



## Zadání bakalářské práce

<b>Název:</b>	Automatické vyhodnocení úkolů dětí předškolního věku pomocí obrazové analytiky
<b>Student:</b>	Adam Formánek
<b>Vedoucí:</b>	Ing. Jakub Novák
<b>Studijní program:</b>	Informatika
<b>Obor / specializace:</b>	Znalostní inženýrství
<b>Katedra:</b>	Katedra aplikované matematiky
<b>Platnost zadání:</b>	do konce letního semestru 2022/2023

### Pokyny pro vypracování

Cílem práce je vytvoření aplikace schopné automatizovaně rozpoznat, zda dítě splnilo úkol formou reálného poskládání kostek dle zadání.

Jedná se o navázání na vzdělávací nástroj iSophi SMART (<http://isophi.education/smart/>) a vytvoření automatické zpětné vazby s využitím metod obrazové analýzy. Nástroj využívá tablet k zobrazení úkolu a reálné pomůcky (stavebnice, kostky) k jeho provedení. Práce se zaměřuje na metody měření a vyhodnocení s použitím metod zpracování obrazu.

Úkoly:

- 1) Provedte rešerši v oblasti digitálních vzdělávacích nástrojů a metod snímání a rozpoznání reálných objektů pomocí zpracování obrazu.
- 2) Sestavte měřicí kamerový systém pro snímání materiálu s použitím tabletu.
- 3) Vytvořte metodiku měření včetně specifikace měřených vlastností objektů.
- 4) Navrhněte algoritmy detekce a rozpoznání objektů pomocí metod zpracování obrazu.
- 5) Otestujte výsledky navržených algoritmů a zhodnoťte úspěšnost.
- 6) Vizualizujte výsledky algoritmu na obrazových datech.





**FAKULTA  
INFORMAČNÍCH  
TECHNOLGIÍ  
ČVUT V PRAZE**

Bakalářská práce

# Automatické vyhodnocení úkolů dětí předškolního věku pomocí obrazové analytiky

*Adam Formánek*

Katedra aplikované matematiky  
Vedoucí práce: Ing. Jakub Novák

13. května 2021





---

## Poděkování

Chtěl bych poděkovat Ing. Jakubovi Novákovi za vzorné vedení této práce, přátelský přístup, užitečné rady a postřehy při vývoji i při psaní. Také bych rád poděkoval za podporu své rodině a zejména mladší sestře, která mi pomáhala s testováním.



---

## Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 13. května 2021

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2021 Adam Formánek. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Formánek, Adam. *Automatické vyhodnocení úkolů dětí předškolního věku pomocí obrazové analytiky*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

---

# Abstrakt

Práce se zabývá automatickým vyhodnocením rozložení stavebnice projektu iSophi SMART pomocí tabletu. Nejdříve je navržena metodika měření tak, aby uživatel viděl na displej tabletu, měl prostor na manipulaci se stavebnicí a zároveň, aby tablet viděl na snímanou plochu. Poté je navržen algoritmus využívající tradičních metod zpracování obrazu, jako jsou mean-shift filtrace, barevná segmentace a marker-controlled watershed segmentace, pro získání informací o stavebnici ze snímku. Algoritmus je nakonec implementován v podobě serverové aplikace, která na vstupu přijímá obrázek a požadavek na vyhodnocení. Snímek se zpracuje a ze získaných informací a referenčního řešení se vyhodnotí výsledek. Navržené řešení bylo otestováno při dvou světelných podmínkách na 116 snímcích. Při denním světle je úspěšnost validace 84,48 % a při osvětlení žárovkou v místnosti 88,79 %.

**Klíčová slova** OpenCV, zpracování obrazu, barevná segmentace, klasifikace pozice stavebnice, mean-shift filtrace, iSophi SMART, serverová aplikace, mobilní zařízení

# Abstract

The bachelor thesis focuses on the automatic evaluation of building kit parts placement using a mobile device such as a tablet. The building kit is a part of iSopfi SMART project. First, measurement methodology is designed in the way the user is able to view the tablet's display, reach for building kit parts and still see the whole scene. Then an algorithm, which extracts information from the image about the building kit parts, is implemented by using traditional image processing methods such as mean-shift filtration, color thresholding, and marker-controlled watershed segmentation. Lastly, the algorithm is implemented in the form of a server application, which takes a request to validate an input image. Information from the image is compared to the reference solution of the task, and the result is returned as a response. The proposed application was tested on 116 images under two light conditions. The success rate under the daylight and the room light is 84,48 % and 88,79 % respectively.

**Keywords** OpenCV, image processing, color segmentation, bulding kit parts position classification, mean-shift filtration, iSopfi SMART, server application, mobile device

---

# Obsah

Úvod	1
<b>1 iSopfi SMART a konkurence</b>	<b>3</b>
1.1 LEGO Education . . . . .	3
1.2 PRIMO . . . . .	3
1.3 littleBits . . . . .	4
<b>2 Výzkum problematiky</b>	<b>5</b>
2.1 Metody snímání . . . . .	6
2.2 Barevná segmentace součástí . . . . .	6
2.2.1 Barevné prahování . . . . .	6
2.2.2 Mean-shift segmentace . . . . .	6
2.3 Počet součástí . . . . .	7
2.3.1 Segmentace pomocí Houghovy transformace . . . . .	7
2.3.2 Marker-controller watershed segmentace . . . . .	8
2.4 Rozpoznání objektů pomocí tvarové charakteristiky . . . . .	9
2.5 Měření rozměrů objektů aproximací na kvádr . . . . .	10
<b>3 Zpracování obrazu</b>	<b>13</b>
3.1 Barevné prostory . . . . .	13
3.1.1 Šedotónové . . . . .	13
3.1.2 Barevné . . . . .	13
3.2 Tradiční metody zpracování obrazu . . . . .	14
3.2.1 Mean-shift filtrace . . . . .	14
3.2.2 Segmentace . . . . .	15
3.2.2.1 Prahování . . . . .	16
3.2.2.2 Marker-controlled watershed segmentace . . . . .	16
3.2.2.3 Segmentace pomocí Houghovy liniové transformace . . . . .	17
3.2.3 Prostorové a centrální momenty . . . . .	18

3.2.4	Kontury . . . . .	18
3.2.4.1	Aproximace pomocí Ramer–Douglas–Peucker algoritmu . . . . .	18
3.2.4.2	Konvexní obálka . . . . .	19
3.2.4.3	Tvarové charakteristiky . . . . .	19
3.3	Perspektiva . . . . .	20
3.3.1	Homogenní souřadnicová soustava . . . . .	20
3.3.2	Transformace . . . . .	21
3.3.3	Úběžnice a úběžník . . . . .	22
3.4	K-means . . . . .	22
3.5	Tesseract OCR . . . . .	22
3.6	Morfologické operace . . . . .	22
<b>4</b>	<b>Sestavení aparatury a metodika měření</b>	<b>25</b>
4.1	Snímací aparatura . . . . .	25
4.1.1	Pozadí . . . . .	26
4.1.2	Upevnění snímacího zařízení a stojan . . . . .	26
4.1.3	Umístění zařízení . . . . .	27
4.2	Sestavení aparatury . . . . .	28
<b>5</b>	<b>Návrh řešení algoritmů</b>	<b>31</b>
5.1	Úprava perspektivy . . . . .	31
5.2	Sjednocení barev . . . . .	32
5.3	Segmentace objektů . . . . .	32
5.4	Algoritmy . . . . .	33
5.4.1	Logické řady a barevné matice . . . . .	34
5.4.2	Rovnice a počítání . . . . .	34
5.4.3	Tvary a velikosti . . . . .	36
5.4.4	Zrakové vnímání . . . . .	39
5.5	Zbylé úkoly . . . . .	41
<b>6</b>	<b>Implementace</b>	<b>43</b>
6.1	Použité nástroje . . . . .	43
6.2	Rozhraní programu . . . . .	44
6.3	Databáze referenčních řešení . . . . .	45
6.3.1	Validators . . . . .	46
6.3.2	Tasks . . . . .	46
6.4	Třída Validator . . . . .	48
6.4.1	Získání informací ze snímku . . . . .	48
6.4.2	Validace získaných informací . . . . .	48
6.4.3	Porovnání výsledků . . . . .	48
<b>7</b>	<b>Výsledky</b>	<b>49</b>
7.1	Zhodnocení výsledků . . . . .	49



7.2 Problémy . . . . .	50
<b>8 Diskuze</b>	<b>53</b>
<b>Závěr</b>	<b>55</b>
<b>Literatura</b>	<b>57</b>
<b>A Seznam Použitých zkratk</b>	<b>63</b>
<b>B Obsah přiloženého CD</b>	<b>65</b>



---

## Seznam obrázků

1.1	Ukázka konkurenčních nástrojů. . . . .	4
2.1	Typy součástek rozdělených do skupin podle aplikace iSopli SMART [1]. <b>(a)</b> Algoritmické myšlení. <b>(b)</b> Matematické dovednosti. <b>(c)</b> Prostorová orientace. <b>(d)</b> Zrakové vnímání. . . . .	5
2.2	Segmentace ruky s parametry $(h_s, h_r, M) = (16, 19, 40)$ [2]. . . . .	7
2.3	Segmentace krvinek pomocí HT [3]. . . . .	8
2.4	Segmentace pomocí MCW [4]. . . . .	9
2.5	Postup [5]. . . . .	10
2.6	Aproximace na kvádr z [6]. <b>(a)</b> Maska rozdělená osou, která vede z úběžníku $q_c$ . Body $q_1$ a $q_2$ reprezentují nejvzdálenější body od osy. <b>(b)</b> Aproximovaný kvádr. . . . .	11
3.1	Reprezentace pixelu v jednotlivých barevných prostorech [7]. . . . .	14
3.2	Šedotónový obrázek a jeho reprezentace v grafu [8]. . . . .	15
3.3	Ukázka mean-shift filtrace [2]. . . . .	15
3.4	Segmentace zelené barvy prahováním. . . . .	16
3.5	Ukázka MCW segmentace [9]. <b>(a)</b> Původní šedotónový obrázek. <b>(b)</b> Topografická reprezentace. <b>(c)</b> Watershed bez značek. <b>(d)</b> Watershed se značkami. . . . .	17
3.6	Ukázka segmentace pomocí houghova algoritmu [8]. . . . .	17
3.7	Příklad eroze a dilatace na masce [10]. . . . .	23
4.1	Porovnání různých pozadí scény. . . . .	26
4.2	Stojan, snímací zařízení a zrcátko pro snímání. . . . .	26
4.3	Porovnání umístění a pohledů tabletu. Každý sloupec obsahuje sestavenou aparaturu a pohled tabletu. <b>(a)</b> Z pohledu uživatele. <b>(b)</b> Ze strany bez zrcátka. <b>(c)</b> Ze strany se zrcátkem. . . . .	27
4.4	Plátno s orientačními body a vyznačenými vzdálenostmi. . . . .	28
5.1	Ukázka perspektivní transformace. . . . .	32

5.2	Originální obrázek k segmentaci viz 5.3. . . . . .	33
5.3	Masky vzniklé po barevné segmentaci. V pořadí červená, modrá, žlutá a zelená. . . . .	33
5.4	Maska vytvořená spojením masek z obrázku 5.3. . . . .	33
5.5	Ukázka součástí ze skupiny 5.4.2. . . . .	34
5.6	Průběh počítání koleček pomocí MCW. <b>(a)</b> Originální obrázek. <b>(b)</b> Masky po barevné segmentaci. <b>(c)</b> Masky po erozi a „zaplnění“ otvorů. <b>(d)</b> Kontury po MCW segmentaci. . . . .	35
5.7	Rozdělení koleček z obrázku 5.6 do clusterů pomocí k-means. . . . .	35
5.8	Oříznuté čtverečky a jejich masky. . . . .	36
5.9	Ukázka součástí ze skupiny 5.4.3. . . . .	36
5.10	Barevné součástky a jejich masky. . . . .	37
5.11	Aproximace kontur dřevěných součástí. . . . .	38
5.12	Ukázka součástí ze skupiny 5.4.4. . . . .	39
5.13	Ukázka zpracování modré krychle. . . . .	40
5.14	Ukázka zpracování trojúhelníkového hranolu. . . . .	40
5.15	Ukázka neřešitelných typů úloh. <b>(a)</b> Zrakové rozlišování podobností a směrů. <b>(b)</b> Nápodoba sestavy. . . . .	41
5.16	Ukázka segmentace čtverečků pomocí HT. . . . .	42
6.1	Vývojový diagram serverové aplikace. . . . .	43
6.2	Formát odpovědi serveru pro správné řešení logické řady. . . . .	44
6.3	Formát odpovědi serveru pro špatné řešení rovnice. . . . .	45
6.4	Objekt reprezentující typ úkolu, zde řazení čísel. . . . .	45
7.1	Počet správně a špatně vyhodnocených snímků jednotlivými validátory. <b>(a)</b> Snímky osvětlené žárovkou v místnosti. <b>(b)</b> Snímky osvětlené denním světlem. . . . .	50
7.2	Procentuálně vyjádřen počet správných vyhodnocení ku všem testům z dané skupiny. <b>(a)</b> Snímky osvětlené žárovkou v místnosti. <b>(b)</b> Snímky osvětlené denním světlem. . . . .	50

---

# Úvod

Vzdělávací nástroj iSophi SMART spojuje tradiční stavebnice a tablet za účelem rozvoje rozumové oblasti dětí předškolního věku pro 21. století. Nástroj se využívá zejména ve školkách a snaží se rozvíjet jak myšlení tak samostatnost. Na tabletu je zadání, které dítě musí splnit. Většinou jde o podobu sestavení, matematické operace nebo označení součástky. Nyní probíhá kontrola splněných úkolů buď dítětem samotným nebo dozorujícím člověkem. Posouzení správnosti od uživatele může vést k nevědomému špatnému řešení nebo podvádění. Zároveň pokud je skupina dětí větší, kontrola může být pro učitele časově náročná. Opakovaný zásah učitele není žádoucí, jelikož by si děti měly s procesem poradit samy. Bylo by tedy vhodné přenést zodpovědnost způsobu opravování a zpětné vazby na zařízení, se kterým dítě pracuje.

Hlavním cílem práce je vytvořit serverovou aplikaci, která dokáže zautomatizovat vyhodnocení rozložení stavebnice nástroje SMART pomocí tradičních metod zpracování obrazu. Snímací zařízení je tablet, který se využívá pro zobrazení zadání uživateli.

Práce se zabývá metodikou měření, návrhem algoritmu a závěrečným nasazením do serverové aplikace. Významnou částí zpracování obrazu je sestavení snímací aparatury, která zajistí dosažení co nejkonzistentnějších podmínek snímání, což je jedním z hlavních předpokladů správné validace. U aparatury je důležité, aby uživatel viděl dobře na obrazovku tabletu a zároveň měl volný přístup ke snímané ploše, kam se stavebnice umísťuje. Algoritmus by měl být schopný ze snímku získat informace o rozložení součástek, které budou využity pro následnou validaci. Validaci zajišťuje serverová aplikace, která na vstupu přijme požadavek se zadáním a porovná výsledky s referenčním řešením.



---

# iSophi SMART a konkurence

Vzdělávací nástroj iSophi SMART, cílený na děti předškolního věku, se řadí do celosvětového trendu podpory STEM (Science, Technology, Engineering and Mathematics) [11]. Jinými slovy se zaměřuje na rozvoj vzdělávání v oborech přírodní vědy, techniky, technologií a matematiky. Nástroj SMART navíc rozvíjí i informatické a algoritmické myšlení.

Myšlenka nástroje spočívá v práci dítěte s tabletem a stavebnicí. Dětem se na tabletu prezentuje zadání, které mají splnit. Jednotlivá zadání se dělí do skupin, zaměřených na jiná odvětví problémů. Každá skupina využívá specifické součástky. Není to ale jediný nástroj na trhu s podobným cílem.

## 1.1 LEGO Education

Program, který přistupuje k rozvoji dětí za pomoci známé stavebnice značky LEGO [12]. Dítě si může se stavebnicí hrát samo nebo pod dozorem učitele. Nástroj poskytuje zároveň i různé úkoly a cvičení, které mohou děti zpracovávat. K tomu slouží aplikace dostupná na mobilní i desktopové platformě [13], nebo web s předpřipravenými plány [14]. Výhodou je propracovaný software a možnost programování součástek. Principiálně jde o velmi podobný přístup jako SMART, tedy interakce chytrého zařízení, stavebnice a dětí.

## 1.2 PRIMO

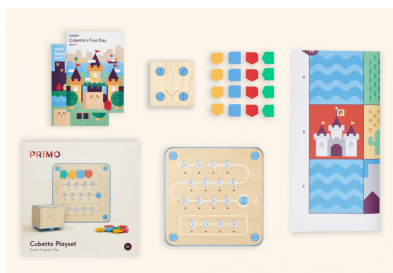
Firma, která je výrobcem vzdělávací sady Cubetto [15]. Sada obsahuje robota, podložku, součástky na programování a příručku. Dítě může pomocí podložky a součástek programovat pohyb robota. Podložky se od sebe různě liší a každá je zaměřená na jiný druh myšlení dítěte. Výhoda pro provoz stavebnice je, že nevyžaduje žádný software.

### 1.3 littleBits

Nástroj se skládá z malých elektronických komponent, které se k sobě magneticky připevňují [16]. Dítě má možnost zkusit různé kombinace komponent podle své fantazie, nebo může postupovat podle přiložené brožurky. K dispozici je i mobilní aplikace, která slouží jako návod i jako inspirace pro tvoření [17].



(a) LEGO Education [18].



(b) Cubetto [19].



(c) littleBits [20].

Obrázek 1.1: Ukázka konkurenčních nástrojů.

