional neural networks, which find rich use in computer vision.

The text of the dissertation in its first half investigates the design of a marker that is specifically suitable for use in augmented reality applications. In these applications, virtual objects are typically added to the real scene. Markers represent one of the ways to obtain the necessary information for the correct and accurate insertion of the virtual objects. When designing the marker, emphasis is placed on its simplicity and easy blending with the environment. Furthermore, it must be possible to acquire information about the position, orientation, and size from the marker, and the author's proposed design of the marker meets all the mentioned criteria.

The work in its second half devotes to the exploration of specific algorithms and techniques that can be used for detection of the designed marker. Both artificial neural networks and classical algorithms are thoroughly inspected and tested. Based on the conducted testing, recommendations are given for specific uses in specific situations. At the end, the work deals with obtaining all the mentioned information from the detected marker, and the proposed method is compared for both detection methods.

Keywords:

artificial neural networks, augmented reality, computer vision, fiducial markers

OBSAH

1	ÚVOD.....	1
2	ČÍLE PRÁCE.....	3
2.1	NÁVRH MARKERU.....	3
2.2	NÁVRH ALGORITMU DETEKCE MARKERU.....	4
3	METODIKA ZPRACOVÁNÍ.....	6
3.1	EVALUAČNÍ METRIKY TESTOVÁNÍ.....	6
3.2	MĚŘENÍ ČASU ZPRACOVÁNÍ ALGORITMŮ NA GPU.....	8
3.3	DATOVÉ SADY.....	10
3.3.1	DATOVÉ SADY VYUŽITÉ PŘI NÁVRHU.....	10
3.3.2	FINÁLNÍ DATOVÁ SADA PRO TESTOVÁNÍ.....	11
3.3.3	TVORBA DATOVÝCH SAD.....	12
3.4	PARAMETRY NEURONOVÉ SÍTĚ.....	13
4	REŠERŠE LITERATURY.....	14
4.1	OBDOBÍ 1996-2008.....	15
4.1.1	SHRnutí OBDOBÍ 1996-2008.....	21
4.2	OBDOBÍ 2009-2015.....	23
4.2.1	SHRnutí OBDOBÍ 2009-2015.....	28
4.3	OBDOBÍ 2016-2020.....	28
4.3.1	SHRnutí OBDOBÍ 2016-2020.....	33
4.4	CELKOVÉ SHRnutí.....	34
5	PROCES NÁVRHU DESIGNU MARKERU.....	35
5.1	PRVNÍ ITERACE TESTŮ.....	35
5.2	DRUHÁ ITERACE TESTŮ – DOPLNĚNY DVA NOVÉ TVARY.....	37
5.3	TŘETÍ ITERACE TESTŮ – DOPLNĚNY DVA NOVÉ TVARY.....	38

5.4	ČTVRTÁ ITERACE TESTŮ – OBRÁZKY INTERIÉRU	39
5.5	PÁTÁ ITERACE TESTŮ – VÍCE RŮZNÝCH BAREV	41
5.6	ŠESTÁ ITERACE TESTŮ – TVAR „T-KŘÍŽEK“	42
5.7	SEDMÁ ITERACE TESTŮ – VĚTŠÍ OBRÁZKY	43
5.8	OSMÁ ITERACE TESTŮ – VĚTŠÍ DATOVÁ SADA	44
5.9	DEVÁTÁ ITERACE TESTŮ – NEJVĚTŠÍ DATOVÁ SADA	45
5.10	DESÁTÁ ITERACE TESTŮ – MARKERY V REÁLNÝCH SCÉNÁCH	46
6	SHRNUTÍ VÝSLEDKŮ NÁVRHU DESIGNU MARKERU	48
6.1	VÝHODY A NEVÝHODY	49
6.2	PŘÍKLADY DETEKČÍ	50
7	ZPŮSOBY DETEKCE NAVRŽENÉHO MARKERU	52
7.1	VYUŽITÍ UMĚLÝCH NEURONOVÝCH SÍTÍ	52
7.1.1	PARAMETRY SÍTĚ, RYCHLOST A PAMĚŤOVÁ NÁROČNOST	53
7.1.2	PŘÍKLADY DETEKČÍ	53
7.2	ALGORITMUS ROZDĚLENÍ OBRAZU NA ČÁSTI	55
7.2.1	PRINCIP ALGORITMU	55
7.2.2	ČASOVÁ KOMPLEXITA	57
7.2.3	RYCHLOST DETEKCE	57
7.3	ALGORITMUS SKENOVÁNÍ ŘÁDKŮ A SLOUPCŮ	59
7.3.1	PRINCIP ALGORITMU	59
7.3.2	SPECIFICKÁ VYLEPŠENÍ ALGORITMU	60
7.3.3	VÝHODY OPROTI PŘEDCHOZÍMU ALGORITMU	65
7.3.4	ČASOVÁ KOMPLEXITA	66
7.3.5	RYCHLOST DETEKCE	67
7.3.6	PŘÍKLADY DETEKČÍ	68
7.4	VYHODNOCENÍ ALGORITMŮ NA FINÁLNÍ DATOVÉ SADĚ	69

7.4.1	UMĚLÁ NEURONOVÁ SÍŤ S ARCHITEKTUROU YOLOv3	69
7.4.2	ALGORITMUS ROZDĚLENÍ OBRAZU NA ČÁSTI	69
7.4.3	ALGORITMUS SKENOVÁNÍ ŘÁDKŮ A SLOUPCŮ.....	71
7.5	SHRnutí A DOPORUČENÍ	77
7.6	ZDROJOVÉ KÓDY	79
8	ZÍSKÁNÍ INFORMACÍ Z DETEKOVANÉHO MARKERU.....	80
8.1	ZPRACOVÁNÍ OBRAZU PRO ZÍSKÁNÍ INFORMACÍ O LINÍCH MARKERU	80
8.2	ZPRACOVÁNÍ LINÍ MARKERU	86
8.3	VÝSLEDKY ZPRACOVÁNÍ	88
8.4	RYCHLOST ZPRACOVÁNÍ	90
8.5	VYBRANÉ PŘÍKLADY DETEKČÍ	91
8.6	ZDROJOVÝ KÓD	92
9	SHRnutí VÝSLEDKŮ A JEJICH ZHODNOCENÍ.....	93
9.1	NÁVRH MARKERU.....	93
9.2	DETEKCE A PŘEČTENÍ MARKERU	94
9.2.1	NÁVRH DETEKČNÍHO SYSTÉMU	94
9.2.2	ŘEŠENÍ PRO ZÍSKÁNÍ INFORMACÍ Z MARKERU	95
9.2.3	SHRnutí PROCESU DETEKCE Z HLEDISKA CÍLŮ PRÁCE	95
10	ZÁVĚR.....	97
10.1	DALŠÍ MOŽNÉ SMĚRY VÝZKUMU	98
11	SEZNAM POUŽITÉ LITERATURY	99
12	VLASTNÍ PUBLIKACE	108
	SHRnutí PUBLIKAČNÍCH VÝSTUPŮ.....	109
13	PŘEHLED ODBORNÝCH VĚDECKO-VÝZKUMNÝCH AKTIVIT	110
	ÚČAST NA PROJEKTECH SPECIFICKÉHO VÝZKUMU	110
	ZAHRANIČNÍ STÁŽE A MEZINÁRODNÍ SPOLUPRÁCE	110

ÚČAST V PROJEKTU SUPERDOKTORAND FIM UHK.....	110
--	-----

SEZNAM OBRÁZKŮ

OBRÁZEK 1 GRAFICKÁ REPREZENTACE METRIKY IOU [AUTOR]	8
OBRÁZEK 2 PŘÍKLADY 20 MARKERŮ Z OBDOBÍ 1992-2007 S JEJICH JMÉNEM (POKUD JE K DISPOZICI) A ROKEM JEJICH PUBLIKACE	22
OBRÁZEK 3 PŘÍKLADY 20 MARKERŮ Z OBDOBÍ 2007-2015 S JEJICH JMÉNEM (POKUD JE K DISPOZICI) A ROKEM JEJICH PUBLIKACE	29
OBRÁZEK 4 PŘÍKLADY 11 MARKERŮ Z OBDOBÍ 2016-2020 S JEJICH JMÉNEM (POKUD JE K DISPOZICI) A ROKEM JEJICH PUBLIKACE	33
OBRÁZEK 5 VYPLNĚNÝ OBDÉLNÍK [AUTOR].....	36
OBRÁZEK 6 OBRYS OBDÉLNÍKU [AUTOR]	36
OBRÁZEK 7 VYPLNĚNÝ TROJÚHELNÍK [AUTOR].....	36
OBRÁZEK 8 OBRYS TROJÚHELNÍKU [AUTOR]	36
OBRÁZEK 9 HVĚZDA TLOUŠŤKY 2 [AUTOR].....	36
OBRÁZEK 10 HVĚZDA TLOUŠŤKY 1 [AUTOR]	36
OBRÁZEK 11 HVĚZDA TLOUŠŤKY 1 V OBRYSU OBDÉLNÍKU [AUTOR]	36
OBRÁZEK 12 SYMBOL ZAVINÁČE (@) [AUTOR].....	36
OBRÁZEK 13 HVĚZDA TLOUŠŤKY 2 VE VYPLNĚNÉM OBDÉLNÍKU [AUTOR].....	40
OBRÁZEK 14 HVĚZDA TLOUŠŤKY 1 VE VYPLNĚNÉM OBDÉLNÍKU [AUTOR].....	40
OBRÁZEK 15 KŘÍŽEK TLOUŠŤKY 2 V OBDÉLNÍKU [AUTOR]	40
OBRÁZEK 16 KŘÍŽEK TLOUŠŤKY 1 V OBDÉLNÍKU [AUTOR]	40
OBRÁZEK 17 „T-KŘÍŽEK TLOUŠŤKY 2“ [AUTOR]	43
OBRÁZEK 18 „T-KŘÍŽEK TLOUŠŤKY 1“ [AUTOR]	43

OBRÁZEK 19 ILUSTRACE PROCESU TAGOVÁNÍ: (A) TŘI ČERVENÉ TEČKY JSOU UMÍSTĚNY PRO OZNAČENÍ HORNÍHO, LEVÉHO A PRAVÉHO ROHU; (B) PO UMÍSTĚNÍ DOLNÍHO BODU JSOU BODY NAHAZENY OHRANIČUJÍCÍM BOXEM. [AUTOR]	47
OBRÁZEK 20 ROC KŘIVKA PRO RŮZNÉ PRAVDĚPODOBNOSTNÍ PRAHY DETEKCE V ROZSAHU OD 0,05 DO 0,7 [AUTOR]	48
OBRÁZEK 21 PŘÍKLADY DETEKOVANÝCH MARKERŮ, KTERÉ BYLY RUČNĚ NAKRESLENY A UMÍSTĚNY V REÁLNÝCH SCÉNÁCH – HORNÍ DVA OBRÁZKY JSOU V INTERIÉRU, DOLNÍ DVA V EXTERIÉRU. [AUTOR]	50
OBRÁZEK 22 PŘÍKLAD DETEKCE MARKERU V OBRÁZKU SE STŘEDNÍM ROZMAZÁNÍM ZPŮSOBENÝM PRUDKÝM POHYBEM KAMERY. OBRÁZEK DOBŘE ILUSTRUJE VÝHODU TVARU „T-KŘÍŽEK“ – JEDNA ČÁRA JE STÁLE TĚMĚŘ OSTRÁ. [AUTOR]	50
OBRÁZEK 23 PŘÍKLADY VŠECH ČTYŘ FALEŠNĚ NEGATIVNÍCH PŘÍPADŮ PŘI POUŽITÍ HODNOTY 0,3 JAKO PRAVDĚPODOBNOSTNÍHO PRAHU DETEKCE. PŘI POUŽITÍ PRAHU O HODNOTĚ 0,2 JSOU MARKERY V HORNÍCH DVOU OBRÁZČÍCH KOREKTNĚ DETEKOVÁNY. [AUTOR]	51
OBRÁZEK 24 PŘÍKLAD JEDINÉ FALEŠNĚ POZITIVNÍ DETEKCE PŘI POUŽITÍ HODNOTY 0,3 JAKO PRAVDĚPODOBNOSTNÍHO PRAHU DETEKCE. ČÁST KOBERCE V PRAVÉ HORNÍ ČÁSTI OBRÁZKU JE MYLNĚ DETEKOVÁNA JAKO MARKER. [AUTOR]	51
OBRÁZEK 25 PŘÍKLADY DETEKČÍ NEURONOVOU SÍTÍ PRO RUČNĚ KRESLENÉ MARKERY SNÍMANÉ V INTERIÉRU [AUTOR]	53
OBRÁZEK 26 PŘÍKLADY DETEKČÍ NEURONOVOU SÍTÍ PRO RUČNĚ KRESLENÉ MARKERY SNÍMANÉ V INTERIÉRU ZA UMĚLÉHO OSVĚTLENÍ [AUTOR]	54
OBRÁZEK 27 PŘÍKLADY DETEKČÍ NEURONOVOU SÍTÍ PRO TIŠTĚNÉ MARKERY SNÍMANÉ V INTERIÉRU [AUTOR]	54
OBRÁZEK 28 PŘÍKLADY DETEKČÍ NEURONOVOU SÍTÍ PRO TIŠTĚNÉ MARKERY SNÍMANÉ V EXTERIÉRU [AUTOR]	54
OBRÁZEK 29 PŘÍKLADY DETEKČÍ NEURONOVOU SÍTÍ. VLEVO RUČNĚ KRESLENÝ MARKER SNÍMANÝ V INTERIÉRU, JEHOŽ BARVA JE TMAVĚ MODRÁ, AVŠAK POD PŘÍMÝM SLUNEČNÍM SVĚTLEM JE TĚMĚŘ ČERNÁ. VPRAVO TIŠTĚNÝ MARKER, KTERÝ JE NA MAGNETICKOU TABULI PŘICHYCEN MAGNETEM. [AUTOR]	54

OBRÁZEK 30 ILUSTRACE PRVNÍHO PRŮCHODU GRAFICKOU PIPELINE PŘI DETEKCI ZELENÉHO MARKERU V RÁMCI ALGORITMU ROZDĚLENÍ OBRAZU NA ČÁSTI. PRO LEPŠÍ ČITELNOST JSOU NA VÝSTUPU VYPLNĚNY POUZE HODNOTY PRVNÍHO PIXELU VE SKUPINĚ, AČKOLIV REÁLNĚ JSOU TYTO HODNOTY VYPLNĚNY DO VŠECH PIXELŮ SKUPINY. [AUTOR].....	55
OBRÁZEK 31 ILUSTRACE DRUHÉHO PRŮCHODU GRAFICKOU PIPELINE V RÁMCI ALGORITMU ROZDĚLENÍ OBRAZU NA ČÁSTI. LEVÁ ČÁST REPREZENTUJE VSTUP PRO ILUSTROVANOU SKUPINU PIXELŮ. PRAVÁ ČÁST REPREZENTUJE VYPOČÍTANÝ VÝSTUP. [AUTOR]	56
OBRÁZEK 32 ILUSTRACE ALGORITMU SKENOVÁNÍ ŘÁDKŮ A SLOUPCŮ [AUTOR]	59
OBRÁZEK 33 PŘÍKLADY DETEKČÍ VYUŽITÍM ALGORITMU SKENOVÁNÍ ŘÁDKŮ A SLOUPCŮ. VŠECHNY ČTYŘI OBRÁZKY PREZENTUJÍ DETEKCE MARKERŮ RŮZNÝCH BAREV (RŮŽOVÁ, ČERVENÁ, MODRÁ) V REÁLNÉM PROSTŘEDÍ, KDE MARKERY OSTATNÍCH BAREV JSOU VŽDY SPRÁVNĚ IGNOROVÁNY. PRAVÝ OBRÁZEK UKAZUJE, ŽE DETEKCE NESTANDARDNÍCH MARKERŮ (ZDE KRUH) JSOU TAKÉ PODPOROVÁNY. [AUTOR]	68
OBRÁZEK 34 PŘÍKLADY DETEKČÍ VYUŽITÍM ALGORITMU SKENOVÁNÍ ŘÁDKŮ A SLOUPCŮ. VLEVO DETEKOVÁNA IKONA POSTAVY. VPRAVO DETEKOVÁN BOD MEZI DVĚMA HVĚZDAMI. V OBOU PŘÍPADECH SE JEDNÁ O TĚŽIŠTĚ JAKO BOD, KDE JE VÁHA DETEKOVANÉ BARVY NEJVYŠŠÍ. [AUTOR]	68
OBRÁZEK 35 PŘÍKLADY DETEKČÍ „T-KŘÍŽEK“ MARKERU S VYUŽITÍM ALGORITMU SKENOVÁNÍ ŘÁDKŮ A SLOUPCŮ. OSTATNÍ MARKERY JSOU VŽDY SPRÁVNĚ IGNOROVÁNY. [AUTOR].....	69
OBRÁZEK 36 A) VLEVO VYŘÍZNUTÝ MARKER Z PŮVODNÍHO SNÍMKU S ID 1584 B) VPRAVO STEJNÝ VÝŘEZ PO APLIKACI GAUSSOVA FILTRU [AUTOR]	70
OBRÁZEK 37 A) VLEVO MASKA VYTVOŘENÁ PODLE HRANICE MARKERU VE VYŘÍZNUTÉM OBRÁZKU B) VPRAVO VYŘÍZNUTÝ MARKER PODLE VYTVOŘENÉ MASKY, TENTO VÝŘEZ PODLE DANÉ MASKY JE POTÉ VYUŽIT PRO VYTVOŘENÍ HISTOGRAMU ODSTÍNŮ BARVY MARKERU [AUTOR]	71
OBRÁZEK 38 HISTOGRAM ODSTÍNŮ VYŘÍZNUTÉHO MARKERU, VÝSLEDNÝ ODSTÍN (HUE V HSV BAREVNÉM MODELU) JE 112 [AUTOR]	72
OBRÁZEK 39 PŘÍKLAD MARKERU S PROBLEMATICKÝM HISTOGRAMEM A) VLEVO VYŘÍZNUTÝ MARKER Z PŮVODNÍHO SNÍMKU S ID 71 B) VPRAVO STEJNÝ VÝŘEZ PO APLIKACI MASKY [AUTOR].....	74

OBRÁZEK 40 UKÁZKA HISTOGRAMU ČERNÉHO MARKERU S ID 71. HISTOGRAM ILUSTRUJE NEJISTOTU PŘI URČENÍ DOMINANTNÍHO ODSTÍNU A PROBLÉM S ČERNOU BARVOU (A OBECNĚ TMAVÝMI ODSTÍNY) PŘI ZJIŠŤOVÁNÍ DOMINANTNÍHO ODSTÍNU V RÁMCI HSV BAREVNÉHO MODELU. [AUTOR].....	74
OBRÁZEK 41 PŘÍKLAD MARKERU SVĚTLÉHO ODSTÍNU S PROBLEMATICKÝM HISTOGRAMEM A) VLEVO VYŘÍZNUTÝ MARKER Z PŮVODNÍHO SNÍMKU S ID 2345 B) VPRAVO STEJNÝ VÝŘEZ PO APLIKACI MASKY [AUTOR]	75
OBRÁZEK 42 UKÁZKA HISTOGRAMU SVĚTLÉHO MARKERU S ID 2345. HISTOGRAM ILUSTRUJE NEJISTOTU PŘI URČENÍ DOMINANTNÍHO ODSTÍNU A PROBLÉM SE SVĚTLÝMI ODSTÍNY PŘI ZJIŠŤOVÁNÍ DOMINANTNÍHO ODSTÍNU V RÁMCI HSV BAREVNÉHO MODELU. [AUTOR]	75
OBRÁZEK 43 SNÍMKY ILUSTRUJÍ DŮVOD, PROČ JE VELIKOST HRANY ČTVERCE NADSAZENA PŘI ODHADU JEJÍHO VÝPOČTU – ABY SE MARKER DO ODHADNUTÉHO BOUNDING BOXU STÁLE VEŠEL I PŘI DIAGONÁLNÍM SNÍMÁNÍ (PRAVÉ SNÍMKY). [AUTOR].....	77
OBRÁZEK 44 HISTOGRAM ODSTÍNŮ VYŘÍZNUTÉHO MARKERU, VÝSLEDNÝ ODSTÍN (HUE V HSV BAREVNÉM MODELU) MÁ HODNOTU 60 [AUTOR].....	82
OBRÁZEK 45 UKÁZKA VYŘÍZNUTÉHO MARKERU Z ORIGINÁLNÍHO SNÍMKU S ID 1518 [AUTOR].	84
OBRÁZEK 46 VLEVO UKÁZKA KOSOČTVEREČNÉ MASKY, KTERÁ JE POUŽITA PRO ODFILTROVÁNÍ PIXELŮ, KTERÉ S MARKEREM NESOUVISÍ. VPRAVO PAK APLIKACE MASKY NA VYŘÍZNUTÝ MARKER. [AUTOR].....	84
OBRÁZEK 47 VÝSTUP ČTVRTÉHO KROKU ZPRACOVÁNÍ, KDY Z PŮVODNÍHO VÝŘEZU JSOU ZACHOVÁNY POUZE PIXELY, KTERÉ MAJÍ ODSTÍN PODOBNÝ ODSTÍNU DETEKOVANÉHO MARKERU (TEDY DOMINANTNÍHO ODSTÍNU V DANÉM VÝŘEZU) [AUTOR]	84
OBRÁZEK 48 VLEVO UKÁZKA VÝSTUPU PO APLIKACI OPERACE EROZE. VPRAVO PAK PO APLIKACI OPERACE DILATACE. PŘI POROVNÁNÍ S PŘEDCHOZÍM OBRÁZKEM (OBRÁZEK 47) DOŠLO K ZACELENÍ DĚR TVOŘENÝCH ŠUMEM. [AUTOR].....	85
OBRÁZEK 49 VLEVO UKÁZKA VÝSTUPU PO APLIKACI GAUSSOVA FILTRU PRO ROZMAZÁNÍ OBRAZU. VPRAVO VÝSTUP PO APLIKACI OTSUOVA ADAPTIVNÍHO PRAHOVÁNÍ. [AUTOR]	85
OBRÁZEK 50 VLEVO UKÁZKA VÝSTUPU PO APLIKACI CANNYHO DETEKTORU HRAN. VPRAVO NAVÍC VYKRESLENY VÝSTUPNÍ LINIE PO JEJICH IDENTIFIKACI HOUGHOVOU TRANSFORMACÍ. [AUTOR]....	85

OBRÁZEK 51 VŠECHNY FINÁLNÍ ZJIŠTĚNÉ INFORMACE O MARKERU ZAKRESLENÉ DO PŮVODNÍHO VÝŘEZU. ČERVENÝ KRUH OZNAČUJE ZJIŠTĚNÝ STŘED MARKERU. MODRÁ ŠIPKA ZNAČÍ SMĚR ORIENTACE MARKERU, KDY PŘÍMO ÚHEL ORIENTACE JE ZAPSÁN TAKÉ V LEVÉM HORNÍM ROHU (158 STUPŇŮ). ZELENÝ OBDÉLNÍK OZNAČUJE CELÝ ZJIŠTĚNÝ MARKER. [AUTOR].....	86
OBRÁZEK 52 PŘÍKLADY VIZUALIZACÍ ZJIŠTĚNÝCH INFORMACÍ Z DETEKOVANÝCH MARKERŮ. VLEVO MARKER S ID 0635, VPRAVO S ID 1197. [AUTOR]	91
OBRÁZEK 53 PŘÍKLADY VIZUALIZACÍ ZJIŠTĚNÝCH INFORMACÍ Z DETEKOVANÝCH MARKERŮ. VLEVO MARKER S ID 2804, VPRAVO S ID 2879. [AUTOR]	91
OBRÁZEK 54 PŘÍKLADY VIZUALIZACÍ ZJIŠTĚNÝCH INFORMACÍ Z DETEKOVANÝCH MARKERŮ. VLEVO MARKER S ID 3197, VPRAVO S ID 3460. [AUTOR]	91
OBRÁZEK 55 PŘÍKLAD „T-KŘÍŽEK“ MARKERU	93

SEZNAM TABULEK

TABULKA 1 TABULKA POPISUJÍCÍ VZTAHY MEZI JEDNOTLIVÝMI PRAVDĚPODOBNOSTNÍMI HODNOTAMI POUŽITÝMI JAKO METRIKY [AUTOR]	7
TABULKA 2 TABULKA S ČETNOSTMI MARKERŮ FINÁLNÍ DATOVÉ SADY NAPŘÍČ DEFINOVANÝMI 72 KATEGORIEMI [AUTOR]	12
TABULKA 3 POROVNÁNÍ VÝSLEDKŮ OSMI TESTOVANÝCH TVARŮ MARKERU V PRVNÍ ITERACI TESTŮ PRO DEVĚT UČÍCÍCH EPOCH [AUTOR]	37
TABULKA 4 POROVNÁNÍ VÝSLEDKŮ DESETI TESTOVANÝCH TVARŮ MARKERU V DRUHÉ ITERACI TESTŮ. PRO KAŽDOU VARIANTU JE UVEDEN NEJLEPŠÍ VÝSLEDEK BEZ OHLEDU NA POČET EPOCH, KTERÝ JE BUĎ 3, 6 NEBO 9. MODŘE JSOU OZNAČENY NOVÉ VARIANTY. ČERVENĚ JSOU OZNAČENY VARIANTY, KTERÉ JSOU PRO DALŠÍ ITERACE TESTŮ VYŘAZENY. [AUTOR]	38
TABULKA 5 POROVNÁNÍ VÝSLEDKŮ SEDMI TESTOVANÝCH TVARŮ MARKERU V TŘETÍ ITERACI TESTŮ PRO 21 UČÍCÍCH EPOCH. MODŘE JSOU OZNAČENY NOVÉ VARIANTY. ČERVENĚ JSOU OZNAČENY VARIANTY, KTERÉ BYLY PRO DALŠÍ ITERACI VYLOUČENY. [AUTOR]	39
TABULKA 6 POROVNÁNÍ VÝSLEDKŮ ČTYŘ TESTOVANÝCH TVARŮ MARKERU VE ČTVRTÉ ITERACI TESTŮ S INTERIÉROVÝMI OBRÁZKY PRO 21 UČÍCÍCH EPOCH. Z TABULKY JE VIDĚT, ŽE VÝSLEDKY JSOU SROVNATELNÉ S UČENÍM S EXTERIÉROVÝMI OBRÁZKY (TABULKA 5). [AUTOR]	41
TABULKA 7 VÝSLEDKY TRÉNOVÁNÍ S VÍCE RŮZNÝMI BARVAMI MARKERU S EXTERIÉROVÝMI OBRÁZKY PRO 21 UČÍCÍCH EPOCH. Z TABULKY JE VIDĚT, ŽE VÝSLEDKY JSOU SROVNATELNÉ S UČENÍM S DVĚMA FIXNÍMI BARVAMI (TABULKA 5). [AUTOR]	42
TABULKA 8 VÝSLEDKY TRÉNOVÁNÍ S VÍCE RŮZNÝMI BARVAMI MARKERU S INTERIÉROVÝMI OBRÁZKY PRO 21 UČÍCÍCH EPOCH. Z TABULKY JE VIDĚT, ŽE VÝSLEDKY JSOU SROVNATELNÉ S UČENÍM S DVĚMA FIXNÍMI BARVAMI (TABULKA 6). [AUTOR]	42
TABULKA 9 VÝSLEDKY TRÉNOVÁNÍ PRO „T-KŘÍŽEK“ TVAR MARKERU S DVĚMA FIXNÍMI BARVAMI A S INTERIÉROVÝMI OBRÁZKY PRO 21 UČÍCÍCH EPOCH. [AUTOR]	43
TABULKA 10 VÝSLEDKY TRÉNOVÁNÍ PRO „T-KŘÍŽEK“ TVAR MARKERU S DVĚMA FIXNÍMI BARVAMI A S INTERIÉROVÝMI OBRÁZKY SE ŠÍRKOU 416 PIXELŮ. TRÉNOVÁNÍ PROBĚHLO PRO 26 EPOCH. [AUTOR]	44

TABULKA 11 VÝSLEDKY TRÉNOVÁNÍ PRO „T-KŘÍŽEK“ TVAR MARKERU S DVĚMA FIXNÍMI BARVAMI A S EXTERIÉROVÝMI OBRÁZKY SE ŠÍRKOU 416 PIXELŮ. TRÉNOVÁNÍ PROBĚHLO PRO 26 EPOCH. [AUTOR].....	44
TABULKA 12 VÝSLEDKY TRÉNOVÁNÍ PRO „T-KŘÍŽEK“ TVAR MARKERU S DVĚMA FIXNÍMI BARVAMI NA KOMBINOVANÉ DATOVÉ SADĚ 7 248 OBRÁZKŮ, JEJICHŽ ŠÍŘKA JE 416 PIXELŮ. [AUTOR]	45
TABULKA 13 VÝSLEDKY TRÉNOVÁNÍ PRO „T-KŘÍŽEK“ TVAR MARKERU S DVĚMA FIXNÍMI BARVAMI NA KOMBINOVANÉ DATOVÉ SADĚ 10 697 OBRÁZKŮ, JEJICHŽ ŠÍŘKA JE 416 PIXELŮ. [AUTOR].....	45
TABULKA 14 VÝSLEDKY TRÉNOVÁNÍ PRO „T-KŘÍŽEK“ TVAR MARKERU S DVĚMA FIXNÍMI BARVAMI NA KOMBINOVANÉ DATOVÉ SADĚ 10 697 OBRÁZKŮ, JEJICHŽ ŠÍŘKA JE 416 PIXELŮ. [AUTOR].....	49
TABULKA 15 NAMĚŘENÉ ČASY BĚHU RENDEROVACÍ SMYČKY JEDNOTLIVÝCH ČÁSTÍ DETEKCE MARKERU PŘI POUŽITÍ ALGORITMU ROZDĚLENÍ OBRAZU NA ČÁSTI A ROZLIŠENÍ OBRAZU 1280 × 720 PIXELŮ. [113].....	58
TABULKA 16 NAMĚŘENÉ CPU ČASY BĚHU JEDNOTLIVÝCH ČÁSTÍ DETEKCE MARKERU PŘI POUŽITÍ ALGORITMU ROZDĚLENÍ OBRAZU NA ČÁSTI A ROZLIŠENÍ OBRAZU 1280 × 720 PIXELŮ. [AUTOR]	58
TABULKA 17 TABULKA OBSAHUJE VÝSLEDKY TESTOVÁNÍ RŮZNÝCH PRAHOVÝCH HODNOT SATURACE A INTENZITY BAREVNÉHO MODELU HSV. VÝSLEDKY UKAZUJÍ, ŽE PRAH KOLEM 0,3 POSKYTUJE NEJLEPŠÍ DETEKCI. [114].....	61
TABULKA 18 TABULKA OBSAHUJE VÝSLEDKY TESTOVÁNÍ OPERACE DILATACE. OBĚ VERZE VÝSTUPŮ (PRŮMĚR, MAXIMUM) JSOU UVEDENY V TABULCE. Z TABULKY VYPLÝVÁ, ŽE DILATAČNÍ OPERACE NEZLEPŠUJE VÝSLEDKY ANI U JEDNÉ VARIANTY OPROTI SITUACI, KDY OPERACE NENÍ POUŽITA (TABULKA 17). [114].....	64
TABULKA 19 TABULKA OBSAHUJE VÝSLEDKY TESTOVÁNÍ OPERACE EROZE VE VARIANTĚ, KDY SE PRACUJE S POČTEM SOUSEDNÍCH PIXELŮ. Z TABULKY VYPLÝVÁ, ŽE OPERACE NEZLEPŠUJE VÝSLEDKY OPROTI SITUACI, KDY OPERACE NENÍ POUŽITA (TABULKA 17). [114].....	64
TABULKA 20 TABULKA OBSAHUJE VÝSLEDKY TESTOVÁNÍ OPERACE EROZE VE VARIANTĚ, KDY SE PRACUJE S VÁHOU SOUSEDNÍCH PIXELŮ. Z TABULKY VYPLÝVÁ, ŽE OPERACE ZNAČNĚ ZLEPŠUJE VÝSLEDKY OPROTI SITUACI, KDY OPERACE NENÍ POUŽITA (TABULKA 17). [114].....	65

TABULKA 21 TABULKA OBSAHUJE VÝSLEDKY TESTOVÁNÍ PŘEDCHOZÍHO ALGORITMU PRO RŮZNÉ PRAHOVÉ HODNOTY PRO SLOŽKY RGB BAREVNÉHO MODELU PŘI POUŽITÍ STEJNÉ METODOLOGIE JAKO PŘI TESTOVÁNÍ DÍLČÍCH VYLEPŠENÍ DRUHÉHO ALGORITMU (TABULKA 17). [114]	66
TABULKA 22 SHRNUTÍ TESTOVANÝCH VARIANT ZLEPŠENÍ ALGORITMU SKENOVÁNÍ ŘÁDKŮ A SLOUPCŮ A JEJICH VÝSLEDKY [114]	66
TABULKA 23 NAMĚŘENÉ ČASY BĚHU RENDEROVACÍ SMYČKY JEDNOTLIVÝCH ČÁSTÍ DETEKCE MARKERU PŘI POUŽITÍ ALGORITMU SKENOVÁNÍ ŘÁDKŮ A SLOUPCŮ. [113]	67
TABULKA 24 NAMĚŘENÉ CPU ČASY BĚHU JEDNOTLIVÝCH ČÁSTÍ DETEKCE MARKERU PŘI POUŽITÍ ALGORITMU SKENOVÁNÍ ŘÁDKŮ A SLOUPCŮ A ROZLIŠENÍ OBRAZU 1280 × 720 PIXELŮ. [AUTOR]	68
TABULKA 25 SOUČET POČTŮ FALEŠNĚ NEGATIVNÍCH (FN) A FALEŠNĚ POZITIVNÍCH (FP) DETEKČÍ. ČERVENĚ JSOU VYZNAČENY VÝSLEDNÉ HODNOTY PROBLÉMOVÝCH ODSTÍNŮ. [AUTOR] .	73
TABULKA 26 POROVNÁNÍ KVALITY DETEKCE POPSANÝCH ŘEŠENÍ [AUTOR]	78
TABULKA 27 POROVNÁNÍ RYCHLOSTI POPSANÝCH ŘEŠENÍ DETEKCE MARKERU V JEDNOM SNÍMKU O ROZMĚRECH 1280 × 720 PIXELŮ. OBRÁZEK PRO VSTUP DO NEURONOVÉ SÍTĚ ZMENŠEN NA ROZMĚRY 416 × 416 PIXELŮ. [AUTOR]	78
TABULKA 28 TABULKA PREZENTUJE VELIKOST TESTOVACÍCH SAD A POČTY NALEZENÝCH MARKERŮ, VŮČI KTERÝM JE VYHODNOCENÍ PROVEDENO [AUTOR]	88
TABULKA 29 POROVNÁNÍ KVALITY ZÍSKÁNÍ INFORMACÍ O STŘEDU Z DETEKOVANÉHO MARKERU PŘI TOLERANCI 60 PIXELŮ PRO FALEŠNĚ POZITIVNÍ DETEKCE [AUTOR]	89
TABULKA 30 POROVNÁNÍ KVALITY ZÍSKÁNÍ INFORMACÍ O ORIENTACI Z DETEKOVANÉHO MARKERU PŘI TOLERANCI 30 STUPŇŮ PRO FALEŠNĚ POZITIVNÍ DETEKCE [AUTOR]	89
TABULKA 31 POROVNÁNÍ KVALITY ZÍSKÁNÍ INFORMACÍ O OBSAHU Z DETEKOVANÉHO MARKERU PŘI TOLERANCI 10 000 PX ² PRO FALEŠNĚ POZITIVNÍ DETEKCE [AUTOR]	90
TABULKA 32 POROVNÁNÍ METRIKY IOU DETEKOVANÉHO A SPRÁVNÉHO MARKERU PŘI TOLERANCI 0,3 PRO FALEŠNĚ POZITIVNÍ DETEKCE [AUTOR]	90

SEZNAM UKÁZEK KÓDU

UKÁZKA KÓDU 1 REFERENČNÍ IMPLEMENTACE VÁŽENÍ ODSTÍNU BARVY V GLSL (OPENGL SHADING LANGUAGE) FRAGMENT SHADERU [AUTOR]	61
UKÁZKA KÓDU 2 REFERENČNÍ IMPLEMENTACE OPERACÍ DILATACE A EROZE PŘI ZPRACOVÁNÍ JEDNOHO PIXELU V DANÉM ŘÁDKU NEBO SLOUPCI (GLSL FRAGMENT SHADER). REFERENČNÍ IMPLEMENTACE VYUŽÍVÁ PRÁH 6,5 PRO ZAMÍTNUTÍ NEBO POTVRZENÍ VÁHY PIXELU V PŘÍPADĚ EROZE. [AUTOR].....	63
UKÁZKA KÓDU 3 REFERENČNÍ IMPLEMENTACE VYŘÍZNUTÍ MARKERU Z PŮVODNÍHO SNÍMKU PODLE DETEKOVANÉHO BOUNDING BOXU A URČENÉHO PADDINGU. UKÁZKA VYUŽÍVÁ KNIHOVNU OPENCV, KTERÁ JE IMPORTOVÁNA POD STANDARDNÍM ALIASEM cv2. [AUTOR]	81
UKÁZKA KÓDU 4 REFERENČNÍ IMPLEMENTACE PŘEVODU OBRÁZKU Z RGB DO HSV BAREVNÉHO MODELU S VYUŽITÍM KNIHOVNY OPENCV [AUTOR]	81
UKÁZKA KÓDU 5 REFERENČNÍ IMPLEMENTACE VYTVOŘENÍ MASKY A HISTOGRAMU ODSTÍNŮ S NÁSLEDNÝM PŘEČTENÍM NEJČASTĚJŠÍHO ODSTÍNU S VYUŽITÍM KNIHOVEN OPENCV A NUMPY (IMPORTOVÁNA POD STANDARDNÍM ALIASEM np). [AUTOR]	82
UKÁZKA KÓDU 6 REFERENČNÍ IMPLEMENTACE ODSTRANĚNÍ PIXELŮ, KTERÉ NEMAJÍ ODSTÍN PODOBNÝ ODSTÍNU DETEKOVANÉHO MARKERU (TEDY DOMINANTNÍHO ODSTÍNU V DANÉM VÝŘEZU) S VYUŽITÍM KNIHOVEN OPENCV A NUMPY [AUTOR]	83
UKÁZKA KÓDU 7 UKÁZKA APLIKACE OPERACÍ ROZOSTŘENÍ OBRAZU GAUSOVÝM FILTREM, OTSUOVA ADAPTIVNÍHO PRAHOVÁNÍ A CANNYHO DETEKTORU HRAN A POUŽITÝCH PARAMETRŮ S VYUŽITÍM KNIHOVNY OPENCV [AUTOR]	83

SEZNAM POUŽÍVANÝCH ZKRATEK

zkratky jsou seřazeny abecedně

AES – „Advanced Encryption Standard“

BBC – „British Broadcasting Corporation“

DPM – „Deformable Parts Models“

FN – „false negative“, česky falešně negativní (např. detekce)

FP – „false positive“, česky falešně pozitivní (např. detekce)

FPS – „frames per second“

GLSL – „OpenGL Shading Language“

GPU – „Graphics Processing Unit“

HOG – „Histograms of Oriented Gradients“

HOM – „Hoffman marker system“

HSV – barevný model, „Hue, Saturation, Value“

IGD – z německého „Institut Graphische Datenverarbeitung“

IoU – „Intersection over Union“

ISO – „International Organization for Standardization“

KL – Kullback–Leiblerova (divergence)

NMS – „Non-maximum Suppression“

PPV – „Prediktivní hodnota pozitivního testu“

QR kód – „quick response“ kód

R-CNN – „Regions with Convolutional Neural Network features“

RANSAC – „Random sample consensus“

RGB – barevný model, „Red, Green, Blue“

SCR – „Siemens Corporate Research“

SIFT – „Scale-Invariant Feature Transform“

SSD – „Single Shot Detector“

SURF – „Speeded Up Robust Features“

TN – „true negative“, česky správná negativní (např. detekce)

TP – „true positive“, česky správná pozitivní (např. detekce)

USPS – „United States Parcel Service“

WebGL – „Web Graphics Library“

YOLO – „You Only Look Once“

1 Úvod

Rozšířená realita je způsob kombinace reálného a virtuálního prostředí. Jedná se o pohled na reálný svět, který je rozšířen virtuálně generovanými prvky, jakými jsou například grafika, video nebo zvuk. Je v kontrastu s virtuální realitou, která naopak kompletně nahrazuje reálný svět a zcela vtahuje uživatele do vlastního virtuálního prostředí. Rozšířená realita může být považována za mezistupeň mezi reálným a virtuálním světem.

