

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra kybernetiky

## Rozpoznávání vizuálního vzoru pro automatické přistávání robotické helikoptéry

**Jan Gärtner**

Vedoucí: Ing. Jan Chudoba  
Studijní program: Kybernetika a robotika  
Duben 2019



## Poděkování

Děkuji vedoucímu práce panu Ing. Janu Chudobovi za vstřícnost, faktickou pomoc a dohled nad celým projektem. Děkuji také své Koze, rodině, spolužákům a kamarádům za to, že mi v průběhu celého studia vytvářeli příjemné prostředí k životu i práci.

## 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.

V Praze 22. května 2019

podpis autora práce

## Abstrakt

Cílem bakalářské práce bylo vytvořit vizuální vzor určený ke značení přistávacího místa helikoptéry a navrhnout algoritmus sloužící k jeho detekci v obraze. V rámci práce je představen lokalizační systém *hetect*, který na snímcích pořízených kamerou helikoptéry detekuje přistávací vzor a odhadne relativní polohu helikoptéry vůči přistávacímu místu. Systém je za daných okolností schopen detekovat vzor i v situaci, kdy není celý v zorném poli kamery, nebo je narušen stíny. Jednotlivé kapitoly se zabývají metodami zpracování obrazu, grafickým návrhem vzoru a implementací algoritmu s ohledem na omezení vlivu osvětlení. V závěrečné části práce byly zmíněné vlastnosti systému experimentálně ověřeny.

**Klíčová slova:** helikoptéra, dron, rozpoznání vizuálního vzoru, autonomní přistávání

**Vedoucí:** Ing. Jan Chudoba  
Inteligentní a mobilní robotika CIIRC,  
Jugoslávských partyzánů 1580/3  
Praha 6

## Abstract

The aim of the bachelor thesis was to design a visual pattern used for marking a landing platform for a helicopter and to design an algorithm, which could detect the pattern in a picture. A localization system *hetect* is introduced within this project. It detects the landing pattern in pictures taken by a camera placed on a helicopter and calculates the presumable relative location of the helicopter. Under certain conditions, the system is able to recognize the pattern even in situations, where it is partly out of the camera's field of vision or when it is partly covered by shadows. The chapters are devoted to image processing methods, graphical design of the pattern and an implementation of the algorithm with an emphasis on reducing an impact of lighting conditions on the result of the detection. All the mentioned qualities of the detection system were verified by real experiments in the final part of the thesis.

**Keywords:** helicopter, drone, visual pattern recognition, autonomous landing

**Title translation:** Visual Pattern Recognition for Autonomous Landing of Robotic Helicopter

## Obsah

<b>1 Úvod</b>	<b>1</b>	<b>5 Ověření funkčnosti systému</b>	<b>31</b>
1.1 Příbuzné projekty . . . . .	2	<i>hetect</i>	
1.2 Detekční systém <i>hetect</i> . . . . .	4	5.1 Ověření jednotlivých fází detekčního algoritmu . . . . .	31
<b>2 Metody detekce rohů v obraze</b>	<b>5</b>	5.2 Detekce vzoru za nepříznivého osvětlení . . . . .	32
2.1 Harrisův detektor rohů . . . . .	5	5.3 Vliv prostředí na výsledek detekce	33
2.2 Detektor Shi-Tomasi . . . . .	6	5.3.1 Lokalizace vzoru na různém pozadí . . . . .	33
2.3 Srovnání detektorů rohů . . . . .	7	5.3.2 Cílené umístění rohu šachovnice do záběru kamery . . . . .	33
2.4 Funkce <code>goodFeaturesToTrack</code> . . . . .	8	5.3.3 Zpracování snímků bez přítomnosti vzoru . . . . .	34
<b>3 Návrh grafického vzoru pro přistávání helikoptéry</b>	<b>9</b>	5.4 Vyhodnocení snímků pořízených za letu . . . . .	34
<b>4 Návrh algoritmu pro rozpoznávání přistávacího vzoru</b>	<b>13</b>	5.4.1 Dosednutí helikoptéry na přistávací vzor . . . . .	34
4.1 Předzpracování obrazu . . . . .	13	5.4.2 Snímání vzoru z větší výšky .	36
4.2 Detekce rohů v obraze . . . . .	14	5.5 Odhad vzdálenosti helikoptéry od vzoru . . . . .	37
4.2.1 Zpřesnění pozice rohu . . . . .	15	<b>6 Závěr</b>	<b>41</b>
4.3 Korelační test na určení rohů šachovnice . . . . .	16	<b>A Obrazový materiál</b>	<b>43</b>
4.3.1 Princip korelačního testu . . . . .	16	<b>B DVD</b>	<b>49</b>
4.3.2 Vliv vzájemné polohy kamery a obrazce . . . . .	18	<b>C Literatura</b>	<b>51</b>
4.3.3 Potlačení falešných detekcí . .	18	<b>D Zadání práce</b>	<b>53</b>
4.4 Ověření přítomnosti kruhové hranice . . . . .	19		
4.4.1 Metoda odstředných paprsků	20		
4.4.2 Proklad hraničních bodů elipsou . . . . .	23		
4.5 Využití informací mezi jednotlivými snímky . . . . .	26		
4.6 Odhad relativní pozice helikoptéry vůči přistávacímu vzoru . . . . .	28		