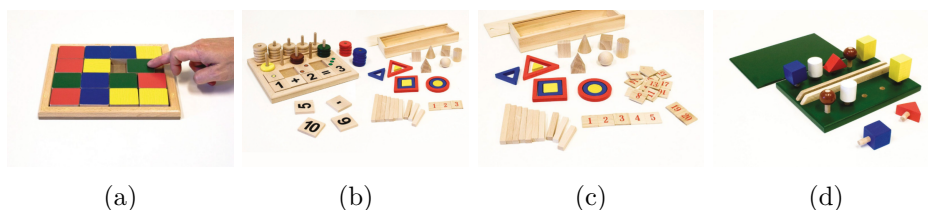


## Výzkum problematiky

Celá stavebnice je rozdělena do modré, červené a zelené tašky. Součástky jsou pro jednotlivé tašky specifické a jsou zaměřeny na jiné odvětví myšlení dítěte. Web iSophi SMART [11] poté dělí stavebnici na pět rozvíjených oblastí:

- algoritmické myšlení – modrá taška
- matematické představy – červená taška
- prostorová orientace – červená taška
- zrakové vnímání – zelená taška
- časová orientace – zelená taška

Jelikož je stavebnice učena pro děti předškolního věku, nevyskytují se zde žádné složité součástky. Zároveň jsou všechny součástky nabarvené, což pomáhá dětem při rozeznávání a plnění úkolů. Každý úkol má také jasně stanovené zadání a druh součástek, který se při řešení využívá.



Obrázek 2.1: Typy součástek rozdělených do skupin podle aplikace iSophi SMART [1]. **(a)** Algoritmické myšlení. **(b)** Matematické dovednosti. **(c)** Prostorová orientace. **(d)** Zrakové vnímání.

Výše uvedené vlastnosti lze využít při hledání správných metodik pro implementaci. Z hlediska metod snímání, zpracování obrazu a vyhodnocení správnosti stavebnice se problémy rozdělí do následujících sekcí.

## 2.1 Metody snímání

Úloha nastavení měřicí aparatury a zvolení správných metod snímání je složitá a individuální v závislosti na případě užití. Nyní se pracovní scéna skládá ze stavebnice a tabletu, kde se uživateli zobrazuje zadání. Z důvodu zachování stejných komponent a úspory je žádoucí, aby byl ke snímání využit zmíněný tablet. Kvůli tomuto omezení se počet možností značně snižuje, jelikož tablet slouží jako snímací zařízení a zároveň prezentuje zadání. Využít se tedy musí zadní kamera.

Kromě klasického pohledu zadní kamery tabletu se nabízí modifikace pomocí zrcátka [21]. To umožní vidět scénu z jiného úhlu bez potřeby natáčet zařízení jiným směrem.

Konfigurace a testování aparatury je popsáno v podsekcí 4.1.3.

## 2.2 Barevná segmentace součástí

Součástky jsou, až na pár výjimek, nabarvené jednolitou barvou. Mezi ně patří součástky dřevěné a tvarové, které mají nabarvený jen rub a líc. Dává proto smysl se zaměřit na segmentaci, která se opírá o barevnou charakteristiku obrázku.

### 2.2.1 Barevné prahování

Barevné prahování je velmi rychlá a naivní metoda pro barevnou segmentaci. Kromě barevné reprezentace obrázku se metoda od klasického prahování nijak neliší — dá se realizovat v libovolném barevném prostoru.

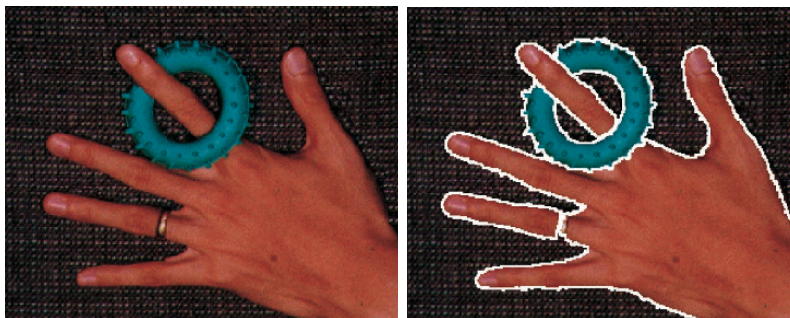
Nejčastější barevná reprezentace obrazu je RGB. Prostor je vhodný pro elektronická zařízení, ale na segmentaci a další zpracování se nehodí. Je tomu tak, kvůli vysoké korelaci mezi kanály R, G a B. Jinak řečeno, pokud se změní jas obrázku, má to dopad na všechny kanály [7].

Vhodný prostor z hlediska zpracování je HSV. Kanál Hue, který reprezentuje barvu, je užitečný při segmentaci barevných objektů. Zároveň oproti RGB nemá jas na barvu ani saturaci vliv, jelikož je reprezentován samostatným kanálem. Nevýhodou je nestabilita hodnot při nižších saturacích [7].

### 2.2.2 Mean-shift segmentace

Metoda se opírá o mean-shift filtraci obrázku a následnou kategorizaci hodnot pixelů. Kategorizace probíhá na základě parametrů  $h_s$ ,  $h_r$  a  $M$ , které se manuálně určí. Parametry  $h_s$  a  $h_r$  vyjadřují, v jak velkém okně se hledá hodnota pixelu, a hranici, ve které se barvy ještě považují za stejné. Po dokončení jednotlivých skupin se odeberou ty, které obsahují méně pixelů, než stanovená hranice  $M$ . Výsledkem je rozdělení obrázku do skupin podle barev [22, 2].

Jedná se o efektivní způsob segmentace, který si v závislosti na zvolené parametry poradí i s nekonzistencí barev a textur. Nevýhoda je časová náročnost a nalezení správných parametrů.



(a) Originální obrázek.

(b) Vysegmentovaný obrázek.

Obrázek 2.2: Segmentace ruky s parametry  $(h_s, h_r, M) = (16, 19, 40)$  [2].

## 2.3 Počet součástí

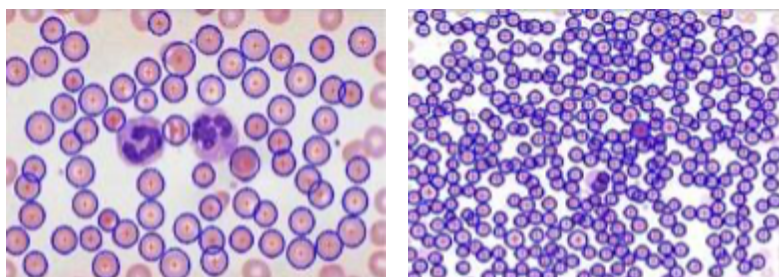
V podúlohách matematických operací se lze setkat s problémem počítání součástí. Uživatel dostane za úkol přiřadit správný počet koleček k danému číslu nebo seřadit hromádky s kolečky. Zde barevná charakteristika nepomůže, jelikož barva o součástkách nic nevyovídá. Také se dost často dotýkají. Pokud se budou kolečka stejné barvy dotýkat, výsledek segmentace bude jedna maska reprezentující více součástí, viz sekce 2.2.

Jelikož barevná složka v tomto případě při počítání nepomůže, je vhodné na tento problém aplikovat jiné metody segmentace. Hlavním cílem je správně rozpoznat objekty stejné barvy, které se mohou dotýkat. Následující metody jsou využívány právě pro řešení tohoto problému.

### 2.3.1 Segmentace pomocí Houghovy transformace

Jednou z možností je využití Houghovy transformace (dále HT) pro segmentaci. Tento typ segmentace se opírá o hranovou detekci v obrázku. Ze získané masky se poté zrekonstruují hrany. Tím se odizolují jednotlivé objekty, které tak lze spočítat.

Algoritmus najde využití v odvětví biologie. Příkladem je určení počtu červených krvinek pomocí HT [3], viz obrázek 2.3. Obrázek bylo potřeba nejdříve předzpracovat, aby měla segmentace co nejlepší výsledky (filtrace, hranové detekce a ekvalizace histogramu). Následně se na předzpracovaný obrázek použila HT. Z nalezených buněk se červené krvinky vyfiltrovaly na základě jejich velikosti a tvaru.



Obrázek 2.3: Segmentace krvinek pomocí HT [3].

Počet červených krvinek v tabulce 2.1 není přesný, jedná se o aproximaci na základě počtu buněk na snímku. Výsledek je porovnáván s manuálním počítáním.

Tabulka 2.1: Porovnání počtu červených krvinek metodou manuálního a automatizovaného počítání za použití HT [3].

Vzorek č.	Počet krvinek manuálně v jedné kapce (v milionech)	Počet krvinek za použití HT v jedné kapce (v milionech)
1	2,8	2,38
2	5,82	5,00
3	3,12	3,02
4	6,40	6,22
5	9,30	9,12

Výsledky algoritmu jsou celkem slibné. I přes to, že je algoritmus schopný rozeznat buňky, které se překrývají, není schopen určit ty buňky, které jsou na obrázku zastoupeny pouze částečně. To se děje kvůli filtraci objektů na základě velikosti a tvaru [3].

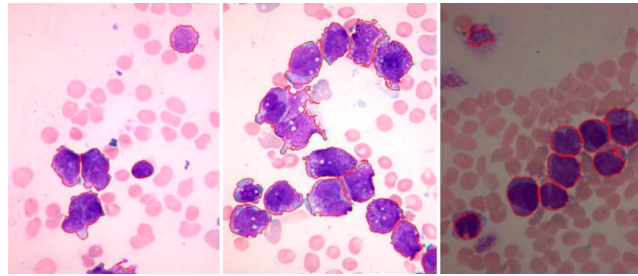
### 2.3.2 Marker-controller watershed segmentace

Segmentace pomocí marker-controller watershed (dále MCW) je metoda, která se opírá o watershed transformaci. Příklad využití lze nalézt opět v biologii. Navržené řešení využívá MCW pro segmentaci bílých krvinek [4], viz obrázek 2.4.

Nejdříve byly zjištěny jádra bílých krvinek, které slouží jako markery pro MCW segmentaci. Po nalezení markerů byla získána maska mean-shift segmentací. Na získanou masku byl následně použit MCW algoritmus, který našel jednotlivé bílé krvinky [4].

V tabulce 2.2 jsou zobrazeny kvantitativní výsledky algoritmu v porovnání s ostatními, které byly použity v [4]. Tučně vyznačená čísla reprezentují nejlepší výsledky.

## 2.4. Rozpoznání objektů pomocí tvarové charakteristiky



Obrázek 2.4: Segmentace pomocí MCW [4].

Lze vidět, že navržené řešení vyšlo nejlépe ze všech porovnávaných. Algoritmus poskytuje robustní a uspokojivé výsledky.

Tabulka 2.2: Porovnání úspěšnosti algoritmů [4].

	<i>TP</i>	<i>FP</i>	<i>FN</i>	<i>OVER SEGM</i>	<i>UNDER SEGM</i>
Navržené řešení	<b>1481</b>	<b>10</b>	<b>69</b>	19	<b>44</b>
K-means shlukování	1332	>800	218	<b>9</b>	511
Barevná a tvarová transformace	3	545	778	25	73
SLIC	4	545	18744	286	57

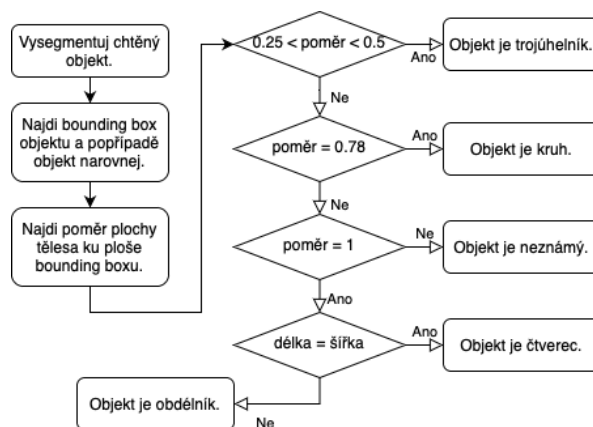
## 2.4 Rozpoznání objektů pomocí tvarové charakteristiky

Celá komponenta prostorové orientace je zaměřená na rozpoznání velikosti a tvaru objektů. Jedná se o jednoduché geometrické tvary jako trojúhelník, čtverec atd. Každá velikost je reprezentována specifickou barvou, například malý čtverec nebo trojúhelník mají zelenou barvu. Jelikož se jedná o jednoduché pravidelné tvary, dává smysl uvažovat použití tvarové charakteristiky pro identifikaci objektů.

Užití metody lze najít v [5]. Nejprve se získala maska objektu, z které lze vypočítat tvarové charakteristiky jako je poměr stran, konvexita nebo třeba kulatost. Aby byly charakteristiky konzistentní, musí být objekt pokaždé ve stejné pozici. Zkontrolovala se tedy rotace a popřípadě adekvátně upravila maska. Z experimentálních výsledků se poté určila hranice tvarových charakteristik a pomocí nich se zjistilo, do jaké kategorie objekt spadá. Postup je vidět na obrázku 2.5.

Jedná se o efektivní a jednoduchou metodu pro klasifikaci triviálních geometrických tvarů [5].

Typy tvarů nejsou omezeny na výše zmíněné. Tvarových charakteristik je spousta a lze jimi popsat nespočet objektů [23].



Obrázek 2.5: Postup [5].

## 2.5 Měření rozměrů objektů aproximací na kvádr

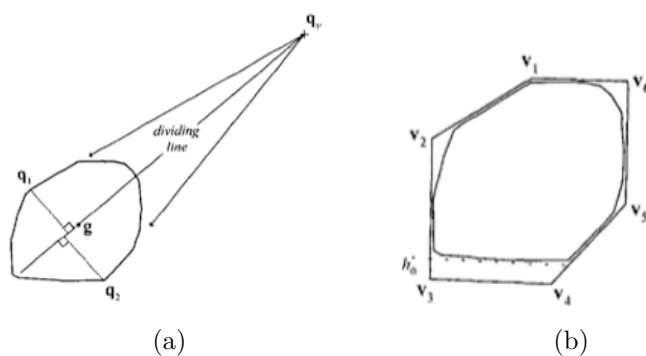
Poslední podproblém při vyhodnocování správného řešení stavebnice je měření a rozpoznání objektů, které nejsou ploché. Při měření se musí uvažovat výškový rozměr, jedná se tedy o 3D objekt reprezentovaný ve 2D obrázku. Nelze proto spoléhat na tvarové charakteristiky, které se mohou výrazně lišit na základě pozice součástky vůči kameře.

Pokud by se zjistily rozměry objektu, lze rozlišit o jaký tvar a typ součástky se jedná. Fotoaparát na mobilním zařízení ale většinou neobsahuje hloubkový senzor.

Jedním z přístupů, kterým lze změřit rozměry objektu bez znalosti hloubkové informace, je aproximace objektu na kvádr [6]. Metoda se opírá o znalost pozice kamery vůči světu a její kalibrace, tedy o znalost vnitřních a vnějších parametrů.

Nejdříve se vysegmentuje chtěný objekt. V masce se pomocí rovnoběžek, které tvoří okraje objektu, najde úběžník. Osa úhlu vytvořeného rovnoběžkami rozděluje objekt na dvě části. V každé se nalezne nejvzdálenější bod v masce od osy. Pomocí těchto znalostí lze zkonstruovat kvádr okolo kontury. Rozměry kvádrů jsou změřeny v pixelech a inverzní transformační maticí, z parametrů kamery, přepočítány na skutečné hodnoty [6].

Vzdálenost tvoří celkem přesný odhad rozměru skutečného objektu, i když se jedná o aproximaci. Potvrzují to i hodnoty z tabulky 2.3.



Obrázek 2.6: Aproximace na kvádr z [6]. **(a)** Maska rozdělená osou, která vede z úběžníku  $q_c$ . Body  $q_1$  a  $q_2$  reprezentují nejvzdálenější body od osy. **(b)** Aproximovaný kvádr.

Tabulka 2.3: Porovnání reálných a odhadnutých rozměrů osobního automobilu [6].

	<i>Hloubka</i>	<i>Šířka</i>	<i>Výška</i>
Reálná vzd.	4820 mm	1765 mm	1465 mm
Odhadnutá vzd.	4462 mm	1575 mm	1306 mm
Error	7,42 %	10,73 %	10,85 %





---

# Zpracování obrazu

Zpracování obrazu je disciplína, která se zabývá prací se signálem a zpracováním obrazových dat. Díky zpracování obrazu je možné obraz upravovat nebo z něj získat informace. V sekcích a podsekcích níže jsou shrnuty některé tradiční metody zpracování obrazu, které jsou využity dále v textu.

## 3.1 Barevné prostory

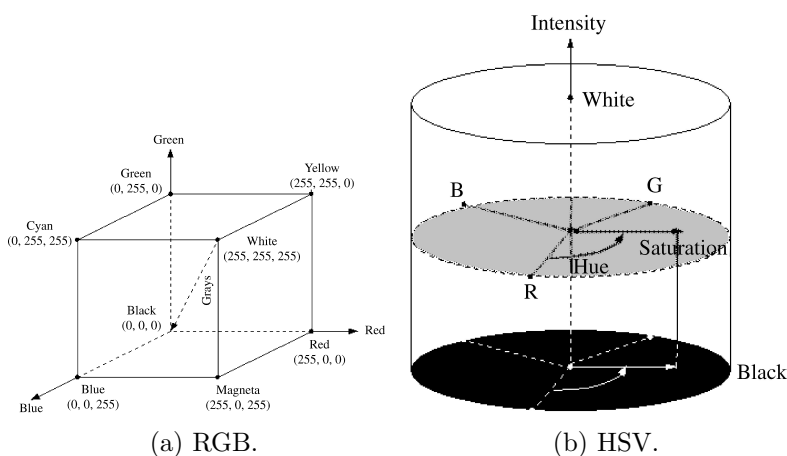
Reprezentace obrazu se liší podle barevného spektra. Nejpřirozenější je rozdělit je do dvou kategorií.