Doplnění virtuálních prvků do scény je nesnadnou úlohou, protože je obtížné zajistit, aby virtuální prvky byly správně umístěny a dobře korespondovaly s reálným okolím, se kterým jsou navíc kvalitativně srovnávány. Existují dva základní způsoby, jak zjistit ze scény dostatek informací pro korektní začlenění virtuálních prvků. Prvním je použití obrazových značek – tzv. markerů. Marker je předem definovaným tvarem, který je ve scéně snadné detekovat a podle něho zjistit pozici, orientaci a další nezbytné informace. Druhým způsobem jsou tzv. „marker-less“ systémy, které se snaží všechny potřebné informace získat pouze pomocí význačných bodů a objektů, které se již ve scéně nacházejí. Jedná se zpravidla o komplikovanější systémy, které ale mají tu výhodu, že scénu není nutné dopředu připravovat. Systémy využívající markery mají naopak výhodu v tom, že jsou použitelné i v situacích, kdy reálná scéna neobsahuje žádné přirozené význačné prvky. Zpravidla je také výhodnější je použít, pokud je vyžadována vyšší přesnost.

Detekce konkrétního markeru je proces vyhledání oblasti v obrazu s unikátním znakem, tvarem nebo barvou. Takové oblasti se používají jako vizuální podněty, které jsou snadno identifikovatelné systémy počítačového vidění. Typickým příkladem markeru jsou tzv. QR kódy (z anglického „quick response“), které se používají pro uchování textové informace v reálném světě ve strojově čitelném formátu. Dalším příkladem je Maxicode [1], [2], který využívá Poštovní služba Spojených států amerických (USPS, United States Parcel Service). Oba tyto systémy jsou také ISO standardy [3], [4].

Hlavním cílem QR kódů a systému Maxicode je uchovat komplexní informace přímo v prostorové scéně. Aplikace rozšířené reality ale typicky takovou funkcionalitu nevyžadují. Jejich hlavní potřebou je lokalizace objektu nebo kamery v reálném světě pro zajištění přesné vzájemné integrace reálného a virtuálního prostředí.

Disertační práce se zabývá dvěma směry výzkumu. Prvním je návrh markeru, jehož design je přesně zaměřen na využití v aplikaci rozšířené reality. Konkrétní potřeby takové aplikace jsou diskutovány dále v této práci. Druhým směrem je prozkoumání algoritmů a řešení, které by dokázaly navržený marker detekovat a zjistit z něj požadované informace. Při detekci je kladen důraz na spolehlivost, přesnost a rychlost.

Práce v následujících kapitolách detailně popisuje cíle disertační práce, použité metodiky a podává rešerši stávajících řešení. Dále jsou popsána specifika detekčních algoritmů a je podrobně popsán proces vytvoření autorem navrhovaného markeru. Na tento proces navazuje detekce markeru a návrh sady algoritmů, které z detekovaného markeru zjišťují výstupní informace. Nakonec jsou shrnuty a zhodnoceny výsledky a jsou nastíněny další možné směry výzkumu.

2 Cíle práce

Cíle práce jsou rozděleny do dvou hlavních kategorií. První z nich je návrh systému obrazových značek (tzv. „markerů“), které jsou díky své jednoduchosti přesně zaměřeny na využití v aplikacích rozšířené reality. Druhou kategorií cílů je prozkoumání možností detekce markerů navržených v rámci prvního hlavního cíle práce.

2.1 Návrh markeru

Prvním hlavním cílem práce je návrh markeru, který umožní detekci konkrétního místa ve snímaném obrazu v systému rozšířené reality. Markerem by typicky bylo označeno konkrétní místo nebo objekt, které je žádoucí v obrazu sledovat. V jedné scéně by se mohlo nacházet i více markerů pro označení více míst či objektů. Markery mají rozsáhlé využití, např. je pak možné dané místo z obrazu odstranit a nahradit jiným, či případně na detekované místo doplnit virtuální objekt.

Návrh vhodného markeru je složitým problémem, protože reálné scény jsou velmi rozmanité. Mnoho již existujících systémů řeší zároveň i jiné problémy a nejsou plně vhodné jen k prosté detekci. Například se zabývají zakódováním složitějších informací v co nejkompaktnější formě. Nutno podotknout, že i existence takových systémů je důležitá, protože mnoho aplikací vyžaduje, aby bylo možné do scény informace uložit. Například může jít o čtení webového odkazu ve strojově čitelném formátu, což je jeden z typických příkladů použití QR kódů. Další společnou nevýhodou stávajících systémů je používání komplikovaných tvarů, které jsou však nezbytné, pokud je nutné ve scéně spolehlivě zakódovat více informací.

Ze zmíněných důvodů je autorem navrhován nový systém, který bude přesně zaměřen na oblast jednoduché lokalizace. V takovém případě není nutné používat komplikované tvary. Navíc složité tvary je zpravidla nutné tisknout. Jednoduchý a pečlivě navržený marker dovede zajistit spolehlivou lokalizaci ve scéně a být méně rušivý. Takový marker by navíc vůbec nemuselo být nutné tisknout a v případě potřeby by ho bylo možné pouze nakreslit například fixou.

Navrhovaný marker musí splňovat několik kritérií, aby bylo možné ze scény získat všechny nezbytné informace pro další práci (například pro doplnění virtuálního objektu). Kritéria jsou definována jako dílčí cíle při návrhu, který je prvním hlavním cílem práce. Kritéria jsou následující:

- schopnost lokalizace markeru (preferovaně jeho středu),
- schopnost detekce orientace jako úhlu rotace v intervalu 0° až 360° ,
- a schopnost získání relativní velikosti pomocí detekce celého markeru.

Splnění všech tří vymezených kritérií je nezbytné dodržet při návrhu markeru. Dále se očekává splnění následujících doplňkových kritérií, které jsou také dílčími cíli prvního hlavního cíle práce:

- jednoduchý tvar markeru včetně zajištění možnosti kreslení rukou,
- a vzor markeru bez vazby na jeho konkrétní barvu.

Splnění doplňkových kritérií není nezbytně nutné, ale díky nim by bylo možné tvořit markery v různorodých přizpůsobitelných variantách, díky čemuž vždy dobře a nerušivě zapadnou do všech rozmanitých scén.

2.2 Návrh algoritmu detekce markeru

Druhým hlavním cílem práce je návrh algoritmů, které dokáží navržený marker detekovat a přečíst. V práci jsou zkoumány dva druhy detekce. Prvním druhem jsou klasické algoritmy, které pracují přímo s obrazovými daty a snaží se oblast s markerem najít. Návrh míří směrem k využití programovatelných shaderů grafických karet, které jsou typem hardware, který je speciálně navržený pro rychlé paralelní zpracování obrazových dat.

Druhým zvažovaným způsobem detekce je použití umělých neuronových sítí. Je předpokládáno, že při použití správně navržené umělé neuronové sítě je možné definovaný tvar markeru ve scéně detekovat. Neuronová síť by měla být schopna i dostatečné generalizace, aby při dobře navržené trénovací sadě dat byla schopna abstrahovat od barvy markeru, pracovat s rozmazaným obrazem a částečně zakrytým markerem. Důležitost takové generalizace podtrhuje i skutečnost, že jedním z definovaných požadavků je možnost kreslení markeru jen rukou, což

s sebou nese mnoho nepřesností a komplikací. Je však možné, že takové nepřesnosti naopak přispějí k lepší detekovatelnosti markeru.

Dále je nutné navrhnout řešení pro získání všech nezbytných informací přímo z detekovaného markeru – informace o pozici, orientaci a velikosti, jak byly definovány v předchozí podkapitole. Tyto informace jsou nezbytné pro dokončení procesu detekce a jsou hlavním výstupem celého řešení.

Zvažované přístupy detekce budou podrobně srovnány. Hlavním kritériem je vždy přesnost detekce, ale důležitá je také její rychlost a spolehlivost. Jednoduchost či složitost implementačních možností jsou také jedním z parametrů.

Proces zkoumání možností detekce navrženého markeru lze shrnout následujícími body, které též představují dílčí cíle druhého hlavního cíle práce:

- prozkoumat možnosti algoritmů pracujících s obrazovými daty,
- prozkoumat možnosti umělých neuronových sítí,
- vyhodnotit uvedená řešení z hlediska jejich schopnosti detekce navrženého markeru,
- a navrhnout způsob získání požadovaných informací z detekovaného markeru.

3 Metodika zpracování

Hlavními cíli práce jsou navržení markeru vhodného pro aplikace rozšířené reality a navržení algoritmu pro spolehlivou a přesnou detekci takového markeru, jak bylo uvedeno v předchozí kapitole. Dále v ní byla též definována tři kritéria, která je nezbytné splnit, aby marker byl plně využitelný z hlediska potřebných informací. Všechna tato kritéria vyžadují, aby marker byl detekován dostatečně přesně. K evaluaci detekce je používáno několik metrik určujících kvalitu klasifikátorů: počet a procento falešně negativních detekcí, počet falešně pozitivních detekcí, senzitivita (anglicky též „recall“), prediktivní hodnota pozitivního testu (anglicky „positive predictive value“ či též v případě klasifikací jako „precision“) a metrika IoU (zkratka z anglického „intersection over union“, někdy též označované jako „Jaccard index“). Dále jsou využívány různé datové sady a parametry neuronové sítě, které jsou blíže představeny v dalším textu této kapitoly.

3.1 *Evaluační metriky testování*

Falešně negativní (FN) detekce je situace, kdy existující marker není detektorem nalezen. Procento chybných detekcí je vyjádřeno jako podíl FN detekcí a celkového počtu markerů, které se v testové sadě nacházejí. Systém by ideálně měl detekovat nulový počet falešně negativních oblastí.

Falešně pozitivní (FP) detekce je situace, kdy je detektorem označena oblast, která žádný marker neobsahuje. Marker by měl být dostatečně odlišitelný od okolí, aby se takovým detekcím předcházelo. Algoritmus by ideálně měl detekovat nulový počet falešně pozitivních oblastí.

Senzitivita je hodnota, která se počítá jako podíl skutečně pozitivních detekcí (používá se zkratka TP z anglického „true positive“) a celkového počtu všech markerů (TP+FN). Lze jí formulovat také jako poměr případů, které jsou správně identifikovány. Její hodnota se pohybuje v intervalu $< 0; 1 >$ a vyšší hodnota je lepší.

$$\text{senzitivita} = \frac{TP}{TP + FN}$$

Prediktivní hodnota pozitivního testu (PPV) je hodnota, která se počítá jako podíl skutečně pozitivních detekcí (TP) a počtu všech detekovaných oblastí bez ohledu na jejich správnost (TP+FP). Její hodnota se pohybuje také v intervalu $< 0; 1 >$ a vyšší hodnota je také lepší.

$$PPV = \frac{TP}{TP + FP}$$

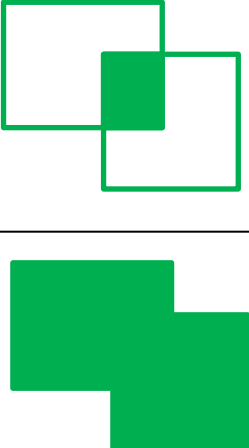
V podobných úlohách je často stanovována specifická, která se počítá jako podíl počtu skutečně negativních případů (používá se zkratka TN z anglického „true negative“) a součtu počtu TN a FP detekcí. V úloze detekce oblasti v obrazu není možné stanovit hodnotu TN. Každá korektně nedetekovaná oblast je skutečně negativním případem. Problém je v tom, že takových oblastí je teoreticky až neomezeně mnoho, protože takových oblastí lze v obrazu i neomezeně mnoho vytvořit. Každá taková teoretická oblast, která není algoritmem detekována, je skutečně negativním případem. Hodnotu specifickosti tedy nelze stanovit, protože čitatel i jmenovatel podílu by byly nekonečno a specifická by byla v limitě 1.

$$specifická = \frac{TN}{TN + FP}$$

Tabulka 1 Tabulka popisující vztahy mezi jednotlivými pravděpodobnostními hodnotami použitými jako metriky [autor]

		Skutečný stav		
Celkový počet		Všechny pozitivní stavy (TP + FN)	Všechny negativní stavy (FP + TN)	
Predikovaný stav	Predikce pozitivního stavu	Skutečně pozitivní (TP)	Falešně pozitivní (FP)	PPV $\frac{TP}{TP+FP}$
	Predikce negativního stavu	Falešně negativní (FN)	Skutečně negativní (TN)	
		senzitivita $\frac{TP}{TP+FN}$	specifická $\frac{TN}{TN+FP}$	

Hodnota IoU se vypočítá jako podíl obsahů průniku a sjednocení dvou oblastí, které představují detekovaný obdélník a obdélník správného řešení pro danou detekci (obecně lze uvažovat čtyřúhelníky). Výsledek je vždy v intervalu $< 0; 1 >$, kde nula znamená falešně pozitivní detekci a jedna znamená perfektní správnou detekci. Celková hodnota IoU pro všechny obrázky v testovací sadě se počítá jako průměr všech IoU hodnot pro jednotlivé markery ve všech obrázcích. V případě falešně negativní detekce se hodnota IoU považuje za nulovou. Pokud je správná oblast označena více správnými detekcemi, tak se celkové průměrné IoU počítá s tou detekcí, jejíž hodnota IoU je nejvyšší.

$$IoU = \frac{\text{obsah oblasti průniku}}{\text{obsah oblasti sjednocení}}$$


Obrázek 1 Grafická reprezentace metriky IoU [autor]

3.2 Měření času zpracování algoritmů na GPU

Čas strávený prováděním algoritmů implementovaných pomocí programovatelných shaderů grafických karet není možné přesně změřit standardním postupem uložení času v rámci kódu prováděného na CPU a následně zjištěním rozdílu od času uloženého po dokončení provádění. Provedená volání, která iniciují operace na GPU, nepopisují skutečnou časovou náročnost operací, protože se jedná pouze o předání příkazů na GPU a provádění programu na CPU po předání pokračuje nezávisle.

Měřené algoritmy jsou implementovány s využitím technologie WebGL a byl tedy zvolen odpovídající způsob měření. Experimenty probíhaly jak s WebGL1, tak také s modernější verzí WebGL2. Vzhledem k faktu, že v práci uvedené výsledky jsou

pouze s využitím WebGL2, i tato kapitola popisuje pouze způsob měření času v rámci WebGL2.

Popsané měření času je v rámci WebGL2 dosažitelné s pomocí tzv. „extensions“ neboli česky rozšíření, které se musí vždy explicitně aktivovat. Konkrétní rozšíření pro měření času se jmenuje `EXT_disjoint_timer_query_webgl2`. Rozšíření je využito v kombinaci s funkcí `beginQuery`, která je v rámci obslužného JavaScriptového programu zavolána před vybranou operací, která má být změřena (například operace vykreslení). Konec měření je pak označeno voláním funkce `endQuery`. Nakonec je nutné naměřený čas v nanosekundách získat voláním funkce `getQueryParameter`.

Použitím popsaného postupu je příkaz pro získání času zařazen do fronty příkazů přesně před moment, než se měřená operace začne provádět. Stejně i čas ukončení je uložen přesně poté, co se provádění operace dokončilo. Správa času je totiž plně pod kontrolou GPU. Přečtení naměřeného času zpět do JavaScriptového programu probíhá s využitím parametru `QUERY_RESULT_AVAILABLE_EXT`, který informuje o tom, zda požadovaná operace již byla dokončena. Obslužný program totiž po spuštění operací na GPU nemá žádnou kontrolu ani zpětnou vazbu, zda požadované operace již byly dokončeny. Musí se tedy periodicky dotazovat, jestli operace dokončená již je, a až poté výsledek vyčíst.

Hlavní nevýhodou všech WebGL rozšíření je, že jejich podpora není v dané implementaci WebGL zaručená. To znamená, že různá zařízení a různé webové prohlížeče mohou rozšíření nepodporovat. Rozšíření vyžadované pro měření času je dostupné pouze ve webových prohlížečích založených na jádru Chromium a dále pouze v jeho desktopové variantě [5] (k březnu 2023). Je tedy možné přesně změřit čas vykonávání v desktopovém systému, ale pro mobilní zařízení není tato varianta k dispozici.

Pro měření na mobilním zařízení je tedy využito méně přesné měření, kdy je využito faktu, že výsledek renderování je možné po dokončení načíst zpět do standardní operační paměti. Takové načtení je blokováno, dokud není renderování

dokončeno. Je tedy možné označit časem začátek renderování v JavaScriptovém kódu a měření zastavit v okamžik, kdy je úspěšně vyčtena spočítaná hodnota. Ačkoliv výsledky jsou méně přesné, tak přesnějších hodnot není možné v rámci využitých technologií dosáhnout.

Všechny uvedené výsledky jsou průměrem 1000 průběhů renderovací smyčkou, kdy průměr prvních 1000 průběhů je ignorován a druhý je použit.

3.3 Datové sady

Během testů bylo použito několik různých datových sad pro trénování neuronové sítě. V prvních iteracích jsou využity obrázky s uměle generovanými markery pro zrychlení procesu, protože ruční vytváření markerů všech zvažovaných variant v reálných scénách by bylo příliš časově náročné. Uměle generované markery byly vkládány do reálných obrázků. Pouze finální datové sady jsou vytvořeny ručně, aby byly zajištěny podmínky podobné reálným situacím.

3.3.1 Datové sady využité při návrhu

První verze datové sady využívá přibližně 3 000 obrázků a s dalšími iteracemi testů byla sada postupně zvětšována až na 9 000 obrázků.

Základní datová sada obsahuje zejména obrázky exteriérů a byla tedy vytvořena další sada, která je složena výhradně z interiérových obrázků. Celkem obsahuje 2 600 obrázků, které byly nalezeny na internetu pomocí anglických výrazů „bedroom“, „kitchen“, „living room“, „office“ a „home office“. V posledních iteracích je využita kombinace s předchozími sadami, čímž vznikla sada s přibližně 11 000 obrázky.

Třetí datová sada je kompletně oddělena od předchozích sad. Jednalo se o ručně vytvořené markery, které byly umístěny v reálných scénách, které byly nasnímány kamerou. Z videí vzniklo 550 obrázků s ručně kreslenými markery, ve kterých byly ručně vyznačeny objímající obdélníky, tzv. „bounding boxy“.

3.3.2 Finální datová sada pro testování

Finální čtvrtá datová sada se skládá z 3330 obrázků, ve kterých se nacházejí jak ručně kreslené markery, tak tisknuté markery – kreslených markerů je 36 a tisknutých také 36, celkem tedy 72 markerů. Všechny byly umístěny v reálných scénách jak v interiéru, tak v exteriéru. Snímání v interiéru proběhlo jak za denního světla, tak i pod umělým osvětlením.

Celkem bylo natočeno 72 videí (pro každý marker jedno video), ze kterých snímky datové sady pocházejí. Každé video je přibližně 15 sekund dlouhé a je snímáno v režimu 60 snímků za sekundu v rozlišení 1920 × 1080 pixelů (FullHD). Z videí pak byl automaticky vybrán každý 20. snímek. Ručním projitím bylo vybráno finálních 3330 markerů, které pak byly ve snímcích ručně tzv. „otagovány“. Tato čtvrtá datová sada je použita pro finální porovnání algoritmů detekce navrženého markeru.

Markery lze také rozdělit do 12 kategorií podle barvy, která je použita jako hlavní podklad. Jedná se o tyto barvy: černá, tmavě hnědá, světle hnědá, červená, oranžová, žlutá, světle zelená, tmavě zelená, světle modrá, tmavě modrá, fialová a růžová. Každá tato podkladová barva je použita třikrát s různou barvou vnitřního znaku, čímž vzniká 36 markerů. Každý z těchto 36 markerů pak existuje ve dvou provedeních – kreslený ručně a vytištěný.

Alternativní rozdělení pracuje s 6 kategoriemi po 12 markerech, kdy každá kategorie má vždy po jednom markeru pro každou podkladovou barvu:

- 12 ručně kreslených markerů snímaných v interiéru pod umělým osvětlením, snímání proběhlo 9. července 2022 kolem 22. hodiny
- 12 ručně kreslených markerů snímaných v interiéru za denního osvětlení, snímání proběhlo 3. července 2022
- 12 ručně kreslených markerů snímaných v interiéru za denního osvětlení, snímání proběhlo 26. června 2022
- 12 tištěných markerů snímaných v interiéru za denního osvětlení, snímání proběhlo 26. června 2022

- 12 tištěných markerů snímaných v exteriéru za slunného dne, snímání proběhlo 26. června 2022 kolem 16. hodiny
- 12 tištěných markerů snímaných v interiéru za denního osvětlení, snímání proběhlo 8. července 2022

Konkrétní počty pro každou ze 72 kategorií shrnuje vždy v kombinaci barva a prostředí následující tabulka:

Tabulka 2 Tabulka s četnostmi markerů finální datové sady napříč definovanými 72 kategoriemi [autor]

Barva	Ruční, Umělé světlo	Ruční, Interiér	Ruční, Interiér	Tištěný, Interiér	Tištěný, Exteriér	Tištěný, Interiér
Černá	92	49	44	49	38	41
Hnědá – tmavá	47	53	50	43	36	44
Hnědá – světlá	39	52	48	49	38	34
Červená	47	41	48	45	49	41
Oranžová	49	48	49	49	33	50
Žlutá	51	43	43	46	41	42
Zelená – světlá	43	46	44	50	41	51
Zelená – tmavá	46	44	49	49	45	44
Modrá – světlá	37	45	47	49	46	50
Modrá – tmavá	47	50	52	39	47	45
Fialová	47	44	47	45	40	50
Černá	51	54	44	47	47	47
Celkem	596	569	565	560	501	539

3.3.3 Tvorba datových sad

Proces vytváření bounding boxů se nazývá „tagování“. Vytvořený podpurný program pro tagování se nachází v autorově repositáři na serveru GitHub: <https://github.com/milankostak/Marker-detection/tree/master/java/src>.

Všechny datové sady byly vždy náhodně rozděleny v poměru 7:2:1 na trénovací, validační a testovací sadu. Výjimkou je pouze finální čtvrtá sada, kde rozdělení je

v poměru 7:1:2, aby byl zajištěn dostatečný a reprezentativní počet pro finální testování.

3.4 Parametry neuronové sítě

Všechny průběžné iterace testů proběhly s využitím architektury neuronové sítě YOLOv3 [6]. Vstupem pro trénování jsou obrázky a odpovídající bounding boxy s kategoriemi objektů, které se v jednotlivých bounding boxech nachází. Výstupem testování je obdobný seznam bounding boxů, kde každý z nich obsahuje souřadnice detekovaného objektu v obrázku, kategorii objektu a pravděpodobnost s jakou je detekovaný objekt klasifikován danou kategorií.

Vstupem při testování je také několik dalších parametrů. Neuronová síť každé detekci přiděluje pravděpodobnost správnosti klasifikace a parametrem lze určovat, které detekce budou považovány za validní a které budou zamítnuty. Výchozí hodnotou prahu je 0,3 (30% pravděpodobnost). Pokud pravděpodobnostní hodnota klasifikace nepřekročí tuto hodnotu, tak je objekt (marker) zamítnut. Nastavením prahu na nižší hodnotu se typicky sníží počet falešně negativních detekcí. Na druhé straně se však zpravidla zvýší počet falešně pozitivních detekcí. Zvýšení prahu má efekt přesně opačný.

Druhou hodnotou, která má přímý vliv na výstupy, je „non-maximum suppression“ (česky „potlačení nemaximálních hodnot“), které je často zkracováno jako NMS [7]. Použitá hodnota 0,45 udává IoU hranici, při které jsou velmi podobné návrhy bounding boxů filtrovány. Pokud se nějaké bounding boxy překrývají a jejich společná hodnota IoU je vyšší než zadaný práh, tak je navržený bounding box s nižší pravděpodobnostní hodnotou odstraněn. Vzhledem ke specifičnosti úlohy, kdy je předpokládáno, že markery nebudou umístěny příliš blízko u sebe, by bylo možné zvážit navýšení NMS hranice.

4 Rešerše literatury

Existuje velké množství literatury, ve které se autoři zabývají návrhem rozmanitých markerů pro různé účely. Kapitola nejprve dává základní pohled na rozdělení systémů a poté následuje podrobný popis 64 nalezených systémů, které jsou nejbližší pojetí v této práci.

Většina zkoumaných systémů zpravidla nspecifikuje prostředí, ve kterém mají být použity. Někteří autoři markerů se však zaměřili na různá prostředí, ve kterých by se marker měl používat. Například dos Santos Cesar a kolektiv [8] provedli evaluaci několika systémů v podmínkách pod vodní hladinou. Bondy a kolektiv [9] navrhli marker, který je určen speciálně pro operace v kosmu.

Je možné také nalézt literaturu, která se zabývá porovnáním existujících systémů. Například Xiang Zhang a kolektiv v roce 2002 porovnali systémy ARToolKit, IGD, SCR a HOM [10]. Shabalina a kolektiv v roce 2019 porovnali systémy ARTag, AprilTag a CALTag [11]. Morar a kolektiv v roce 2020 vypracovali komplexní studii, která se zaměřila na evaluaci celé řady systémů v interiérech [12]. Někteří autoři se zabývali teoretickým návrhem nejlepšího markeru, například Owen a kolektiv v roce 2002 [13] a Fiala v roce 2010 [14].

Je také nutné zmínit, že existuje mnoho systémů, které pro lokalizaci ve scéně nepoužívají markery, ale snaží se význačné prvky extrahovat přímo ze scény, například [15]–[18]. Proces detekce bez markerů je zpravidla komplikovanější a výpočetně náročnější. Několik studií se také zaměřilo na porovnání systémů, které markery využívají a ty které markery nepoužívají [19]–[22].

Zkoumané markery mohou být rozděleny podle několika různých charakteristik:

- podle barvy markeru:
 - černý/šedo-tónový (například ARToolKit, ARTag a Fourier Tag)
 - barevný (například CH-marker a ChromaTag)
- podle tvaru markeru:
 - čtverec (například ARTag, AprilTag a CALTag)
 - kruh (například TRIPTag a RUNE-Tag)

- jiné (například ReacTIVision a Random dot markers)
- podle primární cílové aplikace jde o schopnost:
 - nést komplexní informaci (například QR kód a Maxicode)
 - nést informaci o identifikátoru (například ARTag a AprilTag)
 - lokalizace (například SIFTtag a SURFtag)
 - kalibrace kamery nebo odhad pozice kamery (například CALTag a HArCo)

4.1 Období 1996-2008

Rasmussen a kolektiv [23] v roce 1996 diskutovali možnosti sledování objektů způsobem, který by byl založen pouze na znalosti barvy objektu. Systém byl zaměřen na sledování jednoho nebo více objektů v prostoru na základě ukázky, kterou uživatel poskytl před začátkem sledování. Díky tomu bylo možné jednoduše a rychle objekty detekovat. Autoři také představili způsob opravy chyb, ke kterým může při sledování docházet. Jako konkrétní příklad použití systému uvádějí sledování tenisového míčku. Hlavní nevýhodou systému je neschopnost pracovat při nižším osvětlení scény a při zastínění objektu stínem, což jsou však situace, které v reálných scénách nastávají často. Práce byla publikována před 27 lety a tato skutečnost znamená, že algoritmus nevyužívá výhod moderního hardware – například možnosti paralelizace pomocí grafické karty.

Thomas a kolektiv [24] v roce 1997 pro televizi BBC navrhli a vyvinuli systém pro pozicování kamery v systému virtuální reality. Markery byly umístěny ve známých pozicích, díky čemuž byly kamery schopny vypočítat vlastní pozici.

Matrix [25] je jedním z nejstarších systémů, který jako marker využívá černou čtvercovou oblast s mřížkou menších bílých čtverců uvnitř, které uchovávají informaci (Obrázek 2 (c)). Systém byl publikován v roce 1998.

Moran a kolektiv [26] v roce 1999 vyvinuli systém pro sdílení informací na reálné fyzické zdi. Kamera sleduje povrch zdi a systém pomocí markerů pomáhá vyhledávat a číst jednotlivé informace, které jsou pak doplňovány o elektronické informace z připojené databáze.

ARToolKit [27], [28] je systémem, který cílí na použití v aplikacích rozšířené reality. Marker se skládá z černého čtverce, který ve vnitřním menším bílém čtverci obsahuje znovu černý charakteristický znak (Obrázek 2 (d)). Uživatel si sám definuje sadu znaků a ty jsou při detekci vyhledávány v databázi všech vzorů pomocí korelace. To však vede k pomalému procesu detekce, pokud je databáze vzorů obsáhlá. Systém byl poprvé představen v roce 1999 a inspiroval mnoho budoucích systémů, které ho později překonaly. Aktualizovaná verze ARToolKit Plus [29] je stále jedním z nejpoblárnějších systémů. Aktualizace byla představena v roce 2005 a využívá po vzoru jiných systémů binárně kódované vzory místo korelace.

CyberCode [30] je systém pro sledování objektů navržený v roce 2000, jehož marker se skládá z černých čtverců na bílém pozadí (Obrázek 2 (e)). Každý marker je unikátní a dekodovatelný na konkrétní binární identifikátor. Hlavním cílem autorů bylo navrhnout systém, který by dokázal pracovat i s obrazem z kamery s nízkou kvalitou, které se často nacházejí v mobilních telefonech. Autoři vyvinuli i referenční implementaci s konkrétními aplikacemi – aplikace pro navigaci a provádění muzeem a aplikace pro obecnou navigaci v interiéru. Autoři dále předvedli možnost vylepšení systému pomocí dalších senzorů (například pomocí gyroskopu), které mohou pomoci se sledováním markeru, pokud se ztratí ze zorného úhlu kamery.

Cho a Neumann [31] v roce 2001 publikovali myšlenku markeru složeného z více kruhů. Byl to navíc jeden z prvních systémů, který využíval barevné markery. Předcházející systémy pracovaly na principu detekce pomocí prahování obrazu k nalezení černého markeru. Tento systém využívá prahování zvlášt pro každou barevnou složku RGB modelu. Marker se skládá z několika soustředných kruhů, které se od sebe liší barvou. Kombinace více barev má zajistit unikátnější a lépe rozlišitelné markery. Autoři dále navrhli dva druhy markeru: s konstantní šířkou kruhu a s proporcionální šířkou kruhu (větší průměr kruhu znamená jeho větší šířku). Obrázek 2 (f) je příkladem markeru s proporcionální šířkou.

Zhang a kolektiv [10] se v roce 2002 zabývali čtyřmi různými markerovými systémy – ARToolKit [27], IGD¹ systém, SCR² systém [32], [33] a HOM³ systém. Vybrány byly díky tomu, že všechny byly volně k dispozici a používaly se v aplikacích rozšířené reality. Systémy byly hodnoceny v parametrech přesnosti, spolehlivosti, rychlosti a použitelnosti. Autoři závěrem konstatují, že ARToolKit umožnil nejrychlejší zpracování obrazu, zatímco SCR byl nejrychlejší při zpracování videa. Rozpoznatelnost markerů byla podobná ve všech systémech, ARToolKit dosahoval nejlepších výsledků na malých markerech. Přestože výsledky studie jsou užitečné, je nutné připomenout, že byla publikována v roce 2002 a neevaluuje novější systémy. Appel a Navab v roce 2002 použili HOM systém v aplikaci rozšířené reality [34].

TRIPtag [35] je systém publikovaný v roce 2002. Marker se skládá ze struktury tvaru terče (černá tečka a černý kruh) a dvou černých soustředných kruhů rozdělených do 16 sektorů na bílém pozadí (Obrázek 2 (g)). Systém detekce používá adaptivní prahování a umí zakódovat 3^9 (= 19 683) unikátních identifikátorů.

Naimark a Foxlin [36] v roce 2002 publikovali systém, jehož hlavním cílem je sledování pohybu v obrazu. Tvar markeru se skládá z kruhového černého obrysu, ve kterém se dále nachází černé a bílé části (Obrázek 2 (h)). Autoři se zaměřili hlavně na řešení různých úrovní kontrastu v reálných scénách. Systém pro svou funkčnost vyžaduje viditelnost alespoň tří markerů. Jakmile jsou všechny lokalizovány, systém se přepne do sledovacího režimu a kontroluje pouze části obrazu, ve kterých byly markery detekovány v předchozím snímku, čímž je zajištěno rychlejší zpracování. Systém byl také patentován [37].

Kawano a kolektiv [38] v roce 2003 navrhli detekční systém, jehož marker se skládá ze 32 bitů informace včetně možnosti opravy chyb při detekci. Minimální oblast, se kterou je systém schopen pracovat, je 50×50 pixelů. Marker se skládá

¹ IGD z německého „Institut Graphische Datenverarbeitung“

² SCR z anglického „Siemens Corporate Research“

³ HOM z anglického „Hoffman marker system“

z čtvercového černého rámečku, ve kterém se nacházejí bílé a černé trojúhelníky na černém pozadí (Obrázek 2 (i)). Celkem se jedná vždy o 50 trojúhelníků.

D-touch [39] je systém navržený v roce 2003. Jeho detekční metoda je založena na spojitosti stromového grafu, kterým jsou reprezentovány jednotlivé regiony markeru [40]. Znamená to, že vizuálně různé markery mohou mít stejnou topologickou strukturu. Takové markery se dokonce mohou lišit ve svém tvaru a stále mohou mít stejnou topologii. Obrázek 2 (j) ilustruje vzhled markeru.

VisualCode [41] je systém vydaný v roce 2004. Zaměřuje se zejména na kamery s nízkou kvalitou snímání, které se často nacházejí v mobilních telefonech. Referenční implementace byla vytvořena pro operační systém Symbian. Vzhled markeru je podobný předchozím systémům – černé čtverce na bílém pozadí (Obrázek 2 (k)). Rohs jako jeden z původních autorů pokračoval v práci na projektu a v roce 2005 vydal aktualizovanou verzi [42].

Claus a Fitzgibbon [43] v roce 2004 publikovali systém, jehož cílem bylo využití v přirozených venkovních scénách. Vzhled markeru se od ostatních systémů liší – jedná se o bílé pozadí se čtyřmi černými kruhy v každém z rohů pomyslného čtverce. Každý marker je pak identifikován číslem, které je uvedeno uprostřed bílé oblasti (Obrázek 2 (l)). Proces detekce je založen na kaskádě klasifikátorů a jedná o jeden z prvních systémů, který v nějaké formě využívá strojové učení. Díky tomu je systém robustní vůči změnám osvětlení a měřítka markeru. Autoři uvádějí, že přesnost systému přesahuje 95 % v různých exteriérových a interiérových scénách. V roce 2005 stejní autoři navázali na svou práci a zaměřili se na možnosti automatické kalibrace systému [44].

ARTag [45] je systém, který v roce 2005 představil Mark Fiala. Tvar markeru byl inspirován systémem ARToolKit [27]. Hlavním rozdílem je nemožnost volby při tvorbě sady vnitřních znaků. Systém umožňuje výběr ze sady 2002 předdefinovaných markerů, kde každý z nich reprezentuje jeden unikátní identifikátor. Čtvercový marker se skládá z černého obrysu a pravidelné mřížky černých čtverců, které jsou umístěny uprostřed na bílém pozadí (Obrázek 2 (m)).

System detekuje markery pomocí hranového detektoru. Markery dále obsahují také opravný kód. V roce 2010 Fiala publikoval práci [14], ve které shrnul metody, doporučení a způsoby návrhu velmi spolehlivých markerů.