## Obrázky

1.1 Vizualní vzor použitý v projektu <i>WhyCon</i> [14] . . . . .	3	4.8 Pohled na zkreslený přistávací vzor s narušenou středovou symetrií . . . . .	22
1.2 Ukázky vizuálních přistávacích vzorů . . . . .	3	4.9 Přistávací vzor s ostrým stínem nebo jiným rušivým elementem zakrývajícím hledaný přechod z černé barvy na bílou . . . . .	23
2.1 Základní princip Harrisova detektoru rohů . . . . .	6	4.10 Příklady chybně zastavených paprsků v rizikových situacích . . . . .	24
2.2 Význam vlastních čísel matice $M$ z pohledu Harrisova detektoru rohů a detektoru Shi-Tomasi . . . . .	7	4.11 Hodnota ukazatele $\Delta$ v závislosti na poloze v souřadném systému. . . . .	25
2.3 Referenční vzorky rohových a nerohových struktur v testu [13] . . . . .	7	4.12 Princip dvoustupňového prokladu elipsou . . . . .	26
3.1 Typický vzor používaný pro značení heliportu . . . . .	10	4.13 Zobrazení bodu v prostoru na rovinu snímače pomocí modelu dírkové kamery [2] . . . . .	29
3.2 Prototypy přistávacího vzoru s šachovnicovým rohem . . . . .	11	5.1 Graf průběhu dosednutí helikoptéry . . . . .	35
3.3 Přistávací vzor používaný systémem <i>hetect</i> . . . . .	11	5.2 Odhad relativní polohy dronu v průběhu komplexního testovacího letu . . . . .	38
4.1 Průběh zpracování obrazu z kamery systémem <i>hetect</i> . . . . .	14	5.3 Doba zpracování snímků v průběhu testovacího letu . . . . .	38
4.2 Chyby v detekci středového rohu obrazce na rozostřených snímcích . . . . .	16	5.4 Poměr výpočetní náročnosti jednotlivých fází detekce . . . . .	39
4.3 Ilustrace náběru hodnot intenzity jasové složky v (červených) bodech na kružnici okolo detekovaného rohu . . . . .	17	5.5 Odhad vzdálenosti kamery od přistávacího vzoru . . . . .	39
4.4 Deformace vzoru při šikmém pohledu . . . . .	18	A.1 Jednotlivé fáze detekce zarušeného přistávacího vzoru (2. část) . . . . .	44
4.5 Ilustrace způsobu prohledávání metodou odstředných paprsků . . . . .	20	A.2 Náhlá změna jasu při kontinuálním snímání . . . . .	45
4.6 Příklad pozvolné změny jasu na přistávacím vzoru s použitím metody naivní komparace jasu . . . . .	20	A.3 Porušení přistávacího voru na různých místech . . . . .	45
4.7 Přistávací vzor zarušený ostrými stíny nebo jinými rušivými elementy s kontrastním přechodem . . . . .	22	A.4 Vliv stínů v různých místech přistávacího obrazce . . . . .	45

A.5 Vzor na řídkém trávníku s vysokým počtem kontrastních rohů	46
A.6 Šachovnicové pole mimo prohledávanou oblast snímku	46
A.7 Letecké snímky bez přítomnosti přistávacího vzoru [5]	46
A.8 Kritická vzdálenost, ve které přestává být obrazec detekován	47
A.9 Výřez snímku s rozmazaným přistávacím vzorem	47
A.10 Výřez snímku s falešnou detekcí přistávacího vzoru	48
A.11 Přiblížení k přistávacímu vzoru	48

## Tabulky

5.1 Úspěšnost detekce vzoru na různém pozadí	33
5.2 Výsledky zpracování snímků bez přítomnosti přistávacího vzoru	34
5.3 Úspěšnost detekce vzoru při přistání	36
5.4 Úspěšnost detekce vzoru v rámci komplexního testovacího letu	37
5.5 Příčiny nenalezení vzoru v rámci komplexního testovacího letu	37
5.6 Výpočetní čas spotřebovaný jednotlivými částmi detekčního algoritmu v průběhu komplexního testovacího letu	38





# Kapitola 1

## Úvod

V posledních letech zaznamenáváme významný rozvoj autonomních automobilů, helikoptér, dronů a dalších typů inteligentních zařízení. Mají schopnost zorientovat se v neznámém prostředí a provádět rozhodnutí založená na vnímání okolí. Vývoj v této oblasti přináší obrovský potenciál pro zjednodušení a zlevnění úkonů v dopravě, logistice, distribuci zboží nebo při záchranných akcích.

Pro navigaci autonomních zařízení se většinou využívá kombinace několika principů. Pro příklad uvádím lokalizační systém GPS, lidary, ultrazvukové dálkoměry, barometrické výškoměry, senzory proudění tekutin a další. V případě lokalizace a navádění na známý objekt je možné využít kamer a široké škály metod zpracování obrazu.

Tato práce je zaměřena na přistávání robotických helikoptér a dronů s užitím vizuálního přistávacího vzoru. Koncept je vhodný pro aplikace v budovách a místech s absencí signálu GPS nebo v případech, kdy je požadována vysoká přesnost navigace (v řádu centimetrů). Typickým modelem pro použití metody je helikoptéra, která za pomoci jiného navigačního systémem doletí nad zadanou oblast, kde dojde k přepnutí do režimu vizuálního řízení a přesnému přistání na označené místo.

Naším cílem bylo navrhnout obrazový vzor vhodný pro značení přistávacího místa helikoptéry a vyvinout algoritmus schopný detekovat tento vzor v obraze snímaném kamerou připevněnou na helikoptéře. Na základě detekce vzoru následně určit relativní polohu helikoptéry vůči přistávacímu místu. Při návrhu algoritmu i obrazového vzoru byl kladen důraz na možnost detekce při velkém přiblížení a snížení vlivu osvětlení.