### 3.1.1 Šedotónové

Obraz je reprezentován maticí o dvou dimenzích. Každý pixel je tedy zastoupen hodnotou z intervalu od 0–255, kde 0 znamená černou barvu a 255 bílou. Obecně tedy platí, že čím větší číslo, tím bělejší pixel. Veškeré hodnoty jsou celočíselné — pixel jich může nabývat až 256. Interval je zvolený z hlediska datové reprezentace (1 px = 1 B) a vnímavosti lidského oka.

### 3.1.2 Barevné

Barva je údaj navíc, který nelze vyřešit tak jako v předešlém případě. Proto je barevný obraz řešen maticí o třech dimenzích. Pixel zde není jedno číslo, ale vektor. Na základě typu barevné škály mají hodnoty vektoru různý význam. Mezi nejznámější barevné škály patří například RGB (Red–Green–Blue) nebo HSV (Hue–Saturation–Value). Jak už z názvu vyplývá, jeden pixel v RGB je vektor, kde první hodnota značí míru zastoupení červené barvy, druhá míru zastoupení zelené barvy a třetí míru zastoupení barvy modré. Obdobně tomu je i u HSV. Zde ale hodnoty vektoru značí v pořadí míru zastoupení barvy, hustoty a světlosti.



Obrázek 3.1: Repräsentace pixelu v jednotlivých barevných prostorech [7].

HSV má oproti RGB velkou výhodu. Barva je namísto vektoru značena úhlem v jednotkové kružnici. Nevýhodnou mohou být například nestabilní hodnoty barev při nízké saturaci kvůli nelineární transformaci mezi barevnými prostory [7].

## 3.2 Tradiční metody zpracování obrazu

Tradiční metody se zabývají zpracováním a získáním informací z obrázku. Ty se většinou opírají o nějaké specifické vlastnosti jako je tvar, barva nebo jasová hodnota. Stejně jako pro získání informace lze tyto metody použít na úpravu nebo částečnou opravu snímku.

Postup zpracování obrazu není nutně daný. Je ale dobré se držet nějakého základního schématu. Jedním z takových schémat může být posloupnost filtrace, segmentace a získání informace, například s použitím kontur. Některé metody jsou v následujících podsekcích v pořadí vysvětleny.

### 3.2.1 Mean-shift filtrace

Cílem algoritmu je sjednotit hodnotově podobné oblasti ve snímku. Výsledek mean-shift filtrace je jeho „animovaná“ verze.

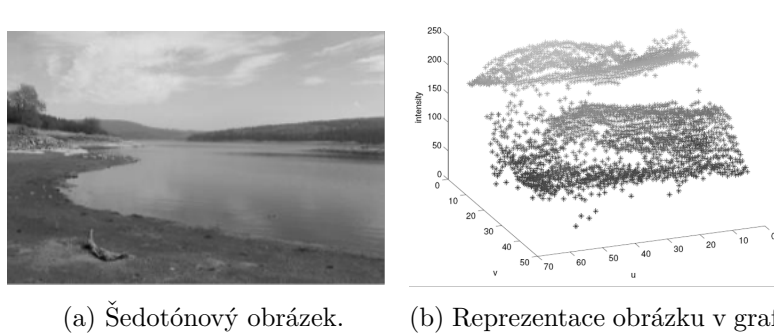
Obrázek je potřeba popsat grafem o  $d$  dimenzích, kde  $d$  závisí na reprezentaci obrázku. Pro dimenzi platí, že [22]:

$$d = \text{počet hodnot reprezentující pixel} + 2. \quad (3.1)$$

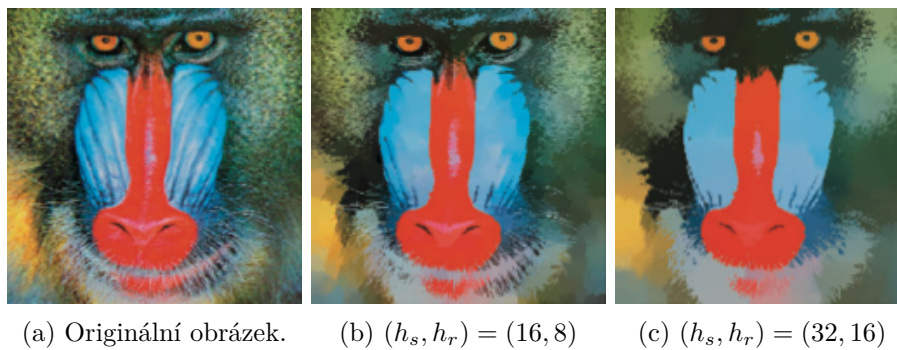
Například u šedotónového obrázku se bude jednat o graf o třech dimenzích (jeden bod =  $(x, y, I(x, y))$ , kde  $I(x, y)$  značí jas bodu) [8]. Na tento graf se následně použije filtrace.

Nechť  $R = \{x_i\}_{i=1,\dots,n}$  je libovolná množina o  $n$  bodech v euklidovském prostoru  $R^d$ . Dále buď oblast  $S_h(x)$  o  $n_k$  bodech, vycentrová na bod  $x$ . Algoritmus v každé iteraci vypočítá v tomto okolí medián, na který se následně vycentruje. Tím se okolí posune a operace se opakuje, dokud se nové středy nestabilizují. Při filtraci se takovýmto způsobem projdou všechny body z množiny  $R$  a přiřadí se jim hodnota nalezeného stabilizovaného středu [22].

Velikost okolí a hranice, která určuje, zda jsou barvy dostatečně hodnotově podobné, jsou reprezentovány parametry  $h_s$  a  $h_r$ . Tyto parametry mají velký dopad na míru filtrace, viz obrázek 3.3.



Obrázek 3.2: Šedotónový obrázek a jeho reprezentace v grafu [8].



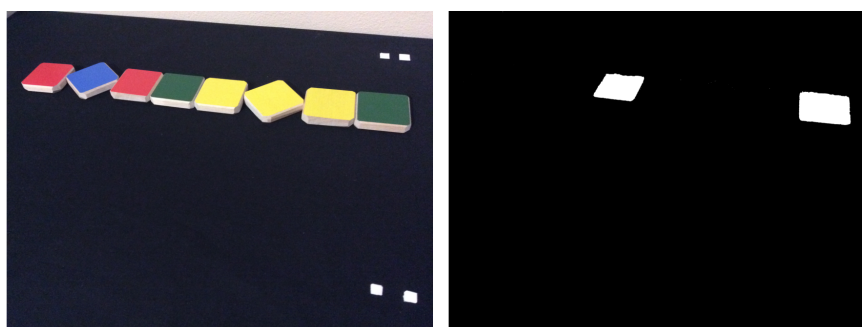
Obrázek 3.3: Ukázka mean-shift filtrace [2].

### 3.2.2 Segmentace

Segmentace je jedna z klíčových částí zpracování obrazu, jelikož značí oblast zájmu. Vysegmentovaná oblast se reprezentuje binární maskou, což není nic jiného než černobílý obrázek. Bílá část značí popředí a černá pozadí. Pokud oblast zájmu zcela neodpovídá sledovanému objektu, může dojít k chybnému zpracování.

### 3.2.2.1 Prahování

Jedná se o jednu ze základních metod segmentace. Nejprve se určí prahy intervalu. Prahý lze určit buď manuálně, nebo vypočítat z histogramu, například Otsuovou metodou nebo adaptivním prahováním [24, 25]. Hodnoty pixelů, které do tohoto intervalu patří, se považují za požadovaný objekt. Ostatní pixely jsou analogicky brány jako pozadí, tedy pixely s hodnotou 0. Prahování je možné aplikovat jak na šedotónový, tak barevný obrázek.



(a) Orig. obrázek.

(b) Maska.

Obrázek 3.4: Segmentace zelené barvy prahováním.

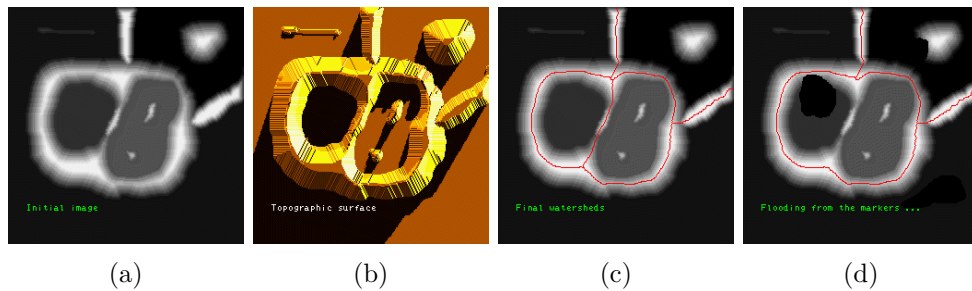
### 3.2.2.2 Marker-controlled watershed segmentace

Marker-controlled watershed segmentace je založena na watershed transformaci. Transformace nevnímá šedotónový obrázek jako matici, ale jako topografický povrch, kde výška bodu je určena jasně.

Bud'  $\Omega \subset \mathbb{Z}^2$  a funkce  $f : \Omega \rightarrow \mathbb{Z}$ , která každému pixelu  $p \in \Omega$  přiřadí jeho výšku. Tedy pixel  $p \in \Omega$  má výšku  $f(p)$  a množinu sousedů  $N(p)$ . Dále nechť  $\exists p$ , pro které platí  $\forall p' \in N(p), f(p') > f(p)$ . Takovému pixelu se říká singulární minimum. Plocha  $P$  je množina sousedních pixelů o stejné výšce. Plocha je minimální, pokud z ní nelze klesnout dolů. Konečně pixel, který je singulární minimum a člen minimální plochy, se nazývá lokální minimum. Tato lokální minima tvoří jednotlivá údolí [26].

Nyní se začne plnit krajina vodou tak, že hladina ve všech údolích bude stejná a bude rovnoměrně stoupat. Postupným zaplavováním dojde ke spojování jednotlivých údolí do větších. Jakmile se spojí dvě údolí, označí se vrchol, který tyto údolí rozděloval jako předěl. Postupně se zaplaví celá krajina. Vzniklé předěly reprezentují hranice jednotlivých údolí (segmentů) [26].

U watershed transformace dochází často k přesegmentování například kvůli šumu a nekonzistencím obrázku. Tento problém řeší umístění značek (marker), které určí, z kterých bodů má zaplavování začít. Markery lze určit, například další formou segmentace, vzdálenostní transformací nebo gradientem [9].



Obrázek 3.5: Ukázka MCW segmentace [9]. (a) Původní šedotónový obrázek. (b) Topografická reprezentace. (c) Watershed bez značek. (d) Watershed se značkami.

### 3.2.2.3 Segmentace pomocí Houghovy liniové transformace

Metoda spočívá v nalezení přímek, které protínají co nejvíce bodů v binární masce, získané nějakou hranovou detekcí, například Cannyho detektorem [27]. Tyto přímky rozdělují snímek na jednotlivé segmenty.

Přímku v kartézské souřadnicové soustavě (KSS) lze vyjádřit rovnicí:

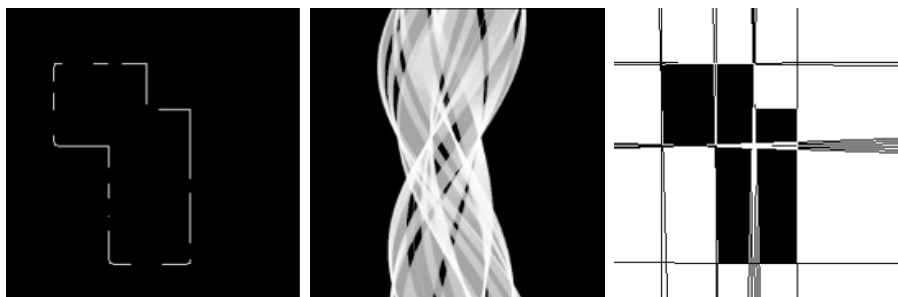
$$y = a \cdot x + b, \quad (3.2)$$

kde pár  $(x, y)$  je souřadnice bodu v obrázku. Pomocí těchto hodnot lze bod popsat vztahem:

$$x \cdot \cos(\theta) + y \cdot \sin(\theta) = p, \quad (3.3)$$

který reprezentuje křivku v polární souřadnicové soustavě (PSS).

Tímto způsobem se popíší všechny body z binární masky a zakreslí se do obrázku. Zakreslené křivky se různě protínají. Místa, kde se křivky protínají nejvíce, značí nalezené přímky, resp. parametry rovnice přímky v KSS [28, 29].



(a) Hranová detekce objektu v KSS. (b) Body hran, převedené do PSS. (c) Detekované hrany algoritmem.

Obrázek 3.6: Ukázka segmentace pomocí houghova algoritmu [8].

### 3.2.3 Prostorové a centrální momenty

Momenty jsou fyzikální pojem, kterým se dá matematicky vyjádřit charakteristika objektu [30]. V OpenCV<sup>1</sup> se rozlišují momenty normalizované centrální, prostorové a centrální [32]. Obecný zápis výpočtu prostorových momentů je:

$$m_{j,i} = \sum_x \sum_y x^j y^i I(x, y), \quad (3.4)$$

kde součet  $j$  a  $i$  udává řád momentu. Proměnné  $x$  a  $y$  jsou v pořadí indexy řádků a sloupců. Funkce  $I(x, y)$  značí hodnotu jasu pixelu. Jedná se o součet hodnot jasů pixelů v obrázku vážených indexy řádků.

Z prostorových momentů lze zjistit charakteristiky obsahu obrázku s ohledem na jeho pozici. Pro charakteristiku objektu samotného, nezávisle na pozici, se využijí centrální momenty:

$$\mu_{p,q} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y), \quad (3.5)$$

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}}, \quad (3.6)$$

kde  $\bar{x}$  a  $\bar{y}$  udávají vážený střed obrázku.

Výpočet momentů dává většinou smysl použít až na binární masku, jelikož intenzity jasu jsou buď 0 nebo 255. Momenty se díky tomu určují na základě plochy vysegmentovaného objektu a ne různých jasových intenzit.

### 3.2.4 Kontury

Pro zjištění kontur se nejčastěji využívá binární maska, vzniklá po segmentaci. Kontury jsou v openCV reprezentovány jako pole bodů jdoucích za sebou [33]. Tyto body označují sledovaný objekt a umožňují podrobnější zpracování. Některé možnosti zpracování jsou popsány níže.

#### 3.2.4.1 Aproximace pomocí Ramer–Douglas–Peucker algoritmu

Myšlenka algoritmu je zjednodušit konturu tak, aby byl stejný objekt reprezentován méně body na úkor přesnosti. Tu si určí uživatel a je vyjádřena vzdáleností  $\varepsilon > 0$ . Jelikož se kontura skládá z konečného množství bodů, seřazených podle toho, jak jdou za sebou, lze ji vnímat jako graf. Vrcholy tohoto grafu tvoří jednotlivé body kontury a hrany jsou vždy mezi sousedními body.

Nechť je graf  $G$ , o délce  $m \geq 2$ , kde  $G$  je z definice cesta.

$$G = (\{0, \dots, m\}, \{\{i, i + 1\} | i \in \{0, \dots, m - 1\}\}). \quad (3.7)$$

<sup>1</sup>OpenCV je open source knihovna, která obsahuje spoustu funkcí užitečných pro zpracování obrazu a strojové vidění [31].

Algoritmus nejdříve pomyslně spojí první a poslední vrchol  $v_z$  a  $v_k$  a označí je za potřebné. Vybere se bod  $v_d$ , který je od pomyslné hrany nejvíce daleko. Pokud je jeho vzdálenost od pomyslné hrany větší než  $\varepsilon$ , algoritmus se rekurzivně zavolá s body:

1.  $v_z$  a  $v_k = v_d$
2.  $v_z = v_d$  a  $v_k$

Pokud podmínka se vzdáleností není splněna, body  $v_z$  a  $v_k$  se označí jako potřebné a algoritmus se vrátí. Po doběhnutí rekurze jsou označeny pouze potřebné body. Ty se spojí od začátečního až po konečný a vznikne aproximovaný graf [34, 35].

$$A = (V, \{\{i, i + 1\} | i \in \{0, \dots, |V| - 1\}\}), \quad (3.8)$$

$$V \subseteq \{0, \dots, m\}. \quad (3.9)$$

#### 3.2.4.2 Konvexní obálka

Konvexní obálka je kontura, která je zbavena vypouklých míst (konvexních defektů) [36]. Jedná se také o jistou míru aproximace.

Analogii lze najít v provázku, který je omotaný kolem pěticípé hvězdy tak, že je co nejvíce napnutý, aby se využilo co nejméně materiálu. Objekt popsaný nataženým provázkem již nezobrazuje pěticípou hvězdu, ale pětiúhelník. Provázek je ekvivalent konvexní obálky.

#### 3.2.4.3 Tvarové charakteristiky

Každá kontura se dá popsat podobností k nějakému tvaru nebo svými statistickými vlastnostmi [32]. Například momenty jsou jednou z metod charakteristiky objektu. Díky tvarovým charakteristikám lze objekty popsat, rozlišovat nebo klasifikovat na základě kontur [23]. Mezi nejzákladnější vlastnosti kontur patří v pořadí kulatost, poměr stran, celistvost a vypouklost:

$$roundness = \frac{4 \cdot area}{\pi \cdot max_d^2}, \quad (3.10)$$

$$aspectRatio = \frac{max_d}{min_d}, \quad (3.11)$$

$$solidity = \frac{area}{convexhull}, \quad (3.12)$$

$$convexity = \frac{perimeter}{convexhull}. \quad (3.13)$$

$Min_d$  a  $max_d$  reprezentují minimální a maximální průměr objektu. Reprezentují také strany nejmenšího obepínajícího obdélníku objektu.  $Min_d$  je kratší strana a  $max_d$  delší.  $Area$  zastupuje obsah kontury, která značí oblast zájmu. A nakonec  $convexhull$ , která vyjadřuje konvexní obálku.