Dell'Acqua a kolektiv [46] v roce 2005 publikovali systém, který je založen na barevných markerech, které by měly být lépe identifikovatelné v komplexních scénách a ve scénách s horšími světelnými podmínkami. Marker se skládá z čtvercových bloků, které jsou diagonálně rozděleny. Celkem tedy obsahuje 50 trojúhelníků, které jsou rozděleny na zelené a modré. Ohraničující obrys je červený (Obrázek 2 (n)). Systém je inspirován prací Kawana a kolektivu [38], jejichž marker má stejný tvar, ale využívá pouze černou barvu.

Cantag [47] je systém s otevřeným zdrojovým kódem publikovaný v roce 2006. Na rozdíl od jiných systémů podporuje více různých tvarů a různé množství uložené informace. Systém nabízí výběr ze čtyř podporovaných tvarů – tři druhy kruhů a jeden druh čtverce. Obrázek 2 (o) představuje příklad tvaru „vnější kruh“. Motivací pro vytvoření systému s podporou více tvarů bylo umožnění výběru nejvhodnějšího tvaru v aktuální situaci a podmínkách.

TriCodes [48] je systém představený v roce 2006. Marker se skládá z černého rámečku s bílou oblastí uvnitř, ve které se nachází jeden čtverec a osm trojúhelníků rozdělených do pravidelné mřížky rozměru 3×3 (Obrázek 2 (p)). Vnitřní čtverec se používá k získání celkové orientace markeru. Každý trojúhelník může být otočen do osmi různých pozic a dva trojúhelníky jsou použity pro kontrolní součet. To znamená, že je možné vytvořit $8^6 = 2^{18}$ (= 262 144) různých variant.

ReactIVision [49] je detekční systém vyvinutý v roce 2007. Základem markeru je nepravidelný amébovitý tvar (Obrázek 2 (q)). Geometrie tvaru byla vygenerována pomocí genetického algoritmu. Pozice markeru je určena jako geometrický střed (centroid) vypočítaný podle všech nalezených vnitřních uzlů. Knihovna je zdarma a má otevřený zdrojový kód [50].

Tateno a kolektiv [51] v roce 2007 publikovali myšlenku rekurzivního vnořování markerů. Díky tomu, že je možné menší variantu markeru vložit do větší varianty a tento postup vícekrát opakovat, lze dosáhnout velmi dobrých detekcí i v situacích, kde je kamera příliš blízko k cílovému objektu nebo je od něj příliš daleko. Autoři konkrétně diskutují, že vyšší vrstva by mohla obsahovat současně více markerů v nižší vrstvě a tento postup dále opakovat.

Fourier Tag [52] je detekční systém vyvinutý v roce 2007. Je založen na frekvenční analýze obrazu, která dekóduje vzory. Konkrétně byla použita Fourierova transformace, od které systém dostal své jméno. Autoři argumentují, že systémy jako ARTag [45] nemají dobré výsledky, pokud se obraz zhorší, například pokud se kamera od markeru příliš vzdálí nebo pokud produkuje příliš mnoho šumu. Fourier Tag se snaží zaměřit na řešení popsanych problémů. Marker je tvořen sadou kružnic, jejichž průměr se zvyšuje se vzdáleností od středu (Obrázek 2 (r)). Xu a Dudek [53] v roce 2011 systém vylepšili o možnost variabilního množství uložené informace a možnost detekce rotace.

CipherCode [54] je systém publikovaný v roce 2007 a od svých předchůdců se odlišuje v tom, že přímo v návrhu podporuje šifrování a parametrizaci množství vložených informací. Šifrovací modul používá AES („Advanced Encryption Standard“) šifrování. Obrázek 2 (s) obsahuje příklad markeru systému CipherCode. V době publikace nebyl systém schopen fungovat v reálném čase, ale lze předpokládat, že na moderním hardware by v reálném čase fungovat dokázal. Autoři své výsledky porovnávají se systémem Semacode, který však již není k dnešnímu dni k dispozici. CipherCode je k použití zdarma a má otevřený zdrojový kód.

BinARyID [55] je systém představený v roce 2007. Tvar markerů je podobný mnoha předchozím systémům s černým pozadím a bílými čtverečky, které se nacházejí uprostřed v pravidelné mřížce rozměru 4×4 (Obrázek 2 (t)). Systém je implementován jako doplněk pro ARToolKit [27] a umožňuje automatické generování markerů, které mají každý unikátní identifikátor. Tímto přístupem je eliminována hlavní nevýhoda systému ARToolKit, pro který je jinak nutné ručně

vnitřní podobu markerů definovat. Je nutné však podotknout, že tento problém byl již v této době vyřešen jinými systémy, například systémem ARTag [45]. Autoři závěrem sdělují, že jejich přístup poskytuje mírně lepší výsledky než obyčejný ARToolKit.

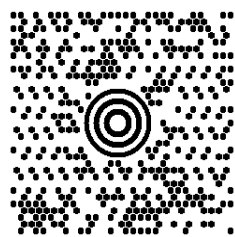
SVMS [9] („Space Vision Marker System“) je systém navržený v roce 2007 speciálně pro použití při kosmických operacích. Kvůli tomu jsou kladeny velké nároky na odolnost vůči zakrytí části markeru a vůči přítomnosti silných stínů a ostrého světla. Marker je složen z bílého obrysu, který je obklopen 44 černými a bílými buňkami. Uvnitř každého markeru je nachází černý kruh, který je obklopen bílými tečkami (Obrázek 3 (a)).

FPI marker [56] je detekční systém publikovaný v roce 2007. Jeho marker se skládá ze sady libovolně uspořádaných trojúhelníků, které jsou černé a každý obsahuje právě jednu bílou tečku (Obrázek 3 (b)). Ostatní systémy zpravidla pro zanesení informace využívají oblast, která musí být dostatečně velká, ale zároveň není využita pro zlepšení přesnosti detekce. FPI marker se snaží tento problém řešit. Autoři v práci přiznávají, že koexistence více markerů v obrazu nebyla testována.

Martin Hirzer v roce 2008 navrhl detekční systém [57], který používá detekci hran a hranové pixely seskupuje pomocí algoritmu RANSAC („Random sample consensus“). Autor uvádí, že díky tomu je detekce rychlá a robustní vůči osvětlení a částečnému zakrytí.

4.1.1 Shrnutí období 1996-2008

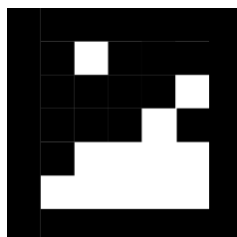
Období, které je zahrnuto podkapitolou 4.1, se vyznačuje tím, že v kontextu dnešní doby se jedná o zastaralé systémy, které nevyužívají výhod moderního hardware. Dále jsou referenční implementace často vyvinuty pro systémy, které dnes již nejsou používány ani podporovány. Mnohdy se jedná o různé „proof-of-concept“ systémy. Avšak většina průkopnických prací, které velmi ovlivnily pozdější systémy, pochází právě z této doby, a proto je nutné se tímto obdobím také zabývat a čerpat z něj informace a postupy, které během něho byly zavedeny a osvědčily se.



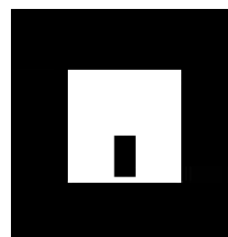
(a) Maxicode [1], 1992



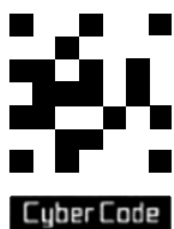
(b) QR kód, 1994



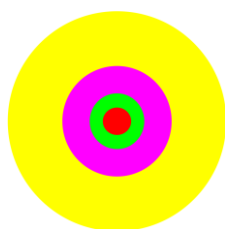
(c) Matrix [25], 1998



(d) ARToolKit [27], 1999



(e) CyberCode [30], 2000



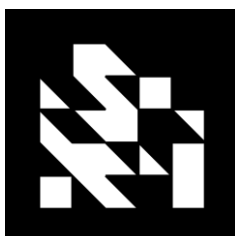
(f) Cho a Neumann [31], 2001



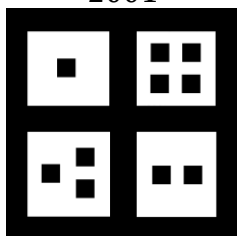
(g) TRIPtag [35], 2002



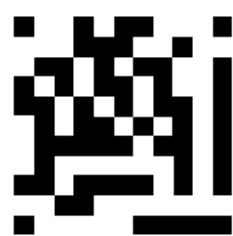
(h) Naimark a Foxlin [36], 2002



(i) Kawano a kolektiv [38], 2003



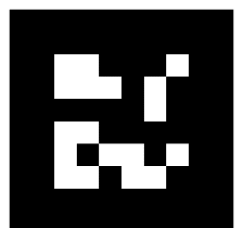
(j) D-touch [39], 2003



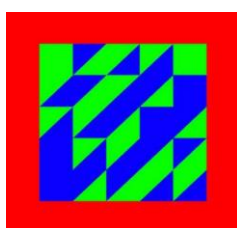
(k) VisualCode [41], 2004



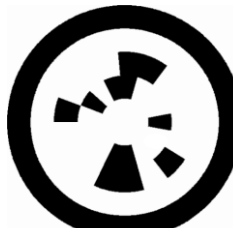
(l) Claus a Fitzgibbon [43], 2004



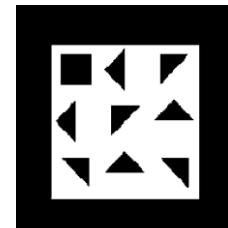
(m) ARTag [45], 2005



(n) Dell'Acqua a kolektiv [46], 2005



(o) Cantag [47], 2006



(p) TriCodes [48], 2006



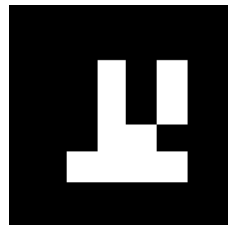
(q) reactIVision [49], 2007



(r) Fourier Tag [52], 2007



(s) CipherCode [54], 2007



(t) binARyID [55], 2007

Obrázek 2 Příklady 20 markerů z období 1992-2007 s jejich jménem (pokud je k dispozici) a rokem jejich publikace

4.2 Období 2009-2015

Bokode [58] je systém vizuálních značek publikovaný v roce 2009. Systém cílí na detekci velmi malých markerů z velké vzdálenosti. Takové markery by byly pro člověka jednoduše přehlednutelné a díky tomu by byly v prostředí velmi nerušivé.

Schweiger a kolektiv [59] v roce 2009 vytvořili tři specifické markery, které jsou velmi dobře detekovatelné pomocí metod SIFT [15] a SURF [60], které jsou běžně využívány při práci se scénami, do kterých se žádné markery neumísťují (tzv. „marker-less“ rozšířená realita). Díky tomu se autoři mohli plně soustředit na návrh markeru a nemuseli řešit návrh detekčního systému. Hlavní nevýhodou přístupu je možnost použití pouze dvou markerů pro metodu SIFT (Obrázek 3 (c)) a jednoho pro metodu SURF (Obrázek 3 (d)). Pokud je však cílem aplikace pouze dobře detekovat oblast bez nutnosti ukládat další informace, tak se jedná o dostačující řešení.

CALTag [61] je marker vyvinutý v roce 2010. Je specificky navržený pro použití při kalibraci kamer. Oblast markeru je tvořena pravidelným střídáním černých a bílých čtverců (jako na šachovnici) a každý roh je využíván jako kalibrační bod (Obrázek 3 (e)). Tímto přístupem je možné dosáhnout dobré detekce, i když je velká část markeru zakrytá. Autoři v textu podrobně vysvětlují, proč jsou systémy jako ARTag [45] nevhodné pro kalibraci kamery.

Wu a kolektiv [62] navrhli v roce 2010 systém, jehož marker se skládá z černé vyplněné čtvercové oblasti, která v sobě obsahuje bílé tečky různých velikostí, které jsou umístěny na diagonálách (Obrázek 3 (f)). Autoři uvádějí, že systém má v porovnání s ARToolKit [27] přibližně 30krát rychlejší detekci.

Neto a kolektiv [63] v roce 2010 publikovali systém pro kalibraci kamery podobný systému CALTag [61]. Marker nového systému má také podobu šachovnice, ale pro tmavé buňky používá modrou a černou barvu a pro světlé buňky žlutou a bílou barvu. Díky tomu bylo možné zvýšit kapacitu markeru a vyloučit situace s perspektivní nejednoznačností. Systém také nevyžaduje kalibraci barev a je

schopen lépe pracovat ve špatných světelných podmínkách. Systém umožňuje zakódovat až 2^{16} (= 65 536) různých markerů.

„Random dot markers“ (česky „náhodné tečkované markery“) [64] je systém publikovaný v roce 2011, jehož markery jsou založeny na náhodně rozprostřených černých tečkách ve čtvercové oblasti (Obrázek 3 (g)). Díky tomu by měl marker být v prostředí méně rušivý. Experimenty autorů ukázaly, že 40 náhodně umístěných teček na jeden marker je použitelných pro vytvoření až jednoho tisíce různých markerů. Systém byl vytvořen s důrazem na flexibilitu a robustnost vůči částečnému zakrytí markeru. Autoři dále diskutují, že náhodně umístěné markery nemusí mít ty nejlepší výsledky a mohlo by být možné nalézt lepší nenáhodné rozdělení.

AprilTag [65] je systém vyvinutý Edwinem Olsonem, který ho publikoval v roce 2011. Tvar markeru je podobný systémům ARToolKit [27] a ARtag [45] (Obrázek 3 (h)). Nově navržený detekční systém je odolnější vůči změnám osvětlení a částečnému zakrytí. Dále je schopen zakódovat až 4 164 různých markerů, což je dvakrát více, než podporuje ARtag. ARToolKit nedokáže pracovat s téměř žádným částečným zakrytím a ARtag zvládá pouze zakrytí malé části markeru. V roce 2016 Wang a Olson [66] aktualizovali detektor pro zlepšení odolnosti vůči změnám v osvětlení a pro zlepšení efektivity výpočtů, čímž se detekce zrychlila 1,5krát. Aktualizovaný detektor se někdy nazývá jako AprilTag 2. Tvar markeru se nezměnil. AprilTag je k dispozici zdarma a má otevřený zdrojový kód [67].

RUNE-Tag [68] je systém publikovaný v roce 2011. Hlavní důraz systému je kladen na velkou odolnost vůči zakrytí části markeru. Tvar markeru je netradiční a je tvořen tečkami, které se nacházejí na pomyslných soustředných kruzích (Obrázek 3 (i)). Verze RENE-129 je schopna detekovat 100 % markerů při 50% zakrytí a 67 % markerů při 70% zakrytí. Silná odolnost vůči zakrytí má však svou cenu v rychlosti detekce a systém je zhruba o řád pomalejší než ARToolKit Plus [29]. Stále je ale schopen pracovat v reálném čase. Dále je nutné podotknout, že zvyšující se pozorovací úhel negativně ovlivňuje schopnost detekce. V roce 2016 autoři dále

svůj systém vylepšili [69]. Pi-Tag je podobný systém s mírně odlišným tvarem markeru, který byl publikován stejnými autory [70].

Belghit a kolektiv [71] v roce 2012 vymysleli barevný marker, který se skládá ze sady čtyř barevných čtverců (zelený, žlutý, modrý a červený), z nichž každý je umístěn v jednom rohu pomyslného objímajícího čtverce (Obrázek 3 (j)). Nevýhodou systému je fakt, že přítomnost jiných objektů stejné barvy může detekci ovlivnit. A vzhledem k tomu, že jsou použity čtyři odlišné a časté barvy, mohou podobné situace nastávat často. Tyto barvy byly ale autory vybrány úmyslně pro jejich jednoduchou rozpoznatelnost v prostoru. Autoři dále přiznávají, že zakrytí jednoho barevného čtverce detekci znemožní.

BlurTags [72] je systém publikovaný v roce 2012, jehož hlavním cílem je detekce v obrazech, které obsahují rozmazání způsobené pohybovou neostrotí (tzv. „motion blur“). Marker je tvořen šachovnicí, jejíž pole jsou na okrajích mírně rozmazaná a obsahují bílé nebo černé čtverce, které jsou hlavním nosičem informace (Obrázek 3 (k)). Autoři své výsledky porovnávají se systémem CALTag [61] a tvrdí, že jejich systém disponuje lepší odolností vůči neostroti obrazu.

CoP-Tag („Connected Points Tag“) [73] je systém vyvinutý v roce 2012. Tvar markeru je tvořen obrysem čtverce, na kterém se nachází 16 teček (na každé straně je vždy pět teček, Obrázek 3 (l)). Detekce spolehlivě funguje s až 62,5 % zakrytých teček, což znamená, že postačuje viditelnost šesti teček. Autoři také tvrdí, že systém je odolný vůči šumu a rozmazání v obrazu.

Chuang a kolektiv [74] v roce 2013 navrhli metodu pro navigaci pro zrakově hendikepované osoby. Hlavním cílem je možnost skenování markeru, pomocí kterého by se uživatel dokázal lokalizovat. Autoři se dále pokusili sloučit údaje z dalších senzorů, které lze nalézt v mobilních telefonech. Experimenty ukázaly, že vyvinutý referenční systém je použitelný jak v interiéru, tak v exteriéru. Detekční metoda byla implementována pomocí knihovny ArUco [75]. Marker se skládá ze zelených a bílých oblastí umístěných uvnitř červeného rámečku (Obrázek 3 (m)).

Pi-Tag [70] je v roce 2013 publikovaný systém, který navazuje na systém RUNE-Tag [68], který byl vyvinut stejným týmem. Design markeru je podobný systému CoP-Tag [73] – skládá se z 12 teček umístěných na hranách pomyslného čtverce, kdy každá hrana obsahuje 4 tečky (Obrázek 3 (n)). Detekce markeru trvá 10 až 150 milisekund a je přibližně o řád pomalejší než detekce systému ARToolKit Plus [29]. Systém je schopen fungovat se střední úrovní zakrytí a poskytuje uspokojivé výsledky i v situacích, kdy je polovina teček zakrytá.

Liu a kolektiv [76] v roce 2013 prezentovali systém, který využívá barevné rovnostranné trojúhelníkové markery (Obrázek 3 (o)). Mnoho předchozích systémů pracuje s černobílými markery, které jsou díky vysokému kontrastu dobře extrahovatelné. Na druhé straně ale mají často nízkou schopnost pracovat za špatných světelných podmínek. Použití barevných markerů cílí na řešení takových problémů. Primární použití systému je při sledování objektů v reálném čase a následném registrování virtuálních objektů na sledované objekty. Autoři tvrdí, že RGB a HSV barevné modely nejsou dostatečně flexibilní a představují novou metodu, která má detekci markeru v obrazu řešit lépe. Hlavní nevýhodou práce je porovnání nové metody pouze se systémem ARToolKit [27], který byl v době publikace 11 let starý a překonaný jinými řešeními. Nová detekční metoda je navíc pomalejší než metoda, kterou používá ARToolKit.

Mono-spectrum marker [77] je systém publikovaný v roce 2013. Primárním účelem markeru je schopnost spolehlivé detekce v nezaostřeném a rozmazaném vstupním obrazu. Podle autorů jsou podobné situace časté, pokud se kamera pohybuje nebo pokud je stacionární a má nastavenou fixní ohniskovou vzdálenost. Detekční systém je však výpočetně náročný a autoři doporučují využití grafické karty, což by však v dnešní době již neměl být problém. Základem markeru je černý čtverec, ve kterém se v pravidelné mřížce nachází devět kruhových oblastí, jejichž okraje jsou rozmazané (Obrázek 3 (p)).

Suzuki a kolektiv [78] v roce 2013 navrhli marker, který je speciálně použitelný na válcovém povrchu. Ostatní systémy často předpokládají, že se marker bude nacházet na rovném povrchu a takové systémy selhávají, pokud je marker umístěn

na netypickém povrchu, například na plechovce nebo lahvi. Tvar markeru se skládá z černého rámečku, ve kterém se nachází bílé tečky, které pomáhají detekovat pozici a zahnutí markeru (Obrázek 3 (q)).

Garrido-Jurado a kolektiv [79] v roce 2014 navrhli systém, jehož primární použití je v aplikacích rozšířené reality a při lokalizaci robotů. Systém je založen na běžných čtvercových černo-bílých markerech (Obrázek 3 (r)). Práce byla implementována pomocí knihovny ArUco [75]. Autoři se zaměřili na automatické generování sady markerů s libovolnou velikostí markeru a s libovolným množstvím markerů v sadě, díky čemuž mohou být markery vygenerovány podle potřeb aktuální situace. Pokud je vyžadován pouze malý počet různých markerů, je lepší použít malou množinu markerů, protože spolehlivost detekce bude vyšší. Situace vyžadující více markerů mohou využít větší množinu, ale musí však počítat s tím, že spolehlivost detekce může být nižší. V roce 2016 byl systém generování dále vylepšen [80] pomocí metody tzv. „celočíselného programování“ (anglicky „mixed integer linear programming“).

BullsEye [81] je v roce 2014 vyvinutý systém, který byl navržen speciálně pro výpočet na grafické kartě. Cílovou aplikací autorů je interakční systém na ploše stolu a detekční systém tedy musí být velmi přesný. Autoři konstatují, že navržený systém pracuje s přesností desetiny pixelu, je schopen ze scény získat pozici středu, úhel rotace a unikátní identifikátor a je dostatečně efektivní pro běh v reálném čase. Systém je porovnáván zejména se systémem reacTIVision [49]. Tvar markeru se skládá z kruhového rámečku, který je místy přerušen. V jeho středu se nachází další černý kruh, v jehož středu je další bílý kruh (Obrázek 3 (s)).

Prasad a kolektiv [82] v roce 2015 publikovali detekční systém, který cílí na použití v dronech, které potřebují informace o svém okolí pro spolehlivou navigaci. Vzhledem k rychlému pohybu dronů je obraz z jejich kamery často rozmazaný. Navržený marker se skládá ze sady soustředných kruhů, které by měly být snadno detekovatelné v rozmazaném obraze (Obrázek 3 (t)). V době publikace nebyl systém schopen zpracovat jeden snímek v reálném čase a na jeho zpracování vyžadoval přibližně 300 milisekund.

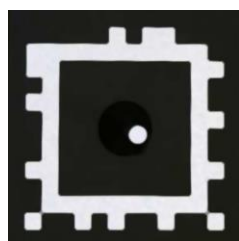
4.2.1 Shrnutí období 2009-2015

Období 2009-2015 se neslo v duchu vylepšování stávajících postupů a řešení. Zejména se jedná o odolnosti vůči částečnému zakrytí, pohybu/rozmazání, šumu, nízkému osvětlení atp. Dalším význačným znakem navržených systémů je snaha o rychlé zpracování vstupních dat, což je nezbytné pro nasazení v reálných aplikacích. Vznikly také různé knihovny a frameworky (např. ArUco), které mají dalším lidem zjednodušit a zrychlit práci při návrhu vlastních systémů.

4.3 Období 2016-2020

L-split [83] je systém, který byl v roce 2016 vyvinut speciálně pro použití při montáži letadel. Jeho název vychází z tvaru markeru, který se skládá ze čtyř dílů tvaru písmene „L“, kde se každý díl nachází v rohu pomyslného čtverce (Obrázek 4 (a)). Každý ze čtyř dílů obsahuje část celkové informace a pro úspěšnou detekci je požadována viditelnost tří dílů. Systém je schopen zakódovat až 2^{18} (= 262 144) odlišitelných markerů.

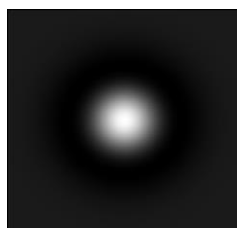
CCTag [84] je v roce 2016 představený systém, který cílí na zlepšení detekovatelnosti v rozmazaných obrazech. Marker se skládá z černých soustředných kruhů na bílém pozadí (Obrázek 4 (b)). Tloušťka jednotlivých kruhů je použita pro zakódování informací. Experimenty ukázaly, že detekční systém je schopen pracovat s velkým množstvím rozmazání a částečného zakrytí. Je ale nutné podotknout, že systém je schopen zpracovat pouze čtyři snímky za sekundu při použití procesoru i5-4590 a 11 snímků při použití grafické karty GTX 980 Ti. Testy byly prováděny pro rozlišení obrázku 1280×720 pixelů. Systém má otevřený zdrojový kód [85]. Předchůdcem je systém C²Tag [86], který stejní autoři publikovali v roce 2012.



(a) SVMS [9], 2007



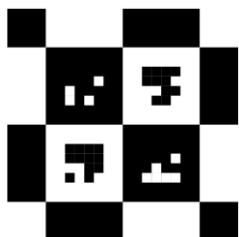
(b) FPI marker [56], 2007



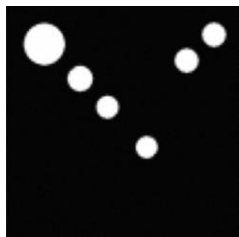
(c) Schweiger a kolektiv [59], SIFT, 2009



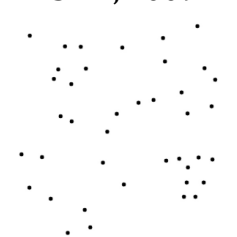
(d) Schweiger a kolektiv [59], SURF, 2009



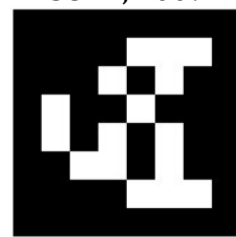
(e) CALTag [61], 2010



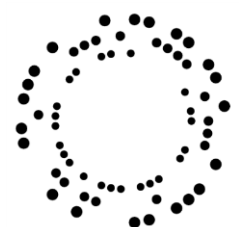
(f) Wu a kolektiv [62], 2010



(g) Random dot marker [64], 2011



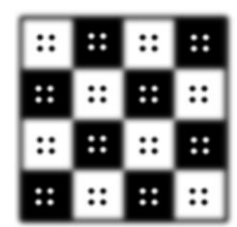
(h) AprilTag [65], 2011



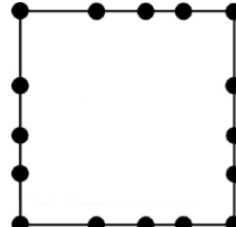
(i) RUNE-Tag [68], 2011



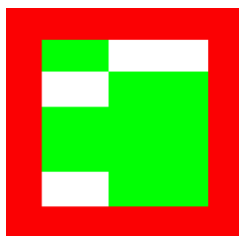
(j) Belghit a kolektiv [71], 2012



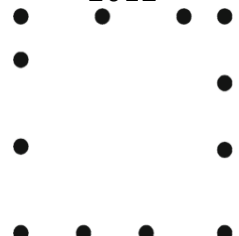
(k) BlurTags [72], 2012



(l) CoP-Tag [73], 2012



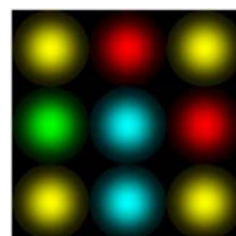
(m) Chuang a kolektiv [74], 2013



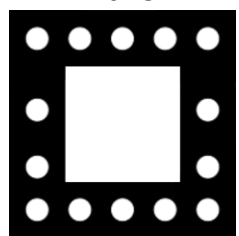
(n) Pi-Tag [70], 2013



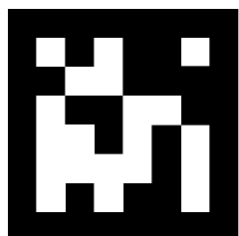
(o) Liu a kolektiv [76], 2013



(p) Mono-spectrum marker [77], 2013



(q) Suzuki a kolektiv [78], 2013



(r) Garrido-Jurado a kolektiv [79], 2014



(s) BullsEye [81], 2014



(t) Prasad a kolektiv [82], 2015

Obrázek 3 Příklady 20 markerů z období 2007-2015 s jejich jménem (pokud je k dispozici) a rokem jejich publikace

CH-marker [87] je v roce 2017 navržený systém, jehož hlavním cílem je spolehlivá detekce i při částečném zakrytí markeru. Marker se skládá z 49 barevných čtverců, které jsou umístěny do pravidelné čtvercové mřížky 7×7 (Obrázek 4 (c)). Devět vnitřních čtverců (3×3) má žlutou nebo bílou barvu a kódují hodnotu nula nebo jedna. Vnější čtverce jsou modré nebo červené a ukládají také hodnotu nula nebo jedna. Díky tomu existuje osm detekovatelných rohů namísto klasických čtyř, což je užitečné v situacích, kdy je marker částečně zakrytý. Systém je marker schopen detekovat, pokud je zakryto maximálně 37 % jeho plochy. Rychlost detekce je srovnatelná se systémy ARToolKit [27] a ARTag [45], pokud je v obrazu nízký počet markerů (jeden až tři). Rychlost se však snižuje s rostoucím počtem markerů ve scéně.

ChromaTag [88] je systém publikovaný v roce 2017. Autoři se pomocí barevných gradientů snažili zrychlit proces detekce, která je tak 92krát rychlejší než CCTag [84], 22krát rychlejší než RUNE-Tag [68] a 16krát rychlejší než AprilTag [65]. ChromaTag zároveň dosahuje podobné úspěšnosti detekce jako zmíněné systémy. Při použití procesoru i7 3. generace systém dokáže zpracovat 926 snímků za sekundu (FPS), což ho činí jedním z nejrychlejších systémů. V případě správných detekcí dosahuje 709 FPS a pro falešně negativní detekci jsou obrázky zpracované s 2616 FPS. Jako hlavní nevýhodu systému autoři uvádějí větší chybovost při velké vzdálenosti od markeru. Pro takové situace doporučují AprilTag. Obrázek 4 (d) obsahuje příklad ChromaTag markeru.

HArCo [89] je v roce 2018 navržený systém, jehož hlavní myšlenkou je hierarchická struktura, která dovoluje vnořování více markerů. Podobné řešení již bylo navrženo v roce 2007 [51]. HArCo se však liší tím, že je možné uvnitř jednoho markeru vnořit více markerů na stejné úrovni (Obrázek 4 (e)). Systém je schopen zpracovat jeden obrázek za přibližně čtyři milisekundy.

Transparent Random Dot Markers (česky „průhledné náhodné tečkové markery“) [90] je systém publikovaný v roce 2018, který pracuje s markery, které jsou vytisknuty na průhledný papír. Uchiyama zde navázal na svůj předchozím systém „Random Dot Markers“ [64], který publikoval v roce 2011.

Susan a kolektiv [91] v roce 2018 prozkoumali možnost využití Kullback-Leiblerovy (KL) divergence [92] pro detekci markerů. Autoři svou práci porovnávají se systémem ARToolKit [27], který markery detekuje pomocí korelace vůči známé sadě markerů. Nový systém se snaží proces korelace nahradit KL divergencí, pomocí které se hledá podobnost v sadě předem definovaných markerů. KL divergence je použita pro výpočet vzdálenosti dvou pravděpodobnostních rozdělení, která jsou definována rozdělením pixelů zpracovávaných markerů. Podle autorů je nová technika přesnější než použití korelace.

LiDaRTag [93] je v roce 2019 publikovaný systém, jehož hlavním přínosem je aplikace lidarů v procesu detekce markeru. Nově vytvořený marker navíc dovoluje detekci jak lidarem, tak i klasickou kamerou. Použití lidarů znamená, že změny v úrovních ambientního osvětlení nemají vliv na detekci. Autoři v textu práce diskutují nezbytná opatření, která jsou nutná při vývoji detekčního systému, který využívá lidar. Dále uvádějí, že markery systémů AprilTag [27], ARTag [45] a systém od Naimarka a Foxlina [36] jsou všechny použitelné v systému LiDaRTag. Na druhé straně markery typu CyberCode [30], CALTag [61] a systém od Cho a Neumanna [31] pro takové použití vhodné nejsou. Autoři dále navrhli svůj vlastní marker (Obrázek 4 (f)) a podrobněji ho porovnali s markerem systému AprilTag. Nový marker je detekován v reálném čase a detekce je mírně pomalejší ve venkovním prostředí. Přesnost detekce je 99 % ve venkovním i vnitřním prostředí.

STag [94] je detekční systém navržený v roce 2019. Primárním cílem bylo navržení markeru a detekčního procesu, které by byly dobře odolné vůči rychlým změnám a nestabilitám v obrazu. Proto v sobě marker obsahuje jak čtvercovou, tak i kruhovou strukturu, a je jedním z prvních systémů, který takovou myšlenku aplikuje (Obrázek 4 (g)). Černý čtverec v sobě obsahuje bílý kruh, ve kterém se nacházejí černé tečky, které nesou informace. Vnitřní černé tečky jsou navíc spojené, protože jsou na ně opakovaně aplikovány morfologické operace dilatace a eroze. Velikost knihovny markerů je variabilní mezi 6 až 22 309 markery. Autoři svůj systém porovnali s ARToolKit Plus [29], prací od Garrido-Jurada a kolektivu [79] a systémem RENE-Tag [68]. V porovnání s těmito systémy vychází STag jako

system s nejpřesnější detekcí markeru a s mírně pomalejším detekčním systémem, kdy je však stále schopen běhu v reálném čase. Systém je poskytován zdarma a má otevřený zdrojový kód [95].

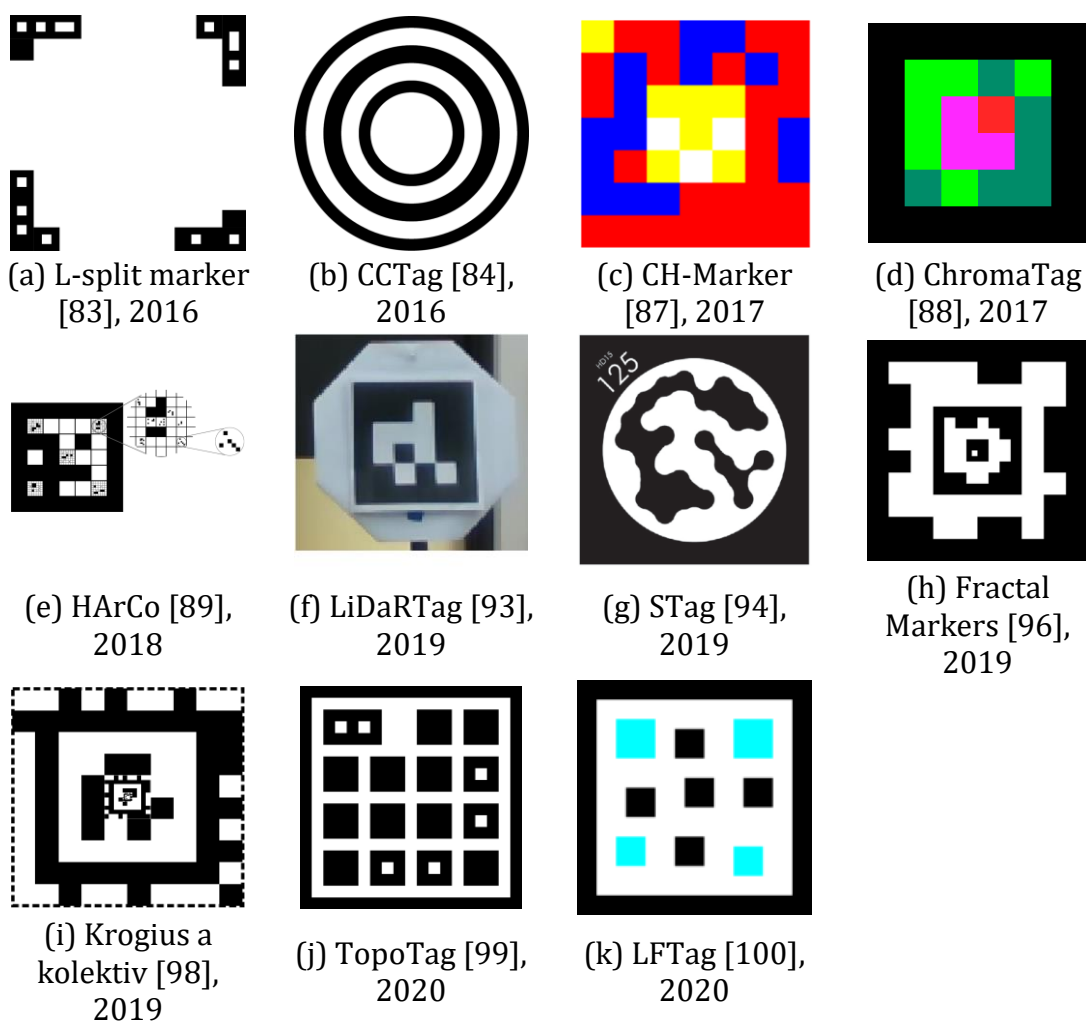
Fractal Markers [96] je detekční systém, který využívá rekurzivní vnořování markerů a který byl publikován v roce 2019. Mnoho markerů trpí špatnou detekovatelností, pokud jsou částečně zakryty. Vnoření více markerů do sebe problém elegantně řeší. Navíc je řešen i problém příliš velké či příliš nízké vzdálenosti kamery od markeru. Obrázek 4 (h) představuje příklad 3vrstvého markeru. Systém je zdarma a jeho zdrojový kód je k dispozici jako součást knihovny ArUco [97].

Krogius a kolektiv [98] v roce 2019 publikovali systém, který se zaměřuje na jednoduchou adaptabilitu pro různé cílové aplikace. Marker může mít variabilní hustotu uložené informace, mít kruhový nebo čtvercový tvar a být rekurzivní (Obrázek 4 (i)). Výsledky práce byly zakomponovány do systému AprilTag [65], [66] a nový systém je též někdy nazýván jako AprilTag 3.

TopoTag [99] je v roce 2020 představený systém, jehož hlavním rysem je velká přizpůsobitelnost. Základní tvar se skládá z černého rámečku, ve kterém se na bílém pozadí nachází černé čtverce, z nichž některé dále obsahují menší bílý čtverec (Obrázek 4 (j)). Vnější černý rámeček ale může mít libovolný tvar a autoři dávají jako příklad obrys ve tvaru motýla. Vnitřní černé čtverce mohou mít také libovolný tvar a autoři dávají příklad kruhu a šestiúhelníku. Velikost knihovny markerů je variabilní a autoři uvádějí, že jejich generátor je mnohem rychlejší než generátory v systémech AprilTag [65] a ArUco [79]. Generování podobně velkých sad zabere 4,1 sekundy pro TopoTag, minuty pro ArUco a dny pro AprilTag. Detekce probíhá v reálném čase a je srovnatelná se systémy ARToolKit [27], AprilTag a ChromaTag [88]. Testy byly provedeny na 169 713 obrázcích reálných scén. Detekční systém je schopen pracovat s až 10% zakrytím markeru.