## 1.1 Příbuzné projekty

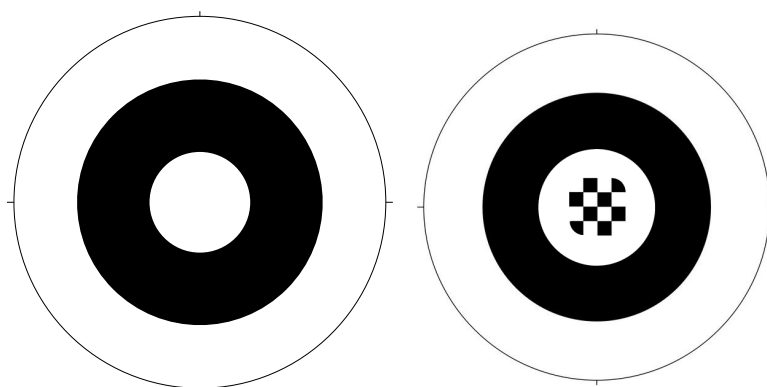
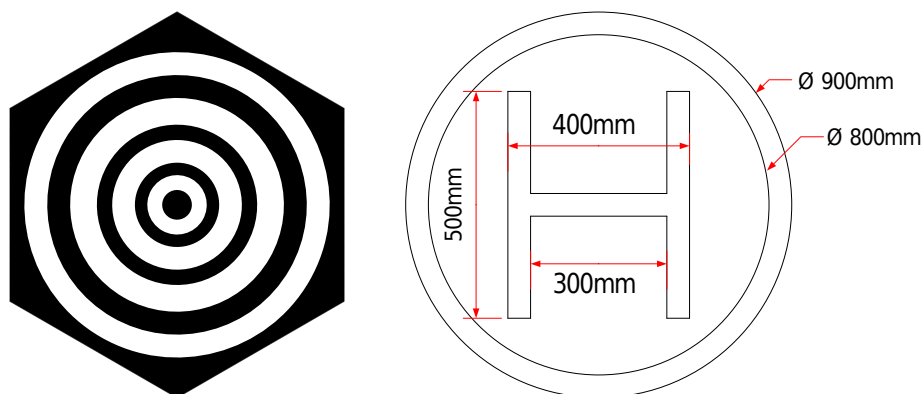
Tématu vizuálního navádění helikoptér při přistávání se věnuje celá řada jiných projektů. Většina z nich ([14], [17], [18]) vyžaduje pro detekci viditelnost celého vzoru. Vzor proto může být vyhotoven jen v takových rozměrech, aby se při dosednutí vešel do záběru kamery. To zásadně limituje maximální pracovní vzdálenost, neboť z větších výšek není malý vzor rozpoznatelný. Použitím většího vzoru bychom se připravili o možnost přesně lokalizovaného přistání, jelikož algoritmus při větším přiblížení ztratí informaci o poloze středu obrazce.

Ve spolupráci vědců z *Texas A&M University* a *University of Southern California* vznikl projekt [18] představující systém pro detekci přistávacího vzoru ve tvaru písmene **H**. Vzor je detekován metodou segmentace. Obrazové pixely jsou nejprve rozděleny do dvou skupin, na černé a bílé. Tím vzniknou spojitě oblasti pixelů, pro než jsou následně ověřovány geometrické vlastnosti, např. poměr obvodu a plochy. Tyto vlastnosti nezávisí na přiblížení a rotaci obrazce.

Projekt *Pusan National University* [17] je jedním z projektů užívajících mezinárodního vzoru pro značení heliportů, viz obr. 1.2b, který zpravidla vyžaduje přítomnost celého vzoru v zorném poli kamery. První fází detekce je opět segmentace pixelů na černé a bílé. Algoritmus následně hledá spojitě oblasti pixelů s poměrem výšky a šířky přibližně 1 : 1, předpokládá tedy kolmé natočení kamery k zemi. Mezi těmito oblastmi se pokouší identifikovat vnější mezikruží, uvnitř kterého pak za pomoci EDF (*Edge Distribution Function*) hledá písmeno **H**.

Algoritmus *WhyCon* [14] vyvinutý ve spolupráci *Fakulty elektrotechnické Českého vysokého učení technického v Praze*, *University of Lincoln* a *University of Buenos* pracuje s jednoduchým vzorem ve tvaru mezikruží, viz obr. 1.1a. Je založen na principu zvaném *blob detection* a dokáže vyhodnotit tisíce snímků za sekundu. Prohledávání obrazu započne v náhodně zvoleném bodě a hledá spojitou oblast černých pixelů o tvaru mezikruží, následně ověřuje přítomnost bílé kruhové oblasti uvnitř mezikruží. Jakmile je vzor detekován, prohledávání následujících snímků je inicializováno v bodě předpokládané pozice středu obrazce. Ve většině případů tedy není třeba prohledávat zbytek obrázku.

Na projekt *WhyCon* [14] navazuje bakalářská práce *Automatický vzlet a přistání robotické helikoptéry* [15] napsaná taktéž na *Fakultě elektrotechnické Českého vysokého učení technického v Praze*. Řeší problémem detekce mezikruží při velkém přiblížení, kterým se doposud zmíněné projekty nezabývaly. Do středu vzoru s mezikružím je umístěna část šachovnice (obr 1.1b). Dojde-li k přiblížení kamery k přistávacímu vzoru, algoritmus začne hledat šachovnici a nevyžaduje přítomnost zbytku obrazce. Složitý vzor však v praxi naráží na špatnou rozeznatelnost, zejména při nepříznivém osvětlení z obrazu zcela vymizí.

