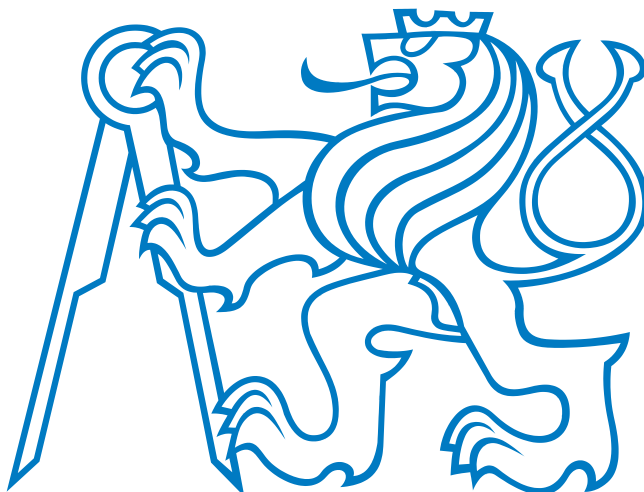


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ

BAKALÁŘSKÁ PRÁCE



Karel Lenc

Počítačové vidění robota hry Eurobot 2010

Katedra kybernetiky, Centrum strojového vnímání

Vedoucí bakalářské práce: doc. Dr. Ing. Jiří Matas

Praha, 2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Karel L e n c
Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření
Název tématu: Počítačové vidění robota hry Eurobot 2010

Pokyny pro vypracování:


1. Seznamte se základními metodami detekce jednobarevných objektů s jednoduchým tvarem (s částmi, které lze aproximovat pomocí koulí, válců a hranolů) založených na barevné segmentaci a na hranové detekci.
2. Navrhněte a implementujte systém rozpoznávání těchto objektů se zaměřením na ty, které se vyskytují ve scénáři soutěže Eurobot 2010.
3. Vytvořte jednoduchý 3D model scény, do kterého budete vkládat informaci o rozpoznávaných a lokalizovaných objektech ve "světových" souřadnicích, tj. v souřadnicích nezávislých na poloze a orientaci robota.
4. Navrhněte metody na efektivní udržování informace o objektech v okolním prostředí.
5. Ověřte funkci systému na datech z reálných scén, vyhodnoťte jeho úspěšnost, analyzujte příčiny selhání.

Seznam odborné literatury:

- [1] Sonka, M.; Hlavac, V.; Boyle, R.: Image Processing, Analysis, and Machine Vision. Toronto: Thomson, 2008.
- [2] Hartley, R.; Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge: Cambridge University, 2003.
- [3] Forsyth, D.A.; Ponce, J.: Computer Vision: a Modern Approach. Upper Saddle River: Prentice Hall, 2003.

Vedoucí bakalářské práce: doc. Dr. Ing. Jiří Matas

Platnost zadání: do konce zimního semestru 2010/2011


prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry





doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 8. 1. 2010

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, SW, projekty atd.) uvedené v příloženém seznamu.

V Praze dne25.5.2010.....

..........
podpis

Abstrakt

Práce popisuje modul vidění robota pro hru Eurobot 2010. Cílem modulu je rozpoznat v okolí předměty tří kategorií a určit jejich pozice. Hledání předmětů je založeno na barvě. Pomocí dichromatického barevného modelu objektů dané kategorie zjišťujeme oblasti, které mohou být obrazem hledaných objektů. Z bodu uvnitř nalezené oblasti hledáme hranové body, ze kterých se ověřuje hypotéza přítomnosti objektu robustním způsobem pomocí metody RANSAC. Pro vidění byl sestaven systém vestavěný do herních majáčků umístěných na okrajích hřiště. Na třiceti testovacích obrázcích z různých herních situací při proměnlivém osvětlení byla dosažena úspěšnost 89% u rajčat a 56% u kukuřic.

Abstract

This thesis describes the vision module created for robot competing in Eurobot games 2010. The vision system is built into three beacons placed in edges of the playground. The aim is to identify three types of objects used in the game and determine their positions. Detection method is based on color. Using dichromatic color model we find areas which could be, with high probability, assigned to particular object's surface. From the point inside the area we look for edge points and with them we try to verify the object presence in robust manner with RANSAC method. Testing of our algorithm, on thirty images from a variety of game situations and in different lightning, resulted in the success rate of 89% for tomatoes and 56% for maize.

Poděkování

Děkuji Doc. Jiřímu Matasovi za vstřícnost, cenné rady a trpělivost při konzultacích k této práci. Dále děkuji Ing. Pavlu Krskovi a Ing. Vladimíru Smutnému za rady a pomoc při sestavování hardwarové části práce. Při výrobě kamerových majáčků a všeobecnou podporou mi také velice pomohli kolegové Lenka Mudrová a Jaroslav Halgašík z týmu FELaaCZech. Děkuji rodině za podporu nejen při studiu.

Obsah

Úvod	1
1 Specifikace problému	2
1.1 Eurobot 2010	2
1.2 Popis robota	3
1.2.1 Hardwarové prostředky	3
1.2.2 Softwarová struktura robota	4
1.2.3 Základní taktiky hry	5
1.3 Požadavky na modul vidění	5
1.3.1 Komunikace modulu vidění s robotem	5
2 Analýza problému počítačového vidění robota	7
2.1 Vizuální vlastnosti snímaného obrazu	7
2.1.1 Osvětlení	7
2.1.2 Hřiště	8
2.1.3 Herní prvky	8
2.1.4 Protivník	8
2.2 Polohové vlastnosti herních prvků	8
3 Detekce herních objektů z obrázku	9
3.1 Dichromatický model odrazivosti	9
3.2 RANSAC	10
3.3 Zpracování vstupního obrazu	10
3.4 Detekce rajčat	10
3.4.1 Detekce oblastí	12
3.4.2 Test tvaru souvislé oblasti	12
3.4.3 Zjištění hranových bodů	14
3.4.4 Hledání parametrů kružnice	14
3.5 Detekce kukuřic	15
3.5.1 Detekce oblastí	18
3.5.2 Určení hranových bodů kukuřice	18
3.5.3 Určení barvy herního prvku	19
3.6 Ověřování přítomnosti herního prvku na hřišti	19
3.6.1 Ověřování přítomnosti pomerančů	19
4 Detekce herní varianty	20
5 Reprezentace prostředí	21
5.1 Souřadnicové systémy	21
5.2 Určení polohy objektu z obrázku	21
5.2.1 Transformace souřadnic	23
5.2.2 Kalibrace kamery	24
5.3 Mapa prostředí	25
5.3.1 Reprezentace objektů v mapě	25
5.3.2 Správa objektů v mapě	26

6 Implementace	27
6.1 Softwarová implementace	27
6.2 Hardwarová implementace	27
7 Experimenty	30
7.1 Ideální průběh detekce	30
7.2 Spolehlivost detekce	33
7.3 Přesnost určení pozice objektu v obrázku	35
7.4 Výpočetní náročnost	35
7.5 Výsledky ze soutěže	35
8 Závěr	37
Literatura	38
Příloha A: Obsah CD	39

Úvod

Cílem práce je navrhnout a realizovat modul vidění pro robota vyvíjeného týmem FEL-aaCzech pro soutěž Eurobot 2010. Soutěž je každoročně organizována sdružením Eurobot a jejími lokálními partnery. Základním úkolem tohoto modulu je po celou dobu hry uchovávat informace o rozmístění herních prvků na hracím hřišti. Díky přesně daným pravidlům hry je herní prostředí poměrně přesně specifikováno. Všechny herní prvky se na začátku hry nacházejí na daných pozicích a některé jsou také snadno barevně rozlišitelné. Modul by ale měl být co nejvíce odolný proti chybám způsobeným změnami v osvětlení snímané scény, proto je kladen důraz na robustnost způsobu detekce objektů z obrázku.

V kapitole 1 je uvedena základní specifikace řešeného problému. Nejprve uvádíme základní pravidla soutěže, následují podstatné informace o robotovi a nakonec definujeme konkrétní úkoly modulu počítačového vidění. Kapitola 2 se věnuje analýze situací, které mohou ve vstupních datech nastat. Kapitola 3 se zabývá řešením problému detekce objektů z obrázku, jehož výsledkem je obecně jejich pozice ve vstupních datech. V kapitole 5 navrhujeme, jak tyto objekty spravovat a jak z polohy objektu v obrázku zjistit v jaké části hřiště se nachází. Implementaci softwarové i hardwarové části práce popisujeme v 6. kapitole. Kapitola 7 obsahuje výsledky experimentů a vypracované řešení zadaného problému.

1 Specifikace problému

1.1 Eurobot 2010

Eurobot je každoročně pořádaná mezinárodní robotická soutěž, jejímž cílem je mimo jiné podporovat kreativitu a výměnu zkušeností mezi soutěžícími. Soutěž se snaží vést k vytváření spolupracujících robotů, kteří se navzájem neohrožují (Eurobot, 2009, s. 4). Každý ročník má jiné téma a pravidla, přičemž některé základní parametry jsou vždy stejné:

- Na hracím stole spolu soupeří vždy dva roboti.
- Hra trvá přesně 90 s a po nich se roboti musí zastavit a neprojevat žádnou činnost.
- Robot v žádném případě nesmí ohrozit či omylem narazit do robota protivníka.
- Délka konvexní obálky půdorysu robota při startu nesmí překročit 1200 mm, během hry nesmí překročit 1400 mm.
- Výška robotů nepřesahuje 350 mm.
- Rozměry hracího stolu jsou 3000 × 2100 mm.

Parametry a schopnost robota zúčastnit se hry kontrolují pořadatelé při homologacích.

Pokud robot v průběhu hry poruší pravidla, může být na základě rozhodnutí rozhodčích penalizován (odečtení soutěžních bodů) nebo až diskvalifikován. Mezi porušení pravidel patří:

- Robot se silně srazí s protivníkem.
- Robot záměrně zabraňuje oponentovi dosažení herního prvku.
- Robot záměrně zabraňuje soupeři dosažení jeho cílové oblasti.
- Robot provádí akci nebo rozložení, která nebyla předem schválena.

Viníka srážky robotů a bodovou diskvalifikaci určuje rozhodčí po skončení zápasu.

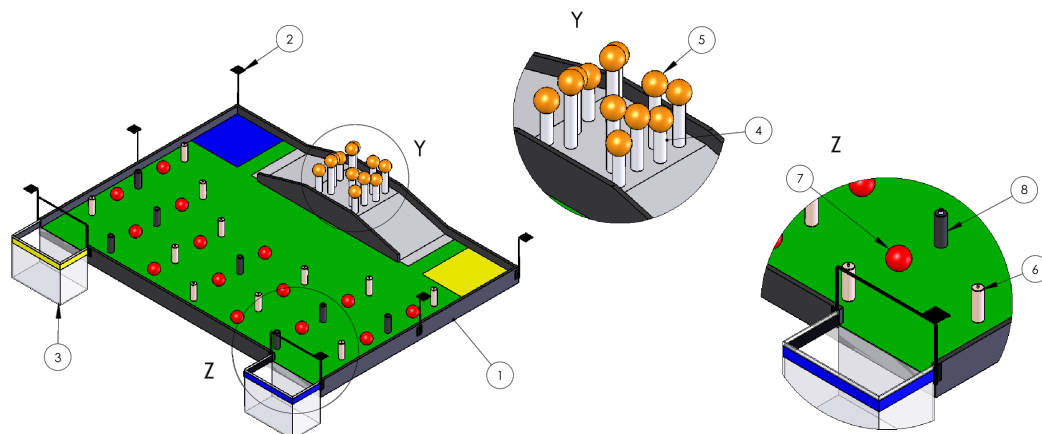
Tématem ročníku 2010 je sklizeň úrody. Cílem hry je sesbírat a odvézt do vlastního kontejneru co nejvíce herních prvků. Ty jsou společné pro oba soupeře a výsledné skóre hry je odvozeno z jejich hmotnosti. Na hřišti se vyskytují tyto objekty:

Rajčata: Červené žonglovací míčky, které jsou před začátkem hry umístěny do 14ti daných pozic. Mají průměr 100 mm a váží 150 g.

Kukuřičné klasy: Dřevěnné válečky zaaretované v 18 daných pozicích. Na náhodně vylosovaných 7 pozic jsou umístěny černé falešné kukuřice, které jsou pevně připevněné ke stolu. Zbylé kukuřice jsou bílé barvy a lze je ze stolu vytáhnout nebo povalit (Eurobot, 2010). Váha kukuřičného klasu je 250 g.

Pomeranče: Oranžové žonglovací míčky umístěné na vyvýšené plošině. Jsou položeny na dutých, pevně připevněných válcích, které představují kmeny stromů. Pomeranče mají nejvyšší hodnotu, váží 300 g a na hřišti je celkem 12 kusů.

Vítězí robot, který dosáhl vyššího skóre (tj. hmotnost ovoce vítězného robota v jeho kontejneru je vyšší než soupeře).



Obrázek 1.1 Hřiště s herními prvky hry Eurobot 2010: 1. Bočnice hřiště, 2. Pozice pro umístění majáčků, 3. Cílový kontejner, 4. Sloupek pomeranče („větev“) 5. Pomeranč, 6. Pravá kukuřice, 7. Rajče, 8. Falešná kukuřice (Eurobot, 2009, s. 36).

Plocha hřiště (zobrazeného na obrázku 1.1) je natřena zelenou barvou s černými bočnicemi. Na hřišti jsou dvě rovnocenné startovací zóny (žlutá a modrá) jejichž přidělení je dopředu známé. Pro oba týmy jsou na krajích hřiště k dispozici sloupky pro lokalizační majáky a je také možné, dle pravidel, umístit na soupeřova robota další lokalizační aparaturu daných rozměrů. Jednotlivé typy prvků jsou snadno barevně odlišitelné s výjimkou falešných kukuřic, které mají stejnou barvu povrchu jako krajnice hřiště. Více lze vidět na obrázku 1.1. Poloha prvků na začátku hry je pevně dána, ale v průběhu hry je možné s nimi, mimo falešných kukuřic, téměř libovolně manipulovat. Nesmí se však házet.

Všechna zmiňovaná pravidla byla čerpána z oficiálních materiálů ke hře (Eurobot, 2009) spolu s dalšími podrobnostmi. Tyto pravidla jsou každoročně vydávána pořadatelem s družením Planète Sciences.

1.2 Popis robota

Zde bychom chtěli popsat základní parametry robota. Jak již bylo uvedeno v sekci 1.1, robot musí splňovat určité rozměrová kritéria. Robot je navržen modulárně po hardwarové i softwarové stránce, což přináší výhody jak ve formě znovupoužitelnosti, tak ve formě snazšího vývoje.

1.2.1 Hardwarové prostředky

Základním výpočetním hardwarem robota je jednodeskový mikropočítač BeagleBoard¹ postavený na procesoru TI OMAP3530. Jedná se poměrně výkonný SoC² s jádrem ARM Cortex-A8 pracujícím na frekvenci 600MHz.