LFTag [100] je systém, který vytvořil Ben Wang v roce 2020. Hlavním cílem bylo řešení jednoznačnosti při detekci rotace a schopnost spolehlivé lokalizace i

v obtížných světelných podmínkách a velkém pozorovacím úhlu. Tvar markeru se skládá z černého rámečku, který v sobě obsahuje černé a světle modré čtverce na bílém pozadí (Obrázek 4 (k)). V porovnání se systémy AprilTag 3 [98] a TopoTag [99], LFTag umožňuje vygenerování větší sady markerů. Sada může kódovat až 100 bitů. Hlavními problémy markeru jsou nízká odolnost vůči částečnému zakrytí a ohýbání. Detekce je dostatečně rychlá pro běh v reálném čase. Systém je zdarma a má otevřený zdrojový kód [101].



Obrázek 4 Příklady 11 markerů z období 2016-2020 s jejich jménem (pokud je k dispozici) a rokem jejich publikace

4.3.1 Shrnutí období 2016-2020

Období po roce 2016 se vyznačuje zejména tím, že při vývoji systémů již není zpravidla problémem nedostatek výpočetního výkonu. Existující dostupné nástroje dovolují snadno zpracovávat data pomocí grafických karet, které dokáží díky silné

paralelizaci zpracovat obrázky za zlomek času v porovnání s CPU. Stejně tak již i mobilní zařízení (smartphones) jsou dostatečně výkonná a kompaktní.

4.4 Celkové shrnutí

Většina představených markerů je podobného typu jako dnes všudypřítomné QR kódy, jejichž cílem je uložit ve scéně co nejvíce (textových) informací, které jsou snadno strojově čitelné. Cílem této práce je návrh markeru, který je vhodný pro aplikace rozšířené reality, které nevyžadují umístění většího množství informací ve scéně. Pro takové aplikace je důležité, aby marker byl dobře detekovatelný a bylo možné z něj zjistit informace o poloze, orientaci a velikosti, což jsou nezbytné informace pro nahrazení markeru virtuálním objektem. Pouze minorita prozkoumaných systémů se zabývá podobným řešením. Jedná se například o SIFTtag a SURFtag [59]. Tyto dva markery však trpí nejednoznačností rotace. Pouze několik z představených systémů podporuje detekci rotace – Fourier Tags [53], BullsEye [81] a LFTag [100]. Některé markery jsou navrženy s důrazem na schopnost dobré detekce v rozmazaných obrazech – BlurTags [72], CoP-Tag [73], Mono-spectrum marker [77], systém od Prasada a kolektivu [82] a CCTag [84].

Další charakteristickou vlastností všech představených systémů je vyžadování komplexních tvarů, které je nutné tisknout pro zajištění přesné funkčnosti detekčních systémů. Jednoduchý marker pro detekci by nemusel být pouze tisknutý, ale jeho jednoduchost by mohla dovolovat i ruční nakreslení.

5 Proces návrhu designu markeru

Prvním ze dvou primárních cílů práce je návrh vhodného markeru. Kapitola je rozdělena do deseti podkapitol, která každá popisuje průběžné výsledky kontinuálního vývoje a myšlenkových pochodů, které vedly k návrhu finálního tvaru markeru, který má dobré výsledky při detekci a splňuje všechna tři definovaná kritéria.

Detekce napříč jednotlivými iteracemi testů návrhů byly prováděny pomocí umělých neuronových sítí, konkrétně s využitím detekční architektury YOLOv3 [6]. Další prozkoumané možnosti detekce navrženého markeru jsou popsány a diskutovány dále v práci v kapitole 7 Způsoby detekce navrženého markeru.

5.1 První iterace testů

První testy byly provedeny s osmi různými tvary markeru, které byly vybrány jako výchozí bod pro další testování. Vybranými tvary byly:

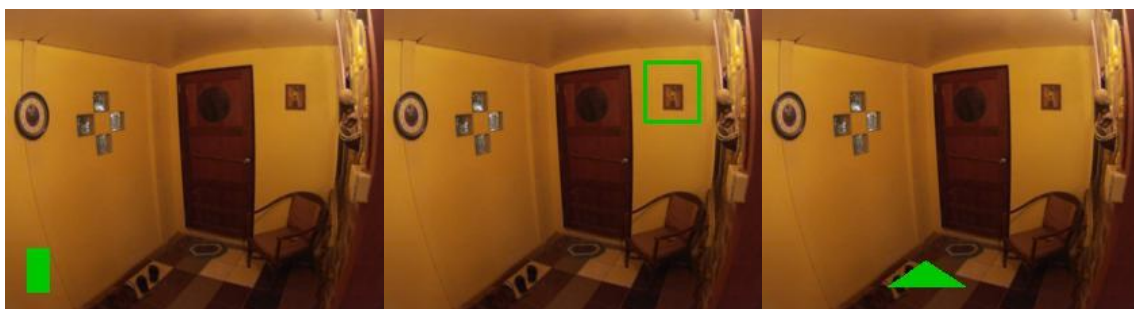
- vyplněný obdélník,
- obrys obdélníku,
- vyplněný trojúhelník,
- obrys trojúhelníku,
- hvězda tloušťky 2,
- hvězda tloušťky 1,
- hvězda tloušťky 1 v obrysu obdélníku,
- symbol zavináče (@).

Všechny zmíněné tvary nesplňují kritéria definovaná pro tvar markeru. Bylo však nezbytně nutné provést podrobné testování různých tvarů, aby se ověřilo, které z nich mají dobré výsledky při detekci. Další iterace testů vždy využily informace z předchozích iterací pro návrh vhodnějších tvarů.

Trénovací sada první iterace testů se skládala z 2 082 obrázků, validační sada z 594 obrázků a testovací sada z 297 obrázků. Pro každý tvar byly provedeny dva testy – první se třemi epochami učení a druhý s devíti epochami. Tabulka 3

obsahuje výsledky z první iterace. Více epoch vedlo k lepším výsledkům a zlepšení bylo srovnatelné napříč testovanými tvary.

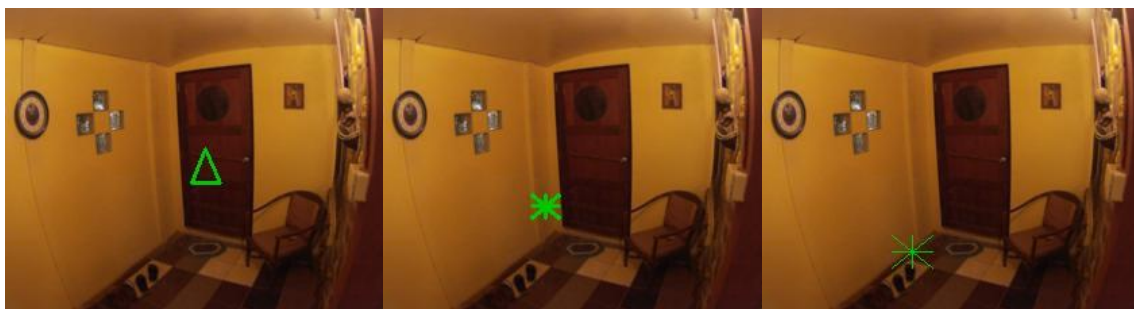
Tvar „vyplněný obdélník“ měl vysoký počet falešně pozitivních detekcí. Je to pravděpodobně způsobeno tím, že v reálných scénách se často nachází jednobarevné obdélníkové oblasti. Znamená to, že tento tvar není vhodný, ačkoliv splňuje některá definovaná kritéria.



Obrázek 5 Vyplněný obdélník [autor]

Obrázek 6 Obrys obdélníku [autor]

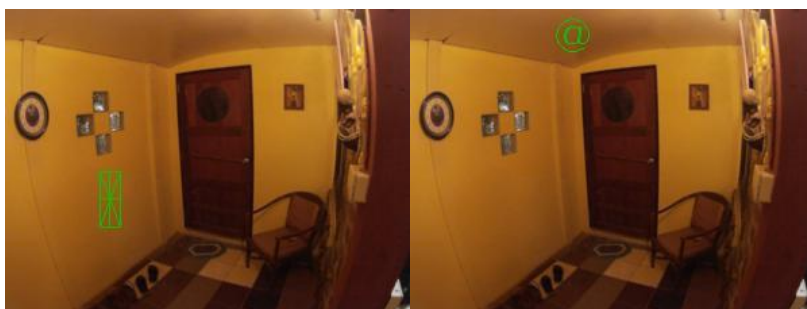
Obrázek 7 Vyplněný trojúhelník [autor]



Obrázek 8 Obrys trojúhelníku [autor]

Obrázek 9 Hvězda tloušťky 2 [autor]

Obrázek 10 Hvězda tloušťky 1 [autor]



Obrázek 11 Hvězda tloušťky 1 v obrysu obdélníku [autor]

Obrázek 12 Symbol zavlnáče (@) [autor]

Tabulka 3 Porovnání výsledků osmi testovaných tvarů markeru v první iteraci testů pro devět učicích epoch [autor]

Tvar markeru	Průměrná IoU	FP	FN	Procent FN	PPV	Senzitivita
<i>Vyplněný obdélník</i>	0,665	15	16	5,4 %	0,949	0,946
<i>Obrys obdélníku</i>	0,666	12	16	5,4 %	0,959	0,946
<i>Vyplněný trojúhelník</i>	0,701	8	8	2,7 %	0,973	0,973
<i>Obrys trojúhelníku</i>	0,679	6	13	4,4 %	0,979	0,956
<i>Hvězda tloušťky 2</i>	0,695	3	9	3,0 %	0,990	0,970
<i>Hvězda tloušťky 1</i>	0,654	13	19	6,4 %	0,955	0,936
<i>Hvězda tloušťky 1 v obrysu obdélníku</i>	0,614	19	21	7,1 %	0,936	0,929
<i>Symbol zavináče (@)</i>	0,699	10	19	6,4 %	0,965	0,936

5.2 Druhá iterace testů – doplněny dva nové tvary

Druhá iterace testů byla navržena na základě výsledků první iterace. K současným osmi tvarům byly přidány dva nové – „hvězda ve vyplněném obdélníku“ pro tloušťku čáry 1 a 2 (Obrázek 13 a Obrázek 14). Tyto tvary nectí původní myšlenku jednobarevného markeru, protože nakreslení čáry do již vyplněného obdélníku vyžaduje druhou barvu. Jednalo se o možný další směr zkoumání, protože detekce celistvé jednobarevné oblasti se ukázala jako náchylná k falešně negativním detekcím a samotná hvězda také nedosáhla očekávaných výsledků.

Všech osm předchozích variant bylo otestováno znovu, aby se potvrdily předchozí dobré či špatné výsledky (Tabulka 4). Testování proběhlo se stejnou datovou sadou.

Obdélníkové tvary měly znovu špatné výsledky a z dalších testů byly vyřazeny. Tvar „hvězda tloušťky 1“ měl stále vysoký počet falešně pozitivních detekcí a nízkou hodnotu IoU. Kvůli tomu byl také vyloučen z dalších testů. Dále „hvězda tloušťky 1 v obrysu obdélníku“ byla komplikovaným tvarem, který měl navíc také konzistentně špatné výsledky. „Symbol zavináče (@)“ má obvykle jednu

z nejlepších hodnot IoU, ale součet jeho falešně pozitivních a falešně negativních detekcí je vysoký a pro svůj komplikovaný tvar byl z dalších testů také vyřazen.

Na druhé straně obě varianty tvaru „hvězda ve vyplněném obdélníku“ měly dobré výsledky. Navíc oba už splňují dvě ze tří definovaných kritérií. Nesplňují pouze kritérium pro detekci úhlu rotace v intervalu 0° až 360°. Je možné ji detekovat pouze v intervalu do 90° v případě čtverce nebo v intervalu do 180° v případě obdélníku.

Tabulka 4 Porovnání výsledků deseti testovaných tvarů markeru v druhé iteraci testů. Pro každou variantu je uveden nejlepší výsledek bez ohledu na počet epoch, který je buď 3, 6 nebo 9. Modře jsou označeny nové varianty. Červeně jsou označeny varianty, které jsou pro další iterace testů vyřazeny. [autor]

Tvar markeru	Průměrná IoU	FP	FN	Procent FN	PPV	Senzitivita
<i>Vyplněný obdélník</i>	0,695	12	15	5,1 %	0,959	0,949
<i>Obrys obdélníku</i>	0,672	16	15	5,1 %	0,946	0,949
<i>Vyplněný trojúhelník</i>	0,699	4	10	3,4 %	0,986	0,966
<i>Obrys trojúhelníku</i>	0,708	4	12	4,0 %	0,986	0,960
<i>Hvězda tloušťky 2</i>	0,725	2	5	1,7 %	0,993	0,983
<i>Hvězda tloušťky 2 ve vyplněném obdélníku</i>	0,717	1	11	3,7 %	0,997	0,963
<i>Hvězda tloušťky 1</i>	0,635	6	25	8,4 %	0,978	0,916
<i>Hvězda tloušťky 1 v obrysu obdélníku</i>	0,676	12	9	3,0 %	0,960	0,970
<i>Hvězda tloušťky 1 ve vyplněném obdélníku</i>	0,687	4	13	4,4 %	0,986	0,956
<i>Symbol zavínáče (@)</i>	0,738	6	11	3,7 %	0,979	0,963

5.3 Třetí iterace testů – doplněny dva nové tvary

Z předchozí iterace postoupilo do dalšího testování pět tvarů. Další dva nové tvary byly doplněny na základě poznatků získaných v předchozí iteraci. Varianty s hvězdou v obdélníku dosáhly dobrých výsledků a jednodušší varianta křížku byla

přidána ve variantách „křížek tloušťky 2 v obdélníku“ (Obrázek 15) a „křížek tloušťky 1 v obdélníku“ (Obrázek 16).

Trénovací sada byla rozšířena o 1 321 obrázků na celkových 3 403, validační sada o 378 na celkových 972 a testovací sada o 189 na celkových 486. Trénování proběhlo pro 21 učicích epoch.

Obě varianty trojúhelníkového tvaru stále dosahovaly horších výsledků než obdélníkové tvary a z dalších testů byly vyřazeny. Tvar „hvězda tloušťky 2“ měl sice vytrvale dobré výsledky, ale z dalších testů byl také vyřazen, protože nesplňuje žádné z definovaných kritérií. Nové tvary s křížkem dosáhly nejlepších výsledků, jak informuje Tabulka 5.

Tabulka 5 Porovnání výsledků sedmi testovaných tvarů markeru v třetí iteraci testů pro 21 učicích epoch. Modře jsou označeny nové varianty. Červeně jsou označeny varianty, které byly pro další iteraci vyloučeny. [autor]

Tvar markeru	Průměrná IoU	FP	FN	Procent FN	PPV	Senzitivita
<i>Vyplněný trojúhelník</i>	0,735	2	16	3,3 %	0,996	0,967
<i>Obrys trojúhelníku</i>	0,728	6	12	2,5 %	0,988	0,975
<i>Hvězda tloušťky 2</i>	0,758	6	7	1,4 %	0,988	0,986
<i>Hvězda tloušťky 2 ve vyplněném obdélníku</i>	0,755	2	9	1,9 %	0,996	0,981
<i>Hvězda tloušťky 1 ve vyplněném obdélníku</i>	0,764	7	11	2,3 %	0,985	0,977
<i>Křížek tloušťky 2 v obdélníku</i>	0,764	1	9	1,9 %	0,998	0,981
<i>Křížek tloušťky 1 v obdélníku</i>	0,764	3	5	1,0 %	0,994	0,990

5.4 Čtvrtá iterace testů – obrázky interiéru

Všechny tři předchozí iterace používaly datovou sadu, které se skládá z autorových fotografií, které byly pořízeny převážně ve venkovním prostředí. Kvůli tomu byla vytvořena nová sada, která se skládá pouze z interiérových obrázků, aby se ověřilo,

zda navržený postup a dosavadní dobré výsledky nebyly ovlivněny podobou datové sady (v oblasti umělých neuronových sítí tzv. bias).

Nová sada obsahuje 2 600 obrázků, které byly vyhledány na internetu pomocí anglických klíčových slov „bedroom“ (ložnice), „kitchen“ (kuchyně), „living room“ (obývací pokoj), „office“ (kancelář) a „home office“. Sada je rozdělena do trénovací, validační a testovací sady v počtech 1 820, 520 a 260 obrázků.

Testy byly provedeny pro stejný počet učicích epoch, aby byly předchozí výsledky alespoň částečně porovnatelné. Výsledky nových testů prokázaly, že obrázky interiérů, které častěji obsahují oblasti jednolitých barev, neovlivňují schopnost a přesnost detekce (Tabulka 6).



Obrázek 13 Hvězda tloušťky 2 ve vyplněném obdélníku [autor]



Obrázek 14 Hvězda tloušťky 1 ve vyplněném obdélníku [autor]



Obrázek 15 Křížek tloušťky 2 v obdélníku [autor]



Obrázek 16 Křížek tloušťky 1 v obdélníku [autor]

Tabulka 6 Porovnání výsledků čtyř testovaných tvarů markeru ve čtvrté iteraci testů s interiérovými obrázky pro 21 učicích epoch. Z tabulky je vidět, že výsledky jsou srovnatelné s učením s exteriérovými obrázky (Tabulka 5). [autor]

Tvar markeru	Průměrná IoU	FP	FN	Procent FN	PPV	Senzitivita
<i>Hvězda tloušťky 2 ve vyplněném obdélníku</i>	0,701	2	8	3,1 %	0,992	0,969
<i>Hvězda tloušťky 1 ve vyplněném obdélníku</i>	0,719	5	8	3,1 %	0,981	0,969
<i>Křížek tloušťky 2 v obdélníku</i>	0,692	0	7	2,7 %	1,000	0,973
<i>Křížek tloušťky 1 v obdélníku</i>	0,743	5	5	1,9 %	0,981	0,981

5.5 Pátá iterace testů – více různých barev

Všechny předchozí testy byly realizovány s uměle generovanými markery, které všechny využívaly stejnou barvu – zelenou pro vyplnění a modrou pro případné nakreslení hvězdy nebo křížku. Průběžný experiment s kamerou naznačil, že neuronová síť dokáže detekovat i markery jiné barvy, než která se vyskytla v trénovací sadě, avšak detailnější testování muselo být provedeno pro potvrzení této domněnky.

Pro zrychlení postupu byly zvoleny pouze obě varianty křížku. Barvy byly generovány náhodně v RGB barevném modelu, kde všechny jeho barevné komponenty byly v rozsahu od 200 do 255, aby bylo zajištěno, že generované barvy jsou dostatečně saturované. Pro každý marker byly náhodně vybrány dvě barevné komponenty RGB pro vyplnění obdélníku a jedna pro nakreslení vnitřního křížku. Výsledky na stejné datové sadě prokázaly, že použití více různých barev v procesu trénování nemá vliv na výslednou schopnost detekce (Tabulka 7). Podobný test byl proveden i se sadou interiérových obrázků a výsledky jsou také srovnatelné (Tabulka 8).

Tabulka 7 Výsledky trénování s více různými barvami markeru s exteriérovými obrázky pro 21 učících epoch. Z tabulky je vidět, že výsledky jsou srovnatelné s učením s dvěma fixními barvami (Tabulka 5). [autor]

Tvar markeru	Průměrná IoU	FP	FN	Procent FN	PPV	Senzitivita
<i>Křížek tloušťky 2 v obdélníku</i>	0,764	5	8	1,7 %	0,990	0,984
<i>Křížek tloušťky 1 v obdélníku</i>	0,751	6	10	2,1 %	0,988	0,979

Tabulka 8 Výsledky trénování s více různými barvami markeru s interiérovými obrázky pro 21 učících epoch. Z tabulky je vidět, že výsledky jsou srovnatelné s učením s dvěma fixními barvami (Tabulka 6). [autor]

Tvar markeru	Průměrná IoU	FP	FN	Procent FN	PPV	Senzitivita
<i>Křížek tloušťky 2 v obdélníku</i>	0,674	2	14	5,4 %	0,992	0,946
<i>Křížek tloušťky 1 v obdélníku</i>	0,729	6	3	1,2 %	0,977	0,988

5.6 Šestá iterace testů – tvar „T-křížek“

Po předchozích iteracích bylo zřejmé, že dvoubarevné markery jsou vhodnou cestou pro návrhu markeru. Oba dvoubarevné tvary markery dosahovaly skoro stejných výsledků a křížek byl jako jednodušší tvar preferován vůči hvězdě. Problém křížku (a také hvězdy) ale spočíval v tom, že není možné detekovat úhel rotace v intervalu 0° až 360°. Po rotaci o 90° již není v případě čtverce možné poznat, jaká rotace probíhá. V případě obdélníku je limit 180°. Kvůli tomu byla navržena úprava, jejímž cílem je zajistit plnohodnotnou detekci rotace. Základem je tvar křížku, jehož jedna část vedoucí do jednoho ze čtyř rohů je vynechána (Obrázek 17) – odtud název „T-křížek“. Díky této změně je již možné mít kompletní přehled o provedené rotaci v plném rozsahu 0° až 360°.

Trénink na stejné datové sadě a se stejným počtem učících epoch ukázal, že nový tvar má srovnatelné výsledky jako původní tvar s plným křížkem. Počty falešně pozitivních a negativních případů jsou podobné a hodnota IoU je přibližně o 4-5

setin nižší (Tabulka 9). „T-křížek“ je zvolen finálním tvarem markeru, protože má dobré výsledky a splňuje všechna tři definovaná kritéria:

- schopnost lokalizace markeru podle jeho středu,
- detekce úhlu rotace v rozlišení 360° relativně k pozorovateli z jeho tří linií,
- a detekce celého markeru pro získání relativní velikosti z jeho vyplněné oblasti.

Tabulka 9 Výsledky trénování pro „T-křížek“ tvar markeru s dvěma fixními barvami a s interiérovými obrázky pro 21 učicích epoch. [autor]

Tvar markeru	Průměrná IoU	FP	FN	Procent FN	PPV	Senzitivita
<i>T-křížek tloušťky 2</i>	0,689	6	10	3,9 %	0,997	0,962
<i>T-křížek tloušťky 1</i>	0,664	5	13	5,0 %	0,980	0,950



Obrázek 17 „T-křížek tloušťky 2“ [autor] **Obrázek 18** „T-křížek tloušťky 1“ [autor]

5.7 Sedmá iterace testů – větší obrázky

Všechny předchozí iterace testů byly prováděny na obrázcích, jejichž šířka byla 300 pixelů a výška byla dopočítána tak, aby došlo k zachování poměru stran obrázku. To je také důvod, proč jsou markery v obrázcích tak velké. Šířka markeru je náhodně generována v rozsahu od 5 do 60 pixelů a výška od 6 do 50 pixelů. Uvedené rozsahy jsou stejné pro všechny iterace testů, které byly dosud provedeny. Vzhledem k faktu, že finální tvar markeru byl již stanoven, vstupní obrázky jsou zvětšeny tak, že jejich šířka je 416 pixelů a výška je znovu dopočítána pro zachování poměru stran obrázku. Stanovená šířka přesně odpovídá velikosti,

na kterou jsou obrázky upraveny před vstupem do architektury neuronové sítě YOLOv3. Velikost generovaných markerů je stále ve stejném rozsahu, a tudíž jsou menší vzhledem k velikosti obrázků.

Trénink na stejných datových sadách (exteriér a interiér) umožnil porovnání s předchozími výsledky. Nové výsledky byly horší, ale to bylo po zvětšení obrázků očekáváno (Tabulka 10 a Tabulka 11). Předpokládá se, že přidáním dalších obrázků do datové sady společně se zvýšením počtu učicích epoch povede ke zlepšení výsledků na jejich předchozí úroveň. Pro srovnání byl proveden i trénink s předchozími tvary a ukázalo se, že všechny tvary vykazují podobné zhoršení výsledků. Lze tedy usoudit, že tvar „T-křížek“ nemá vliv na úroveň zhoršení.

Tabulka 10 Výsledky trénování pro „T-křížek“ tvar markeru s dvěma fixními barvami a s interiérovými obrázky se šířkou 416 pixelů. Trénování proběhlo pro 26 epoch. [autor]

Tvar markeru	Průměrná IoU	FP	FN	Procent FN	PPV	Senzitivita
<i>T-křížek tloušťky 2</i>	0,602	2	34	13,1 %	0,991	0,869
<i>T-křížek tloušťky 1</i>	0,570	3	39	15,0 %	0,987	0,850

Tabulka 11 Výsledky trénování pro „T-křížek“ tvar markeru s dvěma fixními barvami a s exteriérovými obrázky se šířkou 416 pixelů. Trénování proběhlo pro 26 epoch. [autor]

Tvar markeru	Průměrná IoU	FP	FN	Procent FN	PPV	Senzitivita
<i>T-křížek tloušťky 2</i>	0,699	1	26	5,6 %	0,998	0,944
<i>T-křížek tloušťky 1</i>	0,686	6	26	5,6 %	0,986	0,944

5.8 Osmá iterace testů – větší datová sada

Osmá iterace byla provedena s datovou sadou, která kombinuje všechny předchozí sady. Nová větší sada se skládá z 5 075 trénovacích, 1 449 validačních a 721 testovacích obrázků. Celkem se jedná o 7 245 obrázků. Trénink proběhl s obrázky, které jsou zmenšeny na 416 pixelů v šířce.

Testování pro obě varianty tvaru „T-křížek“ prokázalo, že zvětšení datové sady má pozitivní vliv na výsledky (Tabulka 12), které jsou porovnatelné s tréninkem s obrázky o šířce 300 pixelů. Obě varianty dosáhly PPV nad 0,98. Počet falešně

negativních případů byl kolem pěti procent. Průměrné hodnoty IoU byly 0,721 a 0,746 pro obě varianty. Obrázky s interiérovými scénami byly ručně zkontrolovány a poměr falešně negativních případů byl stejný jako ve zbytku sady.

Tabulka 12 Výsledky trénování pro „T-křížek“ tvar markeru s dvěma fixními barvami na kombinované datové sadě 7 248 obrázků, jejichž šířka je 416 pixelů. [autor]

Tvar markeru	Počet epoch	Průměrná IoU	FP	FN	Procent FN	PPV	Senzitivita
<i>T-křížek tloušťky 2</i>	37	0,721	7	41	5,7 %	0,988	0,943
<i>T-křížek tloušťky 1</i>	39	0,746	3	31	4,3 %	0,996	0,994

5.9 Devátá iterace testů – největší datová sada

Poslední trénink s uměle generovanými markery byl proveden na dále zvětšené datové sadě. Celkových 10 697 obrázků bylo rozděleno na 7 560 trénovacích, 2 058 validačních a 1 079 testovacích. Testování ukázalo, že další zvětšování sady má pozitivní vliv na výsledky (Tabulka 13).

Dále byl proveden i trénink s tvarem, jehož vnitřní čáry měly náhodnou šířku jedna nebo dvě. Jedná se o přiblížení reálnějším situacím a je tedy důležité, že výsledky jsou srovnatelné s předchozími.

Tabulka 13 Výsledky trénování pro „T-křížek“ tvar markeru s dvěma fixními barvami na kombinované datové sadě 10 697 obrázků, jejichž šířka je 416 pixelů. [autor]

Tvar markeru	Počet epoch	Průměrná IoU	FP	FN	Procent FN	PPV	Senzitivita
<i>T-křížek tloušťky 2</i>	35	0,780	5	33	3,1 %	0,995	0,969
<i>T-křížek tloušťky 1</i>	35	0,743	10	62	5,8 %	0,990	0,943
<i>T-křížek s náhodnou tloušťkou</i>	31	0,762	5	40	3,8 %	0,995	0,963

Celkové výsledky jsou uspokojivé. PPV je ve všech testech vyšší nebo rovna 0,99. Falešně pozitivní detekce se téměř nevyskytují, ale množství falešně negativních

detekcí se zdá být obtížné dostat pod čtyři procenta. Je to problém, který bude nutné dále řešit. Průměrná IoU se pohybovala v rozsahu od 0,74 do 0,78, což znamená excelentní přesnost detekcí.

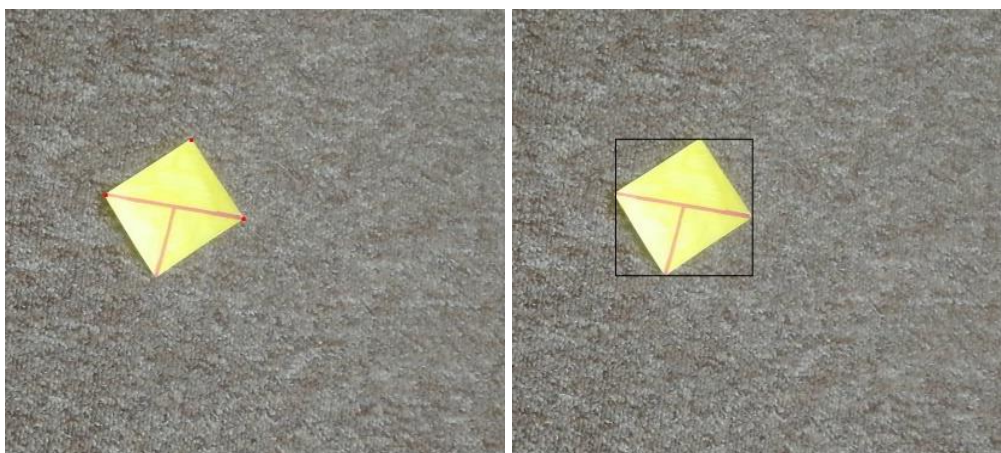
5.10 Desátá iterace testů – markery v reálných scénách

Poslední testy byly provedeny s reálnými markery v reálných scénách. Celkem 18 markerů složených z devíti různých barev bylo ručně nakresleno. Dále bylo nasnímáno video každého nakresleného markeru. Pět videí bylo natočeno ve venkovním prostředí a 13 v interiéru. Venkovní videa byla natočena za běžných světelných podmínek obyčejného letního dne kolem 11. hodiny dopoledne. Počasí bylo jasné s drobným oparem, takže sluneční světlo nebylo přímé. Všechna videa byla natočena na běžný mobilní telefon (Honor 9) v rozlišení 1280 × 720 pixelů s 30 snímky za sekundu. Z natočených videí bylo celkem získáno 550 obrázků – byl vybrán každý desátý snímek z natočených videí. Snímky neobsahující žádný marker byly vyřazeny a každý z 550 obrázků tedy obsahoval právě jeden marker.

Provedení testu s reálnými markery, které jsou získány z videí, bylo důležité, protože se jedná o běžný scénář, ve kterém se detekce markeru očekává. Přibližně polovina obrázků byla mírně rozmazána vlivem rychlejšího pohybu kamery, který byl záměrný pro simulaci této situace. Malé množství obrázků mělo velké rozmazání, jehož délka byla přibližně polovina rozměru markeru. Jednalo se tedy o náročné podmínky, které se však v reálných situacích mohou často vyskytovat, a proto je bylo nutné otestovat. Pro tvar „T-křížku“ se projevila jedna z jeho dalších možných výhod v situacích s rozmazaným obrazem. Tvar obsahuje dvě čáry, které se protínají v úhlu blízkém 90°, což znamená, že při rozmazání v libovolném směru je jedna z čar stále dostatečně ostrá, jak ilustruje Obrázek 22.

Bounding boxy pro všech 550 markerů byly ručně vyznačeny. Při procesu tagování byla využita metodologie, kterou navrhli Papadopoulos a kolektiv [102]. Metoda spočívá v tom, že se klikáním označí části objektu, které jsou nejvíce nahoře, dole, vpravo a vlevo. V případně obdélníkového markeru se jedná o jeho rohy, jak ilustruje Obrázek 19. Autoři metodologie uvádějí, že tato metoda má být rychlejší než přetahování připravených linií kolem tagovaného objektu.

Sada 550 obrázků byla rozdělena na 385 trénovacích, 110 validačních a 55 testovacích obrázků. Proces trénování byl spuštěn pro 50 učicích epoch s nejlepšími výsledky kolem 40. epochy. Podrobný popis a interpretace výsledků se nachází v následující kapitole 6, která shrnuje výsledky návrhu a prvního cíle disertační práce.

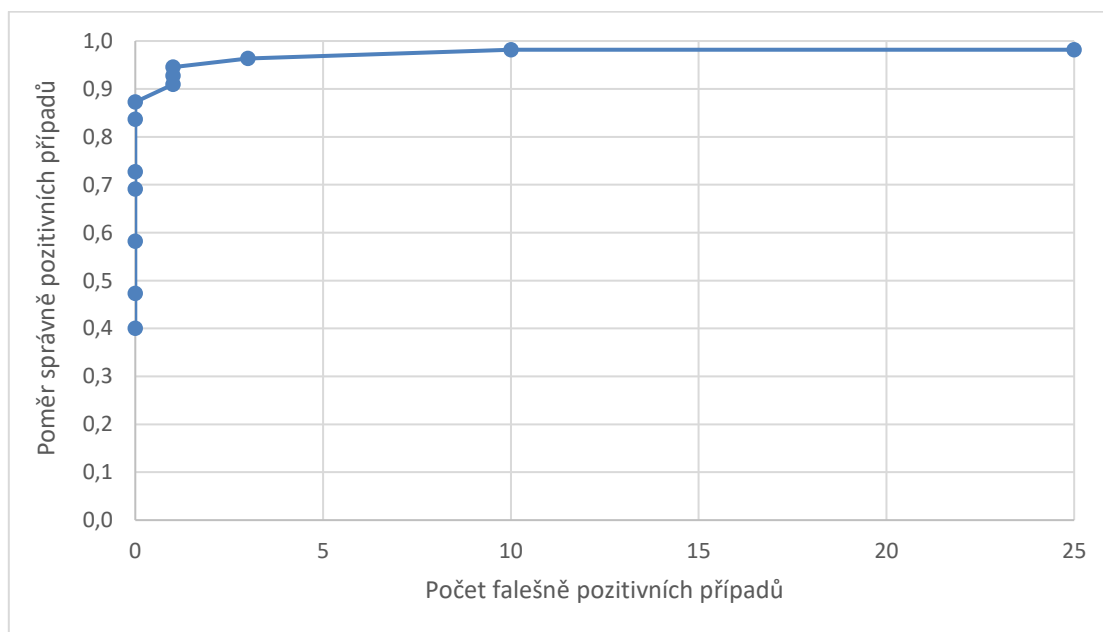


Obrázek 19 Ilustrace procesu tagování (a) vlevo tři červené tečky umístěny pro označení horního, levého a pravého rohu (b) vpravo jsou po umístění dolního bodu všechny body nahrazeny ohraničujícím boxem. [autor]

6 Shrnutí výsledků návrhu designu markeru

Uvedené testy představily, že je možné navrhnout marker, který splňuje definovaná kritéria a má vhodné vlastnosti pro kvalitní detekci. Navržený marker „T-křížek“ umožňuje spolehlivou detekci v rozmazaném obrazu a zároveň i možnost detekce orientace. Kombinace zmíněných vlastností se v existujících prozkoumaných systémech nevyskytuje (podrobnější informace v kapitole 4.4 Celkové shrnutí). Dále je navržený marker využitelný pro detekci pozice a velikosti. Díky svému návrhu a použité architektuře neuronové sítě je marker možné nejen tisknout, ale také pouze kreslit rukou.

Úspěšnost detekcí desáté iterace s reálnými markery ukázala, že zvolený finální marker „T-křížek“ představuje správný směr. Proběhlo také vyhodnocení s různými pravděpodobnostními prahy, které určují, jaké výstupní bounding boxy budou považovány za validní a jaké budou zamítnuty (více informací v kapitole 3.4 Parametry neuronové sítě). Výchozí hodnotou je 0,3 a nejlepších výsledků bylo dosaženo s hodnotou 0,25 (Tabulka 14).



Obrázek 20 ROC křivka pro různé pravděpodobnostní prahy detekce v rozsahu od 0,05 do 0,7 [autor]

Tabulka 14 Výsledky trénování pro „T-křížek“ tvar markeru s dvěma fixními barvami na kombinované datové sadě 10 697 obrázků, jejichž šířka je 416 pixelů. [autor]

Práh	Průměrná IoU	TP	FP	FN	Procent FN	PPV	Senzitivita
0,05	0,710	54	127	1	1,8 %	0,298	0,982
0,10	0,710	54	25	1	1,8 %	0,684	0,982
0,15	0,707	54	10	1	1,8 %	0,844	0,982
0,20	0,692	53	3	2	3,6 %	0,946	0,964
0,25	0,685	52	1	3	5,5 %	0,981	0,945
0,30	0,673	51	1	4	7,3 %	0,981	0,927
0,35	0,659	50	1	5	9,1 %	0,980	0,909
0,40	0,628	48	0	7	12,7 %	1,000	0,873
0,45	0,600	46	0	9	16,4 %	1,000	0,836
0,50	0,524	40	0	15	27,3 %	1,000	0,727
0,55	0,501	38	0	17	30,9 %	1,000	0,691
0,60	0,416	32	0	23	41,8 %	1,000	0,582
0,65	0,332	26	0	29	52,7 %	1,000	0,473
0,70	0,280	22	0	33	60,0 %	1,000	0,400

Při použití vhodného prahu detekce jsou téměř eliminovány falešně pozitivní detekce. Počet falešně negativních detekcí se daří držet na nízkých hodnotách kolem 5 %, zatímco i počet falešně pozitivních detekcí je stále nízký. Hodnota IoU se blíží hodnotě 0,7, takže přesnost detekce je velmi dobrá.