### 3.3 Perspektiva

Fotografie je dvoudimenzionální projekce reálného světa. Perspektiva vytváří iluzi vzdáleností objektů od kamery a „nahrazuje“ hloubkovou informaci. Objekty, které jsou od kamery dál, se na obrázku jeví menší. Ty, co jsou blíž, naopak větší [37].

#### 3.3.1 Homogenní souřadnicová soustava

Homogenní souřadnicové soustava se používá v projektivní geometrii. Velké využití najde při transformacích, protože operace, jako například rotace nebo posunutí, lze vyjádřit maticovým násobením. Další výhodou je možnost reprezentovat body v nekonečnu pomocí konečných souřadnic [38, 39].

Bod  $x$  je homogenní, pokud  $x = \lambda x$  a  $\lambda \neq 0$ .

$$x = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \omega x \\ \omega y \\ \omega \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ \omega \end{bmatrix}. \quad (3.14)$$

$$x' = \omega x, \quad y' = \omega y. \quad (3.15)$$

Další výhodou homogenních souřadnic je reprezentace přímek a vztah přímky s bodem. Přímku lze zapsat jako vektor koeficientů obecné rovnice.

$$ax + by + c = 0 \quad \rightarrow \quad \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad (3.16)$$

Pro zjištění, zdali nějaký bod  $x$  leží na přímce  $l$  stačí vypočítat jejich skalární součin.

$$x \cdot l = 0. \quad (3.17)$$

Pokud je výsledek roven 0, bod leží na přímce, jinak neleží.

Pomocí vektorového součinu lze z dvou bodů  $x$  a  $y$  zjistit přímku  $l$ , která jimi prochází:

$$x \times y = l, \quad (3.18)$$

nebo průsečík  $x$  dvou přímek  $l$  a  $p$ :

$$l \times p = x. \quad (3.19)$$



### 3.3.2 Transformace

Díky transformacím lze obrázek různě posouvat, rotovat nebo i měnit úhel pohledu. Většinou se počítají v homogenní soustavě, především díky možnosti jakoukoliv transformaci reprezentovat pomocí matice a maticového násobení. Transformace se dá rozdělit do skupin podle míry úprav

1. euklidovské – posunutí, rotace
2. podobnostní – škálování
3. afinní – ničí původní tvary, zachovává poměry vzdáleností a ploch, rovnoběžnost a kolinearitu
4. projektivní – libovolná transformace, zachovává pouze dvojpoměr a kolinearitu

Jak ze seznamu vyplývá, mezi transformacemi funguje jistá hierarchie. Platí, že operace, které lze provést transformací dané skupiny, lze provést i transformací ze skupiny vyšší (větší číslo znamená vyšší skupina) [39].

Transformaci v dvoudimenzionálním prostoru lze zapsat takto:

$$x' = Hx. \quad (3.20)$$

Body  $x'$  a  $x$  jsou v homogenní soustavě a  $H$  matice transformace. Operace posunutí se zapíše jako:

$$H = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} I & t \\ 0^T & 1 \end{bmatrix}. \quad (3.21)$$

Obdobně lze zapsat matici rotace.

$$H = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & 0 \\ 0^T & 1 \end{bmatrix}. \quad (3.22)$$

Každá operace je ovlivněna jinými parametry. Například  $t_x$  a  $t_y$  reprezentují posunutí a  $\varphi$  rotaci. Kombinací těchto parametrů lze vytvořit operaci posunutí a rotace současně:

$$H = \begin{bmatrix} \cos\varphi & -\sin\varphi & t_x \\ \sin\varphi & \cos\varphi & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}. \quad (3.23)$$

Přidáváním dalších parametrů lze docílit složitějších operací. Afinní operace vyžaduje parametrů 6, projektivní až 8 [38].

#### 3.3.3 Úběžnice a úběžník

Jelikož jsou fotografie reprezentovány pouze dvěma dimenzemi, ztrácí se oproti reálnému světu informace o hloubce. Hloubka je v obraze reprezentována například zmenšením objektů vůči ostatním. To lze pozorovat například na fotografii cesty, která mizí v dáli. I když jsou okraje cesty rovnoběžné, na obrázku se sbíhají do jednoho bodu. Tomuto bodu se říká úběžník. Úběžník je tedy bod, ke kterému konvergují čáry, které jsou v reálném světě rovnoběžné [37].

Přímka protínající dva úběžníky se nazývá úběžnice. Pomocí úběžnice lze zjistit pozici kamery vůči fotografii. Pokud je k dispozici i znalost nějakého rozměru v obraze, lze zjistit výškové rozměry i bez hloubkové informace [40].

#### 3.4 K-means

K-means je příklad algoritmu nesupervizovaného učení. Jeho účel je rozdělit data do  $n$  skupin. Rozdělení probíhá na základě účelové funkce, která ohodnocuje rozklad množiny dat na podmnožiny (čím menší hodnotu má, tím lepší je rozdělení).

Mějme množinu  $M$ , která popisuje data. Nejdříve se náhodně zvolí  $n$  středových bodů. Shluky se vytvoří tak, že každý bod  $m \in M$  se zařadí do skupiny reprezentované nejbližším středovým bodem. Z těchto shluků se spočítají geometrické středy, které reprezentují nové středové body. Algoritmus se opakuje do té doby, než se účelová funkce dostane na určitou hodnotu nebo přestane klesat [41].

#### 3.5 Tesseract OCR

Tesseract OCR je open source nástroj, který je schopný, pomocí neuronových sítí, rozeznat znaky. OCR je poměrně dost konfigurovatelné a obsahuje spoustu možností pro dosažení nejlepšího výsledku. Jedna z užitečných možností je trénování enginu na vlastním fontu [42].

#### 3.6 Morfologické operace

Morfologické operace slouží k úpravě binární masky. Různým zmenšováním nebo zvětšováním oblasti zájmu pomocí strukturního elementu lze masku různě upravovat nebo ji zbavit nežádoucího šumu. Přes masku se postupně prochází strukturním elementem a podle jeho velikosti a tvaru se vysegmentovaný objekt mění. Mezi základní operace patří eroze a dilatace.

Eroze je zjednodušeně zmenšení oblasti zájmu. Pokud se všechny pixely daného segmentu vejdu do strukturního elementu, jsou v obraze ponechány, jinak jsou odstraněny. Opačná operace je dilatace, která naopak oblast zájmu

rozšíří, pokud je ve strukturním elementu alespoň jeden pixel, který do oblasti patří.

Složením eroze a dilatace vznikají další operace jako je otevření a zavření. Otevření provede nejdříve erozi a poté dilataci. Tím se lze zbavit vypouklých částí masky nebo šumu bez změny velikosti masky. Zavření je naopak nejdříve dilatace a poté eroze, což má za následek například zaplnění „otvorů“ ve vysegmentovaném objektu [43, 10].



(a) Orig. maska.      (b) Eroze.      (c) Dilatace.

Obrázek 3.7: Příklad eroze a dilatace na masce [10].



## Sestavení aparatury a metodika měření

Jedna ze stěžejních částí zpracování obrazu je kvalita pořízených fotografií. Aby algoritmus poskytoval co nejlepší výsledky, je potřeba správně vybrat měřící aparaturu. Zároveň je potřeba brát v potaz uživatele a jeho potřeby. Na aparaturu jsou kladeny následující požadavky:

- záběr celé pracovní plochy
- volný přístup uživatele ke stavebnici
- volný pohled uživatele na obrazovku tabletu
- snadné sestavení

### 4.1 Snímací aparatura

Cílem je vytvořit co možná nejvíce konzistentní prostředí, ve kterém se scéna bude snímat. To bohužel kvůli charakteristice problému nelze vždy zaručit, jelikož v každé místnosti je jiné osvětlení. Dodržení konzistence se lze alespoň přiblížit, a to výběrem správných parametrů scény.

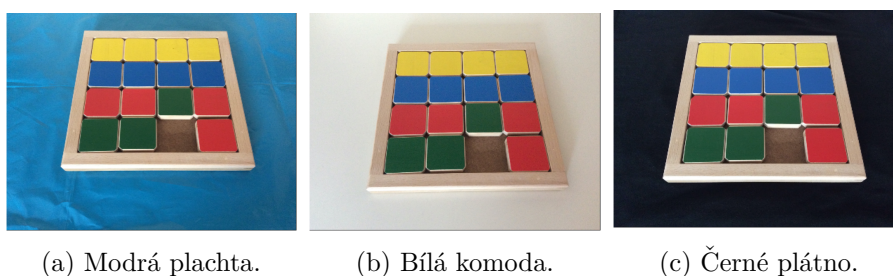
Zařízení použité pro snímání je tablet iPad mini 2. Parametry z oficiální specifikace jsou popsány v tabulce 4.1.

Tabulka 4.1: Specifikace zadní kamery tabletu - iPad mini 2 [44].

<i>Parametr</i>	<i>Hodnota</i>
Autofocus	Ano
Rozlišení	5 MP
Clona	$f/2,4$

### 4.1.1 Pozadí

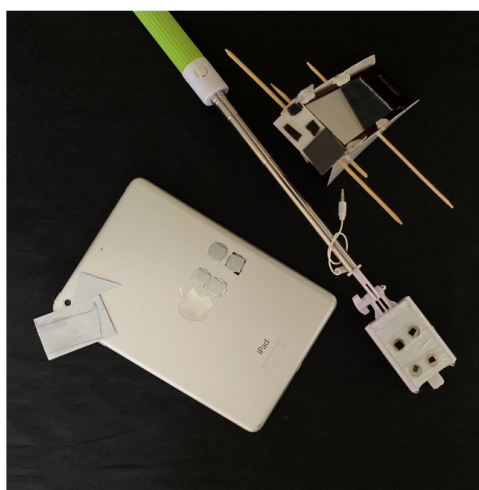
Výběr vhodného typu pozadí je klíčové pro zjednodušení segmentace scény. Kvůli nekozistenci světelných podmínek je nutný výběr matného pozadí. Jelikož scéna bude segmentována podle barevných charakteristik, pozadí musí mít barvu, která se ve stavebnici neobjevuje. Z modré plastové plachty, bílé plochy komody a bavlněného černého plátna je nejlepší možností černé plátno. Jak je vidět na obrázku 4.1c, vybraná možnost splňuje jak matnost, tak dostatečnou disjunktivitu od barev použitých ve stavebnici.



Obrázek 4.1: Porovnání různých pozadí scény.

### 4.1.2 Upevnění snímacího zařízení a stojan

Pro experimentální účely by měl být stojan flexibilní, tedy měl by mít upravitelnou výšku a možnost rotace držáku snímacího zařízení. V tomto případě se jedná o „selfie“ tyč, která všechny podmínky splňuje. Tablet je k tyči připevněn magnety, což je výhoda při opětovném sestavování aparatury, viz obrázek 4.2.



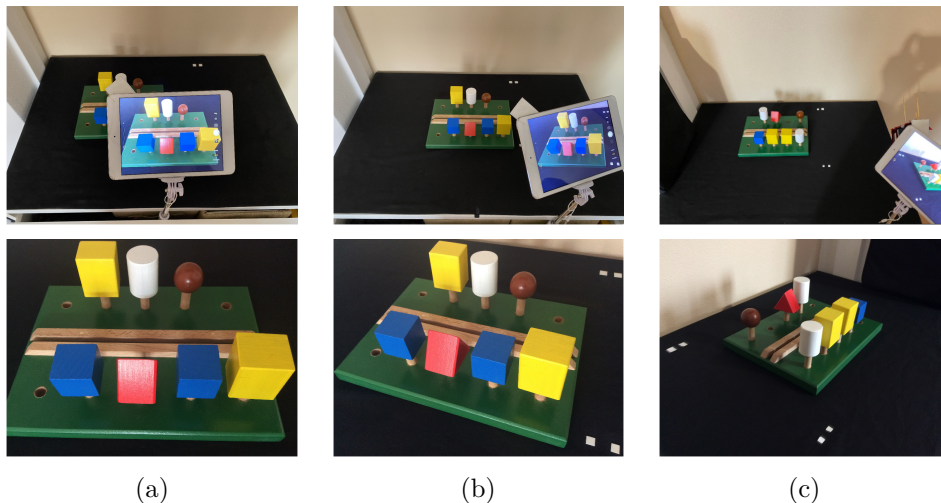
Obrázek 4.2: Stojan, snímací zařízení a zrcátko pro snímání.

### 4.1.3 Umístění zařízení

Po výběru pozadí a stojanu je potřeba vybrat vhodné umístění snímacího zařízení tak, aby snímač viděl na scénu a zároveň nepřekážel uživateli v práci se stavebnicí. Zároveň je důležité, aby uživatel viděl na displej tabletu, jelikož se zde plánuje promítat zadání úlohy. Otestováno bylo několik možností umístění tabletu, viz obrázek 4.3:

1. snímání z pohledu uživatele
2. snímání ze strany bez asistence zrcátka
3. snímání ze strany s asistencí zrcátka

Všechny tyto metody byly otestovány s reálným uživatelem věkové kategorie pod 8 let. Poznatky tedy zahrnují i zpětnou vazbu.



Obrázek 4.3: Porovnání umístění a pohledů tabletu. Každý sloupec obsahuje sestavenou aparaturu a pohled tabletu. **(a)** Z pohledu uživatele. **(b)** Ze strany bez zrcátka. **(c)** Ze strany se zrcátkem.

První možnost přináší velmi hezké výsledky nasnímané scény, bohužel je nepraktická pro uživatele. Tablet je poměrně velký, a i když dítě vidí na displej hezky, manipulace se stavebnicí je nešikovná.

Druhá možnost dává uspokojivé výsledky jak z pohledu tabletu, tak z pohledu manipulace se stavebnicí. Tablet je sice lehce natočený od uživatele, ale úhel není tak velký, aby se musel nějak zásadně naklánět. Nejedná se o ideální řešení z hlediska použitelnosti, ale o přijatelný kompromis obou požadavků.

Poslední možnost je z uživatelského hlediska nejvhodnější. Obrazovka je natočená směrem k dítěti a zároveň mu nebrání v manipulaci se stavebnicí. Nevýhodou je pohled zařízení. Scéna je sice nasnímana celá, ale ze strany.

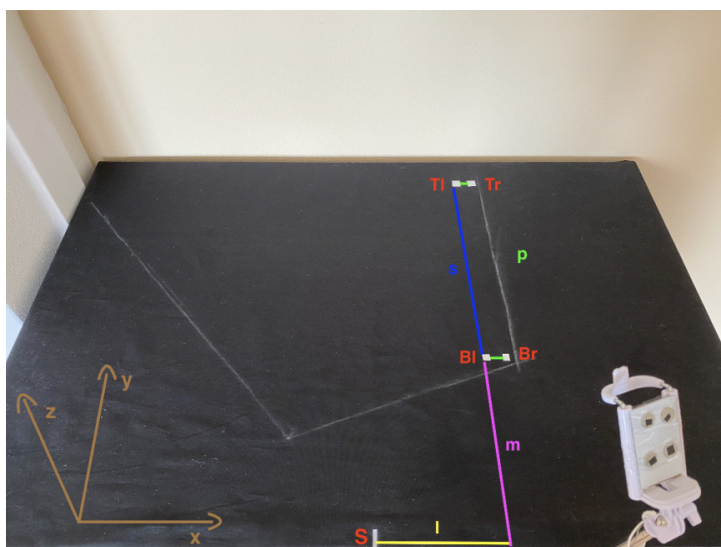
To je nežádoucí u prostorových součástek, které se často překrývají. Dalším problémem je rozostření nasnímaného obrázku kvůli typu zrcátka. Většina zrcátek má vrstvu odrážející světlo až za ochranným sklem. To ale způsobí nejdříve lom světla přes první vrstvu skla a poté teprve odraz světla. Řešením je použití „first surface“ zrcadla, které má odrazovou vrstvu jako první.

Z výše popsaných metod nejlépe vyšel druhý způsob snímání. Byl tedy zvolen pro implementaci a sestavení aparatury.

### 4.2 Sestavení aparatury

Sestavení aparatury by mělo být přímočaré a jednoduché na replikaci. Proto jsou na plátně umístěny značky, označeny na obrázku 4.4, sloužící pro kalibraci scény. Bílá čárka na kraji značí střed plátna  $S$ . Čtyři body  $T_l, T_r, B_l, B_r$  (od levého horního po dolní pravý) po pravé straně slouží ke správnému natočení tabletu.

Uživatel umístí stojan podle parametrů v tabulce 4.2. V tabulce jsou popsány i vzdálenosti jednotlivých bodů již umístěných na plátně<sup>2</sup>. Poté si otevře fotoaparát a natočí stojan s tabletem tak, že těsně u pravého kraje snímku budou zmíněné bílé čtverečky, viz obrázek 4.3b.



Obrázek 4.4: Plátno s orientačními body a vyznačenými vzdálenostmi.

---

<sup>2</sup>Bod  $S$  nemusí být nutně uprostřed stolu. Důležité je, aby se zachovaly vzdálenosti mezi jednotlivými body.



Tabulka 4.2: Parametry umístění tabletu a bodů na plátně, viz obrázek 4.4.

<i>Přímka</i>	<i>Vzdálenost</i>
Vzd. stojanu od S, doprava (osa x)	17 cm
Spodní okraj tabletu od kraje stolu (osa z)	15,3 cm
Vzd. $S$ od $B_l - l$ (osa x)	12,3 cm
Vzd. $S$ od $B_l - m$ (osa y)	18,3 cm
Vzd. $B_l$ od $T_l - s$ (osa y)	26,2 cm
Vzd. $B_l$ od $B_r - p$ (osa x)	2 cm



## Návrh řešení algoritmů

Cílem práce je vytvoření vyhodnocovacího systému, který na vstupu přijme obrázek vyfocený mobilním zařízením a obsahující stavebnici a na výstupu vyhodnotí, zda je řešení na obrázku validní podle zadání. Problém zpracování snímku stavebnice lze rozdělit do několika podproblémů, viz kapitola 2. Lze tedy využít získaných znalostí a z popsanych algoritmů poskládat algoritmus pro získání informací.