(a) : Původní vzor projektu *WhyCon* [14](b) : Upravený vzor *WhyCon* s šachovnicí [15]**Obrázek 1.1:** Vizuální vzor použitý v projektu *WhyCon* [14]

(a) : Vzor soustředných mezikruží [16]

(b) : Mezinárodní vzor pro heliport [17]

**Obrázek 1.2:** Ukázky vizuálních přistávacích vzorů

Článek [16] a v něm popsaný detekční algoritmus jsou dalším příkladem projektů zabývajících se problémem detekce vzoru ve velkém rozsahu vzdáleností. Je použit obrazec tvořený sadou soustředných mezikruží (obr. 1.2a). Algoritmus vždy detekuje největší z viditelných mezikruží. Z poměru poloměrů vnitřní a vnější kružnice daného mezikruží je schopen identifikovat, které z mezikruží našel, a tuto informaci využívá k určení vzdálenosti od přistávacího místa. Systém provádí segmentaci obrazu s pevnou mezní hodnotou mezi černou a bílou barvou, není proto přizpůsoben změnám absolutních hodnot barev.

Prostudované projekty jsou více či méně odolné vůči změnám okolního osvětlení a posunu absolutních hodnot barev. Žádný z jmenovaných projektů se však podrobněji nezabývá možným narušením vizuálního vzoru. Helikoptéra v reálných podmínkách při přistání často vrhá stíny, které mohou ovlivnit průběh detekce.

## 1.2 Detekční systém *hetect*

Představujeme systém *hetect* (heliport-detection) určený pro detekci vizuálního přistávacího vzoru v obraze a relativní lokalizaci helikoptéry. Byl navržen tak, aby co nejlépe kombinoval následující vlastnosti:

- vysoký rozsah vzdálenosti, ze kterých je možno vzor nalézt
- přesné zaměření středu obrazce při velkém přiblížení
- zpracování snímků v reálném čase
- funkčnost za proměnlivého osvětlení
- schopnost detekce částečně narušeného vzoru stíny nebo jinými artefakty

Vstupem systému jsou snímky pořízené kamerou umístěnou na helikoptéře, která je namířena kolmo dolů a snímá scénu pod helikoptérou. Výstupem je odhad relativní pozice helikoptéry vůči středu přistávacího vzoru. Souřadnice mohou následně sloužit jako vstup regulátoru pro navádění helikoptéry k přistání. Návrh regulátoru není součástí této práce.

Systém umožňuje detekci přistávacího vzoru při velkém přiblížení, tedy i v situaci, kdy není obrazec celý v zorném poli kamery. Maximální pracovní vzdálenost systému je tedy omezena jen fyzickými rozměry použitého vzoru, avšak při velkém přiblížení je výstupem pouze informace o vychýlení od středu obrazce v rovině a ztrácíme informaci o výšce helikoptéry.

Algoritmus systému *hetect* nevychází přímo z žádného existujícího projektu, byl celý navržen v rámci této práce. Je založen na detekci rohů v obraze a na rozdíl od projektů zmíněných v úvodu nevyužívá metody segmentace. Od tohoto přístupu si slibujeme lepší možnosti nalezení vzoru narušeného například vlivem stínů.

## Kapitola 2

### Metody detekce rohů v obraze

Jádrem systému *hetect* je detektor rohů v obraze. Využíváme funkci *goodFeaturesToTrack* knihovny OpenCV implementující detektor rohů Shi-Tomasi. Ten je založen na principech široce užívaného Harrisova detektoru.

#### 2.1 Harrisův detektor rohů

Základem většiny rozšířených algoritmů na detekci rohů je tzv. Harrisův detektor, publikovaný ve článku *A Combined Corner and Edge Detector* [12]. Systému *hetect* pracuje s verzí implementovanou v knihovně OpenCV 3.0 [3].

Filozofie problému je následující. Potřebujeme-li rozhodnout, zda bod v obraze představuje roh, zaměříme se na jeho okolí. Vytvoříme masku obsahující kýžený bod a sledujeme rozdíl intenzity barev (jasových složek) při posunu masky ve všech směrech. Označíme-li tento rozdíl jako  $E$ , můžeme psát

$$E(u, v) = \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2, \quad (2.1)$$

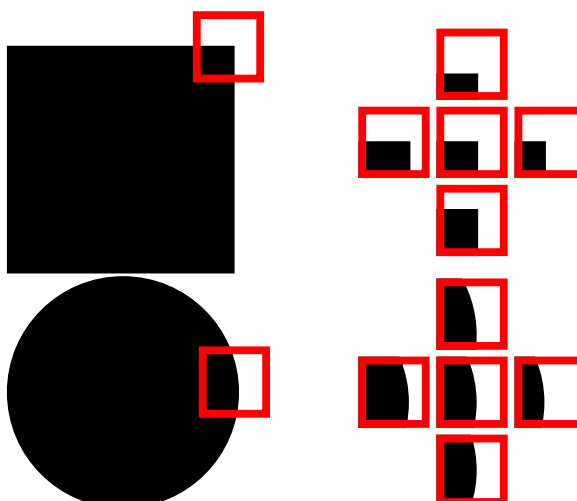
kde  $x, y$  značí souřadnice bodu v obrázku,  $u, v$  posun masky ve směru  $x, y$ ,  $I(x, y)$  je intenzita jasové složky v bodu  $x, y$  a  $w(x, y)$  je maska okolí bodu. Hodnota  $E$  je tím větší, čím ostřejší je roh v obrázku. Princip posunu masky je demonstrován na obr. 2.1

Výraz (2.1) aproximujeme Taylorovým polynomem I. řádu a převedeme do tvaru

$$E(u, v) \approx \sum_{x,y} w(x, y) [I(x, y) + I_x u + I_y v - I(x, y)]^2 \quad (2.2)$$

$$E(u, v) \approx \sum_{x,y} w(x, y) [u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2], \quad (2.3)$$

kde  $I_x, I_y$  jsou derivace intenzity jasové složky obrazu ve směru  $x, y$ .