6.1 Výhody a nevýhody

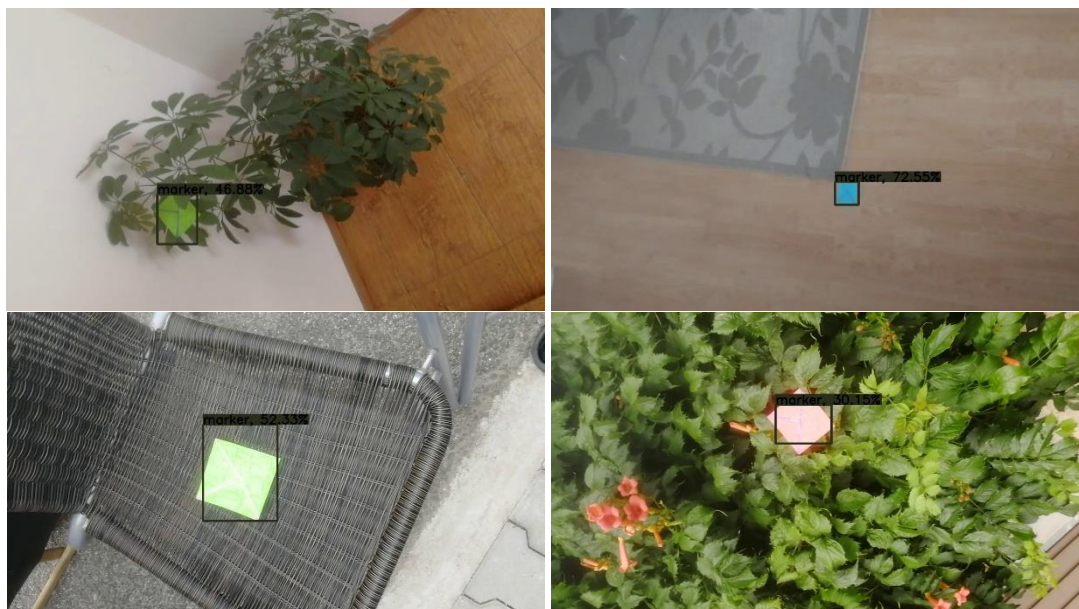
Hlavní předností systému je marker („T-křížek“), který je navržen takovým způsobem, aby z něj bylo možné detekovat pozici, orientaci a velikost, zatímco může být zároveň výrazně rozmazaný vlivem pohybu snímané kamery.

Test provedený s reálnými markery prokázal, že navržený systém je schopen spolehlivě pracovat jak ve vnitřním, tak ve venkovním prostředí, a detekovaný marker může být i nakreslen ručně. Výsledky dále ukázaly, že detekce je přesná a s minimálním počtem falešně pozitivních i falešně negativních případů.

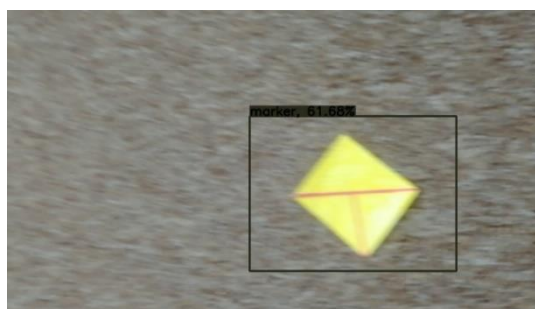
Nevýhodou systému je v současné době pomalejší detekce neuronovou sítí. Průměrná doba zpracování jednoho obrázku je 44 milisekund na grafické kartě NVIDIA GTX 1060. Ačkoliv se stále jedná o detekci v reálném čase, tak je nutné podotknout, že mnohé jiné systémy poskytují rychlejší schopnost detekce.

Druhou nevýhodou je neschopnost systému pracovat s částečným zakrytím markeru. Žádná z navržených datových sad nesimulovala zakrytí markeru, a tedy ani výsledná natrénovaná neuronová síť neumí se zakrytím pracovat. Je přepokládáno, že popsany problém může eliminován návrhem datové sady, která obsahuje i markery, které jsou částečně zakryté.

6.2 Příklady detekcí



Obrázek 21 Příklady detekovaných markerů, které byly ručně nakresleny a umístěny v reálných scénách – horní dva obrázky jsou v interiéru, dolní dva v exteriéru. [autor]



Obrázek 22 Příklad detekce markeru v obrázku se středním rozmazáním způsobeným prudkým pohybem kamery. Obrázek dobře ilustruje výhodu tvaru „T-křížek“ – jedna čára je stále téměř ostrá. [autor]



Obrázek 23 Příklady všech čtyř falešně negativních případů při použití hodnoty 0,3 jako pravděpodobnostního prahu detekce. Při použití prahu o hodnotě 0,2 jsou markery v horních dvou obrázcích korektně detekovány. [autor]



Obrázek 24 Příklad jediné falešně pozitivní detekce při použití hodnoty 0,3 jako pravděpodobnostního prahu detekce. Část koberce v pravé horní části obrázku je mylně detekována jako marker. [autor]

7 Způsoby detekce navrženého markeru

Detekce objektů je jednou z hlavních oblastí počítačového vidění. Člověk je v obrazu schopen téměř instantně rozeznat ohromné množství různých objektů v různých situacích. Pro počítač je to ale obtížný problém, protože i objekty stejné kategorie se mohou od sebe velmi lišit (lidé, zvířata, budovy atp.). V minulosti bylo různými autory vytvořeno mnoho algoritmů a systémů řešících problém detekce a kategorizace objektů v obrazu. Mezi starší metody patří například „Haar features“ [103], SIFT¹ [15], HOG² [104] a DPM³ [105].

V posledních dvou dekadách se v počítačovém vidění začaly široce uplatňovat také umělé neuronové sítě. První principy výpočtových modelů založených na neuronových sítích byly popsány již v 60. letech minulého století. Pravděpodobným zlomovým bodem se stala práce Rumelharta a kolektivu [106], kteří představili rychlý způsob výpočtu tzv. zpětné propagace, který je dodnes základní technikou procesu učení umělých neuronových sítí. Mezi nejznámější metody využívající neuronové sítě pro detekci a kategorizaci objektů v obrazu patří R-CNN⁴ [107]–[109], YOLO⁵ [6], [110], [111] a SSD⁶ [112].

Kapitola nejprve popisuje využití neuronových sítí pro tvorbu detektoru. Dále jsou pak popsány vlastní algoritmy pro detekce markeru v obrazu. Tyto algoritmy využívají programovatelné shadery grafických karet.

7.1 Využití umělých neuronových sítí

Pro návrh designu markeru byla využita architektura neuronové sítě YOLOv3 [6]. Tato architektura je založena zejména na tzv. konvolučních vrstvách, u kterých se obecně ukazuje, že pracují dobře s obrazovými daty.

Vstupem pro učení jsou celé obrázky a textový soubor, kde každý řádek reprezentuje jeden obrázek s jeho bounding boxy a kategoriemi objektů, které se

¹ SIFT z anglického „Scale-Invariant Features“

² HOG z anglického „Histograms of Oriented Gradients“

³ DPM z anglického „Deformable Parts Models“

⁴ R-CNN z anglického „Regions with Convolutional Neural Network features“

⁵ YOLO z anglického „You Only Look Once“

⁶ SSD z anglického „Single Shot Detector“

v definovaných bounding boxech nachází. Na výstupu neuronové sítě je seznam bounding boxů, kde každý z nich obsahuje souřadnice v obrázku, kategorii objektu a pravděpodobnost s jakou je detekovaný objekt klasifikován.

7.1.1 Parametry sítě, rychlost a paměťová náročnost

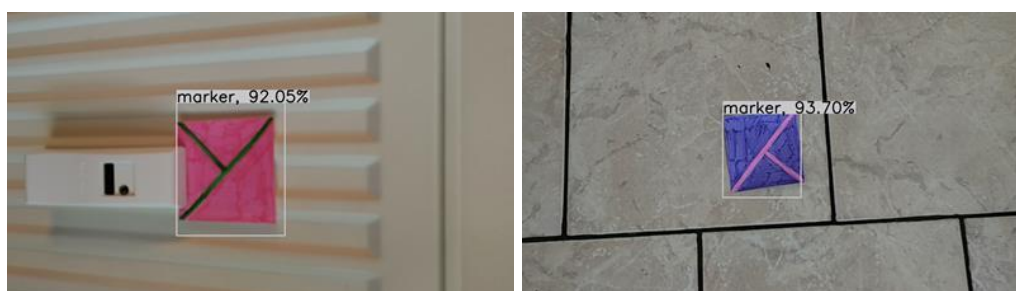
Vzhledem k výpočetní a paměťové náročnosti trénování byla zvolena výchozí velikost obrázku 416×416 pixelů. Výsledná neuronová síť má 62 milionů parametrů. Její trénování pro 2331 obrázků trénovací sady a 333 obrázků validační sady trvalo 36 minut na grafické kartě NVIDIA GeForce RTX 2080.

Pro reálné použití je však důležitá časová a paměťová náročnost při provozu. Popsaná síť dokáže zpracovat jeden obrázek v daném rozlišení 416×416 pixelů za 44 milisekund na grafické kartě NVIDIA GeForce GTX 1060. To umožňuje zpracování zhruba 23 snímků za sekundu (FPS). Případně větší obrázek je zmenšen na požadovanou velikost.

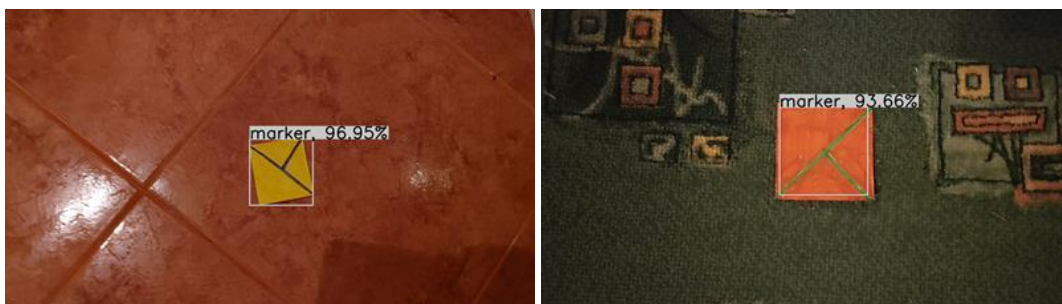
Z hlediska paměťové náročnosti vyžaduje spuštění programu a nahrání sítě 2 GB standardní operační paměti a 490 MB paměti grafické karty. Uložená neuronová síť na disku zabírá 330 MB.

7.1.2 Příklady detekcí

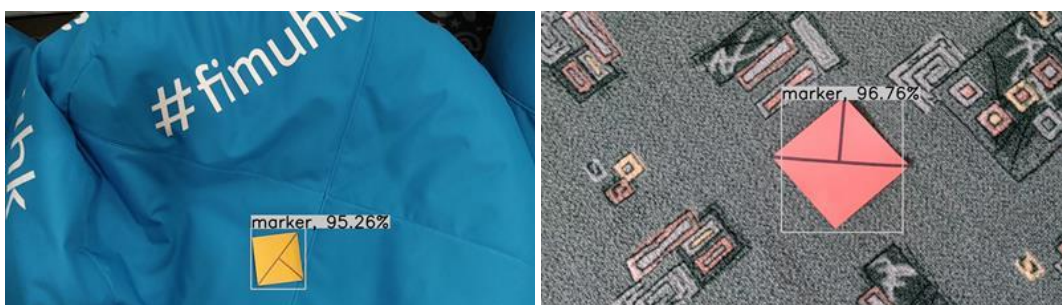
Podkapitola prezentuje několik skupin příkladových detekcí. První skupina jsou detekce ručně kreslených markerů v interiéru, dále následuje skupina ručně kreslených markerů snímaných pod umělým osvětlením. Třetí je skupina tištěných markerů snímaných v interiéru a následně v exteriéru. Pátá (poslední) skupina jsou vybrané markery snímané se speciálními okolnostmi.



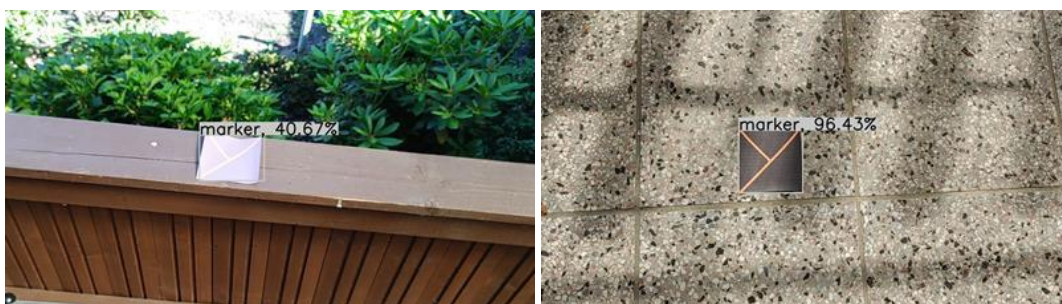
Obrázek 25 Příklady detekcí neuronovou sítí pro ručně kreslené markery snímané v interiéru [autor]



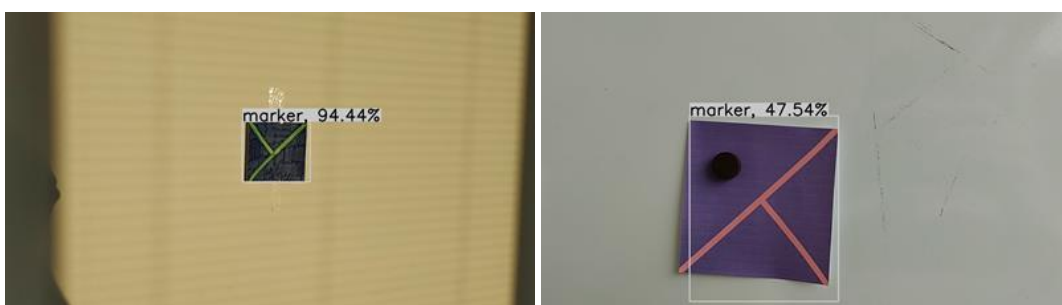
Obrázek 26 Příklady detekcí neuronovou sítí pro ručně kreslené markery snímané v interiéru za umělého osvětlení [autor]



Obrázek 27 Příklady detekcí neuronovou sítí pro tištěné markery snímané v interiéru [autor]



Obrázek 28 Příklady detekcí neuronovou sítí pro tištěné markery snímané v exteriéru [autor]



Obrázek 29 Příklady detekcí neuronovou sítí. Vlevo ručně kreslený marker snímaný v interiéru, jehož barva je tmavě modrá, avšak pod přímým slunečním světlem je téměř černá. Vpravo tištěný marker, který je na magnetickou tabuli přichycen magnetem. [autor]

7.2 Algoritmus rozdělení obrazu na části

Zpracování celého obrazu najednou je zpravidla výpočetně náročně. Princip prvního zkoumaného algoritmu tedy spočívá v rozdělení vstupního obrazu na části, aby došlo ke zrychlení zpracování. Celý algoritmus je rozdělen na dva průchody renderovací pipeline.

Kapitola nejprve uvádí popis algoritmu. Dále je uveden rozbor časové complexity, a následně konkrétní naměřené časy rychlosti detekce algoritmu.

7.2.1 Princip algoritmu

V prvním průchodu je vstupní obraz ve fragment shaderu rozdělen na čtvercové oblasti o 16 pixelech (4×4). V každé oblasti je zjištěn počet pixelů s cílovou barvou a tato hodnota je společně se souřadnicemi levého horního pixelu skupiny zapsána do každého pixelu skupiny. Při uvažování RGB barevného modelu na výstupu je jako červená složka pixelu vložen zjištěný počet, jako zelená složka je vložena souřadnice X a jako modrá složka je vložena souřadnice Y. Postup ilustruje následující Obrázek 30.

Y\X	12	13	14	15	16	17	18	19	20	21	22	23
36	4,12,36				5,16,36				1,20,36			
37												
38												
39												
40	7,12,40				15,16,40				0,20,40			
41												
42												
43												
44	0,12,44				3,16,44				2,20,44			
45												
46												
47												

Obrázek 30 Ilustrace prvního průchodu grafickou pipeline při detekci zeleného markeru v rámci algoritmu rozdělení obrazu na části. Pro lepší čitelnost jsou na výstupu vyplněny pouze hodnoty prvního pixelu ve skupině, ačkoliv reálně jsou tyto hodnoty vyplněny do všech pixelů skupiny. [autor]

Jsou dva důvody, proč je stejná informace zapsána do 16 pixelů zároveň. Prvním je povaha paralelního zpracování, kdy není možné kontrolovat pořadí zpracování jednotlivých pixelů fragmentovým programem. Druhým důvodem je odlišné chování napříč zařízeními při zmenšování výstupního obrazu, protože výstupní textura má čtvrtinové rozměry v obou dimenzích a různá zařízení vzorkují obrazové body z různých pozic na výstup. Při testování nebyl prokázán žádný vliv na výkon a výhodou je jednotné chování napříč všemi zařízeními.

Po dokončení prvního průchodu grafickou pipeline okamžitě navazuje průchod druhý. Zde je vstupem výstup prvního průchodu, jehož rozlišení je navíc sníženo na jednu šestnáctinu – 4 pixely v řádku a 4 pixely ve sloupci, jak výše ilustruje Obrázek 30. Proces je podobný jako u prvního průchodu, liší se způsobem výpočtu výsledných hodnot pro zapsání do pixelu. Funguje ve skupinách po devíti pixelech (3×3), jak znázorňuje následující Obrázek 31.

		3			4			5					
Y\X	r	g	b	r	g	b	r	g	b	POČET	X	Y	
9	4	12	36	5	16	36	1	20	36	36	540	1424	
10	7	12	40	15	16	40	0	20	40		15,0	39,6	
11	0	12	44	3	16	44	2	20	44				

Obrázek 31 Ilustrace druhého průchodu grafickou pipeline v rámci algoritmu rozdělení obrazu na části. Levá část reprezentuje vstup pro ilustrovanou skupinu pixelů. Pravá část reprezentuje vypočítaný výstup. [autor]

Ze skupin, které obsahují alespoň dva zájmové pixely (v ilustraci zelené pixely), se vypočítá vážený aritmetický průměr na základě počtu nalezených zájmových pixelů a jejich souřadnic. Pro předchozí obrázek (Obrázek 30) jsou výsledné souřadnice $X = 15,0$ a $Y = 39,6$ pro 36 zájmových pixelů (v pravé horní skupině je pouze jeden, takže je skupina ignorována). Tyto hodnoty se zapisují do všech devíti pozic zpracovávané skupiny ze stejných důvodů, jaké byly popsány výše při vysvětlení prvního průchodu.

Během zpracování je rozlišení obrázku zmenšeno během obou průchodů a výsledná textura je tedy celkem 144krát menší. Hlavním důvodem pro co nejvyšší

redukci rozlišení, a tedy celkového počtu pixelů, je poslední krok, kdy je nutné data přenést zpět z paměti grafické karty. Testování ukázalo, že tato operace kopírování trvá přibližně 90 % celkového času zpracování a čas se zvyšuje s množstvím pixelů, které je nutné přenést do jiné paměti. Po překopírování dat již dojde pouze k finálnímu nalezení oblasti s nejvíce zájmovými pixely a odpovídající souřadnice jsou souřadnice středu markeru.

7.2.2 Časová komplexita

Časová komplexita z hlediska počtu operací prováděných ve fragment shaderech obou průchodů renderovací pipeline je konstantní. V obou průchodech je proveden vždy stejný počet operací bez ohledu na to, jak velký je vstupní obrázek. Jedná se však o paralelní algoritmus a je tedy nutné započítat také jeho cenu, která vyjadřuje počet potřebných výpočetních jader. Aby tedy bylo možné spustit všechny výpočty pro všechny skupiny zároveň, tak je vyžadován takový počet výpočetních jader, jaký je počet skupin. V prvním kroku je počet roven počtu pixelů obrázku. V druhém kroku je počet 16krát menší. V obou případech se jedná o lineární cenu. Časová komplexita algoritmu je tedy $\mathcal{O}(n)$. Avšak nakonec je nutné ještě zpracovat informace v programu na CPU, aby byl získán výsledek. I tato část má časovou komplexitu $\mathcal{O}(n)$.

7.2.3 Rychlost detekce

Byla vytvořena referenční implementace a algoritmus byl průběžně testován z hlediska rychlosti detekce. Na závěr bylo provedeno podrobné měření, které je rozděleno na dvě části. Přesné měření s využitím WebGL rozšíření (více v 3.2 Měření času zpracování algoritmů na GPU) bylo aplikováno na desktopu. Méně přesné měření pak bylo použito jak pro desktop, tak i pro mobilní zařízení. Testování bylo provedeno na čtyřech různých zařízeních:

- Notebook s grafickým čipem Intel UHD 630, Windows 10
- Notebook s grafickým čipem NVIDIA GeForce GTX 1060, Windows 10
- Mobilní telefon Honor 9 s čipem Kirin 960 a GPU Mali-G71 MP8, Android 7
- Mobilní telefon Nokia 8.3 5G s čipem Snapdragon 765G a GPU Adreno 620, Android 12

Tabulka 15 Naměřené časy běhu renderovací smyčky jednotlivých částí detekce markeru při použití algoritmu rozdělení obrazu na části a rozlišení obrazu 1280 × 720 pixelů. [113]

Zařízení	1. krok	2. krok	Celkem
<i>Intel UHD 630</i>	0,61 ms	0,11 ms	0,72 ms
<i>NVIDIA GTX 1060</i>	0,20 ms	0,04 ms	0,24 ms

Jak bylo vysvětleno v metodologii měření, přesně měření není dostupné pro mobilní zařízení. Je tedy provedeno alternativní porovnání, kde rozhodující čas je dokončení čtení spočítané informace z GPU. Je nutné zdůraznit, že časy nejsou porovnatelné s měřením přímo na GPU. Avšak poskytují alternativní možnost porovnat desktopová zařízení s mobilními na stejné metrice.

Tabulka 16 Naměřené CPU časy běhu jednotlivých částí detekce markeru při použití algoritmu rozdělení obrazu na části a rozlišení obrazu 1280 × 720 pixelů. [autor]

Zařízení	1. krok	2. krok	Čtení	Celkem
<i>Intel UHD 630</i>	0,17 ms	0,09 ms	5,02 ms	5,28 ms
<i>NVIDIA GTX 1060</i>	0,08 ms	0,02 ms	3,50 ms	3,65 ms
<i>Honor 9</i>	0,59 ms	0,50 ms	4,22 ms	5,31 ms
<i>Nokia 8.3 5G</i>	0,58 ms	0,45 ms	5,39 ms	6,42 ms

Z tabulky je možné vidět, že nutnost přečtení výsledku z GPU je značnou nevýhodou algoritmu. Jediný způsob, jak čtení zrychlit, by bylo snížit množství dat, která jsou kopírována z grafické karty, případně ho úplně eliminovat. Zároveň však snížení množství dat snižuje celkové rozlišení algoritmu pro nalezení markeru. Bylo tedy nutné najít kompromis mezi oběma protichůdnými požadavky.

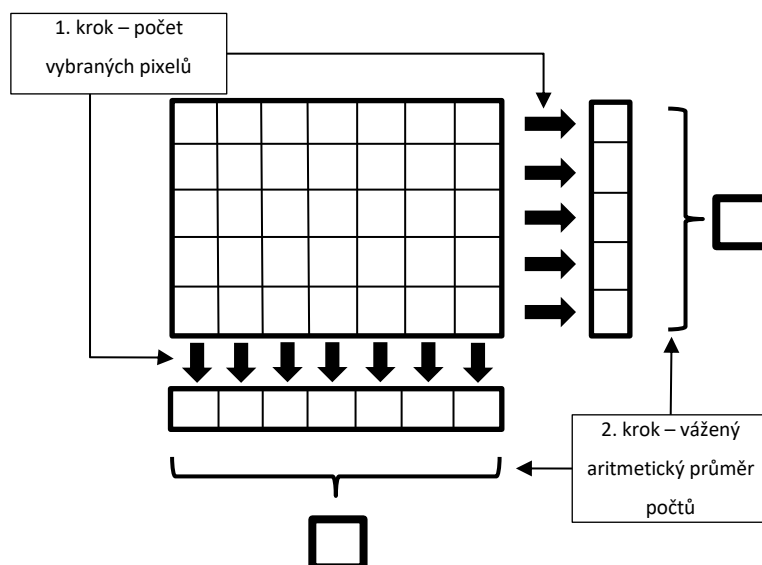
Výše uvedená tabulka (Tabulka 16) obsahuje měření pro stav, který byl zvolen jako ideální, kdy je detekce markerů stále spolehlivá a rychlost detekce je stále přijatelná. Vstupní rozlišení obrazu kamery bylo pro testování nastaveno na 1280 × 720 pixelů a algoritmus zmenší obraz nejprve čtyřikrát a poté třikrát v každém z rozměrů. Výsledkem je obraz o rozměru 107 × 60 pixelů obsahující celkem 6 420 pixelů, které je nutné načíst z paměti GPU.

7.3 Algoritmus skenování řádků a sloupců

Podkapitola také nejprve uvádí popis algoritmu a následně představuje dílčí vylepšení části algoritmu zjišťování zájmových pixelů. Popis je následován porovnáním s předchozím algoritmem, zejména s důrazem na výhody druhého řešení. Následně je uveden rozbor časové komplexity, na který navazují konkrétní naměřené časy výkonu algoritmu a rychlosti detekce. Nakonec je uvedeno několik ukázkových detekcí.

7.3.1 Princip algoritmu

Algoritmus skenování řádků a sloupců se také skládá ze dvou průchodů grafickou pipeline (Obrázek 32). V prvním kroku se pro každý řádek a každý sloupec zjistí počet vybraných pixelů. Pixely obrázku jsou vybírány podle podobnosti jejich odstínu s předem definovaným odstínem značky, která je hledána. Počet zjištěných pixelů v každém řádku a sloupci vytváří výstup prvního kroku.



Obrázek 32 Ilustrace algoritmu skenování řádků a sloupců [autor]

V rámci druhého průchodu jsou na vstupu dvě jednorozměrná pole počtů zájmových pixelů vybraných ve všech řádcích a sloupcích. Všechny hodnoty těchto polí se zvlášť využijí k výpočtu váženého aritmetického průměru. Souřadnice daného sloupce nebo řádku je vážena počtem vybraných zájmových pixelů.

Výstupem druhého kroku jsou pouze dva pixely, z nichž každý obsahuje ve svých čtyřech složkách RGBA následující informace pro danou osu (X nebo Y):

- souřadnice markeru jako složka R,
- složka G jako součet využitý při výpočtu váženého aritmetického průměru jako dělenec,
- složka B jako váha všech zájmových pixelů využita jako dělitel při výpočtu váženého průměru,
- a počet všech zájmových pixelů jako složka A.

7.3.2 Specifická vylepšení algoritmu

Při zpracování obrazu byla testována a aplikována celá řada dalších vylepšení s cílem zkvalitnění výsledku. Pro tento účel byla vytvořena speciální datová sada přesně vhodná pro dané testování. Datová sada použita pro finální porovnání, které je v této práci popsána, není pro takové porovnání vhodná a došlo tedy pouze k převzetí předchozích výsledků z [114].

7.3.2.1 Úprava práce s odstínem a ostatními složkami HSV modelu

Jak bylo vysvětleno výše, princip algoritmu tkví ve výpočtu váženého aritmetického průměru souřadnic zájmových pixelů a jejich počtu. Nejedná se o prostý počet pixelů daného odstínu, ale započítávají se také pixely s blízkým okolím. Referenční implementace využívá toleranci 20 stupňů oběma směry při předpokladu rozsahu odstínu 0° až 360°. Avšak namísto prostého počtu zájmových pixelů se ukázalo být lepší udělovat pixelům váhu podle toho, jak se jejich odstín liší od cílového. Pokud je tedy odstín pixelu ve správném rozsahu, jeho váha se spočítá následujícím nelineárním výpočtem:

$$\text{váha souřadnice pixelu} = \sqrt{\frac{\text{tolerance} - \text{abs}(\text{cílový odstín} - \text{aktuální odstín})}{\text{tolerance}}}$$


```

float hueWeight(float targetHue, float currentHue) {
    float diff = abs(targetHue - currentHue);
    float ratio = (hueThreshold - diff) / hueThreshold;
    float weight = pow(ratio, 0.5);
    return weight; // in range <0;1>
}

```

Ukázka kódu 1 Referenční implementace vážení odstínu barvy v GLSL (OpenGL Shading Language) fragment shaderu [autor]

Vylepšena byla také práce s ostatními složkami HSV barevného modelu – složky S (saturace, anglicky „saturation“) a V (intenzita, anglicky „value“). Proběhl experiment s možností dynamického zjišťování daných hodnot podobně jako pro hodnotu odstínu. Avšak již v předběžném testování se ukázalo, že s hodnotami nelze pracovat tímto způsobem, protože detekce je pak velmi chybová, nestabilní a až nepoužitelná. Důvodem jsou pravděpodobně změny v prostředí, které se při snímání neustále dějí. Kamery se snaží tyto změny kompenzovat a upravovat výstupní obraz tak, aby byl stále barevně stabilní. To však ale znamená, že hodnoty saturace a intenzity jsou kolísavé i v rámci jednoho snímání. Namísto dynamické hodnoty tedy bylo provedeno testování za účelem získání vhodné hodnoty hranice, která se pro obě hodnoty ukázala být kolem 0,3 (při předpokladu rozsahu obou hodnot od 0 do 1). Výsledky popisuje následující Tabulka 17.

Tabulka 17 Tabulka obsahuje výsledky testování různých prahových hodnot saturace a intenzity barevného modelu HSV. Výsledky ukazují, že práh kolem 0,3 poskytuje nejlepší detekci. [114]

Práh pro hodnoty saturace a intenzity	PPV	Průměrná vzdálenost od středu	Medián vzdálenosti od středu	FP	FN
<i>0,20</i>	48,6 %	4,08 px	1,65 px	72	0
0,25	56,4 %	3,83 px	1,76 px	61	0
0,30	56,4 %	4,11 px	1,81 px	60	1
0,35	56,4 %	5,37 px	2,16 px	59	2
<i>0,40</i>	52,1 %	5,99 px	2,64 px	58	9
<i>0,45</i>	46,4 %	5,72 px	2,43 px	55	20
<i>0,50</i>	51,4 %	5,87 px	3,33 px	30	38

7.3.2.2 Operace dilatace a eroze obrazu

Navržený algoritmus neimplementuje operace dilatace a eroze plně, ale jejich princip je využit při výpočtu váhy pixelu. Jak bylo popsáno výše, váha souřadnice pixelu se vypočítá jako rozdíl jeho odstínu od cílového odstínu. Vstupní obraz však obvykle obsahuje mnoho chyb, jako jsou malé otvory uvnitř oblasti markeru a malé oblasti „správně“ zbarvených pixelů, které nejsou součástí markeru. Tyto operace se provádějí za účelem vyhlazení případných děr v markeru a penalizací jednotlivých pixelů podobného odstínu. Aby byl ušetřen výpočetní čas, operace nejsou prováděny samostatně, ale jsou sloučeny s prvním krokem algoritmu. Obě operace jsou otestovány samostatně a je provedeno i kombinované testování.

Operace dilatace

Je-li váha pixelu nenulová, tak se operace dilatace neprovádí. Pokud je váha pixelu nulová, tak algoritmus zjistí váhu všech osmi okolních pixelů a průměrnou váhu nenulových hodnot použije jako výstup zpracovávaného pixelu. Druhá varianta pracuje na výstupu s funkcí maxima namísto průměru.

$$weight = \begin{cases} weight; & \text{if } weight \neq 0 \\ \frac{\sum_{k=1}^8 w_k}{\sum_{k:w_k \neq 0} 1}; & \text{if } weight == 0 \end{cases}$$

Operace eroze

Operace eroze se provádí pouze pokud je váha odstínu zpracovávaného pixelu nenulová. Princip spočívá v tom, že pokud okolní pixely nemají dostatečnou váhu z hlediska odstínu, váha zpracovávaného pixelu je vynulována. Byly provedeny dvě varianty testů. První používá pouze počet sousedních pixelů, které mají odstín v rozsahu tolerance odstínu. Tato varianta neměla tak dobrý účinek jako druhá varianta, která pracuje s vážením odstínu sousedních pixelů. Důvodem je pravděpodobně fakt, že počet zahrne i pixely, jejichž odstín je méně podobný cílovému odstínu, zatímco použití vážení zajistí, že jsou zahrnuty pouze pixely, které mají sousední pixely shodnější s cílovým odstínem.

$$weight = \begin{cases} 0; & \text{if } weight == 0 \\ 0; & \text{if } weight \neq 0 \text{ and } \frac{\sum_{k=1}^8 w_k}{\sum_{k:w_k \neq 0} 1} < X \\ weight; & \text{if } weight \neq 0 \text{ and } \frac{\sum_{k=1}^8 w_k}{\sum_{k:w_k \neq 0} 1} \geq X \end{cases}$$

Ukázka kódu 2 obsahuje ukázkou referenční implementace popsaných operací. Implementace je provedena v GLSL syntaxi v rámci fragment shaderu.

```
float weight = getPixelWeight(texCoords);
// do dilatation operation
if (weight == 0.0) {
    float weights[8];
    readNeighborPixels(texCoords, weights);
    float sum = 0.0;
    float count = 0.0;
    for (int i = 0; i < 8; i++) {
        if (weights[i] != 0.0) {
            sum += weights[i];
            count++;
        }
    }
    weight = count == 0.0 ? 0.0 : sum / count;
}
// do erosion operation
if (weight != 0.0) {
    float weights[8];
    readNeighborPixels(texCoords, weights);
    float sum = 0.0;
    for (int i = 0; i < 8; i++) sum += weights[i];
    if (sum < 6.5) weight = 0.0;
}
sum += weight;
```

Ukázka kódu 2 Referenční implementace operací dilatace a eroze při zpracování jednoho pixelu v daném řádku nebo sloupci (GLSL fragment shader). Referenční implementace využívá práh 6,5 pro zamítnutí nebo potvrzení váhy pixelu v případě eroze. [autor]

Výsledky

Následující tři tabulky (Tabulka 18, Tabulka 19, Tabulka 20) obsahují postupně naměřené výsledky kvality detekce všech popsaných variant implementace operací dilatace a eroze.

Tabulka 18 Tabulka obsahuje výsledky testování operace dilatace. Obě verze výstupů (průměr, maximum) jsou uvedeny v tabulce. Z tabulky vyplývá, že dilatační operace nezlepšuje výsledky ani u jedné varianty oproti situaci, kdy operace není použita (Tabulka 17). [114]

Práh pro hodnoty saturace a intenzity	Výstup dilatace	PPV	Průměrná vzdálenost od středu	Medián vzdálenosti od středu	FP	FN
0,3	průměr	56,4 %	4,48 px	1,82 px	60	1
	maximum	55,7 %	3,29 px	1,79 px	61	1
0,4	průměr	51,4 %	6,80 px	3,00 px	59	9
	maximum	51,4 %	6,84 px	2,99 px	59	9
0,5	průměr	50,0 %	3,70 px	2,57 px	32	38
	maximum	50,0 %	3,70 px	2,57 px	32	38

Tabulka 19 Tabulka obsahuje výsledky testování operace eroze ve variantě, kdy se pracuje s počtem sousedních pixelů. Z tabulky vyplývá, že operace nezlepšuje výsledky oproti situaci, kdy operace není použita (Tabulka 17). [114]

Práh pro hodnoty saturace a intenzity	Počet sousedních pixelů	PPV	Průměrná vzdálenost od středu	Medián vzdálenosti od středu	FP	FN
0,3	8	57,8 %	5,78 px	1,95 px	58	1
	7	57,1 %	4,92 px	1,83 px	59	1
	6	57,1 %	5,14 px	1,81 px	59	1
	5	56,4 %	3,98 px	1,83 px	60	1
0,4	8	50,7 %	5,11 px	2,43 px	58	11
	7	51,4 %	5,36 px	2,57 px	58	10
	6	52,1 %	5,63 px	2,60 px	58	9
	5	52,1 %	5,72 px	2,59 px	58	9
0,5	8	55,0 %	7,18 px	4,00 px	25	38
	7	55,0 %	7,19 px	4,07 px	25	38
	6	55,0 %	7,19 px	3,95 px	25	38
	5	52,9 %	5,13 px	3,37 px	28	38

Tabulka 20 Tabulka obsahuje výsledky testování operace eroze ve variantě, kdy se pracuje s váhou sousedních pixelů. Z tabulky vyplývá, že operace značně zlepšuje výsledky oproti situaci, kdy operace není použita (Tabulka 17). [114]

Práh pro hodnoty saturace a intenzity	Váha sousedních pixelů	PPV	Průměrná vzdálenost od středu	Medián vzdálenosti od středu	FP	FN
0,30	7	74,3 %	5,42 px	1,79 px	22	14
	6,5	75,0 %	5,40 px	2,13 px	33	2
	6	70,7 %	4,28 px	1,74 px	40	1
	5	69,3 %	4,35 px	1,66 px	42	1
0,35	7	75,7 %	5,78 px	1,82 px	20	14
	6,5	79,3 %	4,85 px	2,13 px	25	4
	6	72,1 %	5,34 px	1,93 px	37	2
	5	68,6 %	4,21 px	1,85 px	42	2
0,40	7	72,9 %	6,78 px	2,96 px	19	19
	6,5	76,4 %	5,49 px	3,45 px	22	11
	6	67,9 %	3,91 px	2,41 px	35	10
	5	64,3 %	4,97 px	3,18 px	9	9

Nejlepších výsledků bylo dosaženo s prahem 0,35 pro saturaci a intenzitu a s prahem 6,5 pro váhu sousedních pixelů pro operaci eroze. Operace dilatace neměla na detekci kladný přínos. Výsledný a dále testovaný algoritmus tedy využívá zmíněné hodnot prahů, využívá operaci eroze a nevyužívá operaci dilatace.

7.3.3 Výhody oproti předchozímu algoritmu

Oproti předchozímu algoritmu (Algoritmus rozdělení oblasti na části) je zde výhoda eliminace velikosti výstupní informace na naprosté minimum. Pokud je tedy nutné informace přečíst zpět z paměti grafické karty, tak druhý algoritmus dělá téměř maximum pro zrychlení daného procesu (úplné maximum by byl pouze jeden pixel). Je však možné využít i faktu, že finální informace je dostupná přímo v paměti grafické karty a na algoritmus navázat dalším průchodem grafickou pipeline a výsledky využít i bez nutnosti jejich kopírování z grafické paměti.