Zpracování probíhá v jednotlivých krocích, které jsou za sebou postupně popsány níže. Některé části jsou společné, jako úprava perspektivy obrazu 5.1 nebo segmentace objektů 5.3. Následná práce s vysegmentovanými objekty je specifická pro každou skupinu součástí. Na rozdělení do skupin a popis algoritmů se zaměřuje sekce 5.4.

### 5.1 Úprava perspektivy

Díky vybrané metodě měření, z kapitoly 4, je scéna vyfocena z boku. Avšak referenční řešení i zadání prezentují stavebnici shora<sup>3</sup>. Aby výsledky validace korespondovaly s řešením, je potřeba obrázek perspektivně transformovat.

Čtverečky na plátně, viz sekce 4.2, které slouží pro kalibraci aparatury, mají ještě jedno využití. Body mají mezi sebou přesně definovanou vzdálenost. Zároveň z pohledu uživatele tvoří obdélník. Relaci mezi reprezentací čtyř bodů z pohledu tabletu a z pohledu uživatele lze využít pro namapování bodů mezi jednotlivými pohledy. Pokud je známa relace mezi jednotlivými perspektivami, lze zjistit i projekční matici, která slouží k transformaci mezi perspektivami. Za předpokladu, že se mapuje z pohledu tabletu do pohledu shora, tak lze projekční maticí získat obrázek „narovnaný“. Inverzní maticí lze snímek transformovat zpět do původního stavu.

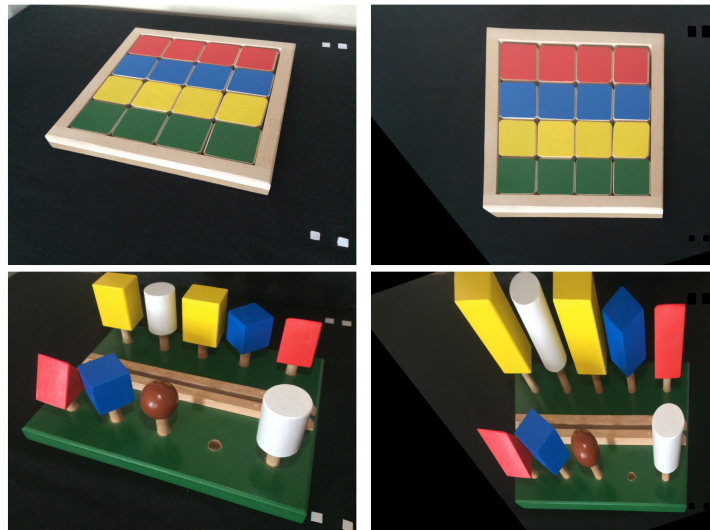
---

<sup>3</sup>Výjimka je komponenta „zrakové vnímání“, kde je zadání někdy vyfoceno z pohledu uživatele

## 5. NÁVRH ŘEŠENÍ ALGORITMŮ

---

Pokud se transformace aplikuje na scénu, která obsahuje pouze součástky ploché, výsledný obrázek je upraven velmi hezky. Problém nastává při úpravě scény obsahující tvary prostorové. Ty jsou často zdeformované a tedy většinou nemá smysl na nich provádět žádné tvarové detekce.



(a) Pohled tabletu.

(b) Pohled shora.

Obrázek 5.1: Ukázka perspektivní transformace.

### 5.2 Sjednocení barev

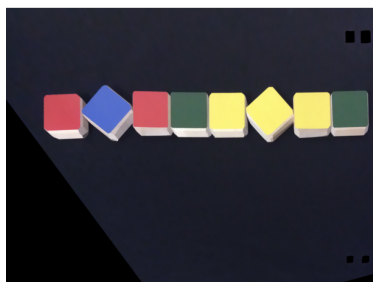
Většina informací o jednotlivých objektech se opírá o barevné charakteristiky. Ať už jde o pozici, informaci o velikosti nebo typu součástky, je velmi důležité, aby se objekty skládaly převážně z jedné stejné nebo z více velmi blízkých barevných hodnot. Pokud by tomu tak nebylo, tak by segmentace nepodávala stabilní výsledky. Sjednocení barev zajistí mean-shift filtrace.

### 5.3 Segmentace objektů

Součástky jsou segmentovány v závislosti na své barvě. Na základě algoritmu se segmentuje podle různých barev. Například pro logické řady a barevné matice se uvažují pouze součástky, které jsou modré, červené, žluté nebo zelené.

Segmentace se provádí manuálním prahováním v barevném prostoru HSV. Barevné prahy jsou určeny experimentálně. Metoda je vhodná, jelikož je rychlá a součástky mají od sebe dostatečně odlišnou barvu.

Jelikož se ale jedná o prahování manuální, občas se kvůli nekonzistenci osvětlení dostane do výsledné masky šum. Menší šum, který se vyskytuje většinou kvůli nestabilním hodnotám HSV, se odstraní pomocí morfologických

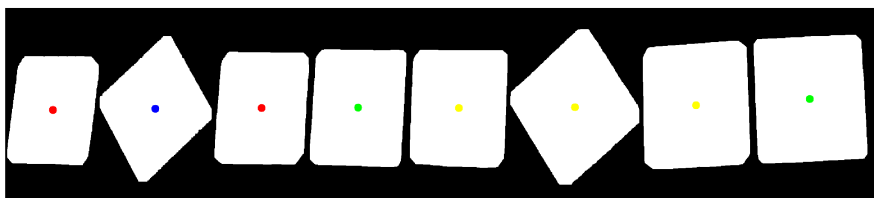


Obrázek 5.2: Originální obrázek k segmentaci viz 5.3.



Obrázek 5.3: Masky vzniklé po barevné segmentaci. V pořadí červená, modrá, žlutá a zelená.

úprav. Pokud je šum větší nebo se po segmentaci v masce nachází nežádoucí objekty, lze je vyfiltrovat podle tvarových charakteristik nebo pozice kontury nechtěného objektu. Například, jak je vidět na obrázku 5.1, už při transformaci se odstraní okrajová část, která s jistotou tvoří pozadí. Odstranění okrajových částí je založeno právě na pozici kontur.



Obrázek 5.4: Maska vytvořená spojením masek z obrázku 5.3.

Z obrázku 5.4 lze vypořozovat, že jedna z výhod tohoto stylu segmentace je přímý zisk informace o barvě objektů.

## 5.4 Algoritmy

Po získání kontur je potřeba algoritmus rozdělit na jednotlivé skupiny. Každá skupina obsahuje jiné podúkoly a součástky, které vyžadují jiné typy zpracování a validace. Zpracování kontur se tedy rozdělí do následujících skupin na základě typu úkolů, co dítě musí splnit, nebo typů součástek.

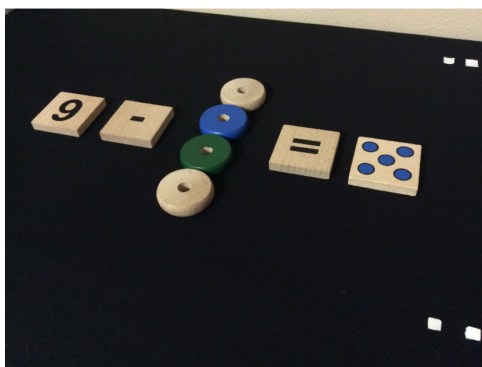
### 5.4.1 Logické řady a barevné matice

Skupina obsahuje pouze čtverečky červené, zelené, modré nebo žluté barvy. V aplikaci iSophi SMART spadají pod skupinu „algoritmické myšlení“. Z hlediska zpracování je potřeba pro logické řady určit pořadí barev a rotaci jednotlivých součástí, a pro barevné matice pořadí a popřípadě chybějící dílek.

Tady zpracování defakto končí segmentací. Po segmentaci je totiž známo dostatečné množství informací o součástkách, a to barevně rozlišené kontury. Z kontur lze poté snadno pomocí momentů dopočítat pozici v obraze, která je reprezentována jako střed dané kontury. Rotace se zjistí metodou `minBoundingRect` [36]. Tyto informace stačí pro validaci jednotlivých snímků. Vizualizace pořadí kontur lze vidět opět na obrázku 5.4.

### 5.4.2 Rovnice a počítání

V této skupině se vyskytují součástky jako kolečka a čtverečky, na kterých je namalované buď číslo, nebo puntíky, jejichž počet udává dané číslo. V aplikaci iSophi SMART je lze najít pod záložkou „matematické představy“. Úkolem dítěte je vyřešit rovnici, označit počet nebo seřadit množiny.



Obrázek 5.5: Ukázka součástek ze skupiny 5.4.2.

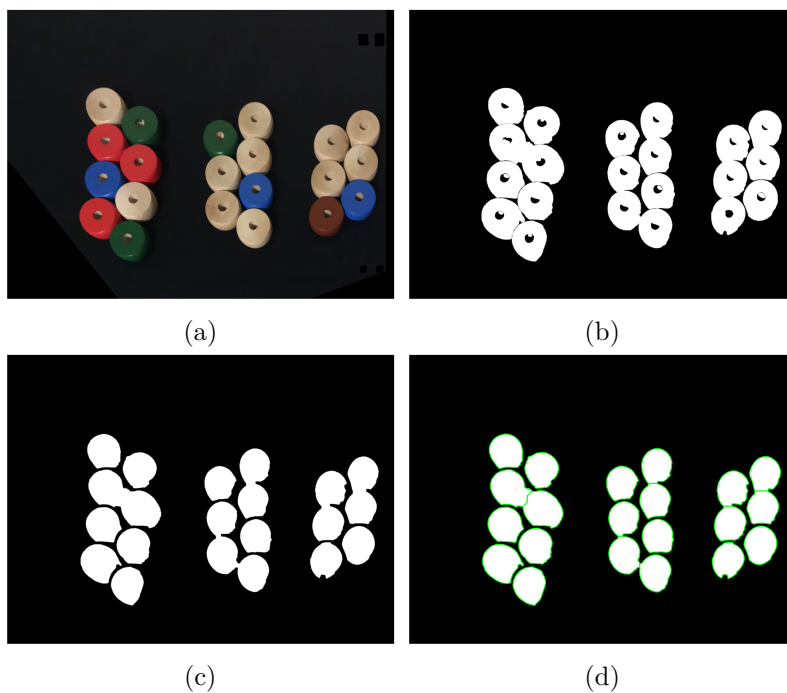
První úkol je počítání koleček. Barevná segmentace sama problém nevyřeší, jelikož součástky se často dotýkají, což způsobuje spojení kontur do jedné větší. To lze vyřešit použitím segmentace pomocí MCW.

Překážkou použití MCW segmentace je ale fakt, že kolečka mají uprotřed otvor. Segmentace pomocí barev zaručí, že v masce budou pouze části, které kolečko opravdu reprezentují. Kolečka vedle sebe se tedy budou dotýkat pouze velmi malou plochou. Plochu lze ještě zmenšit nebo rovnou odstranit erozí. Z masky se poté vezme jen externí kontura — otvory se uzavřou a zůstanou pouze kontury připravené pro MCW segmentaci.

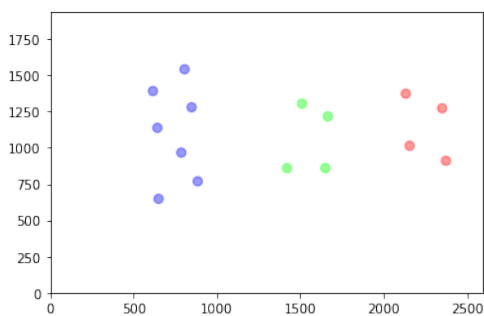
V této fázi je většina koleček, díky erozi, už oddělená. To sice pomůže při počítání, ale nelze z masky poznat, která kolečka patří k sobě, a která už ne. Skupiny lze rozeznat pomocí k-means algoritmu, který kontury rozdělí

do clusterů podle jejich středů. Se získanými informacemi lze aplikovat MCW segmentaci a spočítat kolečka v jednotlivých hromádkách.

Celý postup segmentace je možné vidět na obrázku 5.6 a rozdělení kontur do skupin na obrázku 5.7.



Obrázek 5.6: Průběh počítání koleček pomocí MCW. (a) Originální obrázek. (b) Maska po barevné segmentaci. (c) Maska po erozi a „zaplnění“ otvorů. (d) Kontury po MCW segmentaci.



Obrázek 5.7: Rozdělení koleček z obrázku 5.6 do clusterů pomocí k-means.

Druhý úkol je detekce a počítání koleček na čtverečkách. Součástky jsou ze dřeva a znaky, které jsou na nich natištěné, mají vždy jinou barvu. Segmentace čtverečků tedy probíhá šedotónovým prahováním a následnou filtrací kontur

podle tvaru. Získanou konturu lze využít pro oříznutí požadované části pro lepší zpracování. Z oříznutého obrázku je poté potřeba spočítat kolečka nebo rozeznat znak. Vysegmentovaná součástka s korespondující maskou je vidět na obrázku 5.8.

Kolečka se uvnitř součástky nedotýkají. Jejich počtu odpovídá počet kontur po šedotónové segmentaci.

Pro detekci čísel lze využít Tesseract OCR, který z dobře vysegmentovaného obrázku vrátí daný znak. Předzpracování zahrnuje snížení rozlišení obrázku a vyčištění případného šumu pomocí morfologických operací. Masku se na závěr invertuje, jelikož OCR přijímá binární obrázek, kde je oblast zájmu označena černou barvou a pozadí bílou. Mezi znaky pro rozeznání se zahrnují i matematické operace jako  $+$ ,  $-$  nebo  $=$ .

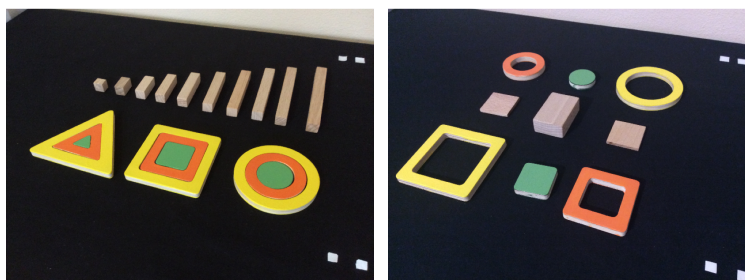


Obrázek 5.8: Oříznuté čtverečky a jejich masky.

Ze získaných informací z obrázku lze provést finální validaci pořadí kontur. K tomu opět poslouží jejich středy.

### 5.4.3 Tvary a velikosti

Skupina zahrnuje součástky zaměřené na procvičení rozeznávání tvarů a velikostí. V iSophi SMART je zahrnuta pod záložkami „matematické představy“ a „prostorová orientace“. Ukázku lze vidět na obrázku 5.9.

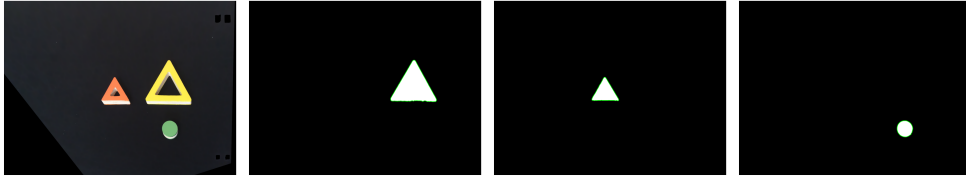


Obrázek 5.9: Ukázka součástek ze skupiny 5.4.3.

Úkolem dítěte je součástky správně poskládat, seřadit nebo označit tvar, který je něčím specifický. Skupina se ale neskládá pouze z barevných částí, je proto potřeba k úkolu přistupovat opět po částech.



První část sestává z rozpoznání tvarů a velikostí nabarvených kusů. Tvar lze zjistit pomocí tvarových charakteristik kontur. Informaci o velikosti lze získat přímo z barevné segmentace, protože jí reprezentuje barva součástky.



Obrázek 5.10: Barevné součástky a jejich masky.

Větší problém nastává při rozpoznávání součástek ze dřeva. Je to materiál, který má přirozeně spoustu nekonzistencí v zabarvení, jako jsou například letokruhy. Další komplikace způsobuje fakt, že například tyčky jsou dost vysoké. Po perspektivní úpravě snímku lze tedy očekávat, že maska těchto objektů bude zkreslená, viz obrázek 5.1. To je nežádoucí efekt při jejich určování. Pro správnou validaci je potřeba oba podproblémy vyřešit.

Jako první je segmentace, která je nekonzistencí materiálu postížena nejvíc. Pro získání kvalitní masky součástek je potřeba využít šedotónového prahování. Tím se v masce ale vyskytnou i barevné tvary, které jsou zde nežádoucí. Řešením je porovnání masek získaných z barevné a šedotónové segmentace. Objekty, které se opakují, jsou vyfiltrovány a v masce zbydou pouze dřevěné kusy stavebnice.

Druhý podproblém tvoří zkreslení součástek po transformaci snímku. Kvůli tomu nelze pro klasifikaci objektů použít tvarové charakteristiky kontur. Lze si ale povšimnout, že všechny dřevěné součástky jsou kvádry. Pro klasifikaci je tedy vhodné použít aproximaci kontur na kvádr. Z kontur lze poté určit typ součástky, například na základě poměru stran.

Aproximace na kvádr popsána v sekci 2.5 využívala kalibrace a pozice kamery vůči scéně. U mobilního zařízení je potřeba vnitřní parametry získat v době, kdy je fotografie pořízena, jelikož kamera využívá automatické zaostřování. Po snaze získat parametry pomocí frameworku [45] bylo zjištěno, že použité snímací zařízení nepodporuje funkci [46], která parametry vrací. Aproximaci na kvádr je tedy potřeba provést jiným způsobem. Zároveň jsou měřené rozměry v pixelech a zkreslené. To není ideální, ale zároveň ani fatální, jelikož není potřeba objekt přesně změřit, ale zařadit ho do nějaké kategorie. Navíc vzdálenosti jsou potřeba pouze pro součástky typu tyčka, zbytek se určí například z poměrů stran kvádrů (tyček je deset velikostí, zbytek součástek je tvarově unikátních).