Obrázek 2.1: Základní princip Harrisova detektoru rohů

S užitím maticového zápisu píšeme

$$E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}, \quad (2.4)$$

kde

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \quad (2.5)$$

Dále zavedeme parametr

$$R = \det(M) - k (\text{trace}(M))^2 \quad (2.6)$$

$$R = \lambda_1 \lambda_2 - k (\lambda_1 - \lambda_2)^2, \quad (2.7)$$

který značí, jak výrazným rohem bod v obraze je.  $\lambda_i$  jsou vlastní čísla matice  $M$ ,  $k$  je volným parametrem Harrisova detektoru. Význam vlastních čísel  $\lambda_i$  pro detekci znázorňuje obr. 2.2a.

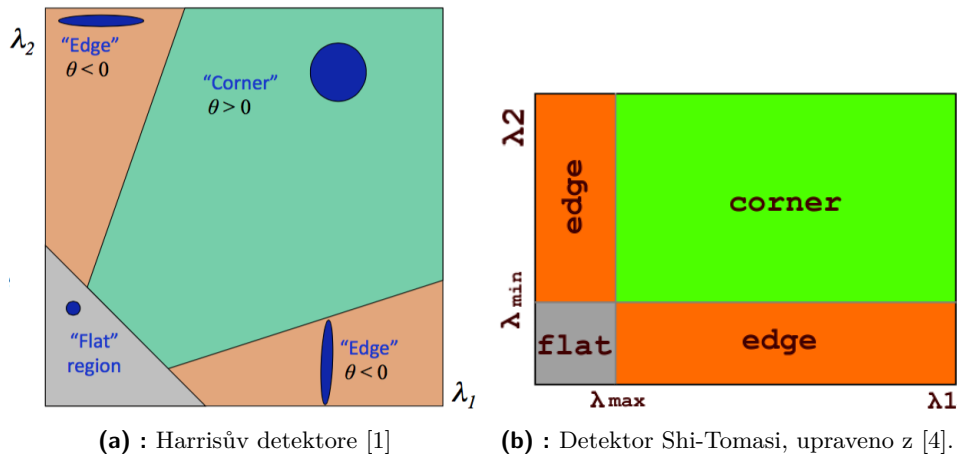
Výpočtem hodnoty  $R$  pro všechny pixely v obraze získáme matici hodnot, která udává, s jakou pravděpodobností je daný pixel rohem. Vhodnou volbou mezní hodnoty dostaneme seznam nalezených rohů v obraze.

## 2.2 Detektor Shi-Tomasi

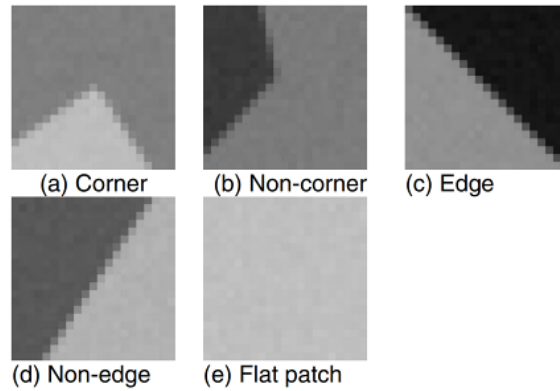
Algoritmus představený v článku *Good Features to Track* [19] vychází z podobného principu jako Harrisův detektor rohů. Liší se v definici parametru  $R$ . Místo výrazu (2.6) definuje parametr jako

$$R = \min \{ \lambda_1, \lambda_2 \}. \quad (2.8)$$

Vliv vlastních čísel na výsledek detekce  $\lambda_i$  je znázorněn na obr. 2.2b.



**Obrázek 2.2:** Význam vlastních čísel matice  $M$  z pohledu Harrisova detektoru rohů a detektoru Shi-Tomasi



**Obrázek 2.3:** Referenční vzorky rohových a nerohových struktur v testu [13]

## 2.3 Srovnání detektorů rohů

Komplexní test [13] srovnává výsledky Harrisova detektoru, detektoru Shi-Tomasi a několika dalších rohových detektorů na přibližně 120 000 vzorcích. Původní Harrisův detektor je hodnocen jako výpočetně nejrychlejší. Detektor Shi-Tomasi ovšem vykázal nižší počet *false positive* i *false negative* nálezů, tedy v nižším počtu případů došlo k nesprávnému vyhodnocení vzorku. Klíč, podle kterého byly vytvořeny referenční výsledky, je znázorněn na obr. 2.3.

Je nutné zmínit, že v systému *hetect* vyžadujeme detekci rohů šachovnice, které nebyly v testu [13] zahrnuty. Při srovnání obou detektorů v našich experimentech provedených na snímcích z testovacích letů však detektor Shi-Tomasi vykázal stejný nebo nižší počet nenalezených rohů (*false negative*) oproti Harrisově detektoru. Množství falešných nálezů (*false positive*) není pro naši aplikaci kritické.