Předchozí algoritmus dále dělil vstupní obraz na podoblasti definované velikosti. Při tomto dělení dochází ke ztrátě přesnosti, která je úměrná velikosti dané podoblasti. V novém návrhu k žádné podobné ztrátě informace nedochází a detekce středu markeru je tedy přesnější (Tabulka 17, Tabulka 21).

Tabulka 21 Tabulka obsahuje výsledky testování předchozího algoritmu pro různé prahové hodnoty pro složky RGB barevného modelu při použití stejné metodologie jako při testování dílčích vylepšení druhého algoritmu (Tabulka 17). [114]

Práh pro složky RGB modelu	PPV	Průměrná vzdálenost od středu	Medián vzdálenosti od středu	FP	FN
0,10	55,0 %	21,81 px	15,86 px	16	47
0,15	57,1 %	18,07 px	14,32 px	24	36
0,20	52,9 %	17,05 px	16,43 px	42	24
0,25	25,0 %	18,89 px	13,02 px	94	11

Nakonec Tabulka 22 shrnuje všechny testované varianty a zlepšení a textově popisuje jejich vliv na detekci.

Tabulka 22 Shrnutí testovaných variant zlepšení algoritmu skenování řádků a sloupců a jejich výsledky [114]

Testovaná varianta	Výsledek
Přechozí algoritmus detekce	Horší výsledky, zejména přesnost
Použití HSV namísto RGB	Přesnější práce s odstínem
Použití váženého odstínu	Lepší výsledky
Dynamické hodnoty saturace a intenzity	Nepoužitelné
Operace eroze s počtem	Drobné zlepšení
Operace eroze s váhou	Výrazné zlepšení
Operace dilatace s obou variantách	Téměř žádná změna

7.3.4 Časová komplexita

Časová komplexita prvního kroku algoritmu je $O(\max(\text{width}, \text{height}))$, což znamená lineární složitost. V případě paralelního algoritmu je nutné uvažovat také cenu, kterou zde představuje vyžadovaný počet výpočetních jader, který je roven

množství potřebných kroků. Cena je pak vynásobena původní časovou složitostí a vychází $\mathcal{O}(\max(\text{width}, \text{height}) * \max(\text{width}, \text{height}))$. Po zjednodušení vychází kvadratická složitost $\mathcal{O}(n^2)$. Dále je nutné zmínit, že suma šířky a výšky bude zpravidla v řádu několika stovek, či maximálně nižších tisíců, a moderní grafické karty mají obvykle podobný řád počtu výpočetních jader. V případě, že by suma byla nižší než dostupný počet výpočetních jader, by výsledná časová složitost byla lineární $\mathcal{O}(n)$, protože cena v podobě počtu jader by byla redukována na konstantu.

Časová komplexita druhého kroku je stejná, tedy $\mathcal{O}(\max(\text{width}, \text{height}))$. Avšak zde jsou vždy potřeba již pouze dvě výpočetní jádra, takže lze cenu algoritmu považovat za konstantu. Časová složitost druhého kroku je tedy lineární $\mathcal{O}(n)$. Případné následné renderovací kroky využívající výstupy algoritmu mohou využít faktu, že informace o poloze markeru se již nachází v paměti grafické karty. Dále je možné druhý krok spojit s dalším zpracováním a eliminovat režii jednoho spuštění renderovací pipeline.

7.3.5 Rychlost detekce

Stejně jako pro přechodí algoritmus je i zde použita stejná metodologie měření, aby bylo možné oba algoritmy porovnat. Nejprve je prezentováno měření operací na GPU, které je z důvodu již popsaných limitací provedeno pouze na desktopu. Následně je uvedeno také měření na CPU, které zahrnuje i mobilní zařízení. I zde je nutné zdůraznit, že časy obou tabulek nejsou mezi sebou porovnatelné.

Tabulka 23 Naměřené časy běhu renderovací smyčky jednotlivých částí detekce markeru při použití algoritmu skenování řádků a sloupců. [113]

Zařízení	1. krok	2. krok	Celkem
<i>Intel UHD 630</i>	1,88 ms	1,02 ms	2,90 ms
<i>NVIDIA GTX 1060</i>	1,04 ms	0,63 ms	1,67 ms

Z tabulky 24 je možné vidět, že přečtení výsledku z GPU zabere většinu času i pro algoritmus skenování řádků a sloupců. Avšak tento algoritmus poskytuje také možnost výsledky ponechat v paměti GPU, protože už se jedná přímo o souřadnice

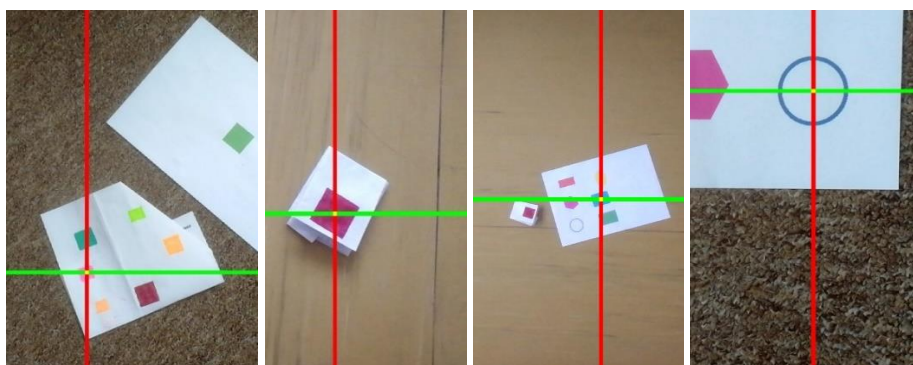
markeru, které je možné rovnou použít v případném dalším procesu. Jedná se o velkou výhodu proti předchozímu algoritmu rozdělení obrazu na části.

Tabulka 24 Naměřené CPU časy běhu jednotlivých částí detekce markeru při použití algoritmu skenování řádků a sloupců a rozlišení obrazu 1280 × 720 pixelů. [autor]

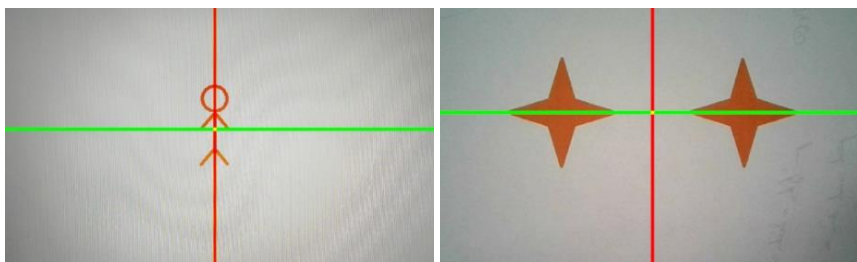
Zařízení	1. krok	2. krok	Čtení	Celkem
<i>Intel UHD 630</i>	0,16 ms	0,07 ms	6,71 ms	6,94 ms
<i>NVIDIA GTX 1060</i>	0,10 ms	0,02 ms	3,21 ms	3,33 ms
<i>Honor 9</i>	0,53 ms	0,47 ms	14,25 ms	15,25 ms
<i>Nokia 8.3 5G</i>	0,51 ms	0,37 ms	13,27 ms	14,15 ms

7.3.6 Příklady detekcí

Podkapitola nejprve prezentuje dvě sady vybraných detekcí markerů různých barev a tvarů. Sada má ukázat, že algoritmus ve svém principu není limitován pouze na vyplněné obdélníkové oblasti. Následně jsou ukázány detekce přímo „T-křížek“ markeru.



Obrázek 33 Příklady detekcí využitím algoritmu skenování řádků a sloupců. Všechny čtyři obrázky prezentují detekce markerů různých barev (růžová, červená, modrá) v reálném prostředí, kde markery ostatních barev jsou vždy správně ignorovány. Pravý obrázek ukazuje, že detekce nestandardních markerů (zde kruh) jsou také podporovány. [autor]



Obrázek 34 Příklady detekcí využitím algoritmu skenování řádků a sloupců. Vlevo detekována ikona postavy. Vpravo detekován bod mezi dvěma hvězdami. V obou případech se jedná o těžiště jako bod, kde je váha detekované barvy nejvyšší. [autor]



Obrázek 35 Příklady detekcí „T-křížek“ markeru s využitím algoritmu skenování řádků a sloupců. Ostatní markery jsou vždy správně ignorovány. [autor]

7.4 Vyhodnocení algoritmů na finální datové sadě

Podkapitola obsahuje údaje o vyhodnocení všech tří referenčních řešení pro detekci navrženého markeru „T-křížek“. V každé další podkapitole je popsáno, jak bylo zajištěno, aby porovnání mezi odlišnými způsoby detekce bylo možné. Dále jsou vždy představeny problémy, které bylo nutné vyřešit, a jsou prezentovány výsledky, které jsou nakonec mezi sebou porovnány.

7.4.1 Umělá neuronová síť s architekturou YOLOv3

Výstupem umělé neuronové sítě jsou přímo informace o souřadnicích bounding boxů detekovaných markerů. Výstupní informace jsou tedy přímo použitelné pro zjištění kvality detekce. Testovací sada obsahuje 666 obrázků a každý obsahuje právě jeden marker. Z těchto markerů pouze 6 nebylo nalezeno, počet falešně pozitivních detekcí je tedy 6, což činí 0,01 %. Falešně pozitivní detekce se v obrázcích nacházejí 2. Hodnota metriky IoU je 0,822, což značí velmi přesné detekce.

7.4.2 Algoritmus rozdělení obrazu na části

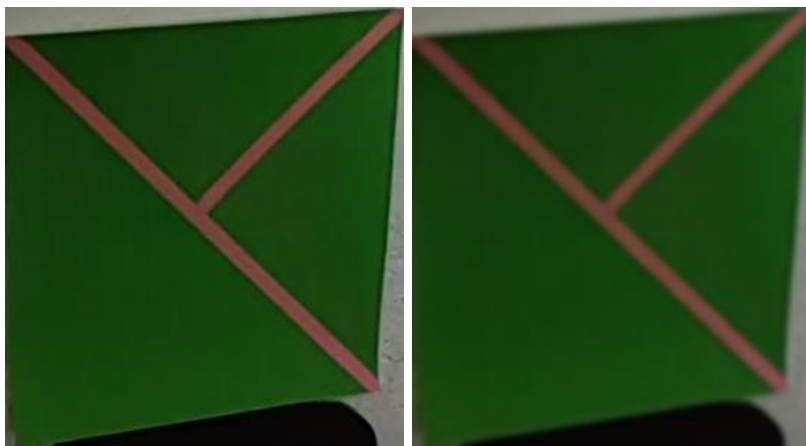
Původní algoritmus pracoval pouze s fixní zelenou barvou markeru. Nezbytné změny algoritmu pro otestování různě barevných markerů datové sady tedy byly

provedeny. Dále aby bylo možné porovnat výsledky s ostatními algoritmy, bylo nezbytné pro každý snímek datové sady barvu získat.

Každý marker v daném snímku je čtyřúhelníková oblast, ke které jsou známy souřadnice jejích rohů (vyznačeny při tagování datové sady). Tyto informace je tedy možné použít a následujícím postupem získat hlavní barvu markeru:

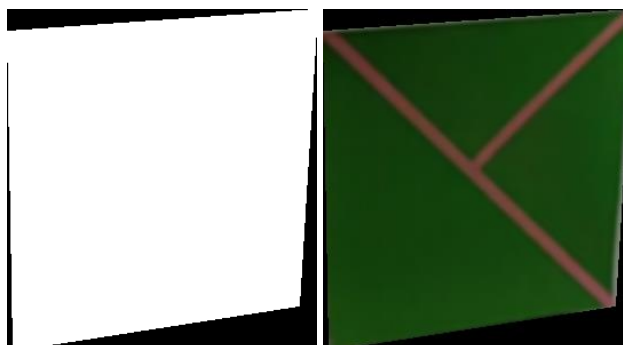
1. Načtení snímku
2. Vyříznutí bounding boxu s markerem (Obrázek 36)
3. Rozmazání snímku Gaussovým filtrem (Obrázek 36)
4. Seřazení bodů čtyřúhelníku tak, aby daný čtyřúhelník opisovaly
5. Vytvoření masky ze seřazených bodů (Obrázek 37)
6. Získání histogramu pro každou složku RGB barevného modelu podle výřezu tvořeným maskou z předchozího bodu (celkem tedy 3 histogramy)
7. Pro každý histogram získání hodnoty s nejvyšším počtem

Se zjištěnými barvami bylo možné detekci na datové sadě spustit. Avšak vzhledem k faktu, že algoritmus byl laděn pro práci s jednou konkrétní barvou (zelenou), tak výsledky nejsou dobré. I přes drobné úpravy čtení výsledné souřadnice bylo správně detekováno pouze 29 markerů z 666. Počet falešně negativních obrázků činí 408, což je 61,3 procent. Počet falešně pozitivních výsledků je 229, což znamená 34,4 procent. Vzhledem k nízké úspěšnosti nebyla hodnota metriky IoU stanovena.



Obrázek 36 a) vlevo vyříznutý marker z původního snímku s ID 1584 b) vpravo stejný výřez po aplikaci Gaussova filtru [autor]

Další výzkum se zabýval zlepšením algoritmu, což vedlo k algoritmu skenování řádků a sloupců, který mnoho problémů řeší a je považován následníkem algoritmu rozdělení obrazu na části. Přesto je algoritmus rozdělení obrazu na části zejména pro účely porovnání také zařazen.



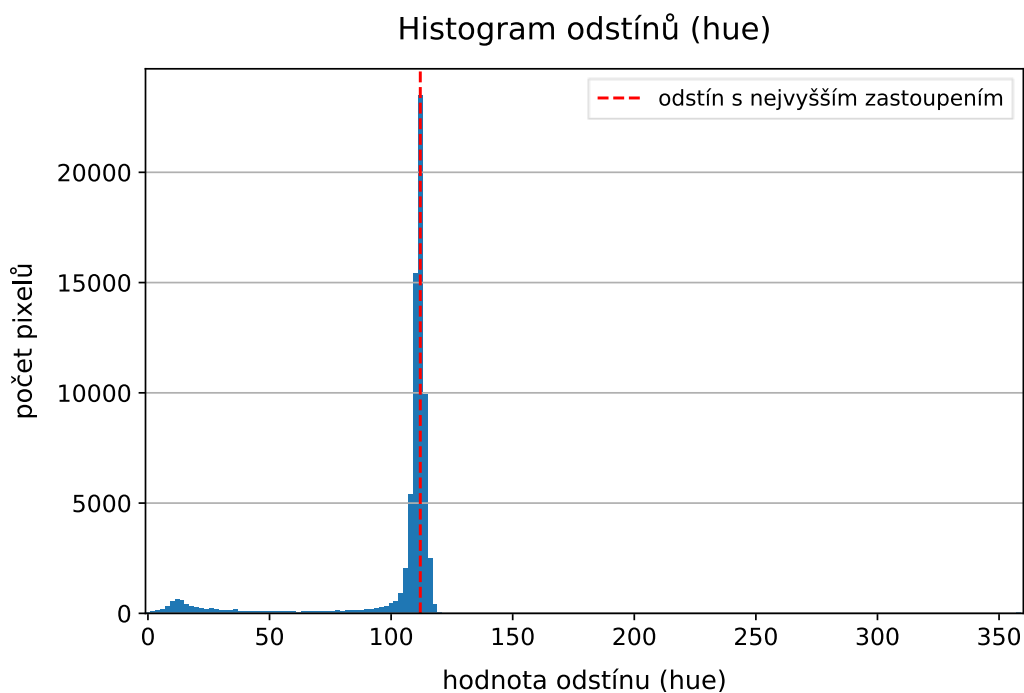
Obrázek 37 a) vlevo maska vytvořená podle hranice markeru ve vyříznutém obrázku b) vpravo vyříznutý marker podle vytvořené masky, tento výřez podle dané masky je poté využit pro vytvoření histogramu odstínů barvy markeru [autor]

7.4.3 Algoritmus skenování řádků a sloupců

Základním principem algoritmu je zvolení dominantního odstínu barvy markeru jeho snímáním předtím, než je spuštěn proces detekce. Při běžném použití by pak nebylo nutné odstín znovu volit při probíhající detekci. Avšak datová sada obsahuje pouze jednotlivé obrázky, nejedná se o video stream. Stejně jako pro předchozí algoritmus, i zde bylo nezbytné navrhnout způsob, jak pro každý snímek datové sady odstín automaticky získat.

Postup je velmi podobný, rozdíl je zejména v tom, že algoritmus pracuje s HSV barevným modelem a je nutné získat pouze dominantní odstín barvy markeru:

1. Načtení snímku
2. Vyříznutí bounding boxu s markerem
3. Rozmazání snímku Gaussovým filtrem
4. Převedení snímku z RGB barevného modelu do HSV barevného modelu
5. Seřazení bodů čtyřúhelníku tak, aby daný čtyřúhelník opisovaly
6. Vytvoření masky ze seřazených bodů
7. Získání histogramu H složky (HSV barevný model) podle výřezu tvořeným maskou z předchozího bodu
8. Hodnota histogramu s nejvyšším počtem představuje dominantní odstín



Obrázek 38 Histogram odstínů vyřiznutého markeru, výsledný odstín (hue v HSV barevném modelu) je 112 [autor]

Následně bylo možné algoritmus na datové sadě spustit. Avšak ukázalo se, že i přes umělé, a tedy s předpokladem přesné, určení odstínu barvy vykazuje detekční algoritmus vysoký počet falešně negativní detekcí, což jsou detekce, kdy žádná oblast ve snímku není označena jako marker. Konkrétně se jednalo o 130 případů, což je 19,5 % snímků.

Pro zjištění počtu falešně pozitivních detekcí byla zvolena maximální tolerance 50 pixelů od správné detekce středu markeru. Pro výpočet je využit princip euklidovské metriky. Podmínka byla stanovena na 50 pixelů, protože vůči velikostem markerů v datové sadě se jedná o hranici, která by měla zajistit, že v dalších krocích půjde z markeru přečíst požadované informace. Vyšší odchylka by vedla k tomu, že informace získat nepůjde. V definované situaci bylo nalezeno 133 falešně pozitivních detekcí – tedy detekcí, kdy střed markeru byl nalezen dále než 50 pixelů od skutečného středu markeru. To představuje 20,0 % snímků testovací sady.

Při odečtení falešně negativních a falešně pozitivních detekcí zbylo z celkových 666 snímků pouze 403 snímků, což představuje 60,5 % snímků. Chybně

detekovaných je tedy 39,5 % snímků. Tento počet dále vyplývá drobně nevýhodně směrem k tištěným markerům, kde tento poměr činil 48 %. Pro ručně kreslené markery se jednalo o 32 % chybně detekovaných snímků.

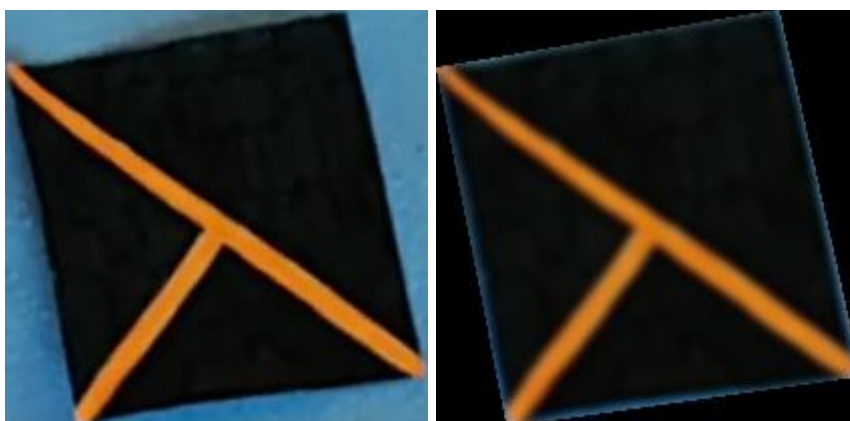
Kromě rozdělení na kreslené a tištěné markery byly dále různé markery snímány v různých podmínkách. Jedna šestina snímků byla provedena pod umělým osvětlením v interiéru. Jedna šestina snímků pochází z exteriéru. Zbývající snímky byly snímány v interiéru za denního světla. Z toho jeden marker byl snímán pod přímým slunečním světlem. Podrobnější popis rozdělení finální datové sady se nachází v kapitole 3.3.2 Finální datová sada pro testování.

Při podrobném prozkoumání výsledků se ukázalo, že falešné výsledky se seskupují zejména v některých skupinách markerů. Bylo použito celkem 12 podkladových barev, která byla každá zopakována 3krát s jinou barvou linie „T-křížek“ markeru. Následující tabulka (Tabulka 25) dává vhled na počty falešných detekcí sečtených podle podkladové barvy markeru.

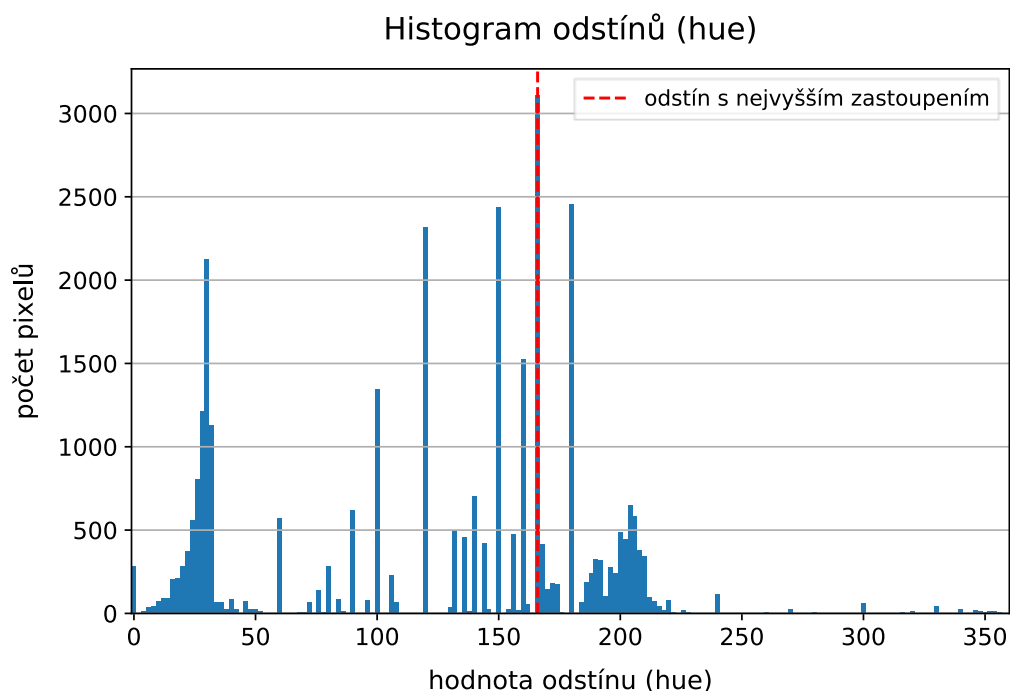
Tabulka 25 Součet počtů falešně negativních (FN) a falešně pozitivních (FP) detekcí. Červeně jsou vyznačeny výsledné hodnoty problémových odstínů. [autor]

Barva	Celkový počet testovacích snímků	Počet FN + FP detekcí	Procent FN + FP detekcí
Černá	62	59	95 %
Hnědá – tmavá	53	41	77 %
Hnědá – světlá	54	29	54 %
Červená	55	12	22 %
Oranžová	54	11	20 %
Žlutá	54	14	26 %
Zelená – světlá	55	19	35 %
Zelená – tmavá	55	2	4 %
Modrá – světlá	56	5	9 %
Modrá – tmavá	55	18	33 %
Fialová	54	26	48 %
Růžová	59	27	46 %

Výsledky ukazují, že zejména markery tmavých odstínů je daným algoritmem problematické detekovat. Důvodem je způsob práce s odstínem barvy v rámci HSV barevného modelu, kdy například černá „barva“ nefiguruje jako odstín (hue složka) v rámci HSV modelu. Černá „barva“ je v tomto modelu reprezentována složkou V, která určuje, jak hodně „tmavá“ bude daná barva vůči odstínu (H složka). Problém dobře ilustruje následující histogram odstínů pro vybraný černý marker (Obrázek 40).

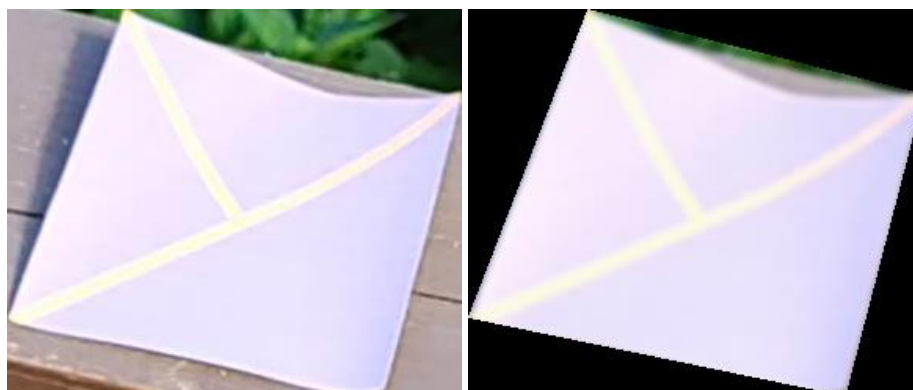


Obrázek 39 Příklad markeru s problematickým histogramem a) vlevo vyříznutý marker z původního snímku s ID 71 b) vpravo stejný výřez po aplikaci masky [autor]

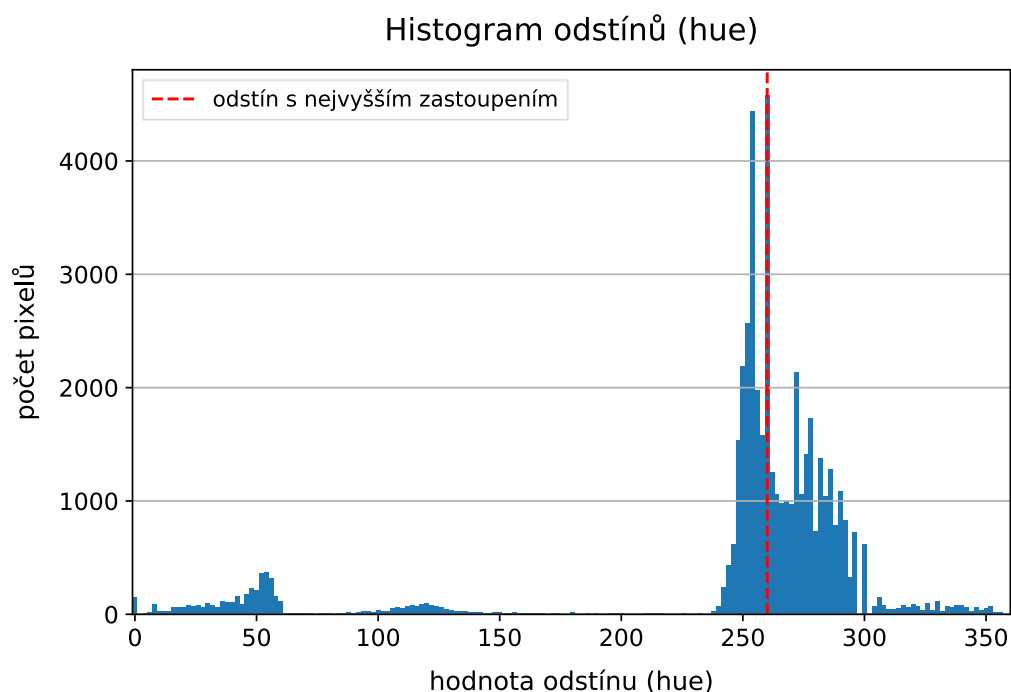


Obrázek 40 Ukázka histogramu černého markeru s ID 71. Histogram ilustruje nejistotu při určení dominantního odstínu a problém s černou barvou (a obecně tmavými odstíny) při zjišťování dominantního odstínu v rámci HSV barevného modelu. [autor]

Podobný problém je dále možné v menší míře pozorovat pro velmi světlé a nevýrazné odstíny, konkrétně fialovou a růžovou barvu – a zejména pak ve venkovním prostředí (jak ilustruje Obrázek 41). Aplikuje se zde podobný princip jako pro tmavé odstíny. Bílá „barva“ není možným odstínem z hlediska H složky HSV modelu. Množství bílé „barvy“ pixelu je zde přeloženo do hodnoty S, která reprezentuje saturaci hlavního odstínu.



Obrázek 41 Příklad markeru světlého odstínu s problematickým histogramem a) vlevo vyříznutý marker z původního snímku s ID 2345 b) vpravo stejný výřez po aplikaci masky [autor]



Obrázek 42 Ukázka histogramu světlého markeru s ID 2345. Histogram ilustruje nejistotu při určení dominantního odstínu a problém se světlými odstíny při zjišťování dominantního odstínu v rámci HSV barevného modelu. [autor]

Jako problémový se ukázal také marker, který sice měl standardní modrou barvu a byl nasnímaný v interiéru, avšak pod přímým slunečním světlem zapadajícího slunce, takže barva je snímána téměř jako černá. Stejný fakt platí i pro téměř celou podsadu markerů snímaných v exteriéru. Světlo bylo intenzivní a odstíny markeru mají velkou tendenci být velmi světlé nebo velmi tmavé a ve většině situací není možné odstín spolehlivě zjistit z důvodů, které jsou výše popsány.

Pro markery s potenciálně problémovým odstínem byl dále proveden test s vyzkoušením všech 360 možných odstínů. Test měl prokázat, zda je vůbec možné najít takový odstín, kterým by bylo možné marker správně detekovat. A zde se ukázalo, že neexistuje možnost nastavit takový odstín, aby vůbec bylo marker možné najít s danou tolerancí 50 pixelů.

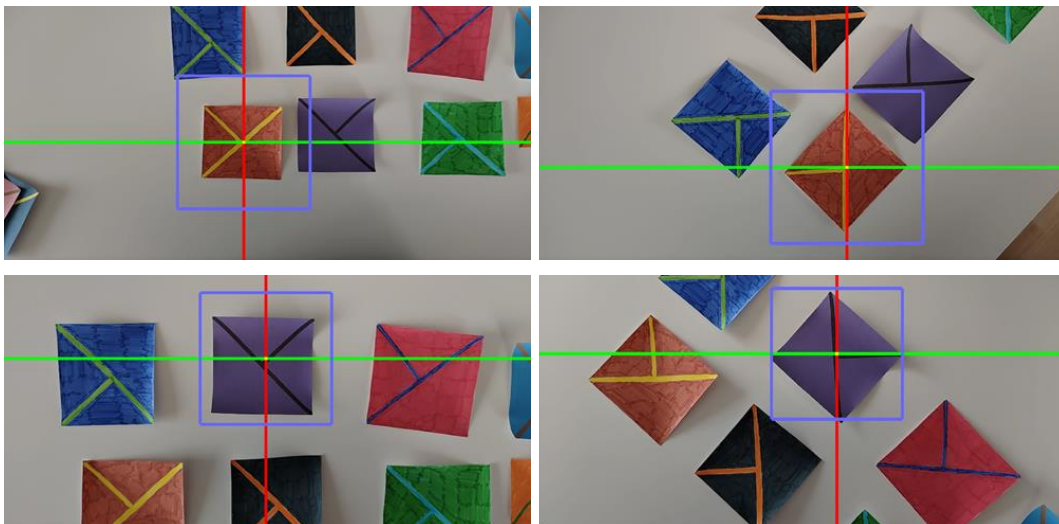
Jedním ze závěrečných doporučení tedy bude, aby při použití algoritmu skenování řádků a sloupců nebyly použity příliš tmavé nebo příliš světlé odstíny a že snímání je omezeno na interiér mimo přímé sluneční paprsky, které z hlediska snímání často velmi zesvětlují odstíny. Z tohoto důvodu je dále pro porovnání použita pouze podmnožina snímků, která podmínky doporučení splňuje.

Definovaná podmnožina datové sady obsahuje celkem 314 snímků. Z toho je pak falešně negativních 9 detekcí (2,9 %) a falešně pozitivních 33 detekcí (10,5 %). Celková chybovost je tedy 13,4 % snímků.

Aby dále bylo možné porovnat i metriku IoU, musel být proveden odhad velikosti, na základě které je pak možné odhadnout bounding box markeru. Algoritmus danou informaci na svém výstupu přímo nemá, ale kromě středu markeru je na výstupu také počet pixelů, které barvu markeru ve snímku mají. Tento počet je možné pro odhad velikosti využít. Ačkoliv se nejedná přímo o informaci o bounding boxu, tak počet je využitelný k výpočtu délky hrany a dále pak společně se středem k odhadu bounding boxu, který lze použít pro výpočet IoU metriky. Vzhledem k tomu, že markery testovací sady jsou převážně čtvercové, je po odhad délky strany a daného čtverce na základě počtu pixelů k využít následující vzorec:

$$a = \sqrt{k} * 2,1$$

Vzorec záměrně mírně nadsazuje velikost strany čtverce, aby bylo zajištěno plné obsáhnutí markeru i v situacích, kdy je vůči standardní poloze otočen o 45° a při použití pouze odmocniny by se rohy, které jsou v dalším zpracování důležité, do bounding boxu nevešly. Situaci ilustrují následující obrázky (Obrázek 43).



Obrázek 43 Snímky ilustrují důvod, proč je velikost hrany čtverce nadsazena při odhadu jejího výpočtu – aby se marker do odhadnutého bounding boxu stále vešel i při diagonálním snímání (pravé snímky). [autor]

Odhadnuté bounding boxy jsou poté porovnány se správnou informací a je vypočítaná průměrná hodnota metriky IoU, která činí 0,469. K výpočtu jsou využity pouze true positive výsledky. Ačkoliv se jedná spíše o mírně podprůměrnou hodnotu pro IoU, tak při přihlédnutí k faktu, že je založena pouze na odhadu velikosti hrany bounding boxu, se jedná o dostačující hodnotu.

7.5 Shrnutí a doporučení

Z hlediska finálního doporučení proti sobě stojí zejména dva faktory – rychlost a kvalita. Kvalitnější výsledky podává s velkým náskokem neuronová síť. Avšak cenou je 10krát až 50krát pomalejší zpracování jednoho snímku.

Další výhody zpracování obrazu umělou neuronovou sítí s využitím YOLOv3:

- možnost využití více markerů v jedné scéně,
- větší odolnost vůči barevným změnám,
- možnost využít širší paletu barev (bez žádného omezení),
- a možnost snímat i scény v exteriéru.

Na druhé straně stojí výhody využití algoritmů na GPU:

- značně rychlejší zpracování jednoho snímku,
- možnost nasazení i na mobilním telefonu,
- snímání více markerů v jedné scéně nachází jejich vizuální těžiště,
- větší kontrola nad procesem snímání,
- a tzv. „vysvětlitelnost“ řešení, protože se jedná o standardní algoritmus.

Oba způsoby detekce tedy dokáží najít uplatnění v závislosti na situaci a požadovaných parametrech detekce a rozhodnutí je na základě daných doporučení až na uživateli.

Tabulka 26 Porovnání kvality detekce popsaných řešení [autor]

Způsob detekce	FN	FP	FN + FP	IoU
YOLOv3	6 (0,1 %)	2	8	0,822
Algoritmus rozdělení obrazu na části	408 (61,3 %)	229 (34,4 %)	637 (95,6 %)	–
Algoritmus skenování řádků a sloupců	9 (2,9 %)	33 (10,5 %)	42 (13,4 %)	0,469

Tabulka 27 Porovnání rychlosti popsaných řešení detekce markeru v jednom snímku o rozměrech 1280 × 720 pixelů. Obrázek pro vstup do neuronové sítě zmenšen na rozměry 416 × 416 pixelů. [autor]

Způsob detekce	NVIDIA GeForce GTX 1060 (laptop)	Nokia 8.3 5G
YOLOv3	44 ms	– *
Algoritmus rozdělení obrazu na části	3,65 ms	6,42 ms
Algoritmus skenování řádků a sloupců	3,33 ms **	14,15 ms **

* Neuronová síť nebyla vzhledem k výpočetní a paměťové náročnosti testována v prostředí mobilních zařízení.

** Uvedené časy algoritmu zahrnují kopírování výsledné informace z paměti grafické karty. V případě nekopírování se časy zpracování pohybují v rozmezí 1-3 milisekund.

7.6 Zdrojové kódy

Zdrojové kódy referenčních implementací použitých pro testování detekce jsou k dispozici v autorových online repositářích na serveru GitHub. Repositář <https://github.com/milankostak/Marker-detection-NN> obsahuje zdrojové kódy, které se týkají detekce neuronovými sítěmi. Repositář <https://github.com/milankostak/Marker-detection> zahrnuje kód, který řeší detekci klasickými výpočtovými algoritmy na GPU. Dále se v něm také nachází použitý tagovací program.

8 Získání informací z detekovaného markeru

Proces detekce markeru v celém obrazu je pouze první částí celého řešení. Po získání souřadnic bounding boxu je nutné udělat výřez z původního obrázku a provést sérii operací, které zjistí potřebné výstupní informace. Operace lze rozdělit do dvou skupin: zpracování obrazu pro získání informací o liniích markeru a následné využití linií markeru pro výpočet informací o středu, orientaci a velikosti.

8.1 Zpracování obrazu pro získání informací o liniích markeru

Celý proces je možné sumarizovat následujícími deseti kroky:

1. Vyříznutí detekovaného markeru z původního obrázku
2. Konverze pixelů obrázku do HSV barevného modelu
3. Vytvoření histogramu odstínů (H složka HSV modelu) a získání odstínu s nejvyšším počtem pixelů
4. Odfiltrování pixelů s odlišným odstínem
5. Aplikace morfologické operace eroze
6. Aplikace morfologické operace dilatace
7. Aplikace rozostření pomocí Gaussova filtru
8. Aplikace Otsuova adaptivního prahování
9. Aplikace Cannyho hranového detektoru
10. Nalezení hran markeru pomocí Houghovy transformace

Prvním krokem je vyříznutí detekovaného markeru z původního celého snímku. Využity jsou informace o detekovaném bounding boxu. Vyříznutí přidává kolem detekovaného bounding boxu padding 15 pixelů na všechny strany, aby bylo zajištěno, že celý marker bude ve výřezu. Zároveň 15 pixelů není příliš velká hodnota, aby se do výřezu dostalo příliš mnoho dalších pixelů, které s markerem nesouvisí. Ukázka kódu 3 obsahuje referenční implementaci popsaného vyříznutí a Obrázek 45 výřez ilustruje.