¹<http://beagleboard.org/hardware>

²System-on-a-chip

Pro regulátory a další nízkourovňové moduly jsou k dispozici dvě desky Arduina Mega³, které jako výpočetní jádro využívají mikro-kontroléry ATmega1280. S těmito deskami se po softwarové stránce komunikuje totožně jako s ostatními moduly. Zprávy se odesílají do speciálního modulu, který zajišťuje příjem a odesílání zpráv přes RS232.

Pro rozpoznávání je také k dispozici Toradex Robin Z530⁴, což je x86 embedded PC o velikosti Nano COM ExpressTM s procesorem Intel[®] AtomTM. V našem případě je použit spolu s Daisy Pico-ITX rozšiřujícím modulem, který umožňuje snadné připojení USB zařízení, Ethernetu RJ-45 a analogového VGA konektoru. Tento hardware byl vybrán pro své malé rozměry a nízkou spotřebu. Zároveň s sebou v tomto malém balení přináší i všechny výhody klasického PC.

1.2.2 Softwarová struktura robota

Operačním systémem hlavní výpočetní jednotky je GNU/Linux. Jednotlivé softwarové moduly jsou reprezentovány jako procesy, takže sdílení času procesoru obstarává operační systém na základě vhodně zvolených priorit. Hlavní modul má na starost pouze spouštění procesů a komunikaci mezi nimi.

Základní hierarchie modulů a směr komunikace mezi nimi je znázorněn na obrázku 1.2. Herní strategii a chování robota řídí modul herního plánování. Ten posílá příkazy modulům brouk a sběrač, které mají za úkol řídit pohyb robota.

Brouk například obdrží od herního plánování úkol dostat se na určitou pozici na hřišti. Na základě informací z modulu pro správu pozice vypočítá nejvhodnější trajektorii a podle nich vygeneruje posloupnost příkazů pro ovládání kol. Pokud narazí na problém, oznámí to nadřazenému modulu. Podobné úkoly řeší i modul sběrač s tím, že jeho úkolem je ovládat manipulátory pro manipulaci s herními prvky.

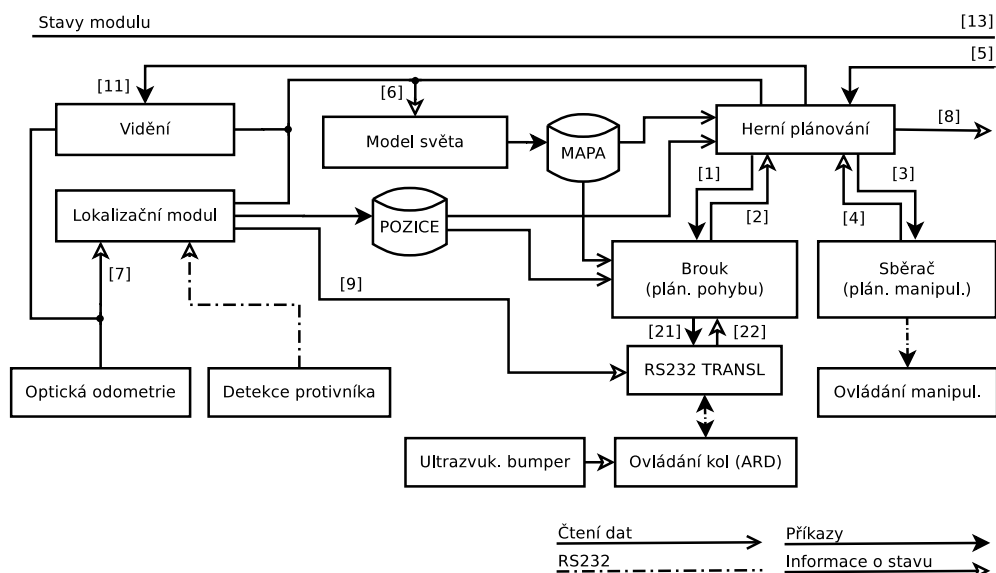
Lokalizační modul má za úkol, na základě přijatých dat od modulů ovládajících lokalizační senzory, určit nejpravděpodobnější pozici a natočení robota na hřišti. Zároveň také uchovává informace o pozici protivníka. Tyto informace ukládá jak do souboru pozice, tak je odesílá na komunikační kanál, odkud jsou pomocí RS232 translatoru přeposílány modulu ovládání kol. Tento modul na jejich základě může provádět určité korekce pohybu.

Model světa má za úkol spravovat soubor mapy, která je využívána především herním plánováním pro určení strategie na základě pozic herních prvků. Tento modul je úzce svázán s modulem vidění, jehož implementace je cílem této práce.

Komunikace mezi moduly je zprostředkovávána hlavním modulem a probíhá přes síťové sockety pomocí jednoduchých zpráv. Zprávy mají pevně danou hlavičku a liší se obsahem podle typu modulu. Spojení jsou logicky uspořádána do několika komunikačních kanálů a hlavní modul odpovídá za přeposílání zpráv mezi nimi. Mezi některými moduly je vložen mezičlen (RS232 translator), který zajišťuje překlad zpráv a jejich rozesílání po jiné fyzické vrstvě. Výhodou tohoto řešení je, že lze velmi snadno zprávy odposlouchávat a tím kontrolovat provoz robota. Dále pokud bychom chtěli rozšiřovat robota o další výpočetní jednotky, lze je jednoduše propojit standardním síťovým rozhraním.

³<http://arduino.cc/en/Main/ArduinoBoardMega>

⁴http://www.toradex.com/En/Products/Robin/Robin_Modules



Obrázek 1.2 Moduly robota a komunikační kanály mezi nimi.

1.2.3 Základní taktiky hry

Při samotné hře se počítá s několika základními taktikami zaměřené na:

Spolehlivost: Po startu nejkratší a nejspolehlivější cestou dopravit do cílové oblasti minimálně jedno rajče a až do konce hry se snažit dopravit do cíle co nejvíce zbývajících herních prvků. V tomto případě je potřeba udržovat aktuální informace o polohách herních prvků na hřišti.

Sběr vlastních pomerančů: Po startu sesbírat pomeranče u startovní pozice a dopravit je do cíle. Následně se pokusit, dle rozložení ostatních herních prvků na hřišti, do cíle dopravit zbývajících ovoce. Stejně jako v minulé strategii je potřeba udržovat aktuální informace o polohách herních prvků na hřišti.

Sběr všech pomerančů: Po odvezení pomerančů ze startovní pozice se pokusit odvézt tyto herní prvky i z protivníkovi startovní pozice. V tomto případě je potřeba zjistit, zda soupeř tyto herní prvky již neodvezl. Následně pokračovat jako v předchozích strategiích.

1.3 Požadavky na modul vidění

Úkolem navrhovaného modulu je správně lokalizovat hrací prvky na herním poli a výsledky detekce odeslat do společného mapového modulu robota.

1.3.1 Komunikace modulu vidění s robotem

Z obrázku 1.2 lze vidět s jakými moduly rozpoznávání komunikuje. Pro nás jsou důležité hlavně kanály 11, 6 a 7.

Na kanálu 11 jsou přijímány požadavky k detekci objektů v okolí, které jsou odesílány z plánovacího modulu a mohou být vyslány například při startu hry, aby měl plánovací modul k dispozici aktuální informace o rozložení herních prvků v úhlu záběru. Dále může přijít od plánovacího modulu požadavek na znovuoověření objektů v zorném poli, které je časově méně náročné (u objektu je známá jeho pozice v obrázku), což se děje například v případě, když se plánovací modul rozhodne vyjet pro konkrétní prvek.

1 Specifikace problému

Na kanálu 6 jsou obecně odesílány zprávy s informacemi o umístění objektů na hřišti. Ty přijímá a zpracovává mapový modul. Ten má na starost správu všech objektů na hřišti spolu s historií pohybu robota protihráče. Na tento kanál by mělo vidění odesílat informace o všech detekovaných objektech.

Posledním kanál, se kterým modul rozpoznávání pracuje, je kanál 7. Zde jsou lokalizačnímu modulu odesílány updaty aktuální polohy, načež lokalizační modul na jejich základě udržuje nejpravděpodobnější aktuální polohu. Aktuální polohu robota a jeho natočení může modul vidění načítat ze sdíleného souboru, který lokalizační modul spravuje (pokud modul rozpoznávání běží na stejném počítači jako lokalizační modul).

2 Analýza problému počítačového vidění robota

Hra probíhá v přesně specifikovaném herním prostředí. Tým má k dispozici samotného herního robota a sloupky pro umístění majáčků v přesně určených okrajových pozicích hřiště. Scénu lze tedy snímat z mobilního robota, či z těchto pevných pozic.

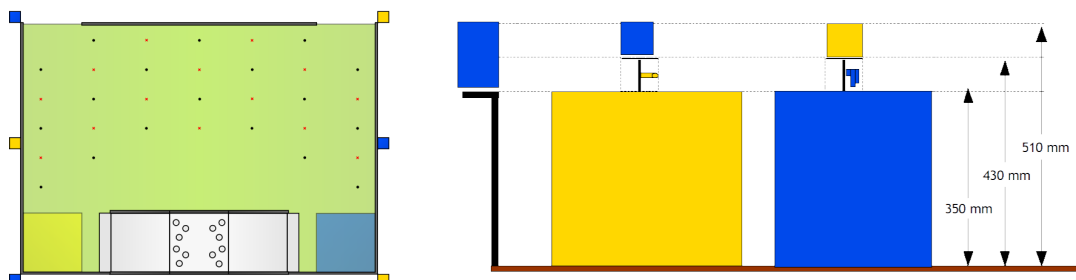
Při snímání obrazu z robota je poloha herních prvků v obraze ovlivněna polohou robota v hřišti. Ten se po většinu hry nachází v konstantní výšce, kdy má pouze tři stupně volnosti. Robot může během hry vyjízďet na vyvýšenou část hřiště pro sběr pomerančů, kdy se mění náklon i vzdálenost od povrchu hřiště.

Majáčky na hřišti jsou umístěny ve výšce 350 mm nad povrchem a v pozicích, které jsou dány startovní variantou. Přesné pozice majáčků na hřišti lze vidět na obrázku 2.1.

2.1 Vizuální vlastnosti snímaného obrazu

2.1.1 Osvětlení

Osvětlení herní plochy není pravidly nijak specifikováno. Hřiště je však během hry běžně osvětleno reflektory z různých směrů. Tyto zdroje světla lze zjednodušit jako blízký bodový zdroj světla (FORSYTH – PONCE, 2003, s. 74). A právě proto, že se jedná o bodové zdroje, tak se na hřišti objevují oblasti zastíněné herními prvky. Další potencionální problémy se mohou objevit díky vzájemné reflexi, jenž vzniká tak, že světlo odražené od jednoho povrchu osvětlí jiný povrch. V našem případě bude tato reflexe například způsobovat barevný nádech herních prvků do barvy povrchu hřiště. Výslednou barvu herních prvků v obrázku také ovlivňuje barevná teplota světelného zdroje.



Obrázek 2.1 Polohy majáčků na hřišti a na robotech (Eurobot, 2009, s. 21).

2.1.2 Hřiště

Optické vlastnosti hřiště jsou určeny materiálem a nátěrem povrchu. Dle herních specifikací je hřiště vyrobené ze dřeva a je natřeno akrylátovou barvou definovanou vzorníkem barev RAL ¹. Detailnější informace o odstínech barev lze nalézt v oficiálních pravidlech soutěže (Eurobot, 2009, s. 43). Díky těmto parametrům se jedná převážně o Lambertovský povrch bez výraznějších spekularit.

2.1.3 Herní prvky

Z hlediska optických vlastností se na hřišti vyskytují dva základní typy herních prvků.

Pomeranče a rajčata: Částečně odrazivé objekty s velmi výraznou barevnou složkou.

Lesklý povrch způsobuje, že část odraženého světelného toku závisí na úhlu dopadu příchozího světelného toku. To v obraze objektu způsobuje odlesky zdroje světla, které jsou charakteristické vysokou intenzitou jasu (FORSYTH – PONCE, 2003, s. 66). Objekty jsou kulovitěho tvaru.

Pravé a falešné kukuřice: Herní prvky vyrobené soustružením z polypropylenu, což způsobuje že jejich povrch je převážně Lambertovský. Odražený světelný tok tedy není tolik závislý na úhlu dopadajícího světla. Odlišit tyto prvky mezi sebou je možné jen na základě rozdílné intenzity, protože jejich barevná složka není definovaná jako u rajčat nebo pomerančů a více se u nich projeví vzájemná reflexe. U falešných kukuřic je to navíc komplikované tím, že mají podobné optické vlastnosti jako okraje hřiště. Objekty jsou válcovitěho tvaru.

2.1.4 Protivník

Podle pravidel lze protivníka definovat pouze jeho maximálními rozměry. Protivník by také neměl na sobě mít barvy, které by byly zaměnitelné s barvami herních prvků. Na protivníka je možné umístit libovolné vlastní zařízení o rozměrech 80 × 80 × 80 mm do přesně specifikované výšky nad hřištěm.

2.2 Polohové vlastnosti herních prvků

Herní objekty, mimo pomerančů, se na hřišti vyskytují převážně na jeho povrchu. U rajčat a stojících kukuřic určujeme pouze polohu na povrchu hřiště. U ležících kukuřic je navíc důležité určit i jejich natočení v ose kolmé k povrchu hřiště. Pomeranče se při startu hry nacházejí v přesně definovaných pozicích nad hřištěm. Všechny herní prvky stejného typu jsou vzájemně zaměnitelné. Dále se prvky mohou nacházet v těle robota, kde již nejsou důležité pro hru našeho robota.

Při snímání objektů kamerou se mohou objekty navzájem zakrývat, či mohou být zakryty herními roboty.

¹<http://www.ral-farben.de/>

3 Detekce herních objektů z obrázku

Všechny herní objekty ze vstupního obrázku detekujeme podobným způsobem:

- V obrázku vyhledáme body, které svojí barvou odpovídají hledanému hernímu prvku.
- Ze souvislých oblastí těchto bodů vybereme jeden bod, který budeme dále nazývat „počátečním bodem“.
- Ze počátečního bodu se snažíme najít okraje objektu takovým způsobem, který nezávisí na jeho pozici v rámci obrazu objektu ve vstupních datech.
- Snažíme se určit nejpravděpodobnější polohu geometrického útvaru, který daný herní prvek znázorňuje (u rajčat se jedná o kružnici, u kukuřic o obdélník). K tomu využíváme RANSAC algoritmus, který blíže popíšeme v sekci 3.2.

3.1 Dichromatický model odrazivosti

Dichromatický model odrazivosti popisuje světlo odražené od bodu objektu jako lineární kombinaci dvou složek:

Surface reflection: Odražené světlo s barevnými vlastnostmi jako světlo zdroje, jež vzniká přímým odrazem od nehomogenního povrchu. Povrch se v tom případě chová téměř jako zrcadlo.

Body reflection: Odražené světlo se změněnými barevnými vlastnostmi, které se vytvoří pohlcením určité části světelného spektra. Pohlcená část spektra je dána optickými vlastnostmi povrchu.