## 2.4 Funkce `goodFeaturesToTrack`

`goodFeaturesToTrack` [4] je funkcí knihovny OpenCV implementující detektor rohů Shi-Tomasi. Pro úplnost uvádím, že funkce umožňuje volbu mezi Harrisovým detektorem a detektorem Shi-Tomasi.

Použití detektoru Shi-Tomasi nevyžaduje nastavení volného parametru  $k$ , který se v definici (2.8) nevyskytuje. Není také třeba hledat vhodnou prahovou hodnotu parametru  $R$ , funkce ji nastaví tak, aby byl detekován zadaný počet rohů. Tím je odstraněn problém vlivu osvětlení, kdy s pevně nastavenou prahovou hodnotou dochází k výraznému kolísání počtu detekovaných rohů. Na problém jsme narazili při testování funkce `conerHarris` obsažené v knihovně OpenCV.

Lze také omezit minimální vzdálenost mezi detekovanými rohy. Toto omezení zabraňuje četným detekcím rohů na složitém pozadí nebo označením jednoho skutečného rohu několika nálezy. Násobná detekce jednoho rohu bývá způsobena rozmazáním obrazu.



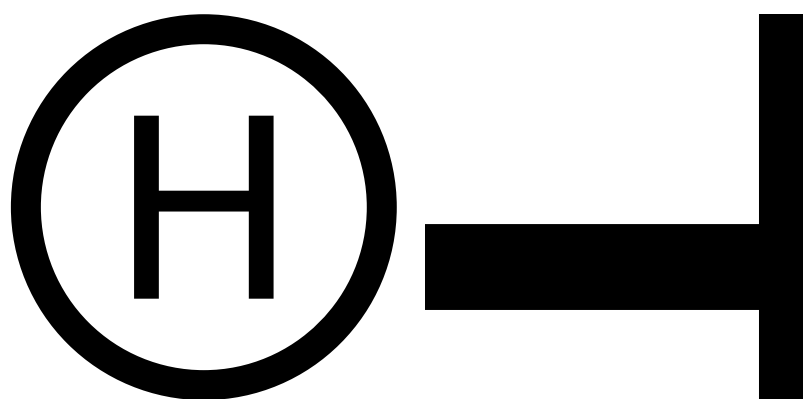
## Kapitola 3

### Návrh grafického vzoru pro přistávání helikoptéry

Při návrhu grafického přistávacího vzoru byly brány v potaz následující skutečnosti. Některé jsou částečně protichůdné, bylo třeba vytvářet kompromisy.

- Vzor musí být jasně rozlišitelný od běžně se vyskytujících útvarů, snažíme se minimalizovat riziko falešných detekcí.
- Vzor by neměl být zbytečně složitý, při pozorování z větší vzdálenosti by docházelo ke ztrátě detailů.
- Řízení helikoptéry v reálném čase vyžaduje co nejrychlejší zpracování snímků. Obrazec navrhujeme tak, aby byl co nejsnadněji rozpoznatelný.
- Symetrie přináší výhody. Nezáleží-li na orientaci vzoru (rotaci v rovině), detekce je snazší a jsou šetřeny výpočetní zdroje.
- Pro potlačení vlivu osvětlení je třeba užívat barvy s velkým kontrastem. Na přeexponovaných či podexponovaných snímcích se slévají podobné barevné odstíny a nejsme je schopni rozlišit.
- Zpracování obrazu v odstínech šedi je rychlejší. Vlastní barvy jsou navíc často silně zkresleny vlivem osvětlení. V kombinaci s požadavkem na vysoký kontrast je vhodné použít černobílý vzor.
- Cílem je maximalizovat rozpětí vzdáleností, ze kterých je možno vzor detekovat. Vzdálenost je shora omezena rozlišením obrazu, vlastnostmi použitého algoritmu a fyzickými rozměry vzoru, má-li být snadno zhotovitelný. Zdola je omezena schopností algoritmu zaměřit střed vzoru při velkém přiblížení kamery. Je tedy třeba, aby byl vzor identifikovatelný i v situaci, kdy není v zorném poli kamery celý.

Na obr. 3.1a je široce užívaný vzor pro heliport, se kterým pracují některé obdobné projekty [17]. Vzor je jednoznačně identifikovatelný, v terénu těžko



(a) : Vzor pro heliport celý v zorném poli kamery

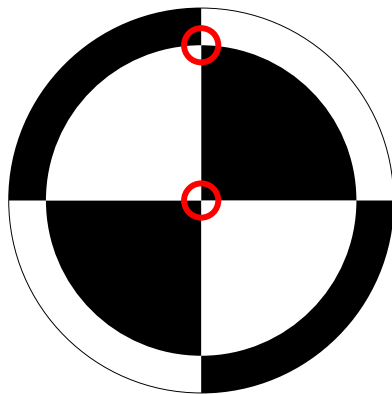
(b) : Vzor pro heliport zachycený při velkém přiblížení

**Obrázek 3.1:** Typický vzor používaný pro značení heliportu

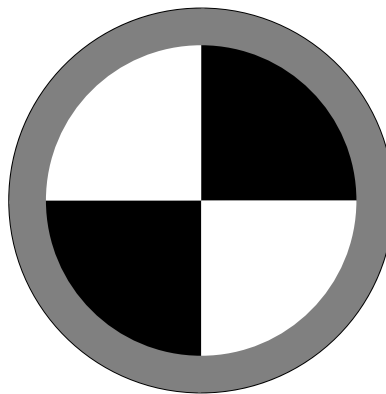
zaměnitelný a je možno jej detekovat v reálném čase [17]. Na obr. 3.1b je však vidět, že při větším přiblížení ztrácíme informaci o poloze středu obrazce. Tento fakt limituje rozsah pracovní vzdálenosti.