Druhým krokem je konverze pixelů z RGB barevného modelu do modelu HSV, jehož složka H obsahuje přímo informaci o odstínu pixelu. S využitím knihovny

OpenCV se může jednat pouze o jeden řádek, jak ilustruje referenční implementace (Ukázka kódu 4).

```
def get_cropped(image_id: str, padding: int = 15):
    matches = (x for x in lines if x[0] == image_id)
    bb = next(matches)
    if len(bb) <= 1:
        return [None, -1, -1]

    x1 = float(bb[1]) - padding
    if x1 < 0:
        x1 = 0
    y1 = float(bb[2]) - padding
    if y1 < 0:
        y1 = 0
    x2 = float(bb[3]) + padding
    y2 = float(bb[4]) + padding

    image_path = f"{base_path}/test/{image_id}.jpg"
    img = cv2.imread(image_path)
    cropped_image = img[round(y1):round(y2), round(x1):round(x2)]
    return [cropped_image, round(x1), round(y1)]
```

Ukázka kódu 3 Referenční implementace vyříznutí markeru z původního snímku podle detekovaného bounding boxu a určeného paddingu. Ukázka využívá knihovnu OpenCV, která je importována pod standardním aliasem cv2. [autor]

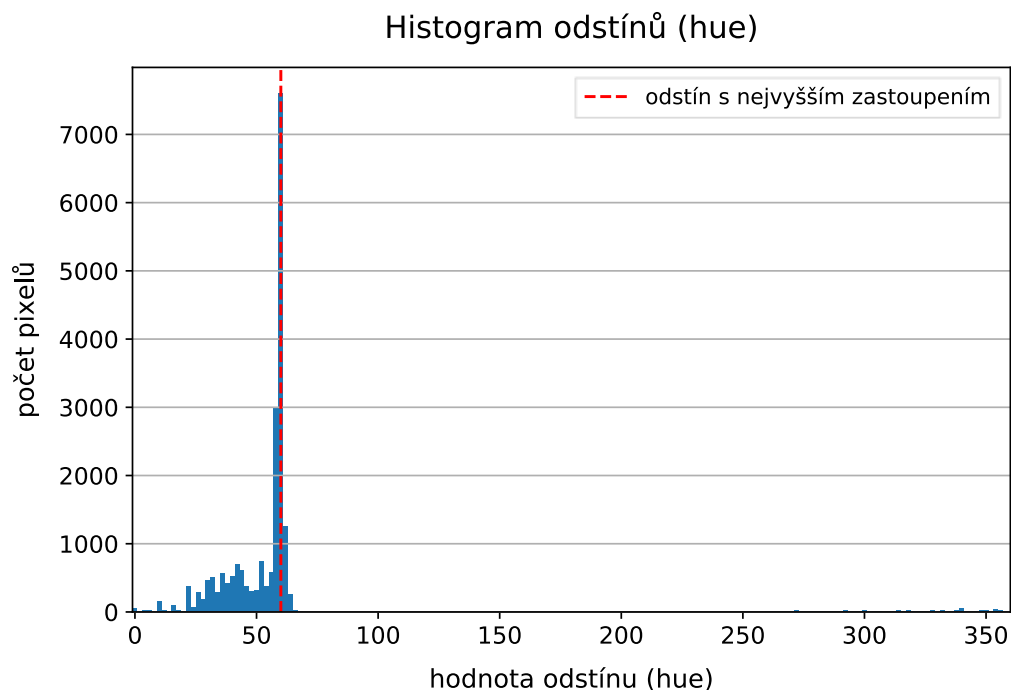
```
hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
```

Ukázka kódu 4 Referenční implementace převodu obrázku z RGB do HSV barevného modelu s využitím knihovny OpenCV [autor]

Cílem třetího kroku je získání dominantního odstínu zpracovávaného markeru. Pro tento účel je nutné vytvořit histogram odstínů (Obrázek 44) a poté načíst ten s nejvyšším zastoupením. Vzhledem k dříve nastavenému paddingu je pravděpodobné, že se v obrázku nachází značné množství pixelů jiného odstínu, než je odstín markeru. Takových pixelů by zpravidla nemělo být dostatečně na to, aby to narušilo zjištění odstínu markeru. Avšak pro zajištění vyloučení popsanych pixelů je při tvorbě histogramu dále aplikovaná maska, kterou vizuálně ilustruje Obrázek 46. Ukázka kódu 5 obsahuje referenční implementaci popsanych operací.

Ve čtvrtém kroku jsou na základě nejčastějšího odstínu z výřezu vyfiltrovány všechny pixely ostatních odstínů. Hodnoty odstínu jsou při zpracování uvažovány v rozsahu od 0 od 180 a tolerance pro zachování pixelu v obrázku je nastavena na 4 v obou směrech. Pro příklad histogramu výše, kde je hodnota odstínu 60, by se

v popsaném rozsahu jednalo o hodnotu 30. Výstup čtvrtého kroku ilustruje Obrázek 47, referenční implementaci prezentuje Ukázka kódu 6.



Obrázek 44 Histogram odstínů vyřiznutého markeru, výsledný odstín (hue v HSV barevném modelu) má hodnotu 60 [autor]

```
mask = np.zeros(shape=img.shape[:2], dtype=np.uint8)
points = np.array([
    [img.shape[1] // 2, 0], [img.shape[1], img.shape[0] // 2],
    [img.shape[1] // 2, img.shape[0]], [0, img.shape[0] // 2]
])
cv2.fillPoly(img=mask, pts=[points], color=255)

hist = cv2.calcHist(images=[hsv], channels=[0], mask=mask,
    histSize=[180], ranges=[0, 180])
target_hue = np.argmax(hist)
```

Ukázka kódu 5 Referenční implementace vytvoření masky a histogramu odstínů s následným přečtením nejčastějšího odstínu s využitím knihoven OpenCV a NumPy (importována pod standardním aliasem np). [autor]

Jak ilustruje Obrázek 47, ve výřezu se může nacházet šum, který by negativně ovlivňoval další zpracování. V rámci kroků pět a šest jsou postupně aplikovány morfologické operace eroze a dilatace, aby došlo k zacelení děr, které šum tvoří. Operace dilatace používá menší masku, aby došlo ke zdůraznění linií markeru v případě, že nejsou dostatečně výrazné. Testování ukázalo, že výsledky se tím mírně zlepšily.

```

h, s, v = cv2.split(hsv)
hue_threshold = 4
# handle circular hue
if hue_threshold <= target_hue <= 180 - hue_threshold:
    h[h < target_hue - hue_threshold] = 0
    h[h > target_hue + hue_threshold] = 0
elif target_hue < hue_threshold:
    # if target_hue is too low
    h[np.logical_and(target_hue - hue_threshold + 180 > h, h >
target_hue + hue_threshold)] = 0
else:
    # if target_hue is too high
    h[np.logical_and(target_hue - hue_threshold > h, h > target_hue
+ hue_threshold - 180)] = 0

```

Ukázka kódu 6 Referenční implementace odstranění pixelů, které nemají odstín podobný odstínu detekovaného markeru (tedy dominantního odstínu v daném výřezu) s využitím knihoven OpenCV a NumPy [autor]

V rámci kroků sedm, osm a devět jsou postupně aplikovány operace rozostření obrazu Gaussovým filtrem, Otsuovo adaptivní prahování a Canny detektor hran. Cílem rozostření a prahování je připravit obrázek, aby detektor hran vrátil co nejkvalitnější výsledek. Zejména se jedná o vyhlazení nerovností jednotlivých hran markeru. Použité parametry jednotlivých operací ilustruje Ukázka kódu 7.

```

blur_dst = cv2.GaussianBlur(src=dilatation_dst, ksize=(11, 11),
sigmaX=0)

threshold_value, threshold_dst = cv2.threshold(
    src=blur_dst, thresh=0, maxval=255, type=cv2.THRESH_BINARY +
cv2.THRESH_OTSU
)

edges_dst = cv2.Canny(image=threshold_dst, threshold1=50,
threshold2=200, edges=None, apertureSize=3)

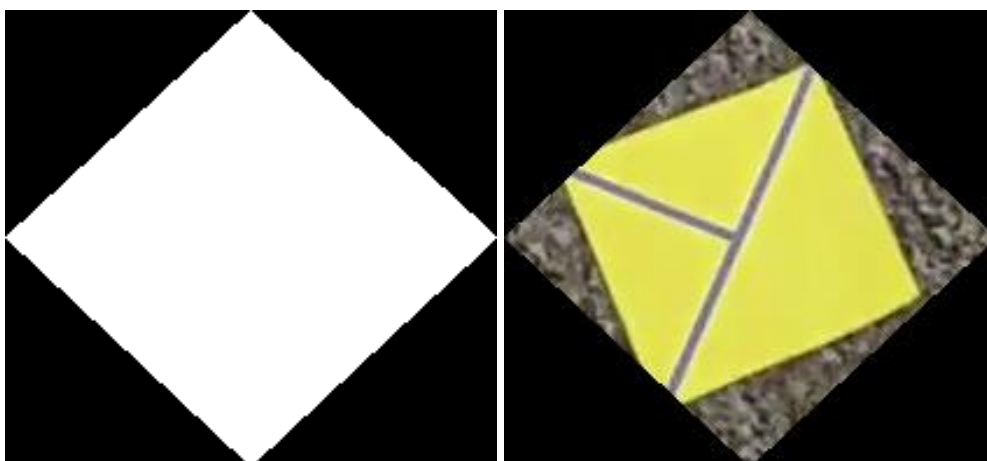
```

Ukázka kódu 7 Ukázka aplikace operací rozostření obrazu Gaussovým filtrem, Otsuova adaptivního prahování a Cannyho detektoru hran a použitých parametrů s využitím knihovny OpenCV [autor]

Desátý krok zpracování spočívá v aplikaci Houghovy transformace pro identifikaci přímk, které tvoří linie markeru po aplikaci Cannyho detektoru hran. Výstupy ilustruje Obrázek 50. S parametry Houghovy transformace se při zpracování pracuje dynamicky, aby byla zajištěna detekce dostatečného množství přímk. V dalším zpracování se pokračuje pouze, pokud je detekováno dostatečné množství přímk pro popsání markeru.



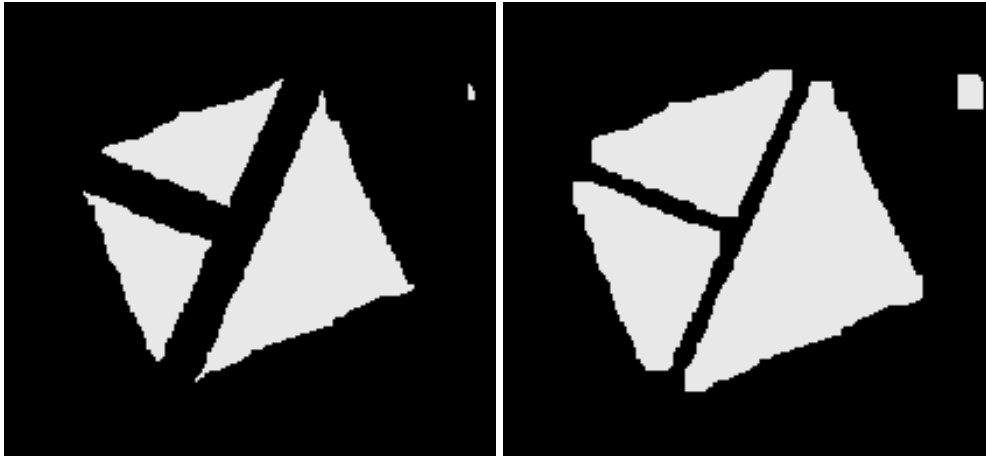
Obrázek 45 Ukázka vyříznutého markeru z originálního snímku s ID 1518 [autor]



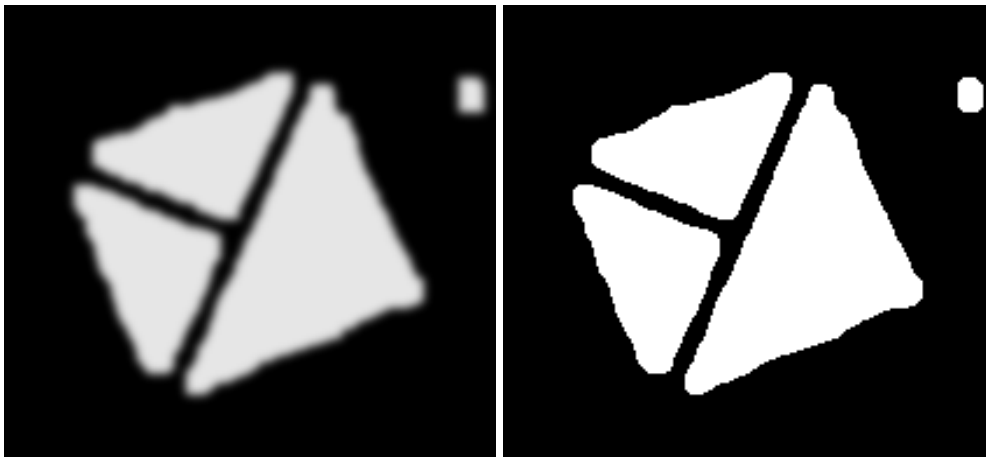
Obrázek 46 Vlevo ukázka kosočtverečné masky, která je použita pro odfiltrování pixelů, které s markerem nesouvisí. Vpravo pak aplikace masky na vyříznutý marker. [autor]



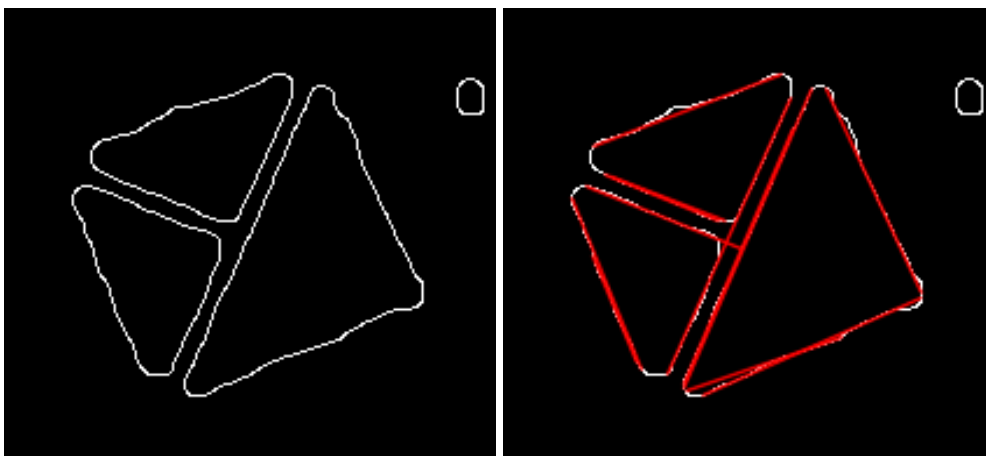
Obrázek 47 Výstup čtvrtého kroku zpracování, kdy z původního výřezu jsou zachovány pouze pixely, které mají odstín podobný odstínu detekovaného markeru (tedy dominantního odstínu v daném výřezu) [autor]



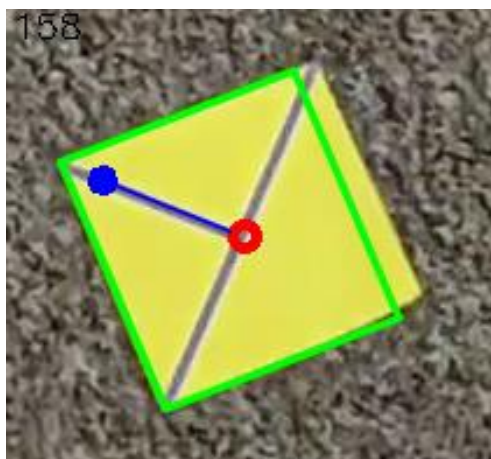
Obrázek 48 Vlevo ukázka výstupu po aplikaci operace eroze. Vpravo pak po aplikaci operace dilatace. Při porovnání s předchozím obrázkem (Obrázek 47) došlo k zacelení děr tvořených šumem. [autor]



Obrázek 49 Vlevo ukázka výstupu po aplikaci Gaussova filtru pro rozmazání obrazu. Vpravo výstup po aplikaci Otsuova adaptivního prahování. [autor]



Obrázek 50 Vlevo ukázka výstupu po aplikaci Cannyho detektoru hran. Vpravo navíc vykresleny výstupní linie po jejich identifikaci Houghovou transformací. [autor]



Obrázek 51 Všechny finální zjištěné informace o markeru zakreslené do původního výřezu. Červený kruh označuje zjištěný střed markeru. Modrá šipka značí směr orientace markeru, kdy přímo úhel orientace je zapsán také v levém horním rohu (158 stupňů). Zelený obdélník označuje celý zjištěný marker. [autor]

8.2 Zpracování linií markeru

Zpracování má na svém vstupu přímky, které jsou výstupem Houghovy transformace. Jedná se o nejkomplicovanější krok, protože pouze informace o přímkách je nutné použít pro zjištění středu markeru, jeho orientaci a jeho velikosti (Obrázek 51). Zpracování lze rozdělit na následující operace:

1. Vypočítání délky úsečky a parametrů k a q (směrnice a průsečík s osou Y)
2. Seřazení úseček podle hodnoty směrnice
3. Vypočítání rozdílů hodnot směrnic seřazených úseček
4. Nalezení tří největších rozdílů
5. Rozdělení úseček do čtyř clusterů podle podobnosti jejich směrnice
6. Zjištění směrodatných odchylek parametrů q pro každý cluster
7. Rozdělení čtyř clusterů na dvě skupiny podle odlišnosti spočítaných směrodatných odchylek, čímž dojde k rozdělení na vnitřní a vnější linie markeru
8. Využití vnitřních linií markeru pro spočítání jejich průsečíku, a tedy středu markeru
9. Rozdělení vnitřních linií na kratší a delší a následně použití kratších k výpočtu orientace společně s již vypočítaným středem k zajištění směru
10. Využití vnějších linií markeru, jejich rozdělení na dvě skupiny podle parametrů q , rozdělení na čtyři hrany markeru a spočítání jejich průsečíků

Samotné zpracování je uvažováno pouze pokud je vráceno alespoň pět linií. Výstupem jsou koncové body úseček, takže prvním krokem je zjištění délek úseček a jejich parametrů k (směrnice) a q (průsečík s osou Y), které reprezentují přímkou, kterou dané dva body tvoří. Pole linií je následně seřazeno podle hodnoty směrnice.

Třetím krokem je vypočítání rozdílů směrnic mezi seřazenými liniemi. Rozdíly nejsou v daném rozsahu rovnoměrně rozloženy vůči prostému rozdílu, je tedy aplikována funkce arkus tangens.

Cílem čtvrtého a pátého kroku je rozdělit linie do čtyř clusterů tak, že dva clustery obsahují linie obrysu tvaru a dva clustery obsahují linie vnitřního tvaru. Hlavním rozpoznávacím znakem těchto skupin jsou odlišné směrnice, které uvnitř clusteru jsou vždy velmi podobné. Po vypočítání rozdílů jsou tedy nalezeny tři největší rozdíly a ty představují hranice pro rozdělení do clusterů.

Po základním rozdělení na čtyři clustery je nutné dále v šestém a sedmém kroku zjistit, které dva clustery obsahují vnější linie a které dva vnitřní linie. K tomu je využit parametr q . Myšlenka spočívá v tom, že oba clustery vnějších linií v sobě obsahují dva druhy linií, kdy oba druhy mají z předchozího rozdělení velmi podobnou směrnici, ale mají velmi odlišný průsečík s osou Y . Je tedy spočítána směrodatná odchylka pro všechny čtyři původní clustery a na základě vypočítaných hodnot je rozhodnuto, které dva clustery obsahují vnější linie a které obsahují vnitřní linie.

Po dokončení sedmého kroku je známé rozdělení mezi vnější a vnitřní linie markeru. Cílem osmého kroku je využití vnitřních linií pro spočítání jejich průsečíku, který představuje střed markeru. Hodnoty linií v obou clusterech vnitřních linií jsou zprůměrovány a následně je spočítán průsečík těchto dvou linií.

V devátém kroku je zjištěna orientace markeru. Hlavní rozdíl dvou clusterů vnitřních linií spočívá v délce jejich linií. Je tedy možné zjistit, který cluster obsahuje směrové linie a následně s touto informací zjistit orientaci markeru.

Posledním krokem je zjištění velikosti markeru. K tomu slouží zbývající dva clustery vnějších linií. Pro každý cluster dojde k rozdělení na dvě další skupiny a z výsledných čtyř skupin je možné spočítat souřadnice rohů markeru.

Na výstupu se tedy nachází všechny tři hledané informace: střed markeru, orientace markeru a velikost markeru.

8.3 Výsledky zpracování

Kvalita výstupu referenčního algoritmu je změřena pro oba algoritmy detekce markeru v obraze. Prvním je neuronová síť s architekturou YOLOv3 a druhým je algoritmus skenování řádků a sloupců. Dále pro neuronovou síť jsou uvažovány dvě situace – první pro celou testovací sadu a druhá pro testovací sadu s obrázky, které mají vhodné odstíny, jak bylo vysvětleno v kapitole 7.4.3. Algoritmus skenování řádků a sloupců je vyhodnocen pouze pro danou podsadu testovací sady, protože výsledky jsou k dispozici pouze pro ni.

Tabulka 28 Tabulka prezentuje velikost testovacích sad a počty nalezených markerů, vůči kterým je vyhodnocení provedeno [autor]

Způsob detekce	Velikost sady	Nalezených markerů
YOLOv3	666	649 (97 %)
YOLOv3	314	308 (98 %)
Algoritmus skenování řádků a sloupců	314	297 (95 %)

Postupně jsou představeny čtyři metriky – vzdálenost od středu, odchylka od orientace, rozdíl obsahu a IoU. Pro každou z metrik je vždy prezentováno množství nalezených markerů a dále medián a průměr chyby pro všechny nalezené markery. Poté následuje aplikace tolerance pro falešně pozitivní detekce a následně medián a průměr chyby pouze pro výsledky v dané toleranci.

Vzdálenost od středu představuje vzdálenost správného středu od detekovaného středu. Vzdálenost je spočítaná jako euklidovská metrika. Tolerance pro správnou detekci je nastavena na 60 pixelů. Tabulka 29 výsledky podrobně prezentuje.

Množství detekcí v toleranci se pohybuje v rozmezí od 50 do 80 procent. Chyba detekcí je hodnota pohybující se kolem 4 pixelů, což značí velkou přesnost.

Tabulka 29 Porovnání kvality získání informací o středu z detekovaného markeru při toleranci 60 pixelů pro falešně pozitivní detekce [autor]

Způsob detekce	Nalezených markerů	Medián chyby	Průměr chyby	Detekce v toleranci	Medián v toleranci	Průměr v toleranci
YOLOv3	649	20 px	46 px	451 (69 %)	4,5 px	14,6 px
YOLOv3	308	6 px	32 px	244 (79 %)	3,6 px	11,3 px
Alg. sken.	297	61 px	110 px	148 (50 %)	4,1 px	11,2 px

Odchylka od správné orientace představuje prostý rozdíl mezi správným úhlem orientace a tím detekovaným. Tolerance pro správnou detekci je nastavena 30 stupňů. Tabulka 30 výsledky podrobně prezentuje. Množství detekcí v toleranci se pohybuje v rozmezí od 45 do 70 procent. Přesnost detekcí je hodnota pohybující se kolem 1 stupně, což značí velkou přesnost. Je ale nutné podotknout, že množství detekcí mimo toleranci je vyšší, než by bylo vhodné.

Tabulka 30 Porovnání kvality získání informací o orientaci z detekovaného markeru při toleranci 30 stupňů pro falešně pozitivní detekce [autor]

Způsob detekce	Nalezených markerů	Medián chyby	Průměr chyby	Detekce v toleranci	Medián v toleranci	Průměr v toleranci
YOLOv3	649	9°	49°	355 (55 %)	1,1°	3,1°
YOLOv3	308	2°	37°	208 (68 %)	1,0°	2,4°
Alg. sken.	297	41°	58°	134 (45 %)	1,0°	3,0°

Rozdíl obsahu představuje prostý rozdíl množství pixelů v detekovaném markeru od množství pixelů správného markeru. Tolerance pro správnou detekci je nastavena na 10 000 obsahových pixelů (px²). Výsledky prezentuje Tabulka 31. Množství detekcí v toleranci se pohybuje v rozmezí od 35 do 55 procent a jedná se tedy spíše méně kvalitní výsledky. Avšak po aplikaci tolerance je přesnost dostačující.

Tabulka 31 Porovnání kvality získání informací o obsahu z detekovaného markeru při toleranci 10 000 px² pro falešně pozitivní detekce [autor]

Způsob detekce	Nalezených markerů	Medián chyby	Průměr chyby	Detekce v toleranci	Medián v toleranci	Průměr v toleranci
YOLOv3	649	10 632 px ²	26 448 px ²	311 (48 %)	3 042 px ²	3 611 px ²
YOLOv3	308	7 187 px ²	22 098 px ²	176 (57 %)	2 652 px ²	3 315 px ²
Alg. sken.	297	15 717 px ²	69 974 px ²	102 (34 %)	2 741 px ²	3 390 px ²

IoU dvou čtyřúhelníků představuje podíl jejich průniku a sjednocení. Vyjadřuje jejich podobnost a překryv bez ohledu na jejich absolutní velikost. Tolerance je nastavena na 0,3 podobně jako při práci s neuronovou sítí. Tabulka 32 ukazuje, že v toleranci se pohybuje 50 až 75 procent detekcí. Při aplikaci tolerance je IoU mezi 0,7 až 0,8, což značí kvalitní výsledky.

Tabulka 32 Porovnání metriky IoU detekovaného a správného markeru při toleranci 0,3 pro falešně pozitivní detekce [autor]

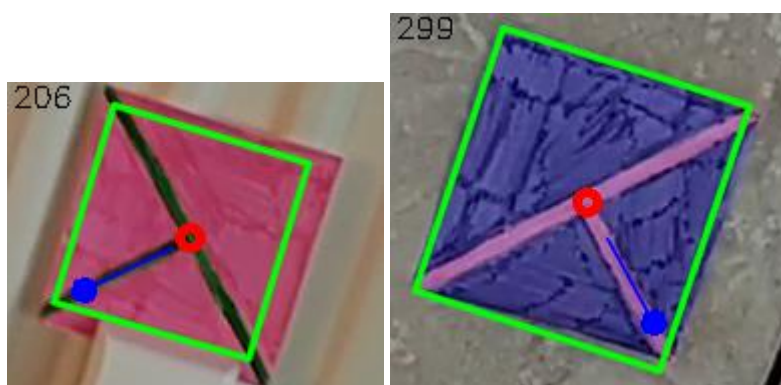
Způsob detekce	Nalezených markerů	Medián chyby	Průměr chyby	Detekce v toleranci	Medián v toleranci	Průměr v toleranci
YOLOv3	649	0,44	0,48	409 (63 %)	0,75	0,69
YOLOv3	308	0,67	0,58	230 (75 %)	0,81	0,74
Alg. sken.	297	0,27	0,38	143 (48 %)	0,74	0,70

8.4 Rychlost zpracování

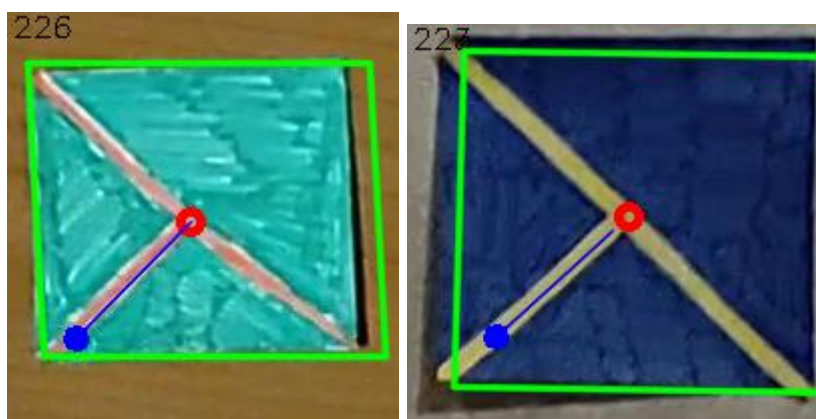
Rychlost zpracování byla změřena pro referenční implementaci vytvořenou v Pythonu s využitím knihoven OpenCV a NumPy. Naměřené časy jsou aritmetickým průměrem časů zpracování všech obrázků testovací sady, tedy 666 obrázků. Časy jsou dále rozdělené na čas zpracování obrazu a čas zpracování linií:

- čas zpracování obrazu: 3,84 ms,
- čas zpracování linií: 0,58 ms,
- a celkový čas: 4,42 ms.

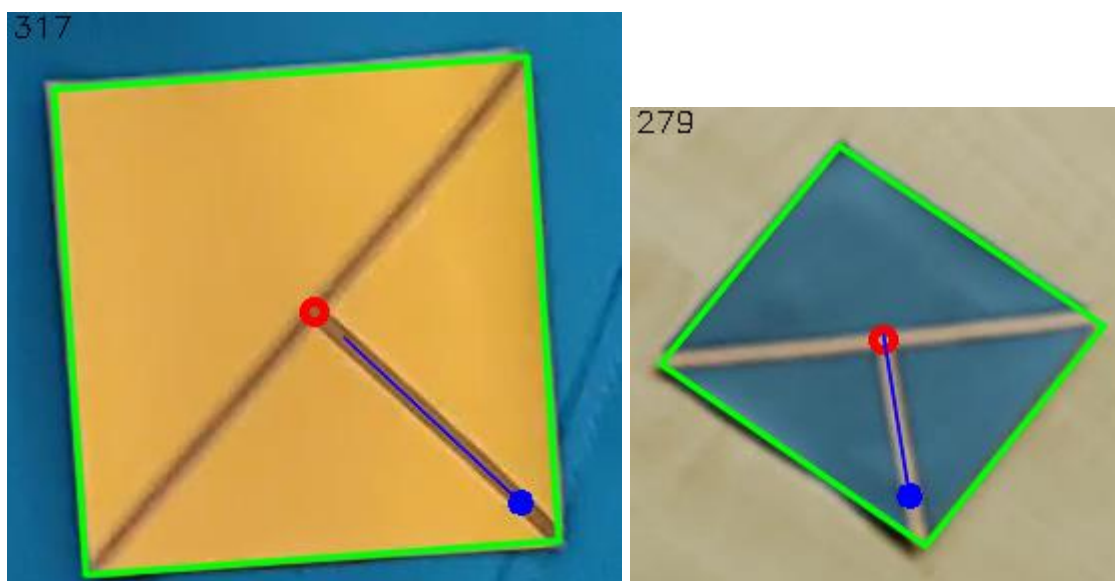
8.5 Vybrané příklady detekcí



Obrázek 52 Příklady vizualizací zjištěných informací z detekovaných markerů. Vlevo marker s ID 0635, vpravo s ID 1197. [autor]



Obrázek 53 Příklady vizualizací zjištěných informací z detekovaných markerů. Vlevo marker s ID 2804, vpravo s ID 2879. [autor]



Obrázek 54 Příklady vizualizací zjištěných informací z detekovaných markerů. Vlevo marker s ID 3197, vpravo s ID 3460. [autor]

8.6 Zdrojový kód

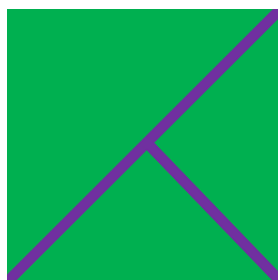
Zdrojový kód implementace k replikaci výsledků je k dispozici v autorově online repositáři na serveru GitHub: <https://github.com/milankostak/Marker-detection-NN>.

9 Shrnutí výsledků a jejich zhodnocení

Kapitola shrnuje na jednom místě všechny dosažené výsledky a následně je vždy provedeno jejich finální zhodnocení z hlediska naplnění stanovených cílů disertační práce.

9.1 Návrh markeru

Výsledkem návrhu je marker „T-křížek“, který ilustruje následující Obrázek 55. Před samotným započítím návrhu byla prozkoumána literatura, která se problematice věnuje. Nabyté znalosti pak byly při návrhu využity. Marker je navrhován iterativním způsobem postupných testů, kdy bylo zkoumáno, jak se jednotlivé zvažované tvary chovají z hlediska možností detekce.



Obrázek 55 Příklad „T-křížek“ markeru

V průběhu iterací byl nakonec navržen i tvar „T-křížek“, který měl jedny z nejlepších výsledků a zároveň splňuje všechna kritéria na něj kladená v rámci cílů práce:

- *Schopnost lokalizace markeru* – tvar má střed, který je dobře detekovatelný, jedná se o průsečík tří čar jdoucích ze tří rohů obdélníku.
- *Detekce úhlu rotace v rozlišení 360° relativně k pozorovateli* – tvar obsahuje tři linie, které umožňují detekci popsané rotace.
- *Detekce celého markeru pro získání relativní velikosti* – základem tvaru je vyplněná obdélníková oblast, která je využitelná pro detekci velikosti.

Splnění všech tří vymezených kritérií bylo nezbytné dodržet. Dále byla definována dvě doplňková kritéria, které jsou také dílčími cíli prvního hlavního cíle práce:

- jednoduchý tvar markeru včetně zajištění možnosti kreslení rukou,
- a vzor markeru bez vazby na jeho konkrétní barvu.

Ačkoliv splnění doplňkových kritérií nebylo definováno jako nezbytně nutné, tak se je také podařilo splnit. Díky nim je možné tvořit markery v rozmanitých variantách, díky čemuž vždy kvalitně a nerušivě zapadnou do různorodých scén.

Na základě shrnutých informací lze konstatovat, že všechny dílčí cíle prvního hlavního cíle práce byly naplněny, a tudíž celý první hlavní cíl disertační práce je považován za splněný.

9.2 Detekce a přečtení markeru

Druhý hlavní cíl disertační práce je rozdělen na dvě části. První částí je prozkoumání detekčních systémů, které dokáží marker v obrazu najít. Druhou částí je návrh řešení, které je schopné z detekovaného markeru přečíst požadované informace.

9.2.1 Návrh detekčního systému

Výsledkem zkoumání jsou dva způsoby detekce markeru „T-křížek“ v obrazu. Prvním je umělá neuronová síť s architekturou YOLOv3. Druhým je algoritmus pro paralelní zpracování na grafické kartě v rámci WebGL, v práci nazývaný jako „Algoritmus skenování řádků a sloupců“.

Na základě prezentovaného testování lze konstatovat, že obě techniky je možné využít k detekci markeru. Zároveň oba způsoby mají různé vlastnosti, díky kterým jsou vhodné pro různé situace. Detekce neuronovou sítí je přesnější a s menším počtem falešných výsledků (FN i FP), které se často pohybují v desetinách procenta. Avšak cenou za vyšší kvalitu je 10krát až 50krát pomalejší zpracování jednoho snímku.

Hlavní výhody zpracování obrazu neuronovou sítí s architekturou YOLOv3 shrnují následující body, jak již byly prezentovány v kapitole 7.5:

- možnost využití více markerů v jedné scéně,
- větší odolnost vůči barevným změnám,
- možnost využít širší paletu barev (bez žádného omezení),
- a možnost snímat i scény v exteriéru.

Detekce s využitím algoritmu skenování řádků a sloupců na druhé straně poskytuje mnohem rychlejší zpracování jednoho snímku při zachování dostatečné kvality. Ale chybovost v porovnání s neuronovou sítí je vyšší, kdy množství falešných výsledků se pohybuje v řádu jednotek procent.

Shrnutí výhod algoritmu skenování řádků a sloupců, jak již byly představeny v kapitole 7.5:

- značně rychlejší zpracování jednoho snímku,
- možnost nasazení i na mobilním telefonu,
- snímání více markerů v jedné scéně nachází jejich vizuální těžiště,
- větší kontrola nad procesem snímání,
- a tzv. „vysvětlitelnost“ řešení, protože se jedná o standardní algoritmus.

Závěrem tedy lze konstatovat, že obě prozkoumané techniky detekce dokáží najít své uplatnění. Konkrétní použití vždy záleží na preferencích uživatele, aktuální situaci a dostupných zařízeních a jejich výkonu.

9.2.2 Řešení pro získání informací z markeru

Poslední řešenou částí disertační práce je návrh způsobu získání informací z detekovaného markeru „T-křížek“. Proces má dvě části – první částí je zpracování obrazu s detekovaným markerem sérií grafických operací, které mají odhalit obrys markeru. Druhou částí je postprocessing, jehož cílem je z detekovaných linií obrysu markeru přečíst informace o středu markeru, orientaci markeru a velikosti markeru.

Navržený způsob je otestovaný pro oba způsoby detekce markeru a výsledky ukázaly, že z detekovaného markeru lze s dostačenou přesností informace přečíst.

9.2.3 Shrnutí procesu detekce z hlediska cílů práce

Druhým hlavním cílem práce byl návrh řešení, která by dokázala z detekovaného markeru přečíst vyžadované informace. Jak bylo v předchozích dvou podkapitolách popsáno, oba navržené postupy marker dokáží v obrazu rychle a

přesně detekovat. Dále výstupy detekce jsou využity pro přečtení informací z markeru a testování ukázalo, že informace jsou z markeru získatelné.

Na základě prezentovaných informací lze konstatovat, že oba dílčí cíle druhého hlavního cíle práce byly naplněny, a tudíž celý druhý hlavní cíl disertační práce je pokládán za splněný.

10 Závěr

Disertační práce ve svém textu nejprve shrnuje cíle práce a definuje metodiky zpracování. V následující kapitole je provedena podrobná rešerše literatury věnující se podobným tématům. Bylo podrobně prozkoumáno 64 různých systémů.