Odražené světlo tedy závisí na vlastnostech světelného zdroje a povrchu objektu.

$$E(\lambda) = c_b \cdot E_b(\lambda) + c_s \cdot E_s(\lambda) \quad (3.1)$$

Kde E je výsledné spektrální osvětlení, E_b spektrální osvětlení charakteristické pro materiál objektu a E_s osvětlení odpovídající vlastnostem světelného zdroje.

Předpokládáme-li, že je obraz snímán kamerou s lineární odezvou, lze změřenou barevnou hodnotu popsat jako:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = c_b \cdot \begin{pmatrix} R_b \\ G_b \\ B_b \end{pmatrix} + c_s \cdot \begin{pmatrix} R_s \\ G_s \\ B_s \end{pmatrix} \quad (3.2)$$

Dle rovnice 3.2 každý povrch generuje body ležící na přímce v RGB barevném prostoru. Přesnou polohu a směrnici přímky ovlivňují vlastnosti zdroje světla. Poloha bodu má tedy jen dva stupně volnosti a leží tedy na rovině (Kravtchenko – Little, 1999).

Problém jsme si zjednodušili zavedením jednoduchého dichromatického barevného systému, který je invariantní na intenzitu osvětlení. Místo hodnot $[R, G, B]$ zavádíme $[r, g, I]$, které jsou dány vztahem:

$$r = \frac{R}{R + G + B} \quad g = \frac{G}{R + G + B} \quad I = R + G + B \quad (3.3)$$

Barevnost světla odraženého od povrchu lze nyní popsat jako:

$$\begin{pmatrix} r \\ g \end{pmatrix} = c'_b \cdot \begin{pmatrix} r_b \\ g_b \end{pmatrix} + c'_s \cdot \begin{pmatrix} r_s \\ g_s \end{pmatrix} \quad (3.4)$$

Z rovnice 3.4 lze vidět, že v tomto barevném systému nezávisí barevnost odraženého světla na intenzitě odraženého světla. To nám umožňuje specifikovat množinu hodnot odpovídajících danému typu povrchu jen ve dvourozměrném prostoru, který se snadno implementuje například pomocí look-up tabulky.

3.2 RANSAC

RANSAC (RANdom SAMple Consensus) je iterativní metoda pro robustní určení neznámých parametrů ze sady dat. Vstupem je množina dat \mathcal{U} , ze kterých neznámá část odpovídá hledanému modelu (tzv. inliers) a zbytek dat z této množiny je chybný (tzv. outliers). Cílem algoritmu je nalézt parametry modelu Θ^* z prostoru parametrů Θ , pro které funkce $J_s(\Theta, \mathcal{U}, \Delta)$ nabývá maximálních hodnot. Výsledek této funkce je ve většině případů definován jako počet datových bodů odpovídajících dané volbě parametrů. Vstupním parametrem algoritmu je Δ , která představuje maximální chybu, kdy je datový bod ještě považován za součást modelu (Chum, 2005, s. 9).

Postupujeme takto:

- Generujeme hypotézu parametrů modelu. Z množiny \mathcal{U} náhodně vybereme podmnožinu dat \mathcal{S}_k , pro kterou platí, že počet datových bodů je minimálně takový, aby bylo možné jednoznačně určit parametry modelu.
- Pro dané parametry se zjistí výsledek hodnotící funkce.
- Pokud je dosaženo nového maximálního ohodnocení, uložíme tyto parametry jako aktuálně nejlepší.

Platí, že výsledek této metody je silně závislý na poměru počtu inlierů vůči outlierům ve vstupních datech a počtu iterací algoritmu (Chum, 2005, s. 11).

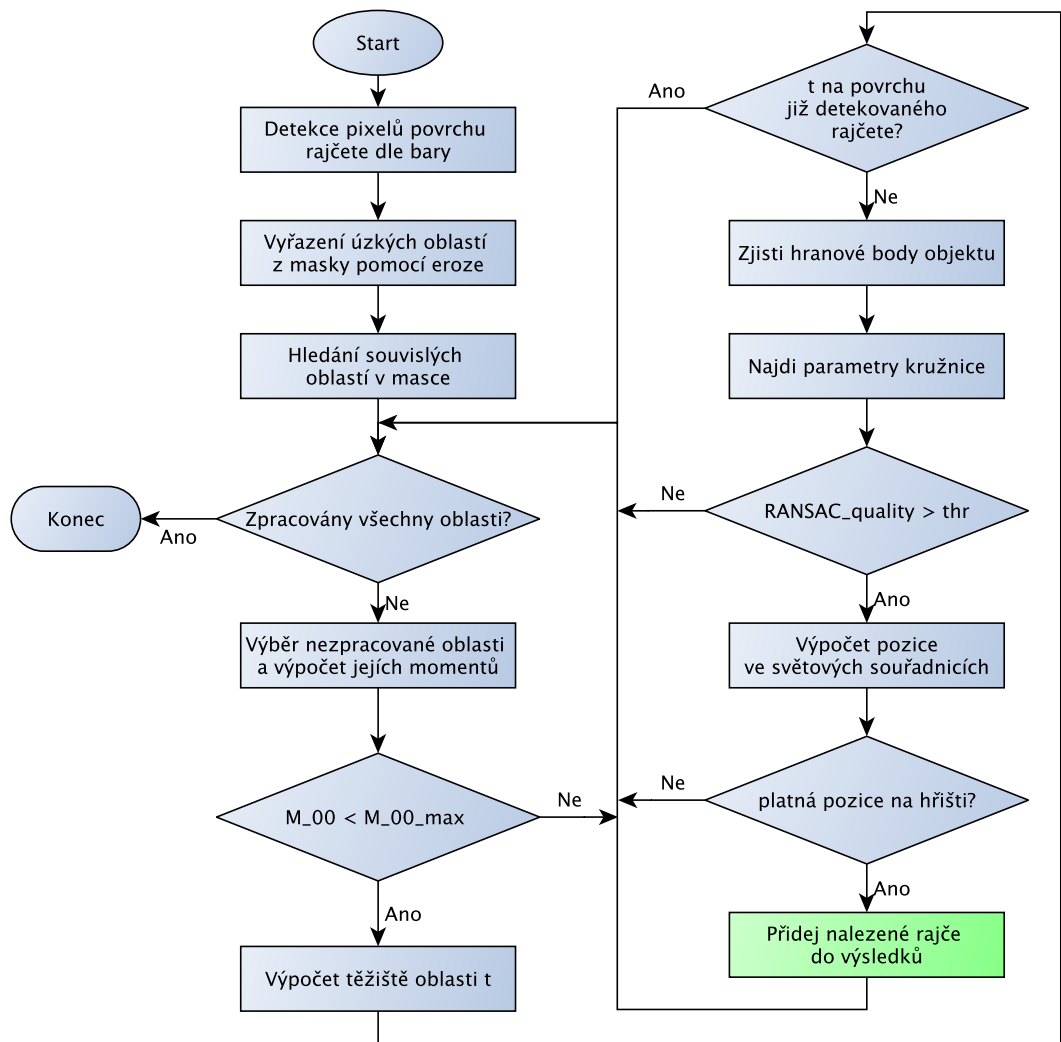
3.3 Zpracování vstupního obrazu

Ze vstupního obrázku jsou nejprve odstraněny oblasti neodpovídající povrchu herního hřiště. To provádíme tak, že na základě nakalibrovaných transformací ze souřadnic hřiště do roviny vršků kukuřic ořízneme z obrázku oblasti za okraji hrací plochy.

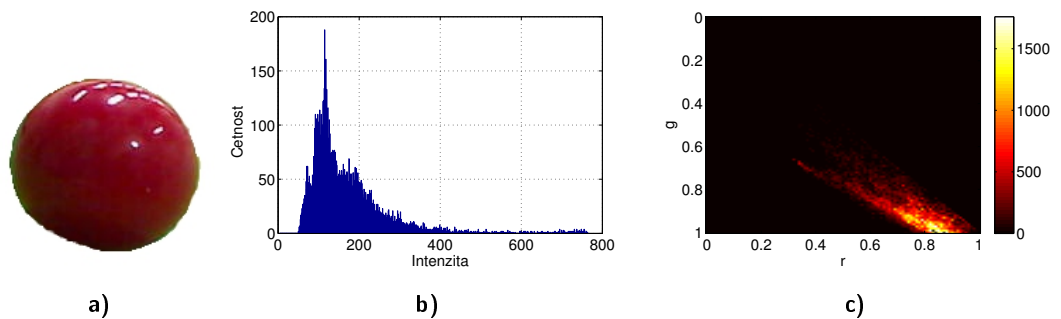
3.4 Detekce rajčat

Jak již bylo zmíněno v kapitole 1, herní prvky znázorňující ve hře rajčata, mají oproti svému okolí velmi výraznou barvu, proto je při hledání startovního bodu využito již popsaného dichromatického modelu.

3 Detekce herních objektů z obrázku



Obrázek 3.1 Postup detekce rajčat.



Obrázek 3.2 a) Obrázek rajčete b) Histogram intenzit c) 2D histogram chromaticit. Histogramy jsou vytvářeny z obrázku bez odlesků.



Obrázek 3.3 Aplikace eroze na oblast detekovaného rajčete.

3.4.1 Detekce oblastí

Pro určení pixelů, které s velkou pravděpodobností přísluší rajčeti, je využito dvou-rozměrné look-up tabulky. Tabulka je vytvořena na základě předem uložených obrázků rajčat, z nichž je vypočítán histogram intenzit a dvourozměrný histogram barevných složek s poměrně nízkým rozlišením (100 na 100 bodů), což vede k nižší datové náročnosti a i menším požadavkům na množství vstupních dat. Hodnoty v histogramu jsou oprahovány na základě uživatelem zadané hodnoty, čímž získáme 2D pole binárních hodnot, kde pravdivá hodnota znamená, že barevná hodnota odpovídá povrchu rajčete.

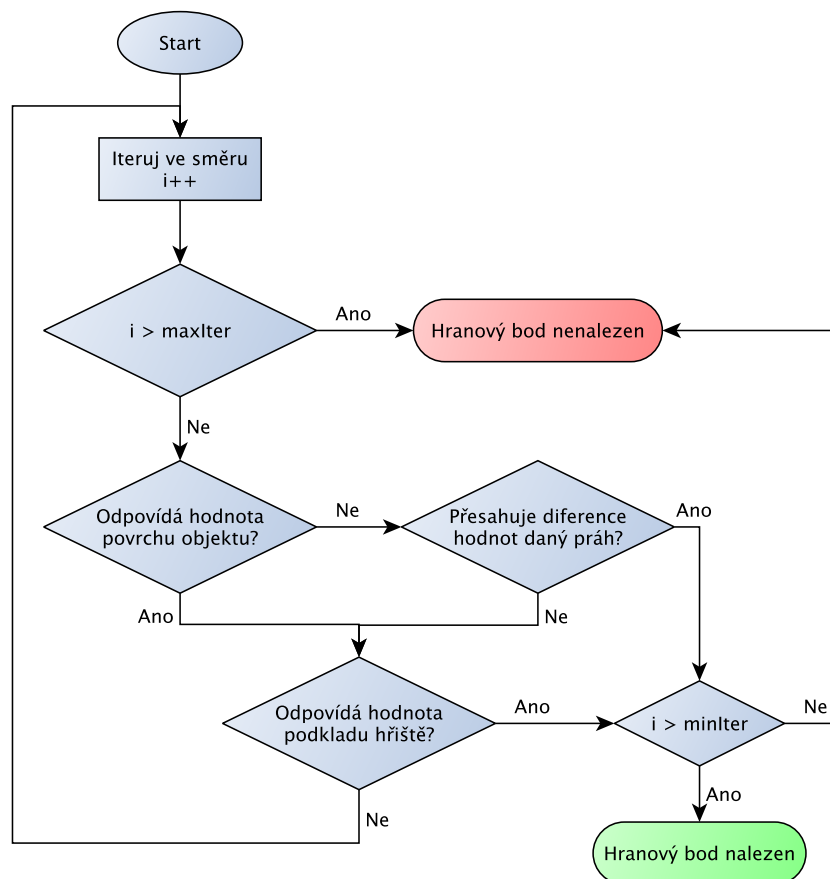
Tato tabulka se vytváří ve speciálním režimu programu a je následně uložena do souboru. Do stejného souboru se také ukládá uživatelem definovaný interval povolených intenzit jasu. Díky tomu je možné tabulky velmi snadno spravovat. Při startu aplikace jsou automaticky všechny tabulky načteny do vlastních objektů, aby mohli být využity v detektorech.

Při samotné detekci je vstupní obrázek nejprve zmenšen na polovinu, abychom snížili dobu zpracování. Pro vstupní data se vytvoří binární obrázek (maska) stejných rozměrů inicializovaný na nulové hodnoty. U každého pixelu obrázku se určí jeho chromaticita spolu s intenzitou a na základě look-up tabulky a intervalu povolených intenzit rozhodneme, zda-li daný pixel odpovídá povrchu rajčete. Pokud ano, patřičný pixel v masce se nastaví na pravdivou hodnotu. Tím vzniknou v binárním obrázku oblasti pixelů, které barvou odpovídají povrchu rajčat.

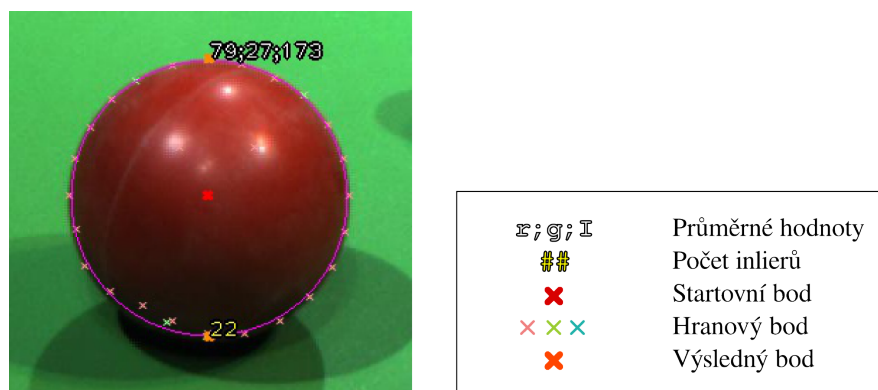
3.4.2 Test tvaru souvislé oblasti

Vstupní data jsou ovlivněna šumem a nedokonalostmi použité kamery. Nejvíce se to projevuje u pixelů s nízkou hodnotou intenzity jasu, kde tyto chyby mají velký vliv na chromaticitu. Proto vzniká určité množství false-positiv souvislých oblastí. Množinu nalezených souvislých oblastí chceme na základě barevných hodnot klasifikovat do dvou tříd - objektů rajčat a objektů neodpovídajících rajčatům. Rajčata definujeme jako oblasti, které při erozi (SHAPIRO – STOCKHAM, 2001, s. 79) strukturálním elementem tvaru kruhu mají nenulovou velikost. Tím určujeme jejich minimální tloušťku. Tato operace zásadně snižuje velikost množiny nalezených souvislých komponent.

Tyto souvislé oblasti detekujeme z masky pomocí knihovní funkce OpenCV použité tak, aby nedetekovala díry. Pro každou oblast je vypočítán objem, který je prahovaný maximální hodnotou, protože oblasti přesahující objemem danou mez, jsou dle pozorování často způsobené špatnou klasifikací barevných hodnot. Dále jsou vypočítány první momenty oblasti pro výpočet těžiště, jež je poté použito jako počáteční bod pro další zpracování.