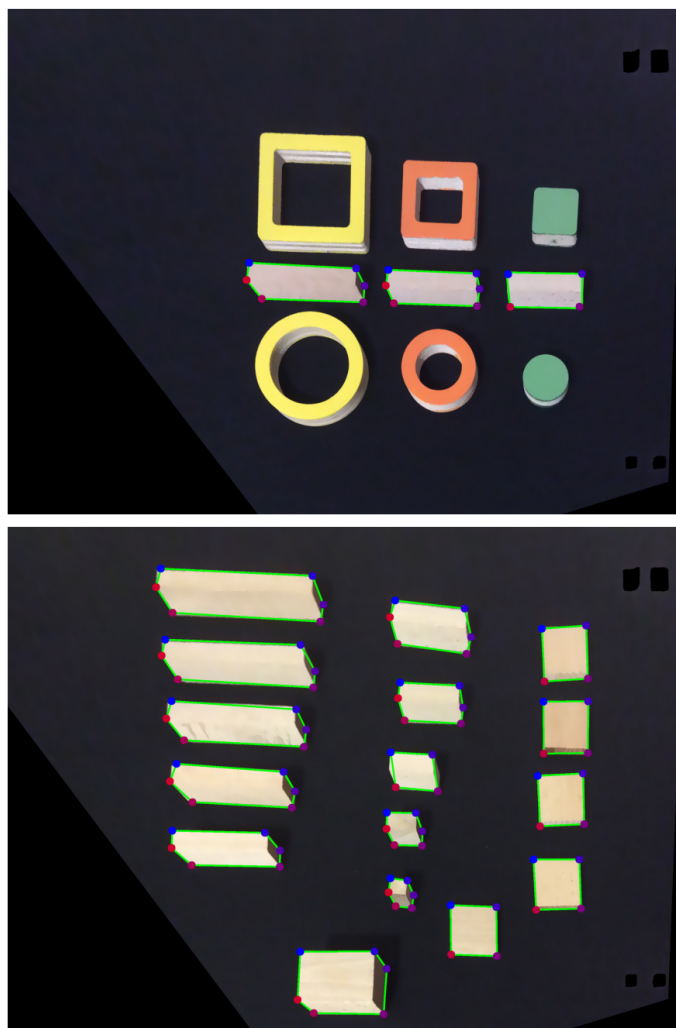
Zjednodušení kontur na kvádr se provede pomocí Ramer–Douglas–Peucke algoritmu. Vzniklá kontura nereprezentuje přímo kvádr, pro účely klasifikace objektů je to ale dostačující. Skládá se ze šesti nebo čtyř bodů v závislosti

## 5. NÁVRH ŘEŠENÍ ALGORITMŮ

---

na aproximované kontuře<sup>4</sup>, viz obrázek 5.11. Kvůli zkreslení je nutné měřit rozměry způsobem, který zkreslení alespoň minimalizuje. Po nalezení rohových bodů aproximované kontury se proto vyberou jen ty, které reprezentují délku objektu a jsou blíže stolu, jelikož takové body nepodléhají většímu zkreslení. Ty se transformují do pohledu shora a vypočítá se délka v pixelech. Podle délky lze určit velikost tyčky.

Validace pozice objektů je vyřešena stejně jako v předchozích případech.



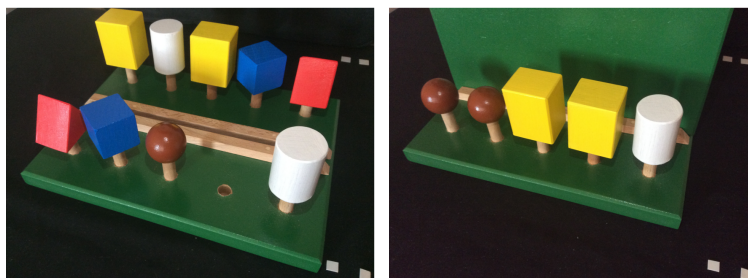
Obrázek 5.11: Aproximace kontur dřevěných součástek.

---

<sup>4</sup>Pokud jsou dva body příliš blízko u sebe, jsou nahrazeny bodem jedním. To se děje například u součástek, které nejsou tak vysoké nebo jsou natočené směrem na kameru tak, že nejsou vidět zbylé hrany.

#### 5.4.4 Zrakové vnímání

Skupina zahrnuje veškeré prostorové součástky, v iSophi webové aplikaci ji lze najít pod stejně pojmenovanou záložkou. Sada sestává ze zelené destičky, kam se součástky upevňují, viz obrázek 5.12.



Obrázek 5.12: Ukázka součástek ze skupiny 5.4.4.

Barevná segmentace vyřeší většinu práce se získáváním informací, jelikož každá součástka ze sady má jednolitou barvu, která zároveň identifikuje její typ. Komplikací je opět deformace obrazu po transformaci, jak je vidět na obrázku 5.1. Deformace je ale přítěž pouze v případě, kdy potřebujeme zjistit rozměry nebo rotaci součástky. To potřebujeme v případě modré krychle, červeného trojúhelníkového hranolu a žlutého kvádru. U zbytku součástek stačí zjistit jejich pozici vůči ostatním součástkám, což lze opět pomocí středů kontur.

Modrou krychli a žlutý kvádr lze aproximovat a změřit stejně jako dřevěné součástky v podsekcí 5.4.3. U krychle je potřeba zjistit, jestli je k uživateli zrotovaná. Z kontury se dá dopočítat čtvrtý bod spodní nebo vrchní strany kvádru, pomocí kterého lze zrekonstruovat čtverec. Rotace čtverce se zjistí totožně jako v podsekcí 5.4.1. Měřením kvádru je třeba zjistit, jestli je natočený delší nebo kratší stranou k uživateli. Postup je identický s měřením dřevěných tyček.

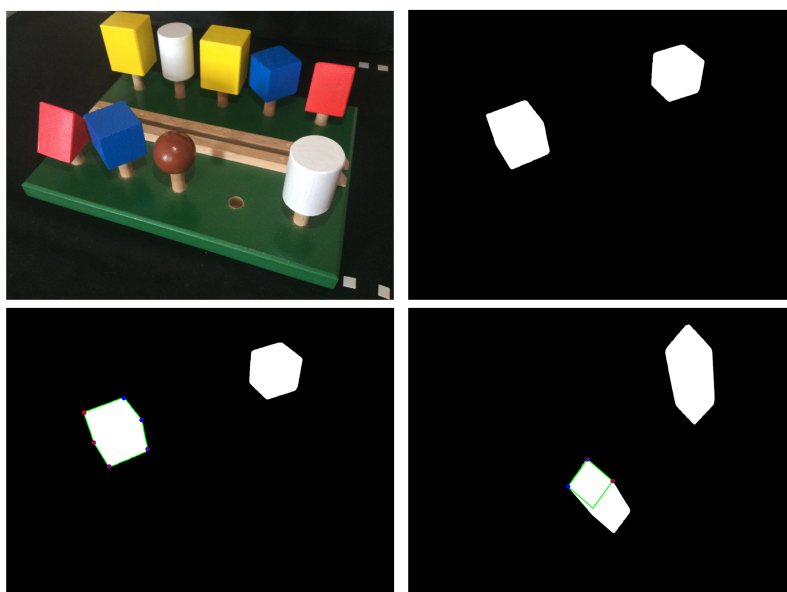
U červeného trojúhelníkového hranolu se aproximace kontury využije také, ale pouze na pět bodů. Je třeba zjistit, jakou stranou je součástka natočena k uživateli. Lze využít faktu, že tablet sleduje scénu z pravé strany. Z kontury se vyberou nejvíce rovnoběžné přímky<sup>5</sup>, z kterých se vytvoří obdélník. Ten indikuje zkosenou stranu. Pátý bod poté slouží k identifikaci natočení. Pokud je bod na pravé straně, skoro leží na levé straně obdélníku<sup>6</sup> nebo je kontura aproximovaná pouze na čtyři body (z důvodu spojení blízkých bodů nebo natočením na kameru), trojúhelníkový hranol je k uživateli postavený ze strany. V ostatních případech je postavený čelem.

<sup>5</sup>Nelze předpokládat, že přímky budou úplně rovnoběžné, jelikož se jedná o aproximovanou konturu.

<sup>6</sup>To je tehdy, když je trojúhelníkový hranol úplně na pravé straně podstavce, tedy nejbliž ke kameře.

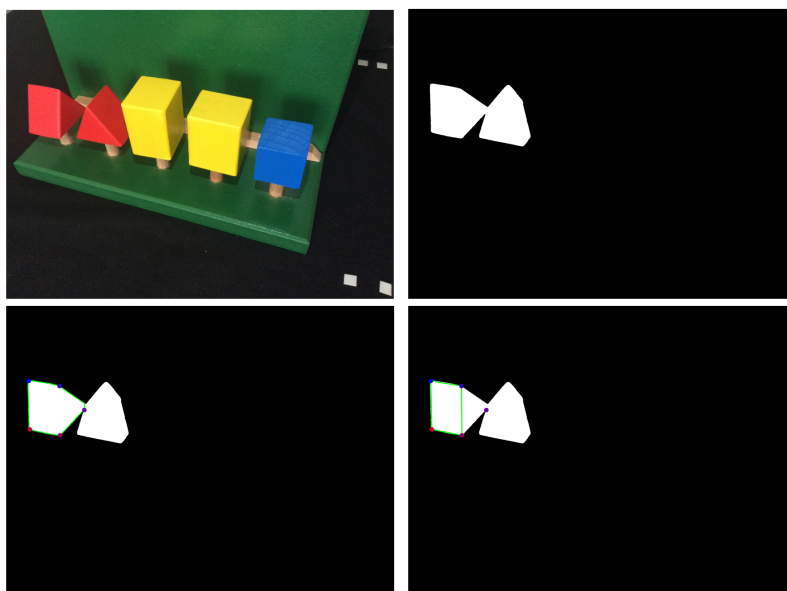
## 5. NÁVRH ŘEŠENÍ ALGORITMŮ

---



Obrázek 5.13: Ukázka zpracování modré krychle.

Jak je vidět na druhém snímku 5.14, objekty se někdy překrývají a maska je poté sjednocená. To lze vyřešit pomocí MCW segmentace.

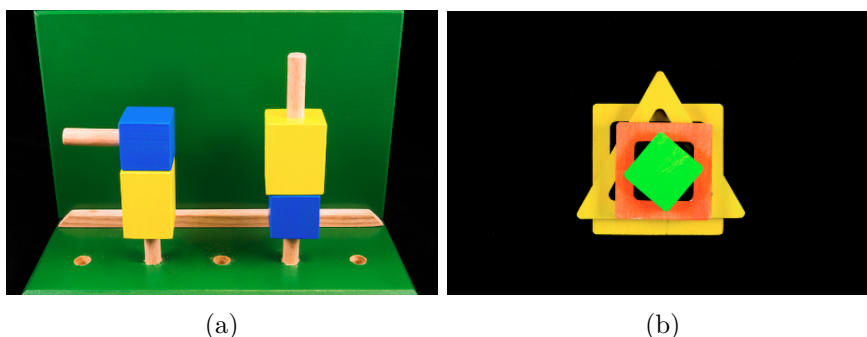


Obrázek 5.14: Ukázka zpracování trojúhelníkového hranolu.

## 5.5 Zbylé úkoly

Popsanými algoritmy v sekci 5.4 nelze vyřešit všechny problémy z jednotlivých skupin.

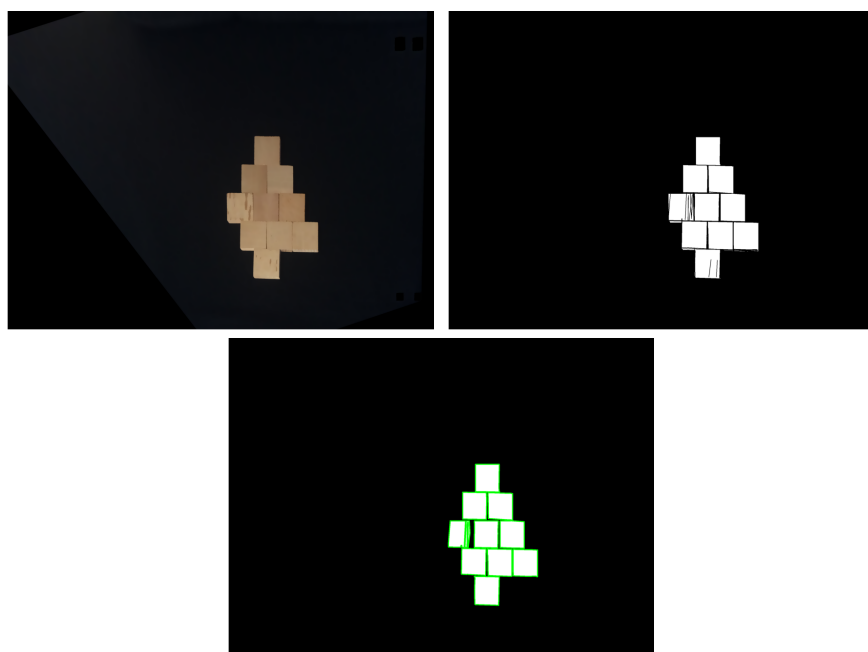
První typ úlohy, který nelze vyřešit se na webu vyskytuje v záložce „Prostorová orientace/Nápodoba sestavy“, viz obrázek 5.15b. Hlavním důvodem je fakt, že se součástky mohou překrývat. Pokud by se překrývaly dvě součástky stejné barvy, nelze je od sebe spolehlivě rozeznat segmentací ani barevnou ani za použití HT nebo MCW. Dalším důvodem je definice referenčního řešení. Způsob, kterým se řešení zaznamenává, neumožňuje nedeterministicky popsat scénu. Součástky musí být vždy seřazené zleva doprava po řádcích, což zde není možné, viz obrázek 5.15.



Obrázek 5.15: Ukázka neřešitelných typů úloh. **(a)** Znakové rozlišování podobností a směrů. **(b)** Nápodoba sestavy.

Podmnožinou tohoto typu úloh je nápodoba sestavy pomocí dřevěných čtverečků. V úloze se součástky nepřekrývají, ale jsou většinou přímo u sebe. To už nelze vždy vyřešit pomocí MCW, protože kontura často vypadá jako jeden celek. Jelikož se ale jedná o hranaté součástky, dává smysl využít segmentaci pomocí HT. Za použití masky z šedotónové segmentace se ořízne oblast zájmu. V té se poté naleznou hrany pomocí Cannyho detektoru, které se předají HT. Zrekonstruované hrany značí předěly jednotlivých čtverečků. Postup lze vidět na obrázku 5.16. I přes získání informace o součástkách, pořád nelze rozumně zadefinovat referenční řešení tohoto typu úlohy.

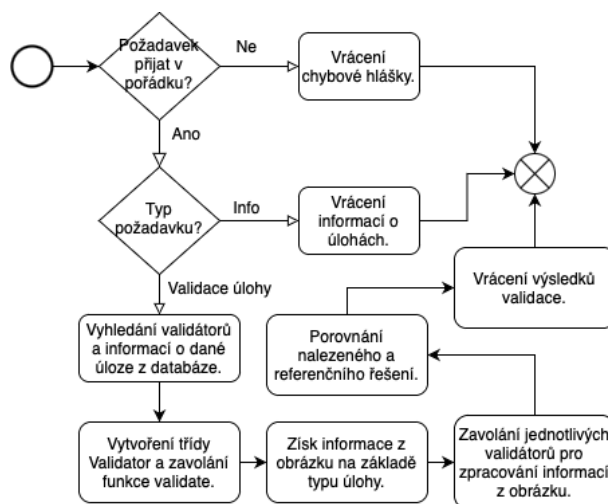
Druhý typ, který nelze spolehlivě vyřešit se vyskytuje pod záložkou „Znakové vnímání/Znakové rozlišování podobností a směrů“, viz obrázek 5.15a. Největším problémem je komplikace při segmentaci, způsobená dřevěnou částí součástky. Kontura je nekompletní a konvexní obálka není vždy spolehlivé řešení. Navíc není možné součástky měřit přesně, jelikož se některé vyskytují ve velké výšce — měření by udávalo velmi zkreslené výsledky.



Obrázek 5.16: Ukázka segmentace čtverečků pomocí HT.

# Implementace

Kapitola popisuje implementační stránku práce. Popsány jsou jak použité nástroje, tak průběh celého procesu vyhodnocení snímku, to znamená od přijetí fotografie po porovnání získaných výsledků s referenčními. Celý proces je vidět na obrázku 6.1 a blíže popsán v následujících sekcích.



Obrázek 6.1: Vývojový diagram serverové aplikace.

## 6.1 Použité nástroje

Pro vytvoření logiky, validace a rozhraní byl použit programovací jazyk Python [47] s příslušnými frameworky jako OpenCV [31] a Flask [48]. Algoritmy byly nejdříve zpracovány v prostředí Jupyter notebook [49] a poté přeneseny do serverové aplikace.

## 6.2 Rozhraní programu

Komunikace probíhá přes REST požadavky. Využity jsou pouze metody POST a GET.

Serverovou aplikaci lze požádat o typy a identifikátory všech úkolů v databázi (metoda GET) nebo validaci úlohy (metoda POST<sup>7</sup>). Pokud si uživatel na vstupu vyžádá validaci, musí dodat obrázek, typ a identifikátor úlohy. Podle typu a identifikátoru jsou z databáze získány referenční informace o úloze. Ty jsou předány společně s obrázkem k validaci, viz sekce 6.4. Obsah a formát informací je popsán v sekci 6.3.