Disertační práce dále podrobně prezentuje návrh tvaru markeru jako první hlavní cíl práce. Záměrem bylo vytvoření markeru, který je vhodný pro aplikace využívající principů rozšířené reality. Důraz byl tedy kladen na to, aby bylo z detekovaného markeru možné získat jeho pozici, orientaci a velikost. Vyjmenované informace jsou nezbytně nutné pro použití v aplikacích rozšířené reality. Typickým použitím je získání informací a následné nahrazení markeru virtuálním objektem, který je do scény vložen s využitím získaných informací. Všechny dílčí cíle prvního hlavního cíle byly v práci splněny, a tudíž celý první hlavní cíl je považován za splněný.

Návrh markeru je pouze první částí celého detekčního systému. Následujícími kroky jsou přesná detekce navrženého markeru a dále získání požadovaných informací, které byly definovány. Jedná se o druhý hlavní cíl práce. Podrobné testování tří referenčních detekčních metod ukázalo, že různé metody mohou nacházet využití v různých situacích podle toho, zda je kladen důraz na rychlost nebo kvalitu detekce.

Natrénovaná umělá neuronová síť YOLOv3 poskytuje kvalitnější výsledky, ale detekce je 10krát až 50krát pomalejší než při použití standardního algoritmu implementovaného pro spuštění na grafické kartě. Na druhé straně pak stojí zmíněný algoritmus pro grafickou kartu, který sice podává horší avšak stále dostačující výsledky, ale zpracování jednoho snímku je mnohem rychlejší.

Všechny dílčí cíle i druhého hlavního cíle práce byly také splněny a celý druhý hlavní cíl je považován za splnění. Společně se splněním i prvního cíle lze konstatovat, že všechny stanovené cíle disertační práce jsou považovány za naplněné.

10.1 Další možné směry výzkumu

Budoucí výzkum by se mohl zaměřit například na přenos informací mezi detekcí jednotlivých snímků videa – tzv. „tracking“. V situaci, kdy je detekce v daném snímku či sérii snímků spolehlivá a v následujícím snímku nastává problém, mohou předchozí detekce asistovat při zlepšení detekce aktuálního snímku.

V případě nasazení výsledné aplikace v mobilním telefonu je možné pro další zlepšení výsledků využít různé senzory, které taková zařízení obsahují. Konkrétně se jedná například o akcelerometr a gyroskop. Tyto senzory mohou dobře posloužit při procesu detekce, protože pohyb zařízení nebude nutné detekovat pouze z obrazu, ale senzory mohou být přidanou hodnotou, vůči které lze výsledky validovat.

Detekce umělou neuronovou sítí je velmi kvalitní, ale zpracování jednoho snímku trvá delší dobu, než je vhodné. Další výzkum by mohl být zaměřen na vyřešení tohoto problému návrhem jiné neuronové sítě. Dále by bylo možné pro přesnější detekce využít segmentační neuronové sítě.

Možný další směr bádání by mohl mířit směrem k lepší výtěžnosti informací z markeru. Příkladem může být perspektivní zkosení, které má nezastupitelnou roli při vnímání prostorových scén.

11 Seznam použité literatury

- [1] D. G. Chandler, E. P. Batterman, a G. Shah, „Hexagonal, information encoding article, process and system“, US4874936A, 17. říjen 1989 Viděno: 6. červenec 2020. [Online]. Dostupné z: <https://patents.google.com/patent/US4874936/en>
- [2] Y. P. Wang a A. Ye, „Maxicode data extraction using spatial domain features“, US5637849A, 10. červen 1997 Viděno: 11. srpen 2020. [Online]. Dostupné z: <https://patents.google.com/patent/US5637849A/en>
- [3] „ISO/IEC 16023:2000: Information technology — International symbology specification — MaxiCode“, ISO. <https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/02/98/29835.html> (viděno 9. červenec 2020).
- [4] „ISO/IEC 18004:2015: Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification“, ISO. <https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/20/62021.html> (viděno 9. březen 2021).
- [5] „EXT_disjoint_timer_query_webgl2 API | Can I use... Support tables for HTML5, CSS3, etc“. https://caniuse.com/mdn-api_ext_disjoint_timer_query_webgl2 (viděno 8. březen 2023).
- [6] J. Redmon a A. Farhadi, „YOLOv3: An Incremental Improvement“, *arXiv:1804.02767*, dub. 2018, Viděno: 3. březen 2020. [Online]. Dostupné z: <http://arxiv.org/abs/1804.02767>
- [7] N. Bodla, B. Singh, R. Chellappa, a L. S. Davis, „Soft-NMS -- Improving Object Detection With One Line of Code“, *arXiv:1704.04503 [cs]*, srp. 2017, Viděno: 14. únor 2021. [Online]. Dostupné z: <http://arxiv.org/abs/1704.04503>
- [8] D. B. dos Santos Cesar, C. Gaudig, M. Fritsche, M. A. dos Reis, a F. Kirchner, „An evaluation of artificial fiducial markers in underwater environments“, in *OCEANS 2015 - Genova*, Genova, Italy, kvě. 2015, s. 1–6. doi: 10.1109/OCEANS-Genova.2015.7271491.
- [9] M. Bondy, R. Krishnasamy, D. Crymble, a P. Jasiobedzki, „Space Vision Marker System (SVMS)“, in *AIAA SPACE 2007 Conference & Exposition*, Long Beach, California, zář. 2007. doi: 10.2514/6.2007-6185.
- [10] Xiang Zhang, S. Fronz, a N. Navab, „Visual marker detection and decoding in AR systems: a comparative study“, in *Proceedings. International Symposium on Mixed and Augmented Reality*, Darmstadt, Germany, 2002, s. 97–106. doi: 10.1109/ISMAR.2002.1115078.
- [11] K. Shabalina, A. Sagitov, L. Sabirova, H. Li, a E. Magid, „ARTag, AprilTag and CALTag Fiducial Systems Comparison in a Presence of Partial Rotation: Manual and Automated Approaches“, in *Informatics in Control, Automation and Robotics*, roč. 495, O. Gusikhin a K. Madani, Ed. Cham: Springer International Publishing, 2019, s. 536–558. doi: 10.1007/978-3-030-11292-9_27.
- [12] A. Morar *et al.*, „A Comprehensive Survey of Indoor Localization Methods Based on Computer Vision“, *Sensors*, roč. 20, č. 9, s. 2641, kvě. 2020, doi: 10.3390/s20092641.

- [13] C. B. Owen, Fan Xiao, a P. Middlin, „What is the best fiducial?“, in *The First IEEE International Workshop Augmented Reality Toolkit*, Darmstadt, Germany, 2002, s. 8. doi: 10.1109/ART.2002.1107021.
- [14] M. Fiala, „Designing Highly Reliable Fiducial Markers“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, roč. 32, č. 7, s. 1317–1324, čvc. 2010, doi: 10.1109/TPAMI.2009.146.
- [15] D. G. Lowe, „Object recognition from local scale-invariant features“, in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, 1999, s. 1150–1157 roč.2. doi: 10.1109/ICCV.1999.790410.
- [16] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, a D. Schmalstieg, „Real-Time Detection and Tracking for Augmented Reality on Mobile Phones“, *IEEE Transactions on Visualization and Computer Graphics*, roč. 16, č. 3, s. 355–368, kvě. 2010, doi: 10.1109/TVCG.2009.99.
- [17] Q. H. Gao, T. R. Wan, W. Tang, a L. Chen, „A Stable and Accurate Marker-Less Augmented Reality Registration Method“, in *2017 International Conference on Cyberworlds (CW)*, zář. 2017, s. 41–47. doi: 10.1109/CW.2017.44.
- [18] C.-W. Chen, W.-Z. Chen, J.-W. Peng, B.-X. Cheng, T.-Y. Pan, a H.-C. Kuo, „A Real-Time Markerless Augmented Reality Framework Based on SLAM Technique“, in *2017 14th International Symposium on Pervasive Systems, Algorithms and Networks 2017 11th International Conference on Frontier of Computer Science and Technology 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC)*, Exeter, UK, čer. 2017, roč. 14, s. 127–132. doi: 10.1109/ISPAN-FCST-ISCC.2017.87.
- [19] J. C. P. Cheng, K. Chen, a W. Chen, „Comparison of Marker-Based and Markerless AR: A Case Study of An Indoor Decoration System“, in *Lean and Computing in Construction Congress - Volume 1: Proceedings of the Joint Conference on Computing in Construction*, Heraklion, Crete, Greece, čvc. 2017, s. 483–490. doi: 10.24928/JC3-2017/0231.
- [20] P. Q. Brito a J. Stoyanova, „Marker versus Markerless Augmented Reality. Which Has More Impact on Users?“, *International Journal of Human-Computer Interaction*, roč. 34, č. 9, s. 819–833, zář. 2018, doi: 10.1080/10447318.2017.1393974.
- [21] A. Gupta, K. Bhatia, K. Gupta, a M. Vardhan, „A Comparative Study of Marker-Based and Marker-Less Indoor Navigation in Augmented Reality“, *IRJET*, roč. 05, č. 04, s. 3569–3572, dub. 2018.
- [22] L. Stridbar a E. Henriksson, *Subjective Evaluation of Marker-Based and Marker-Less AR for an Exhibition of a Digitally Recreated Swedish Warship*. Karlskrona, Sweden: Blekinge Institute of Technology, 2019. Viděno: 8. březen 2020. [Online]. Dostupné z: <http://urn.kb.se/resolve?urn=urn:nbn:se:bth-18500>
- [23] C. Rasmussen, K. Toyama, a G. D. Hager, „Tracking Objects By Color Alone“, s. 7, 1996.
- [24] G. A. Thomas, „A versatile camera position measurement system for virtual reality TV production“, in *International Broadcasting Conference (IBC)*, Amsterdam, Netherlands, 1997, roč. 1997, s. 284–289. doi: 10.1049/cp:19971284.
- [25] J. Rekimoto, „Matrix: A Realtime Object Identification and Registration Method for Augmented Reality“, in *Proceedings. 3rd Asia Pacific Computer*

- Human Interaction*, Shonan Village Center, Japan, čvc. 1998, roč. 3. doi: 10.1109/APCHI.1998.704151.
- [26] T. P. Moran, E. Saund, W. van Melle, A. U. Gujar, K. P. Fishkin, a B. L. Harrison, „Design and Technology for Collaborage: Collaborative Collages of Information on Physical Walls", roč. 1, s. 10, 1999.
- [27] H. Kato a M. Billinghurst, „Marker tracking and HMD calibration for a video-based augmented reality conferencing system", in *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, říj. 1999, s. 85–94. doi: 10.1109/IWAR.1999.803809.
- [28] H. KATO, „ARToolKit : Library for Vision-based Augmented Reality", *Technical report of IEICE. PRMU*, roč. 101, č. 652, s. 79–86, úno. 2002.
- [29] M. Fiala, „Comparing ARTag and ARToolkit Plus fiducial marker systems", in *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, říj. 2005, s. 6 pp.-. doi: 10.1109/HAVE.2005.1545669.
- [30] J. Rekimoto a Y. Ayatsuka, „CyberCode: designing augmented reality environments with visual tags", in *Proceedings of DARE 2000 on Designing augmented reality environments - DARE '00*, Elsinore, Denmark, 2000, s. 1–10. doi: 10.1145/354666.354667.
- [31] Y. Cho a U. Neumann, „Multiring Fiducial Systems for Scalable Fiducial-Tracking Augmented Reality", *Presence*, roč. 10, č. 6, s. 599–612, pro. 2001, doi: 10.1162/105474601753272853.
- [32] Xiang Zhang, Y. Genc, a N. Navab, „Taking AR into large scale industrial environments: navigation and information access with mobile computers", in *Proceedings IEEE and ACM International Symposium on Augmented Reality*, New York, NY, USA, 2001, s. 179–180. doi: 10.1109/ISAR.2001.970531.
- [33] Xiang Zhang, Y. Genc, a N. Navab, „Mobile computing and industrial augmented reality for real-time data access", in *ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.01TH8597)*, Antibes-Juan les Pins, France, 2001, roč. 2, s. 583–588. doi: 10.1109/ETFA.2001.997738.
- [34] M. Appel a N. Navab, „Registration of technical drawings and calibrated images for industrial augmented reality", *Machine Vision and Applications*, roč. 13, č. 3, s. 111–118, čvc. 2002, doi: 10.1007/s001380100066.
- [35] D. López de Ipiña, P. R. S. Mendonça, A. Hopper, a A. Hopper, „TRIP: A Low-Cost Vision-Based Location System for Ubiquitous Computing", *Personal and Ubiquitous Computing*, roč. 6, č. 3, s. 206–219, kvě. 2002, doi: 10.1007/s007790200020.
- [36] L. Naimark a E. Foxlin, „Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker", in *Proceedings. International Symposium on Mixed and Augmented Reality*, říj. 2002, s. 27–36. doi: 10.1109/ISMAR.2002.1115065.
- [37] L. Naimark a E. Foxlin, „Fiducial detection system", US7231063B2, 12. červen 2007 Viděno: 18. červenec 2020. [Online]. Dostupné z: <https://patents.google.com/patent/US7231063B2/en>
- [38] T. Kawano, Y. Ban, a K. Uehara, „A coded visual marker for video tracking system based on structured image analysis", in *The Second IEEE and ACM*

- International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, Tokyo, Japan, 2003, s. 262–263. doi: 10.1109/ISMAR.2003.1240711.
- [39] E. Costanza, S. B. Shelley, a J. Robinson, „D-touch: A Consumer-Grade Tangible Interface Module and Musical Applications”, 2003. Viděno: 11. srpen 2020. [Online]. Dostupné z: <https://eprints.soton.ac.uk/270959/>
- [40] E. Costanza a J. Robinson, „A Region Adjacency Tree Approach to the Detection and Design of Fiducials”, s. 7, 2003.
- [41] M. Rohs a B. Gfeller, „Using camera-equipped mobile phones for interacting with real-world objects”, *Advances in Pervasive Computing*, s. 265–271, 2004.
- [42] M. Rohs, „Real-World Interaction with Camera Phones”, in *Ubiquitous Computing Systems*, Berlin, Heidelberg, 2005, s. 74–89. doi: 10.1007/11526858_7.
- [43] D. Claus a A. W. Fitzgibbon, „Reliable Fiducial Detection in Natural Scenes”, in *Computer Vision - ECCV 2004*, roč. 3024, T. Pajdla a J. Matas, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, s. 469–480. doi: 10.1007/978-3-540-24673-2_38.
- [44] D. Claus a A. W. Fitzgibbon, „Reliable Automatic Calibration of a Marker-Based Position Tracking System”, in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, led. 2005, roč. 1, s. 300–305. doi: 10.1109/ACVMOT.2005.101.
- [45] M. Fiala, „ARTag, a Fiducial Marker System Using Digital Techniques”, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, čer. 2005, roč. 2, s. 590–596. doi: 10.1109/CVPR.2005.74.
- [46] A. Dell'Acqua, M. Ferrari, M. Marcon, A. Sarti, a S. Tubaro, „Colored visual tags: a robust approach for augmented reality”, in *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.*, Como, Italy, čvc. 2005, s. 423–427. doi: 10.1109/AVSS.2005.1577306.
- [47] A. C. Rice, A. R. Beresford, a R. K. Harle, „Cantag: an open source software toolkit for designing and deploying marker-based vision systems”, in *Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06)*, Pisa, Italy, bře. 2006, s. 12–21. doi: 10.1109/PERCOM.2006.13.
- [48] J. Mooser, S. You, a U. Neumann, „Tricodes: A Barcode-Like Fiducial Design for Augmented Reality Media”, in *2006 IEEE International Conference on Multimedia and Expo*, Toronto, ON, Canada, čvc. 2006, s. 1301–1304. doi: 10.1109/ICME.2006.262777.
- [49] M. Kaltenbrunner a R. Bencina, „reactIVision: a computer-vision framework for table-based tangible interaction”, in *Proceedings of the 1st international conference on Tangible and embedded interaction - TEI '07*, Baton Rouge, Louisiana, úno. 2007, s. 69. doi: 10.1145/1226969.1226983.
- [50] „reactIVision”, *SourceForge*. <https://sourceforge.net/projects/reactivision/> (viděno 8. srpen 2020).
- [51] K. Tateno, I. Kitahara, a Y. Ohta, „A Nested Marker for Augmented Reality”, in *2007 IEEE Virtual Reality Conference*, bře. 2007, s. 259–262. doi: 10.1109/VR.2007.352495.

- [52] J. Sattar, E. Bourque, P. Giguere, a G. Dudek, „Fourier tags: Smoothly degradable fiducial markers for use in human-robot interaction”, in *Fourth Canadian Conference on Computer and Robot Vision (CRV '07)*, Montreal, QC, Canada, kvě. 2007, s. 165–174. doi: 10.1109/CRV.2007.34.
- [53] A. Xu a G. Dudek, „Fourier Tag: A Smoothly Degradable Fiducial Marker System with Configurable Payload Capacity”, in *2011 Canadian Conference on Computer and Robot Vision*, kvě. 2011, s. 40–47. doi: 10.1109/CRV.2011.13.
- [54] R. Karodia, S. Lee, A. Mehta, a A. Mbogho, „CipherCode: A Visual Tagging SDK with Encryption and Parameterisation”, in *2007 IEEE Workshop on Automatic Identification Advanced Technologies*, Alghero, Italy, čer. 2007, s. 186–191. doi: 10.1109/AUTOID.2007.380617.
- [55] D. Flohr a J. Fischer, *A Lightweight ID-Based Extension for Marker Tracking Systems*. The Eurographics Association, 2007. doi: 10.2312/PE/VE2007Short/059-064.
- [56] Y. Li, Y.-T. Wang, a Y. Liu, „Fiducial marker based on projective invariant for augmented reality”, *Journal of Computer Science and Technology*, roč. 22, č. 6, s. 890–897, lis. 2007.
- [57] M. Hirzer, „Marker Detection for Augmented Reality Applications”, s. 28, 2008.
- [58] A. Mohan, G. Woo, S. Hiura, Q. Smithwick, a R. Raskar, „Bokode: imperceptible visual tags for camera based interaction from a distance”, *ACM Trans. Graph.*, roč. 28, č. 3, s. 1–8, čvc. 2009, doi: 10.1145/1531326.1531404.
- [59] F. Schweiger, B. Zeisl, P. Georgel, G. Schroth, E. Steinbach, a N. Navab, „Maximum Detector Response Markers for SIFT and SURF”, s. 10, 2009.
- [60] H. Bay, T. Tuytelaars, a L. Van Gool, „SURF: Speeded Up Robust Features”, in *Computer Vision – ECCV 2006*, Berlin, Heidelberg, 2006, s. 404–417. doi: 10.1007/11744023_32.
- [61] R. Koch, A. Kolb, C. Rezk-salama (eds, B. Atcheson, F. Heide, a W. Heidrich, „CALTag: High Precision Fiducial Markers for Camera Calibration”, in *Vision, Modeling, and Visualization (2010)*, Norrköping, Sweden, kvě. 2010, roč. 10. doi: 10.2312/PE/VMV/VMV10/041-048.
- [62] H. Wu, F. Shao, a R. Sun, „Research of quickly identifying markers on Augmented Reality”, in *2010 IEEE International Conference on Advanced Management Science (ICAMS 2010)*, čvc. 2010, roč. 3, s. 671–675. doi: 10.1109/ICAMS.2010.5552875.
- [63] V. F. da C. Neto, D. B. de Mesquita, R. F. Garcia, a M. F. M. Campos, „On the Design and Evaluation of a Precise Scalable Fiducial Marker Framework”, in *2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images*, srp. 2010, s. 216–223. doi: 10.1109/SIBGRAPI.2010.37.
- [64] H. Uchiyama a H. Saito, „Random dot markers”, in *2011 IEEE Virtual Reality Conference*, bře. 2011, s. 35–38. doi: 10.1109/VR.2011.5759433.
- [65] E. Olson, „AprilTag: A robust and flexible visual fiducial system”, in *2011 IEEE International Conference on Robotics and Automation*, kvě. 2011, s. 3400–3407. doi: 10.1109/ICRA.2011.5979561.
- [66] J. Wang a E. Olson, „AprilTag 2: Efficient and robust fiducial detection”, in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, říj. 2016, s. 4193–4198. doi: 10.1109/IROS.2016.7759617.

- [67] „AprilTag“. <https://april.eecs.umich.edu/software/apriltag.html> (viděno 8. srpen 2020).
- [68] F. Bergamasco, A. Albarelli, E. Rodolà, a A. Torsello, „RUNE-Tag: A high accuracy fiducial marker with strong occlusion resilience“, in *CVPR 2011*, čer. 2011, s. 113–120. doi: 10.1109/CVPR.2011.5995544.
- [69] F. Bergamasco, A. Albarelli, L. Cosmo, E. Rodolà, a A. Torsello, „An Accurate and Robust Artificial Marker Based on Cyclic Codes“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, roč. 38, č. 12, s. 2359–2373, pro. 2016, doi: 10.1109/TPAMI.2016.2519024.
- [70] F. Bergamasco, A. Albarelli, a A. Torsello, „Pi-Tag: a fast image-space marker design based on projective invariants“, *Machine Vision and Applications*, roč. 24, č. 6, s. 1295–1310, srp. 2013, doi: 10.1007/s00138-012-0469-6.
- [71] Hayet. Belghit, Nadia. Zenati-Henda, Abdelkader. Bellabi, Samir. Benbelkacem, a Mahmoud. Belhocine, „Tracking Color marker using projective transformation for augmented reality application“, in *2012 International Conference on Multimedia Computing and Systems*, Tangiers, Morocco, kvě. 2012, s. 372–377. doi: 10.1109/ICMCS.2012.6320245.
- [72] A. Reuter, H.-P. Seidel, a I. Ihrke, „BlurTags: spatially varying PSF estimation with out-of-focus patterns“, prezentováno v 20th International Conference on Computer Graphics, Visualization and Computer Vision 2012, WSCG'2012, the Czech Republic, čer. 2012, s. 239--247. Viděno: 11. srpen 2020. [Online]. Dostupné z: <https://hal.inria.fr/hal-00876507>
- [73] Y. Li, Y. Chen, R. Lu, D. Ma, a Q. Li, „A novel marker system in augmented reality“, in *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*, Changchun, China, pro. 2012, s. 1413–1417. doi: 10.1109/ICCSNT.2012.6526185.
- [74] C.-H. Chuang, J.-W. Hsieh, a K.-C. Fan, „A smart handheld device navigation system based on detecting visual code“, in *2013 International Conference on Machine Learning and Cybernetics*, Tianjin, China, čvc. 2013, s. 1407–1412. doi: 10.1109/ICMLC.2013.6890803.
- [75] „ArUco: a minimal library for Augmented Reality applications based on OpenCV | Aplicaciones de la Visión Artificial“. <http://www.uco.es/investiga/grupos/ava/node/26> (viděno 8. srpen 2020).
- [76] J. Liu, S. Chen, H. Sun, Y. Qin, a X. Wang, „Real Time Tracking Method by Using Color Markers“, in *2013 International Conference on Virtual Reality and Visualization*, zář. 2013, s. 106–111. doi: 10.1109/ICVRV.2013.25.
- [77] M. Toyoura, H. Aruga, M. Turk, a X. Mao, „Detecting Markers in Blurred and Defocused Images“, in *2013 International Conference on Cyberworlds*, Yokohama, Japan, říj. 2013, s. 183–190. doi: 10.1109/CW.2013.58.
- [78] A. Suzuki, Y. Manabe, a N. Yata, „Design of an AR marker for cylindrical surface“, in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, říj. 2013, s. 293–294. doi: 10.1109/ISMAR.2013.6671811.
- [79] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, a M. J. Marín-Jiménez, „Automatic generation and detection of highly reliable fiducial markers under occlusion“, *Pattern Recognition*, roč. 47, č. 6, s. 2280–2292, čer. 2014, doi: 10.1016/j.patcog.2014.01.005.

- [80] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, a R. Medina-Carnicer, „Generation of fiducial marker dictionaries using Mixed Integer Linear Programming“, *Pattern Recognition*, roč. 51, s. 481–491, bř. 2016, doi: 10.1016/j.patcog.2015.09.023.
- [81] C. N. Klokmoose, J. B. Kristensen, R. Bagge, a K. Halskov, „BullsEye: High-Precision Fiducial Tracking for Table-based Tangible Interaction“, in *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, New York, NY, USA, lis. 2014, s. 269–278. doi: 10.1145/2669485.2669503.
- [82] M. G. Prasad, S. Chandran, a M. S. Brown, „A Motion Blur Resilient Fiducial for Quadcopter Imaging“, in *2015 IEEE Winter Conference on Applications of Computer Vision*, led. 2015, s. 254–261. doi: 10.1109/WACV.2015.41.
- [83] P. Han a G. Zhao, „L-split marker for augmented reality in aircraft assembly“, *Opt. Eng.*, roč. 55, č. 4, s. 043110, dub. 2016, doi: 10.1117/1.OE.55.4.043110.
- [84] L. Calvet, P. Gurdjos, C. Griwodz, a S. Gasparini, „Detection and Accurate Localization of Circular Fiducials under Highly Challenging Conditions“, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, čer. 2016, s. 562–570. doi: 10.1109/CVPR.2016.67.
- [85] „CCTag“. <https://github.com/alicevision/CCTag> (viděno 8. srpen 2020).
- [86] L. Calvet, P. Gurdjos, a V. Charvillat, „Camera tracking using concentric circle markers: Paradigms and algorithms“, in *2012 19th IEEE International Conference on Image Processing*, zář. 2012, s. 1361–1364. doi: 10.1109/ICIP.2012.6467121.
- [87] J. Liu, J. Zhang, J. Mei, a X. Zhang, „CH-Marker: A Color Marker Robust to Occlusion for Augmented Reality“, *International Journal of Pattern Recognition and Artificial Intelligence*, roč. 32, č. 02, s. 1854004, čvc. 2017, doi: 10.1142/S0218001418540046.
- [88] J. DeGol, T. Bretl, a D. Hoiem, „ChromaTag: A Colored Marker and Fast Detection Algorithm“, in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, říj. 2017, s. 1481–1490. doi: 10.1109/ICCV.2017.164.
- [89] H. Wang, Z. Shi, G. Lu, a Y. Zhong, „Hierarchical fiducial marker design for pose estimation in large-scale scenarios“, *Journal of Field Robotics*, roč. 35, č. 6, s. 835–849, bř. 2018, doi: 10.1002/rob.21780.
- [90] H. Uchiyama a Y. Oyamada, „Transparent Random Dot Markers“, in *2018 24th International Conference on Pattern Recognition (ICPR)*, srp. 2018, s. 254–259. doi: 10.1109/ICPR.2018.8545845.
- [91] S. Susan, S. Tandon, S. Seth, Mohd. T. Mudassir, R. Chaudhary, a N. Baisoya, „Kullback-Leibler Divergence based Marker Detection in Augmented Reality“, in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pro. 2018, s. 1–5. doi: 10.1109/CCAA.2018.8777570.
- [92] S. Kullback a R. A. Leibler, „On Information and Sufficiency“, *Ann. Math. Statist.*, roč. 22, č. 1, s. 79–86, bř. 1951, doi: 10.1214/aoms/1177729694.
- [93] J.-K. Huang, M. Ghaffari, R. Hartley, L. Gan, R. M. Eustice, a J. W. Grizzle, „LiDARTag: A Real-Time Fiducial Tag using Point Clouds“, *arXiv:1908.10349 [cs]*, srp. 2019, Viděno: 6. červenec 2020. [Online]. Dostupné z: <http://arxiv.org/abs/1908.10349>

- [94] B. Benligiray, C. Topal, a C. Akinlar, „STag: A stable fiducial marker system", *Image and Vision Computing*, roč. 89, s. 158–169, zář. 2019, doi: 10.1016/j.imavis.2019.06.007.
- [95] B. Benligiray, „STag". <https://github.com/bbenligiray/stag> (viděno 8. srpen 2020).
- [96] F. J. Romero-Ramirez, R. Muñoz-Salinas, a R. Medina-Carnicer, „Fractal Markers: A New Approach for Long-Range Marker Pose Estimation Under Occlusion", *IEEE Access*, roč. 7, s. 169908–169919, lis. 2019, doi: 10.1109/ACCESS.2019.2951204.
- [97] „Fractal Markers | Aplicaciones de la Visión Artificial". <https://www.uco.es/investiga/grupos/ava/node/68> (viděno 8. srpen 2020).
- [98] M. Krogius, A. Haggemiller, a E. Olson, „Flexible Layouts for Fiducial Tags", in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, lis. 2019, s. 1898–1903. doi: 10.1109/IROS40897.2019.8967787.
- [99] G. Yu, Y. Hu, a J. Dai, „TopoTag: A Robust and Scalable Topological Fiducial Marker System", *IEEE Trans. Visual. Comput. Graphics*, s. 1–1, 2020, doi: 10.1109/TVCG.2020.2988466.
- [100] B. Wang, „LFTag: A Scalable Visual Fiducial System with Low Spatial Frequency", *arXiv:2006.00842*, čer. 2020, Viděno: 6. červenec 2020. [Online]. Dostupné z: <http://arxiv.org/abs/2006.00842>
- [101] B. Wang, „kingoflolz/fiducial". <https://github.com/kingoflolz/fiducial> (viděno 11. srpen 2020).
- [102] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, a V. Ferrari, „Extreme clicking for efficient object annotation", *arXiv:1708.02750*, srp. 2017, Viděno: 22. srpen 2020. [Online]. Dostupné z: <http://arxiv.org/abs/1708.02750>
- [103] C. P. Papageorgiou, M. Oren, a T. Poggio, „A general framework for object detection", in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, led. 1998, s. 555–562. doi: 10.1109/ICCV.1998.710772.
- [104] N. Dalal a B. Triggs, „Histograms of oriented gradients for human detection", in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, čer. 2005, roč. 1, s. 886–893 roč. 1. doi: 10.1109/CVPR.2005.177.
- [105] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, a D. Ramanan, „Object Detection with Discriminatively Trained Part-Based Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, roč. 32, č. 9, s. 1627–1645, zář. 2010, doi: 10.1109/TPAMI.2009.167.
- [106] D. E. Rumelhart, G. E. Hinton, a R. J. Williams, „Learning representations by back-propagating errors", *Nature*, roč. 323, č. 6088, Art. č. 6088, říj. 1986, doi: 10.1038/323533a0.
- [107] R. Girshick, J. Donahue, T. Darrell, a J. Malik, „Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", prezentováno v *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, s. 580–587. Viděno: 25. březen 2020. [Online]. Dostupné z: http://openaccess.thecvf.com/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html

- [108] R. Girshick, „Fast R-CNN“, prezentováno v Proceedings of the IEEE International Conference on Computer Vision, 2015, s. 1440–1448. Viděno: 25. březen 2020. [Online]. Dostupné z: http://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html
- [109] S. Ren, K. He, R. Girshick, a J. Sun, „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks“, in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, a R. Garnett, Ed. Curran Associates, Inc., 2015, s. 91–99. Viděno: 25. březen 2020. [Online]. Dostupné z: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>
- [110] J. Redmon, S. Divvala, R. Girshick, a A. Farhadi, „You Only Look Once: Unified, Real-Time Object Detection“, *arXiv:1506.02640 [cs]*, kvě. 2016, Viděno: 24. březen 2020. [Online]. Dostupné z: <http://arxiv.org/abs/1506.02640>
- [111] J. Redmon a A. Farhadi, „YOLO9000: Better, Faster, Stronger“, *arXiv:1612.08242 [cs]*, pro. 2016, Viděno: 25. březen 2020. [Online]. Dostupné z: <http://arxiv.org/abs/1612.08242>
- [112] W. Liu *et al.*, „SSD: Single Shot MultiBox Detector“, in *Computer Vision – ECCV 2016*, Cham, 2016, s. 21–37. doi: 10.1007/978-3-319-46448-0_2.
- [113] M. Košťák, B. Ježek, a A. Slabý, „Color Marker Detection with WebGL for Mobile Augmented Reality Systems“, in *Mobile Web and Intelligent Information Systems*, Cham, srp. 2019, s. 71–84. doi: 10.1007/978-3-030-27192-3_6.
- [114] M. Košťák, B. Ježek, a A. Slabý, „Adaptive Detection of Single-Color Marker with WebGL“, in *Augmented Reality, Virtual Reality, and Computer Graphics*, Cham, 2020, s. 395–410. doi: 10.1007/978-3-030-58465-8_29.

12 Vlastní publikace

- [1] **Kostak, M.**, Dvorak, J., Selamat, A., Krejcar, O.: *Intelligent Notepad for Windows Phone that Uses GPS Data*. In: Younas, M., Awan, I., and Holubova, I. (eds.) *Mobile Web and Intelligent Information Systems*. pp. 90–103. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-65515-4_8.
- [2] **Košťák, M.**, Ježek, B.: *Mobile phone as an interactive device in augmented reality system*. In: DIVAI 2018 (2018).
- [3] **Košťák, M.**, Ježek, B., Slabý, A.: *Color Marker Detection with WebGL for Mobile Augmented Reality Systems*. In: Awan, I., Younas, M., Ůnal, P., and Aleksy, M. (eds.) *Mobile Web and Intelligent Information Systems*. pp. 71–84. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-27192-3_6.
- [4] Berger, A., **Kostak, M.**, Maly, F.: *Mobile AR Solution for Deaf People*. In: Awan, I., Younas, M., Ůnal, P., and Aleksy, M. (eds.) *Mobile Web and Intelligent Information Systems*. pp. 243–254. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-27192-3_19.
- [5] **Košťák, M.**, Ježek, B., Slabý, A.: *Adaptive Detection of Single-Color Marker with WebGL*. In: De Paolis, L.T. and Bourdot, P. (eds.) *Augmented Reality, Virtual Reality, and Computer Graphics*. pp. 395–410. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58465-8_29.
- [6] **Kostak, M.**, Berger, A., Slaby, A.: *Migration of Artificial Neural Networks to Smartphones*. In: Gervasi, O., Murgante, B., Misra, S., Garau, C., Blečić, I., Taniar, D., Apduhan, B.O., Rocha, A.M.A.C., Tarantino, E., Torre, C.M., and Karaca, Y. (eds.) *Computational Science and Its Applications – ICCSA 2020*. pp. 845–858. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58799-4_61.
- [7] Berger, A., **Košťák, M.**, Ježek, B.: *Online Application for Bitcoin Price Visualization*. In: Bentahar, J., Awan, I., Younas, M., and Grønli, T.-M. (eds.) *Mobile Web and Intelligent Information Systems*. pp. 74–81. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-83164-6_6.
- [8] **Košťák, M.**, Slabý, A.: *Designing a Simple Fiducial Marker for Localization in Spatial Scenes Using Neural Networks*. *Sensors*. 21, 5407 (2021). <https://doi.org/10.3390/s21165407>.

Shrnutí publikačních výstupů

	Celkem publikací	Celkem citací	h-index
<i>Google Scholar</i>	8	16	3
<i>Scopus</i>	7	9	2
<i>Web Of Science</i>	5	4	1

ke dni 7. dubna 2023

13 Přehled odborných vědecko-výzkumných aktivit

Účast na projektech specifického výzkumu

- **2018** – Prostorová lokalizace a vizualizace (dr. Kříž)
- **2019** – ICT podpora kognitivních procesů (doc. Poulová)
- **2020** – ICT jako podpůrný nástroj pro kognitivní procesy (doc. Poulová)
- **2021** – ICT jako podpůrný nástroj pro kognitivní procesy (doc. Poulová)
- **2022** – ICT jako podpůrný nástroj kognitivních procesů (doc. Poulová)
- **2023** – IT jako podpůrný nástroj kognitivních procesů (doc. Poulová)

Mojí oblastí zájmu jsou počítačová grafika a strojové učení a moje výše uvedené publikace se proto týkají speciálních problémů patřících do těchto oblastí. Tematicky reagují převážně na různé problémy, které vznikly v rámci výzkumů, které byly stanoveny v rámci projektů SPEV, kterých jsem se účastnil jako řešitel. V souvislosti s tím bylo i postupně konkretizováno téma mé disertační práce. Proto se publikace týkají částečně vybraných problémů spojených s tématem práce a částečně problémů širších. Projekty SPEV byly použity k financování všech mých publikací s projekty SPEV spojenými.

Zahraniční stáže a mezinárodní spolupráce

- **2019** – Seeds For The Future, ČLR
 - 2týdenní program v Pekingu a Shenzhenu
- **2021** – The Kyoto College of Graduate Studies for Informatics, Japonsko
 - 1 rok; březen 2021–únor 2022
 - stáž na pozici lektora, osobní přítomnost od července do ledna, ostatní měsíce v online formě

Účast v projektu Superdoktorand FIM UHK

- **2019** – Rozšíření materiálů pro výuku předmětu Počítačová grafika 3
- **2020** – Implementace umělé neuronové sítě pro detekci jednoduchého jednobarevného markeru

Přílohy

Gitové repositáře

<https://github.com/milankostak/Marker-detection-NN>

- Zdrojový kód pro spuštění a vyhodnocení neuronové sítě
- Zdrojový kód implementace pro správu markerů
- Zdrojový kód pro přečtení informací z detekovaného markeru

<https://github.com/milankostak/Marker-detection>

- Zdrojový kód detekčních algoritmů na GPU
- Zdrojový kód pro přípravu a vyhodnocení datových sad
- Zdrojový kód pro tagování datových sad

Finální natrénovaná umělá neuronová síť

Archiv obsahuje všechny soubory finální natrénované neuronové sítě pro marker „T-křížek“. Spuštění je možné s využitím kódu a pokynů v prvním gitovém repositáři uvedeném výše.

<https://owncloud.cesnet.cz/index.php/s/lrI0bhggqdN32jX> (294 MB)

Datové sady

Z důvodu velikosti souborů odkazy vždy v páru obsahují obrázky a zdrojová videa. Soubor s obrázky dále obsahuje soubor s informacemi o tagovaných markerech.

<https://owncloud.cesnet.cz/index.php/s/sPikux8cNTRfryn> (90 MB)

<https://owncloud.cesnet.cz/index.php/s/kuF78TKcqXqDhrI> (285 MB)

<https://owncloud.cesnet.cz/index.php/s/LqRYmVk7Q0Cih6M> (1,7 GB)

<https://owncloud.cesnet.cz/index.php/s/IZOr4tu05sdUdWD> (2,7 GB)