Obrázek 3.4 Postup ověřování hodnot pixelů při detekci hranového bodu.



Obrázek 3.5 Znárodnění vytrasovaných bodů rajčete spolu s výsledkem RANSACu.

U tohoto bodu se zkontroluje jemu odpovídající hodnota v masce. V té jsou označeny oblasti, které odpovídají již detekovaným objektům. Takto se zabraňuje vícenásobné detekci stejného objektu.

Zákryty rajčat jsou řešeny tak, že z masky jsou odstraňovány oblasti odpovídající nalezeným objektům. Představuje-li tedy oblast více prvků zároveň, jsou z ní postupně odstraňovány části nalezených rajčat. Detekce končí tehdy, pokud v masce nezůstaly žádné nezkontrolované oblasti.

3.4.3 Zjištění hranových bodů

Z těžiště oblasti, určeného v předchozím bodě, vycházíme při hledání hranových bodů rajčete. Každý hranový bod hledáme iterací po přímce pod daným úhlem. Tyto úhly jsou dány tak, aby rovnoměrně pokrývaly interval $0 - 360^\circ$. Výhodou toho způsobu detekce je nezávislost na počátečním bodě uvnitř objektu.

Okraje objektu hledáme postupnou iterací po přímce o dané směrnici, kdy procházíme jednotlivé pixely a u každého ověřujeme, zda-li stále odpovídá povrchu rajčete. Pokud tomu tak není, iterace se zastaví a aktuální bod je hledaným hranovým bodem objektu. Postup je znázorněn na diagramu 3.4.

Zda-li bod odpovídá povrchu rajčete určujeme pomocí look-up tabulky sestavené pro rajčata. To, jestli bod odpovídá podkladu hřiště zjišťujeme pomocí look-up tabulky, která je vytvořena z obrazu hřiště s ostatními herními prvky.

Protože se ale na povrchu objektu mohou vyskytnout výraznější nespojitosti (například odlesky), které iteraci předčasně zastaví, postup v daném směru několikrát opakujeme. Tím můžeme získat pro každý směr množinu bodů. Přesáhneme-li iterací okraje obrázku, bod není do výsledku přidán. Pokud počet iterací nepřesáhne stanovené minimum nalezený bod není přidán do výsledku.

Výsledkem tohoto kroku jsou množiny bodů v daných směrech. Výsledek hledání hranových bodů je znázorněn na obrázku 3.5.

3.4.4 Hledání parametrů kružnice

Naším úkolem je nalézt optimální parametry kružnice, které odpovídá největší množství nalezených hranových bodů z předchozího detekce. K tomu využíváme metodu RANSAC. Vstupními daty je množina:

$$\mathcal{U} = \{ u \mid u = (x, y)^T \mid x \in \mathbb{R}^+ \mid y \in \mathbb{R}^+ \} \quad (3.5)$$

3 Detekce herních objektů z obrázku

Hledáme parametry $\Theta = (x_0, y_0, r)$ rovnice:

$$r^2 = (x - x_0)^2 + (y - y_0)^2 \quad (3.6)$$

Pro vzdálenost bodu $u = (x, y)^T$ od kružnice definované parametry Θ platí:

$$d(p, \Theta) = \sqrt{(x - x_0)^2 + (y - y_0)^2} - r \quad (3.7)$$

A konečně - hodnotící funkci nalezených parametrů jsme zadefinovali jako:

$$J(\Theta, \mathcal{U}, \Delta) = |\{ p \in \mathcal{U} \mid d(p, \Theta) < \Delta \}| \quad (3.8)$$

Parametry kružnice jsou počítány ze tří náhodně volených bodů, které nesmí být kolineární. Ze vstupní datové struktury, pokud její velikost dosáhne požadovaného minima, jsou nejprve náhodně vybrány tři úhly. V každém úhlu se mohou vyskytovat až tři body, z nichž jeden je opět náhodně vybrán. Takto získáme tři body, z kterých vypočítáme parametry Θ .

Počet iterací je pevně daný. Pokud je počet nalezených inlierů při nejlepších parametrech menší, než uživatelem stanovené minimum, nebere se detekovaný objekt v potaz. U výsledku je také kontrolován maximální povolený průměr nalezeného objektu.

Pokud výsledek projde všemi předchozími testy, výstupem detekce jsou parametry kružnice. Z nich jsou určeny význačné body rajčete:

$$u_{top} = (x_0 + r, y_0)^T \quad u_{bottom} = (x_0 - r, y_0)^T \quad (3.9)$$

Tímto sice zanedbáváme skutečný tvar obrazu rajčete v obrázku, protože se ale jeho skutečná poloha počítá jako průměrná hodnota těchto souřadnic přetransformovaných do souřadnic hřiště, vzniklá chyba se tím sníží.

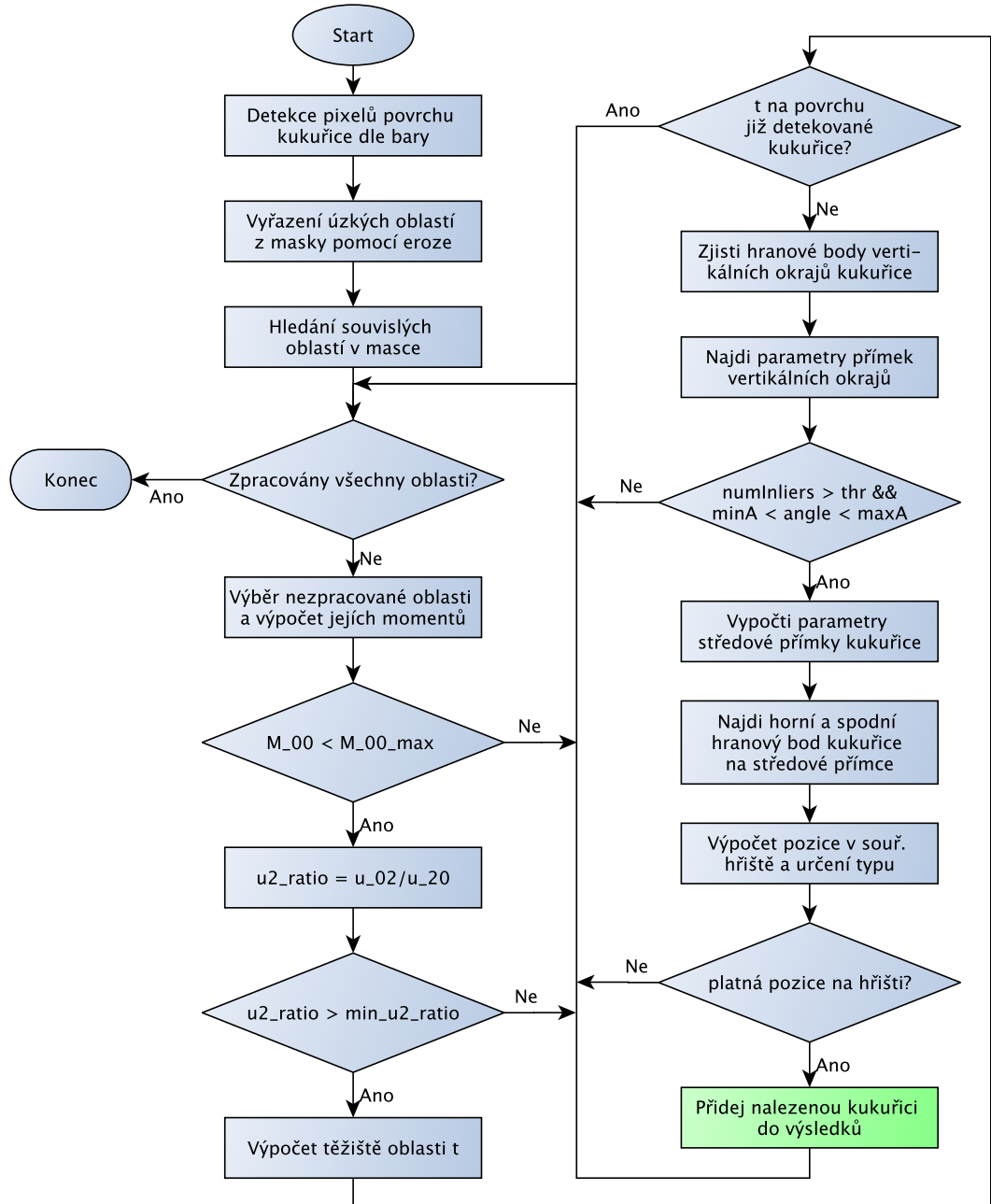
Body uvnitř nalezené kružnice se vyznačí v masce. Tyto body již nemohou být vybrány jako počáteční body pro hledání hranových bodů. Takto zabraňujeme duplicitní detekci stejného objektu (vznikající například při nalezení více oblastí na povrchu jednoho objektu).

3.5 Detekce kukuřic

Kukuřice rozpoznáváme podobným způsobem jako rajčata. Nejprve nalezneme oblasti odpovídající povrchu kukuřic, které použijeme pro určení počátečních bodů, jenž jsou poté využity při hledání hranových bodů kukuřic.

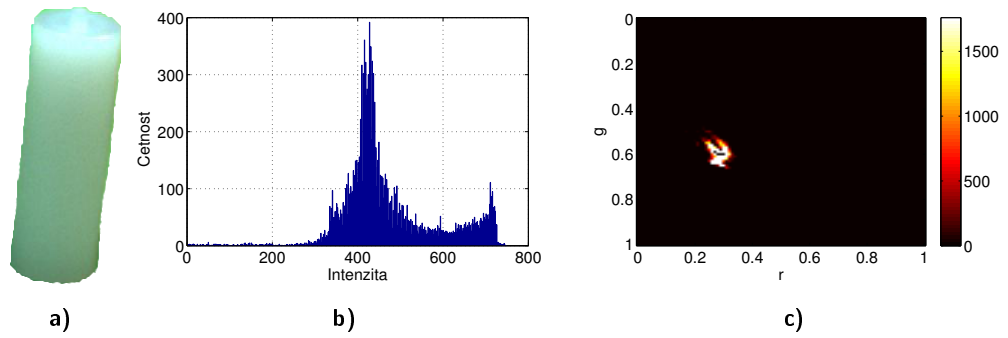
Protože jsou kukuřice vyráběny z materiálu v odstínu šedi, jejich barevná složka je ideálně $r = 1/3$ a $g = 1/3$. Skutečnou hodnotu chromaticit ale velmi ovlivňuje barva osvětlení a světla odraženého od povrchu hřiště. Proto přiřazení pixelů k povrchu kukuřice podle chromaticit není příliš robustní a více využíváme prahování intenzit. Navíc mají falešné kukuřice optické vlastnosti velmi podobné okrajům hřiště. Ležící kukuřice nedetekujeme.

3 Detekce herních objektů z obrázku

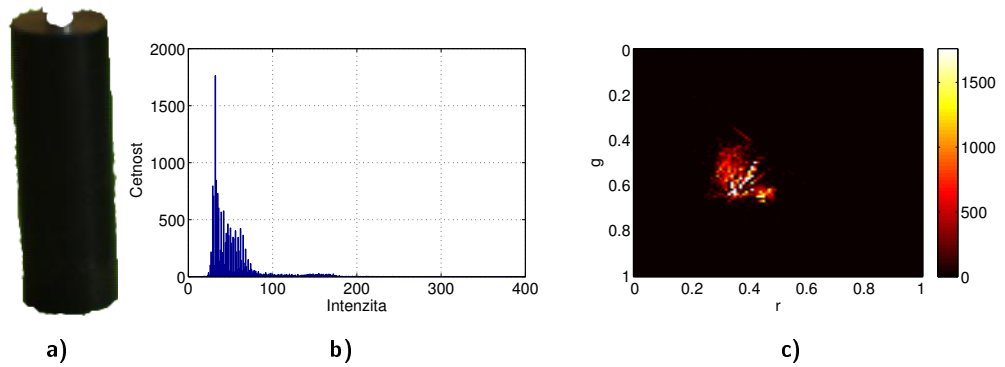


Obrázek 3.6 Postup detekce kukuřic.

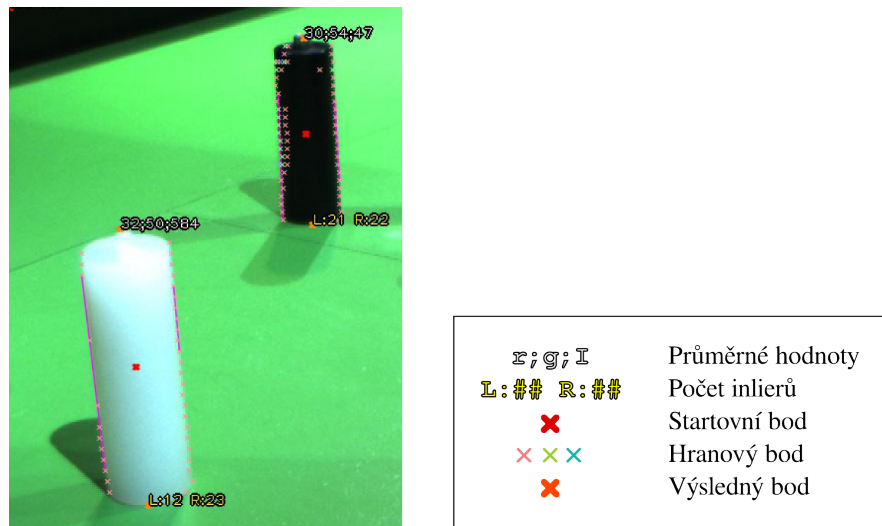
3 Detekce herních objektů z obrázku



Obrázek 3.7 Vlastnosti obrazu pravé kukuřice a) Původní obraz b) Histogram intenzit jasu c) 2D Histogram barevných složek



Obrázek 3.8 Vlastnosti obrazu falešné kukuřice a) Původní obraz b) Histogram intenzit jasu c) 2D Histogram barevných složek



Obrázek 3.9 Znázornění nalezených hranových bodů kukuřic spolu s výsledky RANSACu.

3.5.1 Detekce oblastí

Postup je velmi podobný jako u detektoru rajčat. Pro zjištění příslušnosti pixelu obrázku k bodu povrchu kukuřice se využívá hlavně intenzity jasu, které nejvíce odlišují povrch hřiště od povrchu kukuřice. Pro každý typ kukuřice je na základě předem uložených obrázků kukuřic vytvořena vlastní look-up tabulka. Modely jsou znázorněny na obrázku 3.7 a 3.8. V tabulce jsou tak zaneseny i ty chromaticity odpovídající snímanému povrchu kukuřice spolu s šumem. Použitý postup ale vede k detekci řady false-positive oblastí vznikající například odlesky od povrchu hřiště.

Pixely jsou podle hodnoty rozděleny do tří tříd - pixelů falešných kukuřic, pixelů pravých kukuřic a pixelů neodpovídajících kukuřicím. Spojité oblasti pixelů falešných kukuřic jsou vyznačeny ve vlastním binárním obrázku, stejně jako oblasti pravých kukuřic. Máme tedy dvě binární masky poloviční velikosti původního obrázku.