Po zpracování požadavku nebo vyhodnocení snímku je odeslána odpověď. Pokud se stane chyba, například při zadání adresy neexistujícího úkolu nebo při zpracování, je vrácena chybová hláška. Při správném průběhu je vrácena odpověď ve formátu JSON, která popisuje správnost řešení. Příklady odpovědí jsou vidět na obrázcích 6.3 a 6.2. Jejich struktura je popsána v podsekci 6.4.3.

```
{
  "result": true,
  "message": "You got it right!",
  "detail": {
    "color_order": {
      "result": true,
      "detailed_result": [
        "t", "t", "t", "t", "t", "t", "t", "t"
      ],
      "correct_length": true
    },
    "flat_rotation": {
      "result": true,
      "detailed_result": [
        "t", "t", "t", "t", "t", "t", "t", "t"
      ],
      "correct_length": true
    }
  }
}
```

Obrázek 6.2: Formát odpovědi serveru pro správné řešení logické řady.

<sup>7</sup>POST metody znamenají zápis dat. Zde se reálně nic nezapíše, metoda je využita pouze kvůli tomu, že se posílá obrázek, který se do těla GET požadavku nevejde.



```

{
  "result": false,
  "message": "Better luck next time...",
  "detail": {
    "equation": {
      "result": false,
      "detailed_result": [
        "f","t","f","t","f"
      ],
      "correct_length": false
    }
  }
}

```

Obrázek 6.3: Formát odpovědi serveru pro špatné řešení rovnice.

### 6.3 Databáze referenčních řešení

Jedná se o souborovou databázi. Záznamy jsou uloženy ve složce v JSON objektu. Ten se skládá z dalších objektů, které obsahují informace o popisovaných typech úloh, viz obrázek 6.4. Každý objekt obsahuje atributy popsané v následujících podsekcích.

```

{
  "amounts_order": {
    "validators": [
      "amounts_order"
    ],
    "tasks": [
      {
        "id": 1,
        "num_of_groups": 0,
        "points_instead_of_number": false,
        "points_and_numbers": true,
        "expected_result": {
          "amounts_order": "355"
        }
      }
    ]
  }
}

```

Obrázek 6.4: Objekt reprezentující typ úkolu, zde řazení čísel.

### 6.3.1 Validators

Pole řetězců. Řetězec udává informaci, jaký typ validace bude potřeba provést. Každá skupina má vlastní druhy validátorů, ty jsou vidět v tabulce 6.1.

Například pokud bude pole rovno `["color_order", "flat_rotation"]` znamená to, že při vyhodnocování se bude validovat pořadí barev a rotace součástí ze skupiny logické řady a barevné matice 5.4.1.

Tabulka 6.1: Validátory jednotlivých skupin.

<i>Skupina součástí</i>	<i>Validátor</i>	<i>Popis</i>
Logické řady a barevné matice	<code>color_order</code>	Pořadí barev součástí zleva doprava.
	<code>matrix_layout</code>	Pozice součástí maticově.
	<code>flat_rotation</code>	Rotace součástí.
Rovnice a počítání	<code>equal_amounts</code>	Rovnost dvou čísel/hromádek nebo kombinace.
	<code>equation</code>	Rovnice s operacemi.
	<code>amounts_order</code>	Seřazení čísel zleva doprava.
	<code>mark_amount</code>	Označení daného čísla.
Tvary a velikosti	<code>shapes_order</code>	Pořadí tvarů zleva doprava.
	<code>mark_shape</code>	Označení tvaru.
	<code>shapes_matrix</code>	Pořadí tvarů maticově.
Zrakové vnímání	<code>spacious_shapes</code>	Pořadí a rotace součástí.

### 6.3.2 Tasks

Pole objektů. Každý značí konkrétní úkol (task), který vždy obsahuje atributy:

**id** je integer a identifikuje úkol.

**expected\_result** je objekt, obsahující referenční řešení pro validátory popsané výše v 6.3.1. Způsob zadávání výsledků je většinou řetězcem, kde každý znak nebo podřetězec oddělený znakem `:`, reprezentuje informaci o objektu a jeho poloze vůči ostatním objektům. Pořadí je vždy zapsáno po řádcích shora dolů. Příklady reprezentace výsledků:

- `"rbrgyyyg"` pro obr. 5.2.
- `"9-4=5"` pro obr. 5.5.
- `"side_r:-facing_r:height_y:wide_y:non_rotated_b"` pro obr. 5.14.

Jednotlivé součástky mají svůj reprezentativní název. Všechny názvy jsou popsány v tabulce 6.2.

Tabulka 6.2: Názvy všech součástek.

<i>Součástka</i>	<i>Reprezentace</i>
<b>Logické řady a barevné matice</b>	<b>podle barvy</b>
barevné čtverečky	r, g, b, y
<b>Rovnice a počítání</b>	<b>korespondující počet nebo znak</b>
čtverečky i skupiny koleček	1, 2, ..., 10, +, -, =,
<b>Tvary a velikosti</b>	<b>Podle tvaru a velikosti</b>
trojúhelník	tri_sml, tri_med, tri_lrg
kruh	cir_sml, cir_med, cir_lrg
čtverec	rec_sml, rec_med, rec_lrg
tyčky	sti_u + 1, 2, ..., 10
dřevěný čtverec a kvádr	sqr, cbd
<b>Zrakové vnímání</b>	<b>Podle barvy a rotace</b>
žlutý kvádr	wide_y, height_y *
modrá krychle	rotated_b, non_rotated_b
červený trojúhelníkový hranol	side_r, facing_r **

\* wide = kvádr je natočený širší stranou k uživateli

\*\* side = trojúhelníkový hranol je natočený zkosenou stranou k uživateli

Referenční výsledky mohou být i ve formě čísla, nebo souřadnic. Číslo se použije tehdy, když má uživatel označit nějaký tvar nebo seskupení součástek, funguje tedy jako index. Souřadnice jsou využity pro reprezentaci chybějící součástky u barevných matic.

U tasku mohou být pomocné atributy, které mají efekt na zisk informací z obrazu. Seznam atributů s vysvětlivkami je vidět v tabulce 6.3.

Tabulka 6.3: Speciální atributy podle typu úlohy.

<i>Skupina součástek</i>	<i>Atribut</i>	<i>Popis</i>
Rovnice a počítání	num_of_groups	Integer. Počet skupin (clusterů) koleček.
	points_instead_of_number	Boolean. Říká, jestli jsou na obrázku čtverečky jen s body nebo znaky. *
	points_and_numbers	Boolean. Říká, jestli se na obrázku vyskytují oba typy čtverečků. *

\* Atribut je volitelný. Pouze zrychluje validaci omezením počtu možností.

## 6.4 Třída Validator

Po získání informací z databáze ohledně úlohy je vytvořena třída Validator. Třída zajišťuje volání funkcí pro získání informací z obrazu, zpracování získaných informací i výsledné porovnání výsledků. Celý proces obsahuje metoda `validate`, která je popsána v následujících podsekcích.

### 6.4.1 Získání informací ze snímku

Jako první se zavolá funkce `get_validation_requirements`, která na základě validátorů z databáze zavolá příslušné funkce pro zpracování obrazu. Tedy pokud typ úlohy obsahuje validátor `"equation"`, zavolají se funkce zodpovědné za zpracování součástí ze skupiny rovnice a počítání. Pokud je v poli validátorů například `"shapes_order"`, zavolají se funkce pro součástky ze skupiny tvary a velikosti atd.

Algoritmy použité pro získání informací z obrázku jsou popsány v kapitole 5.

### 6.4.2 Validace získaných informací

Po zpracování snímku projde funkce validátory pro daný typ úlohy a zavolá metody třídy Validator se stejným jménem. Každá taková metoda vezme informace ze snímku a provede na nich validaci. Výsledek uloží do slovníku `validated_solution` s klíčem daným názvem validátoru.

Například, metody `color_order` a `flat_rotation` se zavolají, pokud je pole validátorů ekvivalentní `["color_order", "flat_rotation"]`. Metoda `color_order` vyhodnotí řešení uživatele (pořadí barevných součástí) a uloží do slovníku `validated_solution` s klíčem `color_order`. To samé proběhne v metodě `flat_rotation`, kde se vyhodnotí rotace součástí.

### 6.4.3 Porovnání výsledků

Po kompletní validaci řešení uživatele zbývá pouze porovnání s referenčním řešením. Výsledek může být řetězec, číslo nebo souřadnice. Pro každý validátor se provede porovnání uživatelského a referenčního řešení.

Řetězce se porovnávají po znaku nebo po slovech oddělených `":"`. Finální validace se ukládá do pole. Za každé správné slovo nebo znak se vloží `"t"`, jinak `"f"`. Souřadnice je de facto pole čísel, tedy dá se vyhodnotit stejným způsobem jako řetězec. Výsledek porovnání čísel vrátí `true`, pokud se rovnají, jinak `false`.

Pokud výsledek porovnání pro jeden validátor obsahuje alespoň jeden znak `"f"` nebo `false`, řešení daného validátoru je špatné, tím pádem i řešení celého úlohy. Ukázka je vidět na obrázcích 6.2 a 6.3. Atribut `result` značí správnost celého úlohy a `detail` podrobnější údaje o výsledku všech validací.

---

# Výsledky

Algoritmy byly vyvíjeny postupně a průběžně testovány unit testy. Ty posloužily hlavně pro testování a úpravu experimentálně určených barevných hranic a parametrů tvarových charakteristik. Pro každý typ validátoru bylo použito tři až pět snímků.

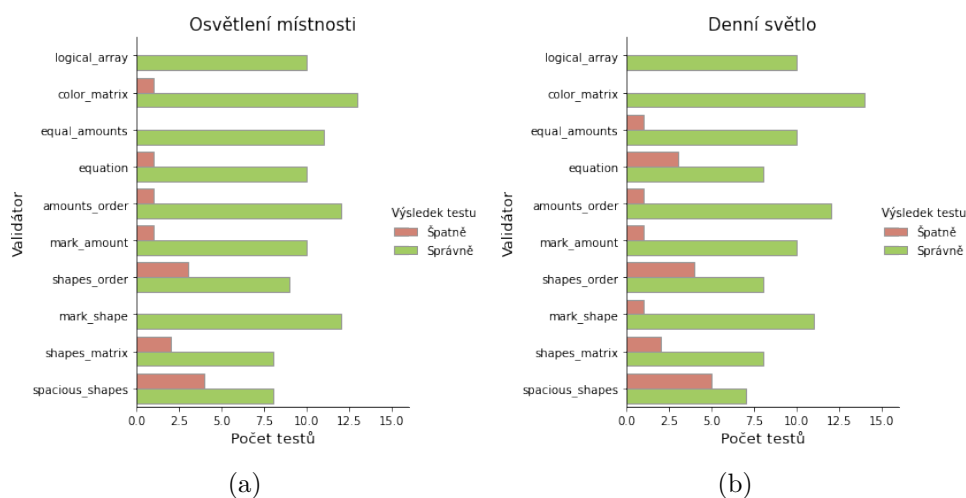
## 7.1 Zhodnocení výsledků

Pro určení úspěšnosti algoritmu byly provedeny testy validátorů. Pro každý bylo pořízeno 10 až 13 snímků, celkově 116, na které byl algoritmus spuštěn. Validace byla testována při dvou různých typech osvětlení. Na obrázku 7.1a jsou vidět výsledky validace na snímcích, které byly pořízeny při osvětlení žárovkou v místnosti, bez přítomnosti denního světla (foceno večer). Na druhém obrázku 7.1b lze pozorovat výsledky při osvětlení denním světlem. Stavebnice a úkoly jednotlivých testů jsou totožné, změna je pouze v osvětlení.

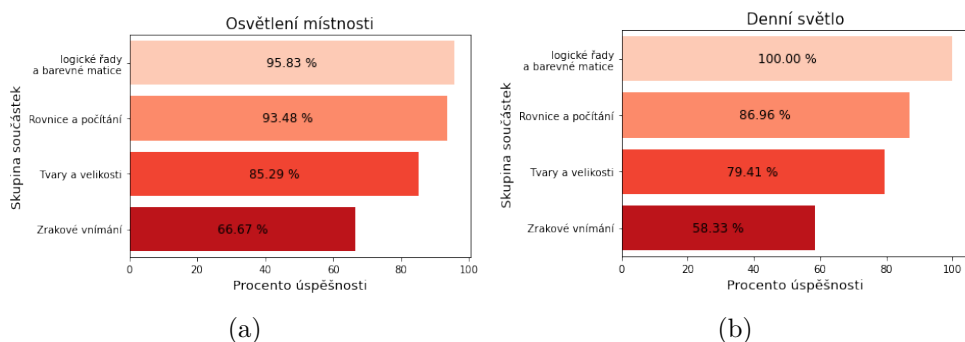
Výsledek každého testu je zadan v databázi a algoritmus musí ze snímku získat totožné řešení. Pokud řešení získané z obrázku nedopovídá tomu v databázi, algoritmus vyhodnotil snímek špatně, tedy test je špatný. Testy jsou sice rozděleny podle validátorů, ale zejména jde o správný zisk informací z obrazu. Úspěšnost lze vidět na grafech 7.2. Ty procentuálně vyjadřují úspěšnost správného zisku informací z obrazu na základě skupiny, do které validátor patří.

Z výsledků je vidět, že nejvíce spolehlivá je validace skupiny 5.4.1, která dosahuje skoro bezchybného vyhodnocení při obou typech osvětlení. Druhá skupina 5.4.2, na kterou bylo aplikováno nejvíce testů, dopadla také velmi dobře. Poslední dvě skupiny 5.4.3 a 5.4.4 obsahují problémové součástky popsané v sekci 7.2. Jejich výsledky jsou tedy o něco hroší. Pokud se problémové součástky na obrázku neobjeví, validace je ve většině případů správná.

## 7. VÝSLEDKY



Obrázek 7.1: Počet správně a špatně vyhodnocených snímků jednotlivými validátory. (a) Snímky osvětlené žárovkou v místnosti. (b) Snímky osvětlené denním světlem.



Obrázek 7.2: Procentuálně vyjádřen počet správných vyhodnocení ku všem testům z dané skupiny. (a) Snímky osvětlené žárovkou v místnosti. (b) Snímky osvětlené denním světlem.

### 7.2 Problémy

Největší problém způsobuje barevná segmentace součástek kvůli nekonzistenci osvětlení a jejich tvarové různorodosti (ploché vs. prostorové).

Práh pro barevnou segmentaci součástek je fixní. Uživatel potřebuje se stavebnicí interagovat, nepřichází tedy v úvahu scénu zakrýt a osvětit. Tím se defakto znemožňuje konzistence segmentace součástek, jelikož osvětlení nebude nikdy stejné. Například pokud se na scénu svítí žárovkou, která má žluté světlo, součástky mají lehce odlišnou barvu, než pokud jsou nasvíceny denním světlem. To nevádí u součástek, které mají od ostatních dostatečně odlišnou barvu (modrá vs. červená), jelikož potom lze práh zadefinovat dostatečně velký

tak, aby do něj různé odstíny spadaly. To se nedá říct o dřevěných objektech, jelikož jsou barevně nekonzistentní a obsahují odstíny barev jako jsou žlutá, oranžová a hnědá. Ve většině případů jsou zároveň i prostorové.

Pokud jsou součástky prostorové, jsou přirozeně světlejší ze strany, z které světlo přichází, a tmavší na stranách odkloněných od zdroje světla. Barevně poté odpovídá jen ta část, která je nasvícená. Pokud se podaří dobře vysegmentovat alespoň nasvícenou část, algoritmus je ve většině případů schopný součástku určit.

Automatické zaostřování tabletu je také problém. Focení snímků pro testování probíhalo za různých světelných podmínek, ale vždy se zaostřením na oblast zájmu. Bohužel nelze předpokládat, že uživatel bude při focení scény manuálně zaostřovat.

Jako poslední se hodí zmínit problém s měřením součástek. Díky tomu, že se rozměry měří pouze v pixelech a není k dispozici žádná informace o hloubce objektů ve snímku, hodnoty jsou velmi nestabilní a závislé na pozici součástky vůči kameře. Komplikace se týká zejména tyček a žlutého kvádrů. Tyček je 10 velikostí, které se od sebe délkou liší o jeden centimetr. Žlutý kvádr má strany velikostně velmi podobné.





---

## Diskuze

Navržený algoritmus funguje, byť ne na 100 %. Přináší s sebou jisté problémy, viz sekce 7.2, a některé úlohy vůbec neřeší, viz sekce 5.5. Hlavními příčinami jsou nekonzistence osvětlení a zaostření, nedokonalé zabarvení, překrývající se součástky a chybějící informace o kalibraci kamery nebo hloubkové informaci, pomocí které by se daly měřit rozměry součástek v reálných velikostech.

Vzdělávací nástroj iSopHi SMART nebyl navržen s úmyslem automatické validace strojem, jelikož některé součástky jsou nedokonale nabarvené. Změna a konzistence zabarvení by vedla ke zlepšení výsledků. Pokud by například tyčky byly nabarvené pouze na vrchní straně jednolitou barvou, nebylo by potřeba konturu aproximovat. Z nabarvené části by se daly, bez perspektivní deformace, změřit rozměry tyčky. Zároveň by se vylepšila segmentace objektu. Taková modifikace by pomohla při segmentaci u všech součástek.

Konzistence osvětlení je složitý problém. Se stavebníci pracují děti ve školce, kde je většinou osvětlení řešeno pouze žárovkou v místnosti nebo denním světlem. Příliš nebo naopak málo světla může způsobit špatné zaostření tabletu a skoro jistě špatnou validaci. To naznačuje další komplikaci, a to snímací zařízení. Snímání pomocí tabletu není kvůli automatickému zaostřování ideální, ale je to logická volba pro co největší využití, jelikož na něm běží aplikace iSopHi SMART. Všechny nekonzistence naznačují, že výměna snímacího zařízení za kalibrovatelnou kameru nebo zajištění umělého osvětlení by jistě vedlo k lepším výsledkům. Nabízí se ale i možnost vyzkoušet odlišný způsob zpracování obrazu, založený na jiných než tradičních metodách.