Vytvořili jsme několik návrhů vzoru s šachovnicovým rohem uprostřed, který netrpí zmíněným omezením. Abychom vzor odlišili od ostatních (náhodných) rohů šachovnice, je obklopen mezikružím s barvami kontrastními k barvám šachovnice. Na obr. 3.2a můžeme vidět vzor obsahující několik podobných klíčových bodů, které mohou vést k problémům v režimu velkého přiblížení. Algoritmus v režimu přiblížení nekontroluje přítomnost mezikruží a může dojít k záměně některého ze zvýrazněných míst za středový roh obrazce. Návrh na obr. 3.2b tento problém řeší, naráží však na problém nízkého kontrastu. Na snímcích v šeru splývá šedý okraj s černými částmi, na přímém slunci se naopak jeví jako bílý.

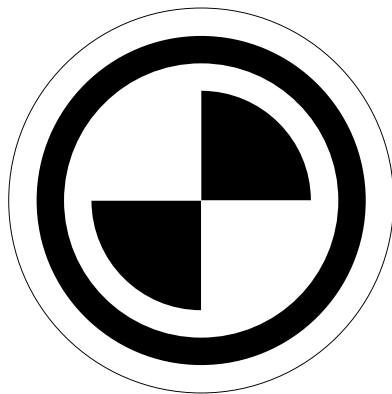
Výsledkem těchto úvah a řady praktických experimentů s prototypy (obr.3.2) je návrh na obr. 3.3. Obsahuje pouze jeden roh šachovnice a netrpí problémy se sníženým kontrastem. Jak je patrné z obr. 3.3b, při přibližování kamery ke vzoru nedochází ke ztrátě informace o poloze středu. Systém *hetect* používá k lokalitaci přistávacího místa právě tento návrh.



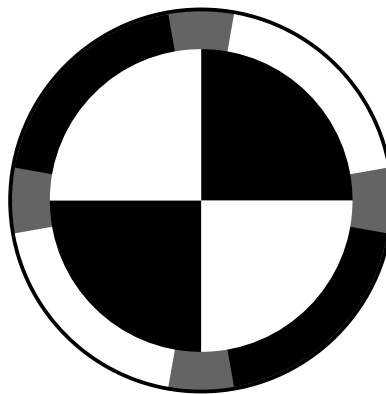
(a) : Návrh s několika rohy šachovnice



(b) : Návrh s nízkým kontrastem barev

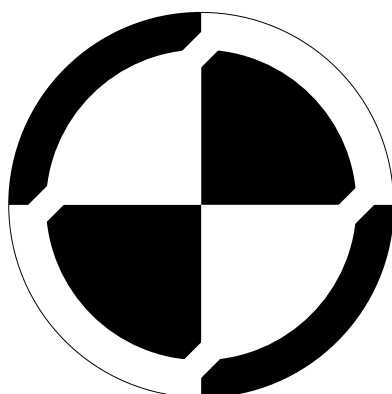


(c) : Návrh s malým vnitřním poloměrem

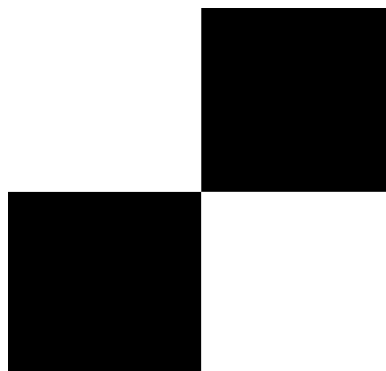


(d) : Návrh s lokálně sníženým kontrastem

**Obrázek 3.2:** Prototypy přistávacího vzoru s šachovnicovým rohem



(a) : Přistávací vzor celý v zorném poli kamery



(b) : Přistávací vzor při velkém přiblížení

**Obrázek 3.3:** Přistávací vzor používaný systémem *hetect*



## Kapitola 4

### Návrh algoritmu pro rozpoznávání přistávacího vzoru

Struktura detekčního algoritmu vychází z charakteru použitého grafického vzoru (obr. 3.3). Počítá se přitom se dvěma scénáři. Zpočátku je vzor od kamery vzdálen natolik, že je celý v zorném poli kamery a je možno jej detekovat jako celek. Dojde-li ke snížení vzdálenosti na kritickou mez, algoritmus přejde do režimu přiblížení a zaměřuje se pouze na střed obrazce.

Prohledávání snímku (v režimu oddálení) probíhá podle schématu na obr. 4.1. Po předzpracování snímku jsou detekovány rohy v obraze funkcí *goodFeaturesToTrack*. Z množiny detekovaných rohů je korelačním testem vybrána podmnožina rohů podobných rohům šachovnice. Na tuto podmnožinu je následně aplikován test kruhovitosti, který zjišťuje přítomnost kruhové hranice vnitřní části vzoru 3.3.