Každou oblast testujeme na tvar pomocí morfologických operací. Zde je voleno takové jádro (obdélník o rozměrech 5×10), aby oblast splňovala podmínku minimální tloušťky ve svislém a vodorovném směru. Oblasti, které toto kritérium nesplňují, jsou z masky odstraněny (jedná se například o okraje hřiště, které jsou detekovány jako oblasti odpovídající falešným kukuřicím).

U každé oblasti, která splnila předchozí podmínku, je vypočítán její obsah. Pokud se nenachází v daném intervalu oblast se dále nezpracovává (vysoký objem má například boční deska kopce s pomeranči). Dále je kontrolován poměr mezi druhými centrálními momenty ve směrech os x a y . Pokud má oblast výrazně vyšší druhý centrální moment ve směru osy x oproti momentu ve směru osy y , považujeme tuto oblast za neplatnou. Pokud je vše v pořádku, těžiště dané oblasti se určí jako počáteční bod hledání hranového bodu kukuřice.

3.5.2 Určení hranových bodů kukuřice

Pro hledání hranových bodů objektu na přímce je použito podobného způsobu jako při detekci rajčat (viz diagram 3.4).

Nejprve se z počátečního bodu iteruje po vertikální přímce v obou směrech, aby se zjistila přibližná výška kukuřice. Získáme tak první odhad horního bodu $u_{te} = (x_{te}, y_{te})$ a spodního bodu $u_{be} = (x_{be}, y_{be})$ kukuřice. Úsečka daná těmito body je rozdělena na několik částí a z každé se po horizontálních přímkách hledají pravé a levé boční hranové body kukuřice. Získáme tak množinu bodů \mathcal{U}_l a \mathcal{U}_r . Výsledek je znázorněn na obrázku 3.9.

Pro oba okraje kukuřice pomocí RANSACu nalezneme parametry rovnice přímky, které lze velmi snadno určit pomocí dvojice náhodně vybraných bodů a jež jsou dány jako:

$$\Theta = (u, \vec{v}) \quad u = (x_u, y_u)^T \quad \vec{v} = (v_1, v_2)^T \quad (3.10)$$

Pro vzdálenost bodu od přímky platí:

$$d(p, \Theta) = \frac{|\vec{v} \times \vec{v}_{pu}|}{|\vec{v}|} \quad \vec{v}_{pu} = (x_p - x_u, y_p - y_u)^T \quad (3.11)$$

Hodnotící funkce je stejně jako u kružnice dána:

$$J(\Theta, \mathcal{U}, \Delta) = |\{ p \in \mathcal{U} \mid d(p, \Theta) < \Delta \}| \quad (3.12)$$

Z množiny \mathcal{U}_l získáme parametry přímky $\Theta_l = (u_l, \vec{v}_l)$ a z množiny bodů \mathcal{U}_r parametry $\Theta_r = (u_r, \vec{v}_r)$, jimž odpovídá největší množství okrajových bodů. Pokud RANSAC

neuspěje, použijeme jako význačné body objektu původně nalezené body u_{te} a u_{be} a dále již nepokračujeme.

Cílem je získat rovnici středové přímky kukuřice, na níž se nachází horní a spodní bod objektu. Úhel této přímky je určen jako střední hodnota z úhlů bočních přímek kukuřice (tyto úhly jsou zároveň kontrolovány, zda-li jsou v daném intervalu). Pro středovou přímku je zvolen bod, pro nějž platí:

$$u_c = \left(\frac{x_{ul} + x_{ur}}{2}, y_c \right) \quad y_c \in (x_{te}, x_{be}) \quad (3.13)$$

Vybrán je bod, jehož y souřadnice odpovídá souřadnici počátečního bodu.

Z tohoto bodu se opět pokoušíme iterací po středové přímce nalézt spodní a horní bod kukuřice u_t a u_b .

Tento způsob hledání okrajů kukuřice není příliš spolehlivý na určování výšky kukuřice, jelikož výsledné horní a spodní body se pokoušíme hledat pouze jednou. Pokud se při tomto trasování zastavíme v oblasti, která ještě odpovídá kukuřici, je výsledná chyba detekce velmi výrazná.

3.5.3 Určení barvy herního prvku

Během hledání okrajových bodů iterací po přímce se sčítají hodnoty intenzit pixelů. Na konci detekce se tyto hodnoty zprůměrují počtem prošlých bodů a z průměrné hodnoty intenzit se určí typ objektu.

3.6 Ověřování přítomnosti herního prvku na hřišti

Pokud je pro plánování důležité pouze ověřit přítomnost aktivních prvků na hřišti, může si plánování od modulu rozpoznávání vyžádat pouze ověření jejich pozice. Tento proces je (oproti plné detekci ze vstupního obrazu) výrazně rychlejší. Využíváme jej také pro detekci typů kukuřic ze vstupních dat od statických kamer, protože pozice kukuřic v obrázku je předem známá.

Pro ověření se využívají stejné postupy jako při detekci daných objektů s tím rozdílem, že souřadnice počátečního bodu odpovídajícího povrchu objektu v obrázku je předem známá (určena zpětnou transformací středového bodu objektu ze souřadnic hřiště do souřadnic obrázku). Vždy se ověřují jen objekty, které jsou v úhlu záběru kamery.

U ověřovaného objektu rajčete je jeho stará poloha vždy aktualizována nově zjištěnými údaji. U kukuřic zároveň kontrolujeme, zda-li byl zachován typ objektu. Pokud není známá varianta, typ ověřené kukuřice oznámíme detektoru variant.

U statických kamer navíc uživatel při kalibraci kamer zadá i body středů kukuřic. Tato možnost byla přidána proto, že u vzdálených kukuřic byla chyba způsobená špatně zkalibrovanými transformacemi již tak velká, že startovní bod v obrázku vycházel mimo tělo kukuřice. Pokud jsou u aktuální kamery nastaveny pevné body středů kukuřic, jsou využity při jejich verifikaci.

3.6.1 Ověřování přítomnosti pomerančů

Tato funkce byla do programu přidána proto, aby měl robot informaci o tom, zda-li jsou na hřišti ještě „nesčesané“ pomeranče. Postup jejich ověřování je stejný jako při ověřování přítomnosti rajčat s tím rozdílem, že jejich startovní body jsou zadány uživatelem při kalibraci kamery.

4 Detekce herní varianty

Na začátku hry je vždy rozhodčími vylosována herní varianta. Herní varianta je určena jako kombinace dvou náhodných losování. V prvním losování se vybírá varianta rozložení falešných kukuřic při okrajích hřiště (devět možností) a ve druhém rozložení ve středu hřiště (čtyři možnosti) (Eurobot, 2009, s. 29).

Varianta je detekována pomocí jednoduchého způsobu zjišťování té nejpravděpodobnější z možných. Na začátku je sestrojena databáze variant (zvlášť jsou počítány středové a okrajové varianty). Ke každé variantě je přiřazena hodnota představující kolik rozpoznaných kukuřic odpovídalo dané variantě (dále „ohodnocení varianty“). Pro každou pozici na hřišti jsou vytvořeny dva seznamy. První z nich představuje seznam variant, které obsahují falešnou kukuřici na dané pozici. Druhý seznam obsahuje výčet všech variant, které na dané pozici nejsou.

Každé pozici na hřišti, kde může být umístěna kukuřice, je přiřazeno unikátní číslo (index), se kterým se pracuje v detektoru variant. Vstupem do detektoru je index pozice kukuřice a její typ. Jedná-li se o falešnou kukuřici, tak se všem variantám obsahujících falešnou kukuřici na dané pozici inkrementuje ohodnocení o jedničku. Poté se naopak variantám, které danou kukuřici neobsahují, ohodnocení o jedničku sníží. Při detekci pravé kukuřice se pracuje se stejnými seznamy s tím rozdílem, že variantám, kterým by falešná kukuřice zvýšila ohodnocení, se ohodnocení sníží a naopak.

Zda-li je varianta známá se určuje tak, že se všechny varianty setřídí sestupně dle ohodnocení a varianta je zdetekována, pokud existuje jen jedna, která má lepší (ne stejné) ohodnocení než ostatní. Toto se provádí separátně pro obě losovací karty.

Výhodou tohoto způsobu určování herní varianty je relativní odolnost proti chybným vstupním datům. Každý herní prvek má zde stejnou váhu a nedojde-li k detekci varianty hned při zpracování první sady dat, další vstupní data jen posílí již nejpravděpodobnější variantu a výsledek se tedy stává o to víc odolný proti dalším chybným detekcím.

5 Repräsentace prostředí

5.1 Souřadnicové systémy

Souřadnice bodu v obrázku: V OpenCV je obrázek reprezentován jako matice pixelů. Souřadnice v obrázku je rovna „pořadí“ pixelu od levého horního rohu obrázku:

$$\mathbf{u}_i = (c, r)^T$$

Kde c značí sloupec a r řádek pole pixelů.

Souřadnice hřiště (světové souřadnice): Tyto souřadnice vyjadřují polohu bodu na hřišti vůči žlutému startovnímu rohu hřiště. Základní jednotkou jsou $[mm]$. Více lze vidět na obrázku 5.1. Světové souřadnice budeme dále značit:

$$\mathbf{u}_w = (x_w, y_w)^T$$

Relativní souřadnice: Tyto pomocné souřadnice slouží k vyjádření polohy bodu vůči aktuální poloze robota (kamery). Počátkem souřadnic je obecný střed robota, jehož poloha na hřišti je nám známá (popsána polohou v souřadnicích hřiště (t_x, t_y) spolu s úhlem natočení α).

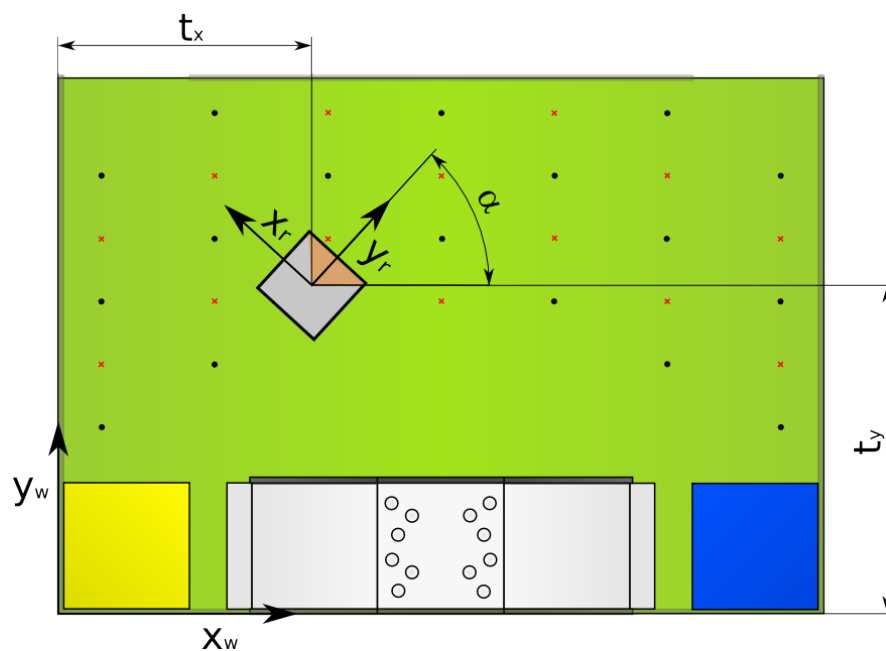
$$\mathbf{u}_r = (x_r, y_r)^T$$

5.2 Určení polohy objektu z obrázku

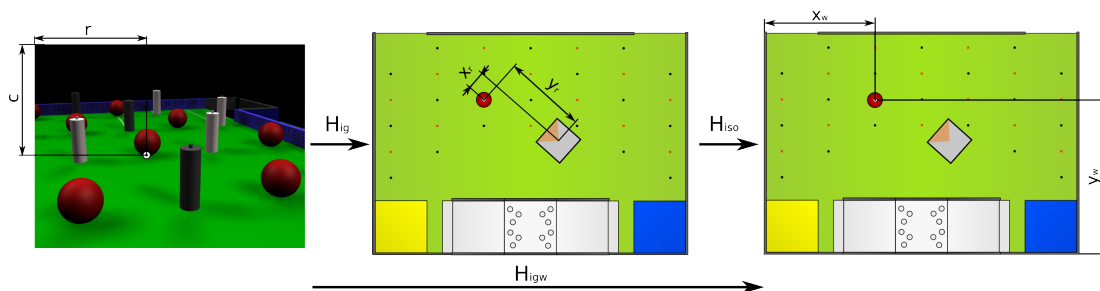
Nedílnou součástí rozpoznání objektu je i určení jeho polohy v herním prostředí. Výstupem algoritmu pro identifikaci objektu je poloha jeho význačných bodů v souřadnicích obrázku. Naším úkolem je nalézt transformaci, která dokáže získaným bodům přiřadit polohu v hřišti.

Obecnou polohu bodu budeme reprezentovat homogenní souřadnicí (Hartley – Zisserman, 2003, s. 27) popsanou vektorem \mathbf{x} . Pro homogenní souřadnici platí, že je dimenze $d + 1$, kde d značí dimenzi prostoru nehomogenní souřadnice. V našem případě se jedná dvou-dimenzionální prostor, tudíž pro převod z nehomogenních souřadnic do homogenních a nazpět platí:

$$\begin{aligned} \mathbf{u}_n = (x_n, y_n)^T &\rightarrow \mathbf{u}_h = (x_n, y_n, 1)^T \\ \mathbf{u}_h = (x_h, y_h, z_h)^T &\rightarrow \mathbf{u}_n = \left(\frac{x_h}{z_h}, \frac{y_h}{z_h} \right)^T \end{aligned}$$



Obrázek 5.1 Zvolené souřadnicové systémy.



Obrázek 5.2 Znázornění transformací souřadnic.

5.2.1 Transformace souřadnic

Homografie (projektivní transformace) je dána jako zobrazení $\mathcal{P}^d \mapsto \mathcal{P}^d$, které je v prostoru homogenních souřadnic \mathcal{R}^{d+1} lineární (SONKA et al., 2008, s. 555). Homografie nám tedy obecně slouží pro transformaci původní souřadnice \mathbf{u} na vektor \mathbf{u}' v jiném souřadnicovém systému. Platí:

$$\mathbf{u}' \simeq H\mathbf{u} \quad (5.1)$$

Kde H je v případě, že uvažujeme dvou-rozměrné souřadnice, matice o rozměrech 3×3 . Cílem je získat takovou matici H , abychom byli schopni pomocí rovnice 5.1 transformovat souřadnice obrázku na souřadnice hracího stolu.

Snímáme-li kamerou objekt v reálné scéně je jeho poloha v obrázku ovlivněna polohou kamery a jejími vlastnostmi. V našem případě předpokládáme, že platí:

1. Poloha kamery má pouze tři stupně volnosti - kamera je umístěna ve fixní výšce na těle robota nebo na majáčkách na okraji hřiště.
2. Vnitřní vlastnosti kamery se v průběhu hry nemění.