I přes to, že zde prezentovaný pohled tabletu není špatný, za vyzkoušení by stálo i upravení pohledu tabletu na scénu pomocí zrcátka za kamerou, viz podsekce 4.1.3, například v kombinaci s jinými metodami zpracování obrazu.



---

## Závěr

Hlavní cíl práce spočíval v navržení serverové aplikace pro validaci rozložení stavebnice na obrázku. Nejdříve byla provedena metodika měření, která zahrnuje případy užití uživatelů. Mezi hlavní požadavky na snímací aparaturu patří, aby děti viděly na tablet, pohodlně dosáhly na pracovní plochu a zároveň, aby snímací zařízení vidělo celou scénu. Na základě vybrané metodiky byl navržen algoritmus, který je pomocí barevných a tvarových předpokladů součástí schopný z fotografie získat informace o objektech.

Algoritmus je implementován formou serverové aplikace, která na vstupu přijímá fotografii se stavebnicí, typ a identifikátor úlohy. Na základě identifikátoru a typu se v databázi najde referenční řešení a potřebné parametry k získání informací z obrázku. Ze získaných informací se zvaliduje uživatelské řešení, které se poté porovná s referenčním. Výsledek je odeslán zpět jako odpověď na požadavek.

Navržené řešení vykazuje dobré výsledky u dvou skupin stavebnice ze čtyř. Algoritmus byl testován za dvou světelných podmínek na 116 snímcích. Při osvětlení žárovkou v místnosti byl algoritmus schopný správně vyhodnotit 88,79 % ze všech snímků, z toho 95,83 % z první a 93,48 % z druhé skupiny. Při osvětlení denním světlem byl algoritmus schopný správně vyhodnotit 84,48 % ze všech snímků, z toho 100,00 % z první a 86,96 % z druhé skupiny. I přes pozitivní statistiky ostatních skupin, u nich nelze, kvůli nekonzistencím barvy a vlastnostem některých součástí, zaručit dobré výsledky.

Práce navazuje na již existující vzdělávací nástroj iSophi SMART, který rozšiřuje o validační systém. Jedná se tedy o studii proveditelnosti.



---

## Literatura

- [1] Aplikace iSopli SMART [online]. iSopli Education s. r. o. [cit. 2021-05-07]. Dostupné z: <https://smart.isopli.education>
- [2] Comaniciu, D.; Meer, P.: Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online], ročník 24, č. 5, 2002: s. 603–619, doi:10.1109/34.1000236, [cit. 2021-04-01]. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/1000236>
- [3] Maitra, M.; Gupta, R. K.; Mukherjee, M.: Detection and counting of red blood cells in blood cell images using Hough transform [online]. *International journal of computer applications*, ročník 53, č. 16, 2012, [cit. 2021-04-08]. Dostupné z: [https://www.researchgate.net/profile/Mausumi-Maitra/publication/258652272\\_Detection\\_and\\_Counting\\_of\\_Red\\_Blood\\_Cells\\_in\\_Blood\\_Cell\\_Images\\_using\\_Hough\\_Transform/links/56e94c5a08aedfed738986b0/Detection-and-Counting-of-Red-Blood-Cells-in-Blood-Cell-Images-using-Hough-Transform.pdf](https://www.researchgate.net/profile/Mausumi-Maitra/publication/258652272_Detection_and_Counting_of_Red_Blood_Cells_in_Blood_Cell_Images_using_Hough_Transform/links/56e94c5a08aedfed738986b0/Detection-and-Counting-of-Red-Blood-Cells-in-Blood-Cell-Images-using-Hough-Transform.pdf)
- [4] Liu, Z.; Liu, J.; Xiao, X.; aj.: Segmentation of White Blood Cells through Nucleus Mark Watershed Operations and Mean Shift Clustering [online]. *Sensors*, ročník 15, č. 9, 2015: s. 22561–22586, ISSN 1424-8220, doi:10.3390/s150922561, [cit. 2021-04-08]. Dostupné z: <https://www.mdpi.com/1424-8220/15/9/22561>
- [5] Rege, S.; Memane, R.; Phatak, M.; aj.: 2D geometric shape and color recognition using digital image processing [online]. *International journal of advanced research in electrical, electronics and instrumentation engineering*, ročník 2, č. 6, 2013: s. 2479–2487, [cit. 2021-04-08]. Dostupné z: [https://www.matlabi.ir/wp-content/uploads/bank\\_papers/gpaper/g517-www.Matlabi.ir.pdf](https://www.matlabi.ir/wp-content/uploads/bank_papers/gpaper/g517-www.Matlabi.ir.pdf)

- [6] Lai, A. H.; Fung, G. S.; Yung, N. H.: Vehicle type classification from visual-based dimension estimation [online]. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, IEEE, 2001, s. 201–206, [cit. 2021-04-08]. Dostupné z: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.858.6530&rep=rep1&type=pdf>
- [7] Cheng, H.; Jiang, X.; Sun, Y.; aj.: Color image segmentation: advances and prospects [online]. *Pattern Recognition*, ročník 34, č. 12, 2001: s. 2259–2281, ISSN 0031-3203, doi:[https://doi.org/10.1016/S0031-3203\(00\)00149-7](https://doi.org/10.1016/S0031-3203(00)00149-7), [cit. 2021-04-01]. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0031320300001497>
- [8] Jiřina, M.; Novák, J.; Brchl, L.: BI-SVZ 9. přednáška. *Segmentace obrazu – Houghova transformace a plošné segmentace* [online]. © 2019 FIT ČVUT. Poslední změna 2021-02-10 [cit. 2021-04-01]. Dostupné z: <https://courses.fit.cvut.cz/BI-SVZ/@B192/lectures/files/bi-svz-09-segmentace-obrazu-plosne.pdf>
- [9] Beucher, S.: Image segmentation and mathematical morphology [online]. 2010, poslední změna 2010-05-18 [cit. 2021-04-02]. Dostupné z: <http://www.cmm.mines-paristech.fr/~beucher/wtshed.html>
- [10] Morphological Transformations [online]. OpenCV docs. [cit. 2021-04-07]. Dostupné z: [https://docs.opencv.org/master/d9/d61/tutorial\\_py\\_morphological\\_ops.html](https://docs.opencv.org/master/d9/d61/tutorial_py_morphological_ops.html)
- [11] iSophi SMART [online]. iSophi Education s. r. o. [cit. 2021-05-07]. Dostupné z: <https://isophi.education/smart/>
- [12] LEGO®Education – About us [online] © 2020 The LEGO Group. [cit. 2021-05-07]. Dostupné z: <https://education.lego.com/en-us/about-us>
- [13] WeDo 2.0 software [software] © 2020 The LEGO Group. [cit. 2021-05-07]. Dostupné z: <https://education.lego.com/en-us/downloads/wedo-2/software>
- [14] LEGO®Education – Get ready with Lessons [online] © 2020 The LEGO Group. [cit. 2021-05-07]. Dostupné z: <https://education.lego.com/en-us/lessons>
- [15] Primotoys education – Coding, STEM numeracy and creativity delivered in a single product [online] © 2017 Primo Toys. [cit. 2021-05-07]. Dostupné z: <https://www.primotoys.com/education/>

- 
- [16] LittleBits – LET’S GET STARTED [online] © 2020 Sphero Inc. [cit. 2021-05-07]. Dostupné z: <https://classroom.littlebits.com/getting-started>
- [17] Welcome to the littleBits classroom [online] © 2020 Sphero Inc. [cit. 2021-05-07]. Dostupné z: <https://classroom.littlebits.com/welcome>
- [18] Obrázek LEGO Education DUPLO stroje [online] STEM Education s. r. o. [cit. 2021-05-07]. Dostupné z: <https://www.jeduedu.cz/wp-content/uploads/2017/12/duplostroje.jpg>
- [19] Obrázek robot Cubetto playset [online] STEM Education s. r. o. [cit. 2021-05-07]. Dostupné z: <https://www.jeduedu.cz/wp-content/uploads/2017/12/Fullcomposition-1-1728x0-c-default-1024x700.png>
- [20] Obrázek STEAMPlusKit littleBits [online] © 2021 Sphero Inc. [cit. 2021-05-07]. Dostupné z: [https://cdn.shopify.com/s/files/1/0306/6419/6141/products/STEAMPlusKit\\_MAIN\\_700x.jpg?v=1597330301](https://cdn.shopify.com/s/files/1/0306/6419/6141/products/STEAMPlusKit_MAIN_700x.jpg?v=1597330301)
- [21] Fauge mobile phone 90 degree periscope camera lens [online] 2021 Amazon Inc. [cit. 2021-05-08]. Dostupné z: <https://www.amazon.com/Semoic-Universal-Detachable-Magnetic-Periscope/dp/B07TV4JM38>
- [22] Comaniciu, D.; Meer, P.: Mean shift analysis and applications [online]. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, ročník 2, 1999, s. 1197–1203 vol.2, doi: 10.1109/ICCV.1999.790416, [cit. 2021-04-06]. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/790416>
- [23] Neal, F. B.; Russ, J. C.: *Measuring shape* [online]. CRC Press, 2012, ISBN 9781439855980, [cit. 2021-04-03]. Dostupné z: <https://books.google.cz/books?id=g4JpsUCFOAwC>
- [24] Image Thresholding [online]. OpenCV docs. [cit. 2021-04-01]. Dostupné z: [https://docs.opencv.org/master/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html)
- [25] Burger, W.; Burge, M. J.: *Principles of digital image processing* [online], ročník 111. Springer, 2009, [cit. 2021-04-01]. Dostupné z: <https://link.springer.com/content/pdf/10.1007/978-1-84882-919-0.pdf>
- [26] Golodetz, S.: Watersheds and Waterfalls (Part 1) [online]. *Overload*, ročník 16, č. 83, 2008: s. 4–9, ISSN 1354-3172, [cit. 2021-04-02]. Dostupné z: [https://ora.ox.ac.uk/objects/uuid:c9a9bb8a-2ac6-493d-8a5c-01c6851cd680/download\\_file?file\\_format=pdf&safe\\_filename=overload83.pdf&type\\_of\\_work=Journal+article](https://ora.ox.ac.uk/objects/uuid:c9a9bb8a-2ac6-493d-8a5c-01c6851cd680/download_file?file_format=pdf&safe_filename=overload83.pdf&type_of_work=Journal+article)

- [27] Canny, J.: A Computational Approach to Edge Detection [online]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník PAMI-8, č. 6, 1986: s. 679–698, doi:10.1109/TPAMI.1986.4767851, [cit. 2021-04-03]. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/4767851>
- [28] Hough Line Transform [online]. OpenCV docs. [cit. 2021-04-01]. Dostupné z: [https://docs.opencv.org/3.4/d9/db0/tutorial\\_hough\\_lines.html](https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html)
- [29] Duda, R. O.; Hart, P. E.: Use of the Hough Transformation to Detect Lines and Curves in Pictures [online]. *Commun. ACM*, ročník 15, č. 1, leden 1972: str. 11–15, ISSN 0001-0782, doi:10.1145/361237.361242, [cit. 2021-04-01]. Dostupné z: <https://doi.org/10.1145/361237.361242>
- [30] Ming-Kuei Hu: Visual pattern recognition by moment invariants [online]. *IRE Transactions on Information Theory*, ročník 8, č. 2, 1962: s. 179–187, doi:10.1109/TIT.1962.1057692, [cit. 2021-04-03]. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/1057692>
- [31] Bradski, G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools* [software], 2000, [cit. 2021-04-14].
- [32] Structural Analysis and Shape Descriptors [online]. OpenCV docs. [cit. 2021-04-06]. Dostupné z: [https://docs.opencv.org/2.4/modules/imgproc/doc/structural\\_analysis\\_and\\_shape\\_descriptors.html](https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html)
- [33] Contours: Getting Started [online]. OpenCV docs. [cit. 2021-04-03]. Dostupné z: [https://docs.opencv.org/3.4/d4/d73/tutorial\\_py\\_contours\\_begin.html](https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html)
- [34] Ramer–Douglas–Peucker algorithm [online] Worddisk. [cit. 2021-04-03]. Dostupné z: [https://worddisk.com/wiki/Ramer\OT1\textendashDouglas\OT1\textendashPeucker\\_algorithm/](https://worddisk.com/wiki/Ramer\OT1\textendashDouglas\OT1\textendashPeucker_algorithm/)
- [35] Douglas, D. H.; Peucker, T. K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, ročník 10, č. 2, 1973: s. 112–122.
- [36] Contour features [online]. OpenCV docs. [cit. 2021-04-06]. Dostupné z: [https://docs.opencv.org/3.4/dd/d49/tutorial\\_py\\_contour\\_features.html](https://docs.opencv.org/3.4/dd/d49/tutorial_py_contour_features.html)
- [37] Jiřina, M.; Novák, J.; Brechl, L.: BI-SVZ 5. přednáška. *Perspektiva a geometrie obrazu* [online]. © 2019 FIT ČVUT. Poslední změna 2021-02-10 [cit. 2021-04-07]. Dostupné z: <https://courses.fit.cvut.cz/BI-SVZ/lectures/files/bi-svz-05-perspektiva-obrazu.pdf>



- 
- [38] Stachniss, C.: Homogeneous Coordinates. In *StachnissLab: Photography and Robotics* [online]. 2020, [cit. 2021-04-03]. Dostupné z: <http://www.ipb.uni-bonn.de/html/teaching/3dcs-ge-2020/stachniss/2020-3dcs-02-homogeneous-coords-4.pdf>
- [39] Burger, W.; Burge, M. J.: Geometric operations [online]. In *Principles of Digital Image Processing*, Springer, 2009, s. 1–64, [cit. 2021-04-03]. Dostupné z: [https://link.springer.com/content/pdf/10.1007/978-1-84800-195-4\\_10.pdf](https://link.springer.com/content/pdf/10.1007/978-1-84800-195-4_10.pdf)
- [40] Criminisi, A.; Reid, I.; Zisserman, A.: Single view metrology [online]. *International Journal of Computer Vision*, ročník 40, č. 2, 2000: s. 123–148, [cit. 2021-04-03]. Dostupné z: <https://doi.org/10.1023/A:1026598000963>
- [41] Klouda, K.; Vařata, D.: BI-VZD 3. přednáška. *Úvod do nesupervizovaného učení. Shluková analýza. Hierarchické shlukování. Algoritmus k-means* [online]. © 2019 FIT ČVUT. Poslední změna 2021-04-19 [cit. 2021-04-28]. Dostupné z: <https://courses.fit.cvut.cz/BI-VZD/lectures/files/BI-VZD-03-cs-handout.pdf>
- [42] Kay, A.: Tesseract: An Open-Source Optical Character Recognition Engine [software]. *Linux J.*, ročník 2007, č. 159, Červenec 2007: str. 2, ISSN 1075-3583, [cit. 2021-04-06]. Dostupné z: <https://dl.acm.org/doi/10.5555/1288165.1288167>
- [43] Jiřina, M.; Novák, J.; Brchl, L.: BI-SVZ 10. přednáška. *Morfologie a tvarové charakteristiky* [online] © 2019 FIT ČVUT. Poslední změna 2021-02-10 [cit. 2021-04-06]. Dostupné z: <https://courses.fit.cvut.cz/BI-SVZ/lectures/files/bi-svz-10-morfologie-a-tvarove-charakteristiky.pdf>
- [44] iPad mini 2 with Retina display – Technical Specifications [online] Apple support: © 2021 Apple Inc. [cit. 2021-04-13]. Dostupné z: [https://support.apple.com/kb/sp693?locale=en\\_US](https://support.apple.com/kb/sp693?locale=en_US)
- [45] Framework AVFoundation [software] Apple developer: © 2021 Apple Inc. [cit. 2021-04-13]. Dostupné z: <https://developer.apple.com/avfoundation/>
- [46] AVCaptureCalibrationData – intrinsicMatrix [online] Apple developer documentation: © 2021 Apple Inc. [cit. 2021-04-13]. Dostupné z: <https://developer.apple.com/documentation/avfoundation/avcameracalibrationdata/2881135-intrinsicmatrix>
- [47] Van Rossum, G.; Drake, F. L.: *Python 3 Reference Manual* [software]. Scotts Valley, CA: CreateSpace, 2009, ISBN 1441412697, [cit. 2021-04-15].

- [48] Grinberg, M.: *Flask web development: developing web applications with python* [software]. O'Reilly Media, Inc., 2018, [cit. 2021-04-15]. Dostupné z: <https://flask.palletsprojects.com/en/1.1.x/>
- [49] Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; aj.: Jupyter Notebooks – a publishing format for reproducible computational workflows [software]. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, editace F. Loizides; B. Schmidt, IOS Press, 2016, s. 87 – 90, [cit. 2021-04-15].

## Seznam Použitých zkratk

**px** Pixel

**B** Bajt

**KSS** Kartézská souřadnicová soustava

**PSS** Polární souřadnicová soustava

**MCW** Marker-controller watershed

**HT** Houghova transformace

**TP** True positive

**FP** False positive

**TN** True negative

**FN** False negative

**REST** Representational State Transfer



---

## Obsah přiloženého CD

	README.md	.....	stručný popis a nastavení aplikace
	environment.yml	.....	konfigurační soubor pro nastavení conda prostředí
	thesis.pdf	.....	text práce ve formátu PDF
	src		
		thesis	..... zdrojový kód práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
		algorithms_jup	..... ukázka algoritmů použitých v aplikaci
		final_results_visualization	..... vizualizace výsledků finálního testování
		final_testing_images	..... dataset pro finální testování
		iSophi_SMART_BE	..... zdrojový kód serverové aplikace