Jelikož i objekty na hřišti mají zpravidla jen dva stupně volnosti (objekty „leží“ na stole), je náš problém zjednodušen, protože se jedná o transformaci z dvou-rozměrného obrázku do dvou-rozměrných světových souřadnic. Abychom však mohli bod přiřadit do správné roviny, musíme vědět v jaké výšce nad herní deskou se nachází. Například pro bod reprezentující vršek rajčete použijeme jinou transformaci souřadnic než pro bod spodní části rajčete. Zavádíme tedy transformace mezi obrázkem a rovinou desky stolu (H_{igw} - *image-ground-world*), obrázkem a rovinou vršků rajčat (H_{itw} - *image-tomato-world*) a obrázkem a rovinou vršků kukuřic (H_{icw} - *image-corn-world*). Toto jsou žádané transformace, které používáme pro převod polohy bodu v obrázku do polohy ve světových souřadnicích.

U kamery, u níž se její pozice na hřišti během hry mění („dynamických kamer“), transformace rozdělujeme do dvou pomocných transformací.

První, projektivní transformace, která zajišťuje transformaci bodu z obrázku do relativních souřadnic kamery, je různá pro různé roviny, čímž tedy získáváme trojici transformačních matic H_{ig} , H_{it} , H_{ic} . Výhodou je, že pokud je kamera pevně připevněna v konstantní výšce, tyto transformace se v průběhu hry nemění a lze je dopředu vypočítat.

Druhá transformace, která převádí podle aktuální pozice robota relativní souřadnice na světové, se skládá ze dvou isometrických transformací (zachovávají eukleidovské vzdálenosti) - rotace a translace. Matice této transformace je dána jako (Hartley – Zisserman, 2003, s. 38):

$$H_{iso} = \begin{pmatrix} \cos \alpha & -\sin \alpha & t_x \\ \sin \alpha & \cos \alpha & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

Kde t_x , t_y , α odpovídají poloze robota jak je znázorněno na obrázku 5.1.

Výsledná transformace mezi obrázkem a rovinou desky stolu bude dána jako:

$$H_{igw} = H_{iso} \cdot H_{ig} \quad (5.2)$$

Obdobně pro transformaci do roviny vršků rajčat a roviny vršků kukuřic.

Postup je tedy takový, že z detektoru zjistíme pozici horního a spodního význačného bodu objektu v obrázku. Na základě aktuální polohy a natočení kamery určíme dvě transformační matice odpovídající těmto bodům (například pro desku stolu a vršky rajčat), z nichž vypočítáme světové souřadnice těchto bodů. V ideálním případě by tyto souřadnice měly být stejné, což lze dále použít například pro zpřesnění. Pokud se

však nalezené souřadnice budou příliš lišit, objekt považujeme za chybně detekovaný. Postup je znázorněn na obrázku 5.2.

U statických kamer, tj. kamer umístěných v majáčkách na okraji hřiště, transformaci nerozdělujeme. Pro výpočet polohy objektu na hřišti máme nakalibrované transformační matice, které se v průběhu hry nemění.

5.2.2 Kalibrace kamery

Pro zjištění polohy objektu je kritické správné zkalibrování kamery. V našem případě je kalibrací myšleno získání správných transformačních matic H_{ig} , H_{it} a H_{ig} . Tyto matice lze získat například z bodových korespondencí (dvojice odpovídajících souřadnic bodů z obrázku a hřiště $(\mathbf{u}'_i, \mathbf{u}_i)$). Problém se tedy zjednodušuje na řešení rovnic:

$$\alpha_i \mathbf{u}'_i = H \mathbf{u}_i \quad (5.3)$$

Kde:

- α_i představuje souhrnné měřítko, na kterém obecně nezáleží (při převodu do nehomogenních souřadnic se zkrátí)
- \mathbf{u}'_i odpovídá světovým souřadnicím bodu
- \mathbf{u}_i jsou souřadnice bodu v obrázku

Pro vyřešení rovnic potřebujeme znát čtveřici bodů (SONKA et al., 2008, s. 558).

Tyto rovnice lze řešit například pomocí DLT algoritmu, který je popsán například v Hartley – Zisserman (2003, s. 88) nebo v SONKA et al. (2008, s. 559). Rovnice se převedou do formy řešitelné pomocí SVD (popis například v Wikipedia (2010)). Výhodou tohoto způsobu řešení je, že se minimalizuje tzv. algebraická vzdálenost (Hartley – Zisserman, 2003, s. 93) a že lze toto řešení použít i pro větší počet dvojic.

V naší implementaci využíváme funkce knihovny OpenCV, která pro výpočet transformací používá výše popsáný postup. Dále ještě využívá metodu RANSAC (Hartley – Zisserman, 2003, s. 117) pro výběr nejvhodnějších korespondenčních dvojic. Hledanými parametry modelu je u této úlohy transformace mezi body a jsou počítány ze čtyř náhodně vybraných korespondenčních dvojic. S odhadnutými parametry se zjišťuje, u kolika dvojic je transformace úspěšná (tzn. pokud je vzdálenost mezi bodem a jeho transformovaným protějškem ze dvojice menší než parametr Δ). Pro výpočet SVD tato knihovna funkce využívá knihovny *lapack*.

Dvojice odpovídajících bodů získáváme jednoduchou metodou, kdy uživatel na obrázku vybírá detekované body (např. body odpovídající spodním částem rajčat) a k nim odpovídající polohy rajčat na hřišti. Po vybrání sady čtyř odpovídajících bodů jsou vypočítány matice pro transformaci bodů z obrázku do hřiště.

U „dynamických kamer“ potřebujeme transformaci nezávislou na aktuální poloze. Pro přepočítání na transformaci do relativních souřadnic postupujeme tak, že matici zleva vynásobíme inverzní maticí odpovídající aktuální poloze (složené opět z rotační a translační matice).

Příklad výpočet matice pro rovinu desky stolu:

$$H_{ig} = H_{iso}^{-1} \cdot H_{igw} \quad (5.4)$$

Jedná se v podstatě pouze o úpravu rovnice 5.4. Obdobně postupujeme i pro matice H_{it} a H_{ig} s tím, že uživatel musí vybrat nové odpovídající dvojice bodů. Tyto matice jsou poté uloženy v XML souborech.

U statických kamer ukládáme přímo matice H_{igw} , protože se jejich poloha v průběhu hry nemění.

5.3 Mapa prostředí

Mapa prostředí má za úkol spravovat aktuální pozice herních prvků na hřišti. Realizujeme jí pomocí dvou datových kontejnerů pro rajčata a kukuřice, které se liší přístupem k přidávání nově nalezených objektů. Zatímco rajčata se mohou vyskytovat v jakékoliv pozici na hřišti, stojící kukuřice mohou být umístěny jen v pevných pozicích.

Další funkcí mapy prostředí je informovat ostatní moduly robota o poloze herních prvků. Proto při každém přidání nového objektu do mapy je poslána zpráva mapovému modulu o jeho pozici a typu. Pokud se herní prvek deaktivuje (tj. nebyl například verifikován), pošle se mapě požadavek na jeho odstranění.

5.3.1 Re prezentace objektů v mapě

Každý nalezený prvek z obrázku je reprezentován instancí třídy, která je potomkem základní třídy herních prvků. Vlastnostmi této třídy jsou horní a spodní souřadnice prvku na hřišti a v obrázku, čas detekce, typ objektu a detekční vzdálenost objektu (euklidovská vzdálenost objektu od kamery). U každého objektu se ukládá informace o jeho validitě. Platí, že každý validní (aktivní) prvek je uložen ve společném mapovém modulu.

Čas detekce je definován jako celočíselná hodnota, která se inkrementuje při každém průběhu hlavní smyčky programu. Informuje tedy o tom, kolikátá sada dat se aktuálně zpracovává.

Skutečná poloha ve světových souřadnicích hřiště je definována jako průměr ze světových souřadnic horního a spodního bodu objektu.

$$\mathbf{u}_w = \left(\frac{x_{wt} + x_{wb}}{2}, \frac{y_{wt} + y_{wb}}{2} \right)^T$$

Kde vektor $\mathbf{u}_{wt} = (x_{wt}, y_{wt})^T$ představuje pozici horního bodu prvku ve světových souřadnicích a $\mathbf{u}_{wb} = (x_{wb}, y_{wb})^T$ pozici spodního bodu.

Herní prvky podle jejich tvaru dále dělíme na třídy rajčat (pro pomeranče a rajčata) a kukuřic (pro falešné a pravé kukuřice). Tyto třídy reprezentující dva základní typy herních prvků ve hřišti (tj. pomeranče - rajčata a falešná - pravá kukuřice) se mezi sebou liší jen hodnotou proměnné vyjadřující typ objektu. Třídy rajčat a kukuřic pouze implementují specifické metody pro daný herní prvek, například ověření polohy herního prvku v obrázku, přepočítání souřadnic obrázku na světové souřadnice apod. U každé kukuřice navíc ukládáme informaci o tom, které pevné pozici ve hřišti byla kukuřice přiřazena.

5.3.2 Správa objektů v mapě

Výstupem detekčního algoritmu je seznam nalezených herních prvků. Úkolem mapy prostředí je tyto objekty správně přiřadit do mapy s ohledem na objekty detekované v předchozích krocích (tj. například zamezit tomu, aby byl jeden herní prvek v mapě dvakrát).

Datový kontejner pro rajčata má specifický způsob přidávání objektu do mapy. Nově nalezený prvek se přidá pouze tehdy, není-li v definovaném kruhovém okolí v mapě nalezen jiný herní objekt. Pokud se v okolí nachází starší objekt, pouze se aktualizuje čas detekce a jeho poloha. Pokud jsou časy detekce stejné (dochází k tomu například v případech, kdy je objekt rozpoznán ze dvou kamer zároveň), objekt s menší detekční vzdáleností se uchová. Jestliže žádný nejbližší prvek nalezen není, herní prvek se přidá do mapy bez úprav.

Datová struktura pro kukuřice řeší správu objektů podobně. Na začátku je však tato struktura naplněna falešnými kukuřicemi s přiřazeným unikátním indexem. Oproti rajčatům se kukuřice do mapy nepřidá, pokud vzdálenost přidávaného prvku od pevné pozice je větší než definované maximum. Tímto způsobem se kukuřice udržují jen na pevných pozicích, ve který mohou být ke stolu připevněny.

Při přidávání nové kukuřice se navíc kontroluje, zda-li je známa herní varianta rozložení kukuřic. Pokud tomu tak není, předá se informace o indexu a typu kukuřice detektoru variant. Pokud se rozpozná herní varianta, všechny dosud nalezené kukuřice se z mapy odstraní a přidají se herní prvky odpovídající nalezené variantě.

Dále je u mapy prostředí implementována možnost kontroly zastaralých herních prvků. Ta se provádí tak, že po přidání celé sady detekovaných prvků se podle času detekce zjišťuje, které objekty v zorném úhlu nebyly nově příchozími daty potvrzeny. Nepotvrzené prvky se deaktivují (tzn. vymažou se ze společné mapy robota).

Verifikací se ověřuje, zda-li je prvek stále přítomný v obrázku. Pokud detekční algoritmus jeho přítomnost nepotvrdí, prvek se deaktivuje. Jestliže je objekt nalezen, aktualizuje se jeho čas detekce a u rajčat i jeho aktuální pozice.

6 Implementace

6.1 Softwarová implementace

Část navrhnutých algoritmů byla prototypována v programu Matlab® R2009a firmy The Mathworks Inc., ale většina algoritmů byla přímo implementována na cílové softwarové platformě.

Program počítačového vidění byl vyvíjen pro operační systém GNU/Linux z toho důvodu, že tento operační systém využívá i výpočetní hardware robota a že na PC Toradex Robin, který jsme měli k dispozici, je vestavěný Solid-State disk o kapacitě jen 2GB. Proto tedy byla vybrána distribuce Debian squeeze¹ (v době psaní práce se jednalo o vývojovou verzi Testing).

Program byl psaný v jazyce C++ a pro kompilaci a debugging zdrojového kódu jsme využili nástrojů ze sady GNU Toolchain. Pro usnadnění práce byla využita řada knihoven. Základem byla knihovna Standard Template Library. Pro serializaci objektů, konfiguraci aplikace a na řadu dalších drobností jsme využily knihovny z projektu Boost C++ Libraries². Pro logování se jevila jako nejvhodnější knihovna *log4cxx*.

Pro práci s obrazem jsme využily knihovnu OpenCV³ verze 2.0 pro C++, vyvíjenou původně společností Intel® a obsahující implementace řady algoritmů pro zpracování obrazu. Tato knihovna mimo jiné také implementuje maticové operace, protože samotné obrázky jsou uloženy jako datové matice. Více informací o této sadě nástrojů lze nalézt na stránce projektu.

Pro práci s kamerami firmy PointGrey bylo využito jejich softwarové knihovny Fly-Capture® SDK⁴. Jedná se o proprietární knihovnu výrobce, která zpřístupňuje snadné API pro ovládání kamer. Každá kamera používaná v programu, má svojí vlastní pozici a je potomkem abstraktní třídy, která implementuje transformace souřadnic a ořezávání nezajímavých částí obrázku. Tuto základní třídu rozšiřuje třída statické kamery, která přidává správu bodů v obrázku, ze kterých by se pro zvýšení přesnosti měli trasovat kukuřice.

Parametry aplikace jsou uloženy v konfiguračních souborech, aby bylo možné co nejjednodušeji editovat. Zároveň jsou v souborech také uloženy transformační matice kamer a startovní body pro ověřování kukuřic a pomerančů u statických kamer. Konfigurační soubory jsou snadno čitelné a uloženy ve formátu konfiguračních souborů Boost nebo v XML. To umožňuje snadnou konfiguraci programu dálkově na cílovém počítači například přes SSH.

6.2 Hardwarová implementace

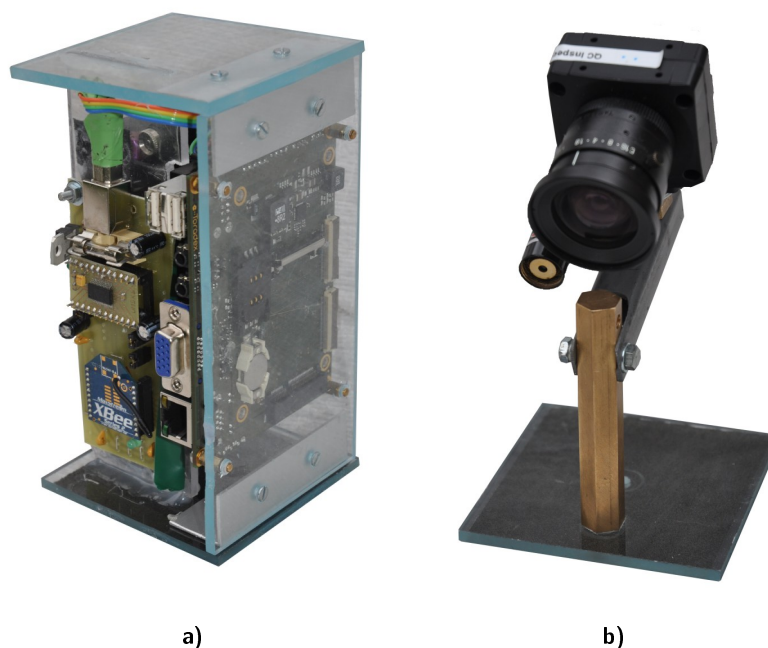
Aplikace je připravena pro umístění kamer do majáčků i na samotného robota. Ve výsledku jsme se však rozhodli, že kamery umístíme pouze do statických pozic v majáčcích, ze kterých je o hře větší přehled.

¹<http://www.debian.org/>

²<http://www.boost.org/>

³<http://opencv.willowgarage.com/wiki/>

⁴<http://www.ptgrey.com/products/pgrflycapture/>



Obrázek 6.1 Hardware rozpoznávání a) Výpočetní jednotka b) Kamera

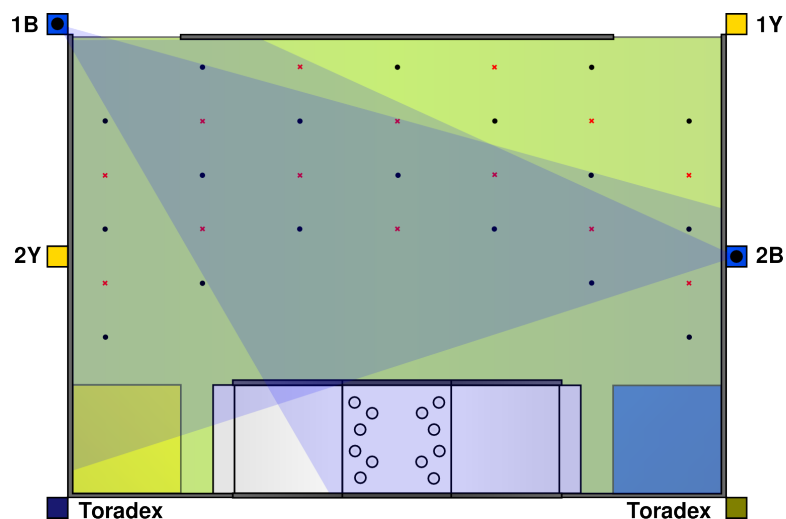
Byl vyroben energeticky soběstačný systém skládající se z trojice majáčků, které mají na starost počítačové vidění robota. Hlavní majáček obsahuje výpočetní Hardware Toradex Robin Z530 spolu s komunikačním modulem a baterií. Ostatní dva majáčky jsou kamerové. Každá kamera je připojena pomocí USB portu k hlavnímu majáčku.

Systém komunikuje s robotem pomocí bezdrátových modulů XBee[®] firmy Digi International Inc.⁵. Dvojice těchto modulů je nakonfigurována v režimu tunelování RS232 tak, že každý z modulů zná hardwarovou adresu toho druhého, čímž se mezi nimi ihned po zapnutí napájení vytvoří plně duplexní spojení. Za zapůjčení modulů a desek s USB-UART převodníkem velmi děkujeme Ing. Petru Novákovi z katedry kybernetiky.

Díky modularitě programového vybavení robota nebyl problém komunikace řešit pomocí zvláštního programového modulu - translátoru zpráv. Tato aplikace po startu naslouchá na nastaveném TCP socketu a všechny příchozí zprávy posílá přes zadané unixové znakové zařízení. Stejně tak druhé vlákno aplikace přijímá data ze zadaného znakového zařízení a (je-li zpráva kompletní) přeposílá zprávu přes TCP sockety na předem určené kanály. Tímto jednoduchým způsobem se přeposílají zprávy z robota do rozpoznávání a zpět jsou přeposílány příkazy do mapového modulu.

Systém je napájen z tříčlánkové modelářské Li-Pol baterie s výstupním napětím 11,1 V a kapacitou 2200 mAh. S touto baterií by měl být systém schopný, dle specifikací výrobce, běžet přes hodinu. Konkrétní testy však nebyly provedeny. Kamery jsou napájeny přes USB port.

⁵<<http://www.digi.com/products/wireless/point-multipoint/xbee-series1-module.jsp>>



Obrázek 6.2 Umístění kamer a znázornění úhlů záběru u modré varianty.

Obraz je snímám kamerami PointGrey Chameleon^{TM6}. Tato kamera byla vybrána pro možnost připojení přes USB port, kterým disponuje použitý výpočetní hardware. S kamerou je použit objektiv o ohniskové vzdálenosti 6.5 mm. S tímto objektivem, díky malé velikosti optického senzoru kamery, dosahuje kamera horizontálního úhlu záběru jen cca 40°. Díky uspořádání hřiště je jedna z kamer umístěna vždy na opačném konci než výpočetní jednotka. Proto je tato kamera připojena přes aktivní 3 m dlouhý USB kabel.

Kamery jsou vzájemně odlišeny pomocí sériového čísla. Spolu s požadavkem na start detekce je od plánování obdržena i informace o startovní barvě, podle které jsou načteny transformační matice kamer.

Nastavení kamer je znázorněno v obrázku 6.2. Toto nastavení bylo zvoleno proto, aby kamera s pořadovým číslem 1 měla v úhlu záběru vyvýšený prostor spolu s pomerančí. Avšak takto nastavené kamery nemají přehled o celém hřišti. Mezi slepé oblasti patří hlavně cílová oblast protivníka, která je však pro našeho robota jen velice málo zajímavá.







Za zapůjčení kamer a výpočetní jednotky děkujeme Centru strojového vnímání při katedře kybernetiky, jmenovitě Ing. Pavlu Krskovi. Za výrobu stojanů kamer velmi děkujeme panu Oldřichu Mudrovi.

⁶<<http://www.ptgrey.com/products/chameleon/index.asp>>

7 Experimenty

V této kapitole se věnujeme zhodnocení výsledků modulu rozpoznávání z několika pohledů. Základní testy jsou provedeny převážně na schopnost detektoru rozpoznat objekty v obrázku. U těchto testů vyhodnocujeme spolehlivost detekce, dosaženou přesnost detekce a časovou náročnost pro zpracování jednoho obrázku.

Ve všech obrázcích jsou detekované objekty znázorněny stejnými symboly:

	Spodní bod	Horní bod
Rajčata		
Pravé kukuřice		
Falešné kukuřice		

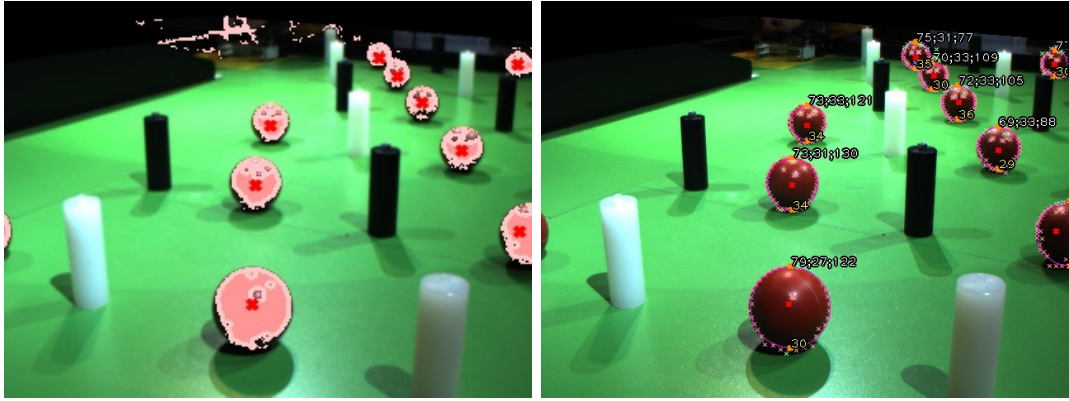
7.1 Ideální průběh detekce

V obrázcích 7.1 až 7.3 je na obrázku se startovním rozložením herních prvků podrobně znázorněn průběh detekce. Při znázornění nalezených oblastí je neplatná oblast ohraničena červeným okrajem. Zároveň je uveden kód chyby, představující podmínku, které detekovaný objekt nevyhověl.

Lze vidět, že způsob detekce oblastí kukuřic, znázorněného v obrázku 7.2, není příliš robustní. V tomto případě jsme přehlédli dvě kukuřice a jedna z nalezených falešných kukuřic nevyhovovala podmínce minimálního počtu inlierů při zjišťování rovnic přímek (kód chyby R). To bylo způsobeno tím, že na pozadí kukuřice je okraj hřiště se stejnými optickými vlastnostmi. Některé z oblastí byly z výsledku vyřazeny z důvodu nevyhovujícího tvaru (kód chyby M). Některé detekované oblasti zase nevyhověly požadavku úhlu nalezených přímek (kód chyby A).

Při detekci ze dvou statických kamer jsou výsledky kombinovány do jedné mapy. Vizualizace mapy robota, spolu s detekovanými objekty ze dvou obrázků, je znázorněna na obrázku 7.4.

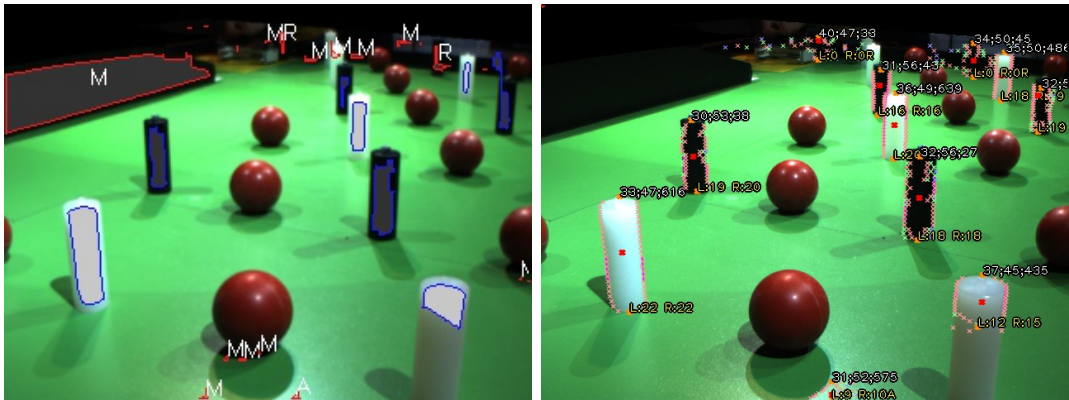
7 Experimenty



a)

b)

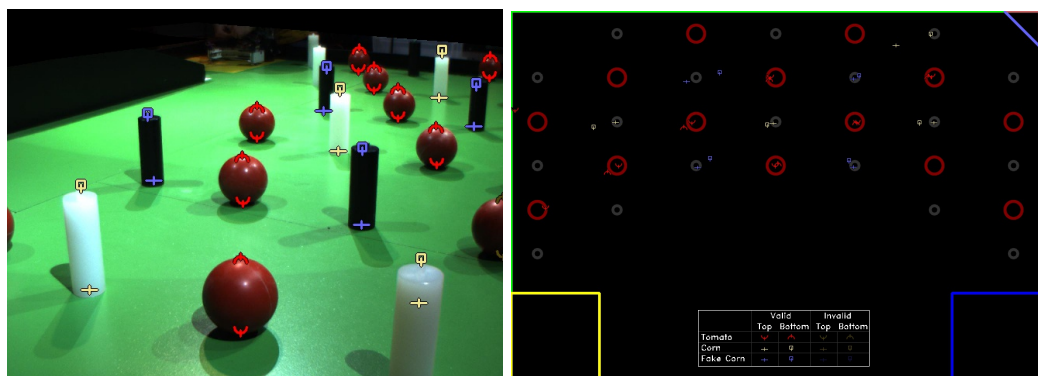
Obrázek 7.1 a) Výsledek hledání oblastí rajčat. Červené křížky znázorňují těžiště oblastí.
b) Výsledek hledání bodů na hranici objektu a parametrů kružnic rajčat.



a)

b)

Obrázek 7.2 a) Výsledek hledání oblastí kukuřic. Modře obtažené oblasti odpovídají nalezeným kukuřicím, zatímco červené některé z podmínek nevyhověly.
b) Výsledek hledání bodů na hranici objektu a parametrů přímek kukuřic.

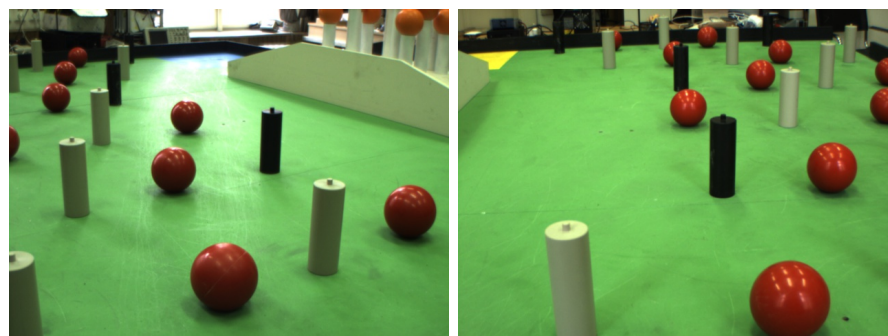


a)

b)

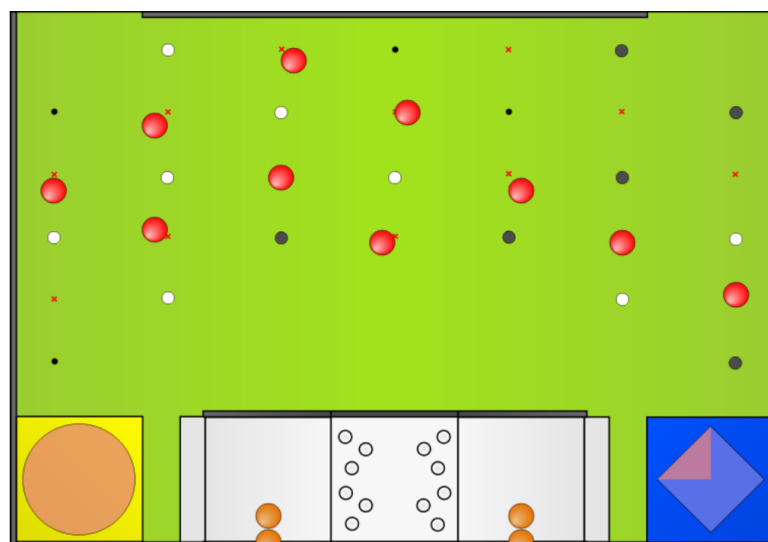
Obrázek 7.3 a) Nalezené objekty v obrázku b) Pozice nalezených objektů na hřišti.

7 Experimenty



a)

b)



c)

Obrázek 7.4 a) Obraz z první kamery b) Obraz z druhé kamery c) Výsledek zanesený v mapě robota.

Obrázek	Rajčata			Kukuřice			Falešné kuk.		
	HR[%]	FP	Δ_i [px]	HR[%]	FP	Δ_i [px]	HR[%]	FP	Δ_i [px]
7.5a	90%	0	4.76	71%	0	6.95	50%	0	4.84
7.5b	90%	0	4.24	100%	0	6.67	50%	0	6.51
7.5c	100%	0	5.29	0%	0	-	67%	3	4.83
7.5d	100%	2	3.84	0%	0	-	75%	0	9.21
7.5e	83%	0	3.66	0%	1	6.95	100%	0	4.79
7.5f	88%	29	25.83	33%	1	8.3	0%	0	-
7.5g	100%	2	3.66	0%	1	-	50%	0	10.82
7.5h	100%	1	4.39	80%	0	5.23	50%	2	5.93
7.5i	100%	0	3.36	50%	0	4.95	67%	3	10.88
7.5j	100%	0	4.01	0%	0	-	67%	0	5.67
7.5k	69%	1	7.42	0%	1	-	67%	1	8.01
7.5l	62%	1	6.94	57%	2	4.7	50%	0	5.87
7.5m	67%	0	5.28	100%	0	10.04	100%	0	12.68
7.5n	100%	4	3.36	0%	0	-	67%	1	8.79
7.5o	67%	0	4.76	0%	0	-	100%	0	15
Průměr:	89%	1.64	5.21	54%	0.65	5.82	58%	0.81	7.31

Tabulka 7.1 Výsledky testování spolehlivosti detekce objektů, HR - procentuální úspěšnost (Hit Rate), FP - Počet klamných detekcí (False Positive), Δ - Absolutní chyba. Průměr je počítán ze 32 obrázků.

7.2 Spolehlivost detekce

Výsledek detekce závisí na kvalitě osvětlení scény. Protože kamera automaticky koriguje vyvážení bílé, jsou některé obrázky hůře detekovatelné. Mezi ně patří například obrázek 7.5f. Zde byla scéna osvětlena silným halogenovým osvětlením a výsledný obraz je načervenalý. To vedlo k nalezení velkého množství klamných rajčat, protože například povrch hřiště měl podobnou chromaticitu jako rajče osvětlené světlem odraženým z hřiště.

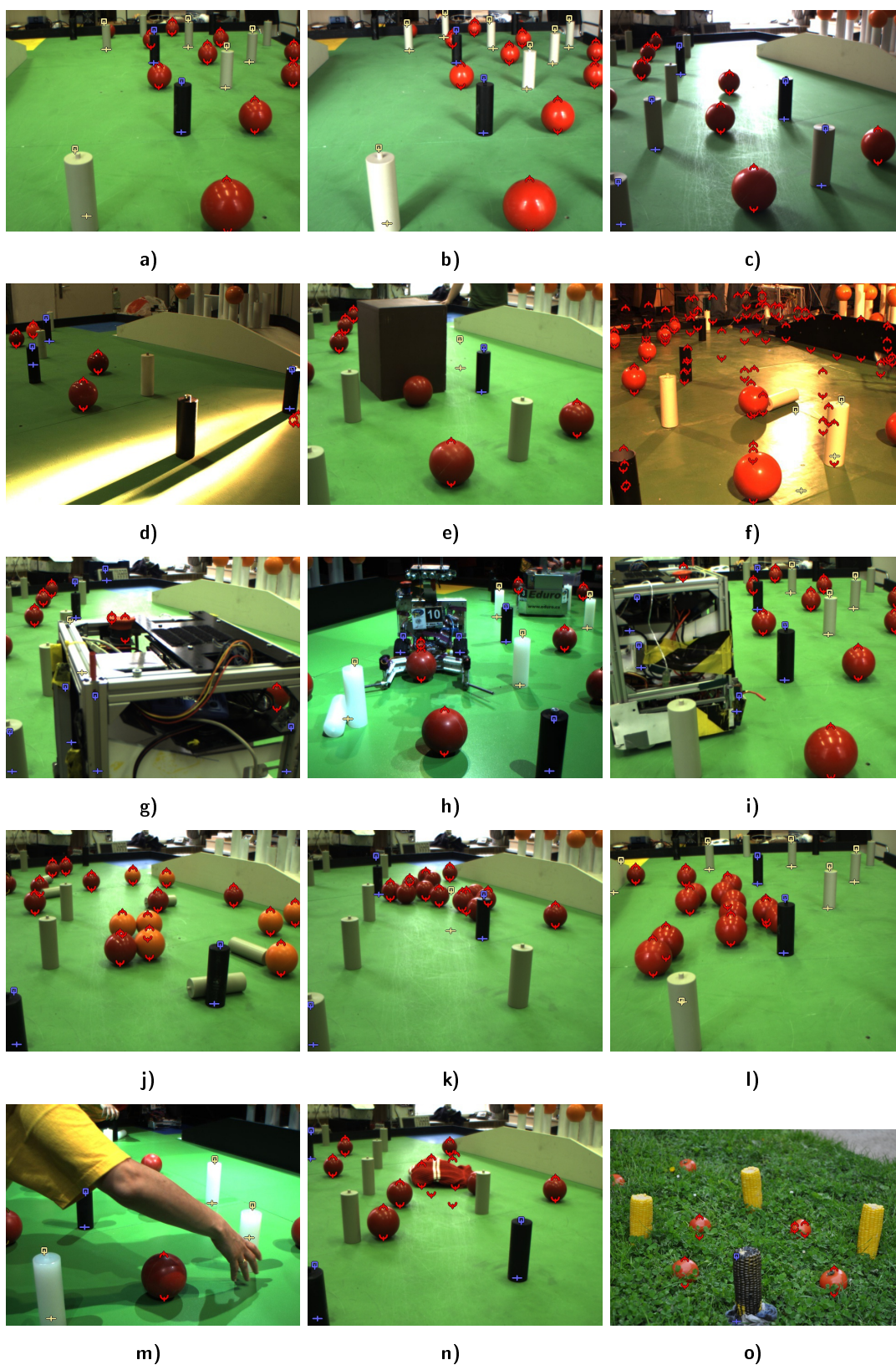
Z výsledků lze také vidět, že detekce kukuřic není příliš robustní na osvětlení. V řadě obrázků nebyly vůbec nalezeny oblasti, které by odpovídaly kukuřicím. V obrázku 7.5c snížení jasu kamery vedlo k tomu, že pravé kukuřice se rozpoznaly jako falešné. V obrázku 7.5k byla část povrchu hřiště nalezena jako pravá kukuřice (v tomto případě to způsobil nízký práh počtu inlierů u RANSACu, takže oblast, omezená rajčaty a průsvitem mezi nimi, se jevila jako pravá kukuřice).

Zákryty rajčat také v některých případech nebyly rozpoznány úplně ideálně. Například v obrázku 7.5l, kde jsou rajčata v řadě za sebou, došlo k tomu, že byla nalezena kružnice odpovídající několika rajčatům zároveň. Tento problém by se dal vyřešit například ověřováním nalezených kružnic.

Detektor kukuřic také není příliš odolný proti cizím prvkům v obraze, což se projevilo v obrázcích 7.5g, 7.5h, 7.5i. Cizí červený objekt byl také určen jako řada rajčat (obrázek 7.5n). Jak již bylo řečeno, tento problém by se dal vyřešit ověřováním barevných hodnot na nalezené kružnici. Tento postup jsme však neimplementovali, protože bychom nemohli detekovat rajčata v zákrytu stejným způsobem jako v obrázku 7.5a.

V 7.5e a 7.5m nebylo rajče u načervenalého objektu detekováno proto, že oblast velkého objektu splynula s oblastí rajčete, které načež bylo vyřazeno kvůli velkému

7 Experimenty



Obrázek 7.5 Vizualizace výsledků testování spolehlivosti detekce objektů.

objemu.

7.3 Přesnost určení pozice objektu v obrázku

Přesnost detekce objektu v rámci obrázku (Δ_i) je hodnota, která popisuje jen vlastnosti detektoru. Přesnost detekce objektu ve světových souřadnicích (souřadnicích hřiště) je závislá jak na kalibraci kamer, tak na detekční vzdálenosti (vzdálenost objektu od kamery). Chyba jednoho pixelu v obrázku u vzdáleného objektu může způsobit chybu na hřišti až v jednotkách centimetrů. Přesnosti detekce z obrázků z 7.5 je uvedena v tabulce 7.2.

Na množině 32 testovacích obrázků byla spočítána průměrná chyba detekce význačných bodů objektů:

	Rajčata	Kukuřice	Falešné kukuřice
Δ_i [px]	5.21	5.82	7.31
Δ_w [mm]	44.03	43.14	57.87
δ_w [mm/m]	21.06	24.76	34.55

Hodnota Δ_w je průměrná absolutní chyba v souřadnicích hřiště. A jak již bylo zmíněno, je ovlivněna vzdáleností objektu od kamery (vzdálenější objekty mají větší chybu detekce než bližší objekty). Proto jsme pro každý nalezený prvek vypočítali relativní chybu vůči euklidovské vzdálenosti půdorysu středu kamery od nalezeného objektu v metrech. V tabulce je uvedena jako hodnota δ_w . Například hodnota 21.06 mm/m u rajčat znamená, že u objektů nalezených 1 m od kamery je průměrná chyba 21.06 mm v souřadnicích hřiště.

7.4 Výpočetní náročnost

Doba potřebná pro zpracování jednoho obrázku (o rozměrech 640×480 pixelů) byla měřena na cílovém výpočetním prostředku. Jedná se o průměrnou hodnotu z 32 testových souborů. Naměřené hodnoty jsou jen orientační, protože byly změřeny pomocí funkce `clock()` s přesností v řádu ms.

	Rajčata	Kukuřice
Detekce oblastí	79.72 ms	63.89 ms
Určení polohy	40.83 ms	41.39 ms
Celkem	225.83 ms	

7.5 Výsledky ze soutěže

České kolo soutěže se konalo dne 1.5.2010. V té době byla implementace systému rozpoznávání funkční, ale bohužel nebyla v samotné soutěži použita. Z technických důvodů, které vznikly dva dny před začátkem soutěže (poškozená odometrie), byl robot schopen pouze základního pohybu s velkými nepřesnostmi. Celá naše strategie se tedy omezila na průjezd hřištěm po diagonále a vysypání nasbíraných rajčat do cílové oblasti. Během hry byl na robotovi spuštěn pouze základní modul ovládání kol spolu s frontou příkazů pro pohyb robota.

Přesto jsme se s touto jednoduchou strategií umístili na čtvrtém místě a na základě pravidel tak postupujeme do evropského kola konaného dne 27.5.2005 ve Švýcarském Rappersvill-Jona.

7 Experimenty

V průběhu soutěžního klání se vyskytly problémy s použitou USB sběrnici u kamery připojené přes aktivní kabel, s níž se nedařilo navázat spojení. Pokud se stejná kamera připojila přes kratší kabel, problém se znovu neobjevil. Bohužel se nám zatím nepodařilo zjistit přesnou příčinu tohoto problému.

Během soutěže byly také pořízeny fotky herního prostředí, které zde používám pro některé experimenty.

8 Závěr

V rámci práce jsem se seznámil se základy zpracování obrazu, především však s metodami segmentace obrazu a implementační stránkou problému.

Výsledkem práce je systém rozpoznávání herních prvků na základě jejich barevných vlastností. Pro detekci byl navržen jednoduchý a rychlý způsob barevné segmentace, který by byl s malými úpravami použitelný i v jiných aplikacích.

Protože mají rajčata v obrázku velmi výraznou barvu oproti okolí, daří se je realizovaným způsobem poměrně spolehlivě detekovat. V případě větších zákrytů však není segmentace schopná rozlišit jednotlivé objekty a proto v těchto případech detekce selhává. Protože jsou objekty zjišťovány na základě barvy, při velké změně barvy osvětlení robustnost algoritmu klesá.

Realizované detekování kukuřic není příliš spolehlivé, protože pro zjišťování polohy v obrázku byl zvolen podobný způsob detekce jako u rajčat. Ten však vychází z barvy objektu, která není u kukuřic natolik výrazná. Navíc je barva těchto objektů ovlivněna odraženým světlem od povrchu hřiště a tak je někdy obtížné je vzájemně odlišit. Ve samotné hře je použito pouze jejich ověření ze statických kamer, které je spolehlivější. Vhodnější metodou pro detekci kukuřic by byla například hranová detekce.

V rámci práce byla také implementována jednoduchá mapa prostředí spravující detekované objekty. Systém dokáže objekty i ověřovat, což vede především k úspoře výpočetního výkonu.

Systém rozpoznávání byl zabudován do majáčků, které jsou umístěny na okrajích hřiště. Pro připojení kamer byla použita sběrnice USB, kde se však u delších kabelů v soutěžním prostředí objevili problémy s komunikací. Pokud by se podařilo tento problém odstranit, například přidáním dalších aktivních prvků, je tato sestava využitelná i pro další ročníky soutěže. Způsob komunikace pomocí XBee[®] bezdrátového rozhraní je snadno použitelný s širokou škálou výpočetní techniky (od stolních počítačů po osmibitové mikrokontroléry).

Literatura

- CHUM, O. *Two-View Geometry Estimation by Random Sample and Consensus: PhD Thesis*. Praha : ČVUT Fakulta Elektrotechnická, 2005. ix, 90 s.
- EUROBOT. *Eurobot 2010 rules* [online]. Planète Sciences, 2009. Dostupné z: <http://www.eurobot.org/commonfiles/docs/2010/E2010_rules_and_drawing_EN.pdf>.
- EUROBOT. *Eurobot 2010 - dodatky* [online]. Katedra softwarevého inženýrství Matematicko-fyzikální fakulty, Univerzita Karlova v Praze, 2010. Dostupné z: <<http://www.eurobot.cz/faq2010.php>>.
- FORSYTH, D. A. – PONCE, J. *Computer Vision: A Modern Approach*. Upper Saddle River : Prentice Hall, 2003. xxv, 693 s. ISBN 0-13-085198-1.
- HARTLEY, R. I. – ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. Cambridge : Cambridge University Press, second edition, 2003. xvi, 655 s. ISBN 0-521-54051-8.
- KRAVTCHENKO, V. – LITTLE, J. J. Efficient Color Object Segmentation Using the Dichromatic Reflection Model. *1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*. 1999, s. 90 – 94.
- SHAPIRO, L. G. – STOCKHAM, G. C. *Computer Vision*. Upper Saddle River : Prentice Hall, 2001. 608 s. ISBN 0-13-030796-3.
- SONKA, M. – HLAVÁČ, V. – BOYLE, R. *Image Processing, Analysis, and Machine Vision*. Toronto : Thomson, third edition, 2008. xxv, 829 s. ISBN 978-0-495-24438-7.
- WIKIPEDIA. *SVD* [online]. Wikimedia Foundation, Inc., 2010. Dostupné z: <http://en.wikipedia.org/wiki/Singular_value_decomposition>.

Obsah CD

Přiložené CD obsahuje zdrojové kódy modulu vidění, text bakalářské práce ve formátu PDF a zdrojové kódy celého textu pro systém L^AT_EX. V následující tabulce je popsána struktura CD.

Adresář	Popis
<code>src</code>	zdrojové kódy modulu
<code>lib</code>	použité knihovny
<code>doc</code>	zdrojové kódy textu bakalářské práce
<code>thesis.pdf</code>	text bakalářské práce

Tabulka .1 Adresářová struktura na CD