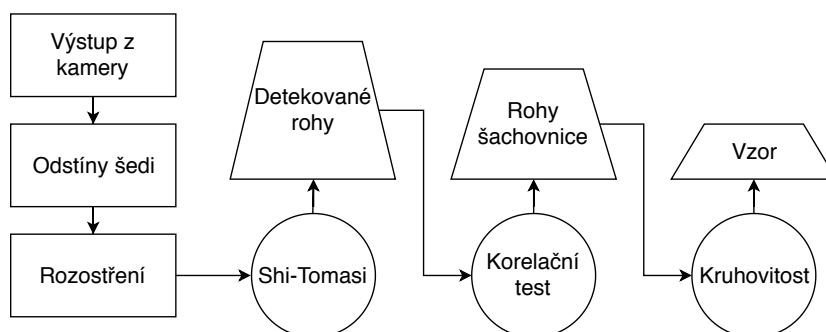
#### 4.1 Předzpracování obrazu

Systém *hetect* dokáže zpracovat snímky libovolného rozlišení, jejich velikost však má vliv na běh algoritmu. Většina experimentů uvedených v kapitole 5 je provedena na obrázcích o rozlišení  $752 \times 480$  (rozlišení kamery umístěné na testovací helikoptěře). V případě výrazně větších snímků je doporučeno přeškálování, omezí se tím negativní vliv na rychlost algoritmu.

Algoritmus v prvním kroku provádí převod barev do odstínů šedi. Funkce *goodFeaturesToTrack* užitá k detekci rohů vyžaduje jako vstup jednobarevný obrázek reprezentovaný maticí hodnot typu *uint8\_t*. Převod vykonává funkce *cvtColor* [6] knihovny OpenCV a probíhá podle předpisu

$$I = 0.299 I_R + 0.587 I_G + 0.114 I_B, \quad (4.1)$$

kde  $I$  je hodnota intenzity jasové složky každého pixelu spočítaná z barevných složek původního obrázku.



**Obrázek 4.1:** Průběh zpracování obrazu z kamery systémem *hetect*

Ve druhém kroku je snímek rozostřen funkcí *blur* [7] knihovny OpenCV. Rozostření zbaví obrázek šumu a náhlých změn jasu vedoucích k výskytu falešných detekcí rohů. Probíhá podle předpisu

$$O_b = (O * K)(x, y), \quad (4.2)$$

kde  $*$  je operátor 2D konvoluce,  $O_b$  rozostřený obrázek,  $O$  původní obrázek a  $K$  konvoluční jádro. V tomto případě je použito jádro

$$K = \frac{1}{S_K} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad (4.3)$$

$S_K$  značí součet prvků matice  $K$ .

V praxi se často setkáváme s tzv. gaussovským rozostřením, jehož jádro  $K$  obsahuje hodnoty odpovídající normálnímu rozdělení pravděpodobnosti. V knihovně OpenCV jej implementuje funkce *GaussianBlur* [7]. Použita nebyla z důvodu vyšší výpočetní náročnosti.

## 4.2 Detekce rohů v obraze

Po předzpracování snímků následuje krok detekce rohů detektorem *Shi-Tomasi* implementovaným ve funkci *goodFeaturesToTrack* knihovny OpenCV. Princip detektoru je popsán v kapitole 2.4. Detekce rohů nám poskytuje prvotní informaci o tom, kde má smysl snímek prozkoumávat a hledat přistávací vzor. Selže-li detekce rohu ve středu obrazce, selže celý detekční algoritmus.

Zásadními parametry ovlivňujícími robustnost a rychlost algoritmu jsou maximální počet detekovaných rohů a minimální vzdálenost mezi rohy. Příliš nízký limit minimální vzdálenosti mezi rohy může vést k násobným detekcím jednoho rohu (za nepříznivých světelných podmínek a na rozmazaných

snímcích). Příliš velký limit může naopak zapříčinit situaci, ve které nebude roh uprostřed hledaného vzoru detekován vůbec. A to ve chvíli, kdy se v jeho blízkosti nacházejí rohy s vyšší hodnotou parametru  $R$  (popsaného v kapitole 2).

Protože jsou pro všechny rohy iterativně ověřovány další vlastnosti, velký počet detekovaných rohů zpomaluje běh algoritmu. V prostředí, kdy je vzor umístěn na jednoduším podkladu, stačí k jeho nalezení zhruba 30 rohů. Objeví-li se ve snímku složitější struktury, je třeba navýšit počet hledaných rohů řádově 10x.

Systém *hetect* nastavuje oba parametry dynamicky. Počet hledaných rohů se při negativní detekci navyšuje, při pozitivní detekci je postupně snižován. Po několika vyhodnocených snímcích začne oscilovat okolo ideální hodnoty pro dané prostředí. V režimu přiblížení dojde ke skokovému navýšení limitu minimální vzdálenosti mezi rohy, jelikož víme, že se v blízkosti hledaného rohu žádný jiný roh nacházet nebude.

### ■ 4.2.1 Zpřesnění pozice rohu

Vlivem nepříznivého osvětlení nebo rozmazání obrazu může nastat situace, kdy je jeden roh v obraze detekován dvakrát. Násobné detekci je možné zabránit nastavením minimální vzdálenosti mezi rohy ve funkci *goodFeaturesToTrack*. Poté je detekován pouze jeden ze dvou zdánlivých rohů, ten ovšem nekoresponduje se středem obrazce, jak ukazuje obr. 4.2.

Jelikož algoritmus *hetect* detekuje cíleně pouze rohy šachovnice, můžeme využít jejich středové symetrie. Definujme faktor asymetrie kolem bodu  $(s_x, s_y)$  jako

$$A(s_x, s_y) = \sum_{i=1}^n \left| I(x_i, y_i) - I(\bar{x}_i, \bar{y}_i) \right|, \quad (4.4)$$

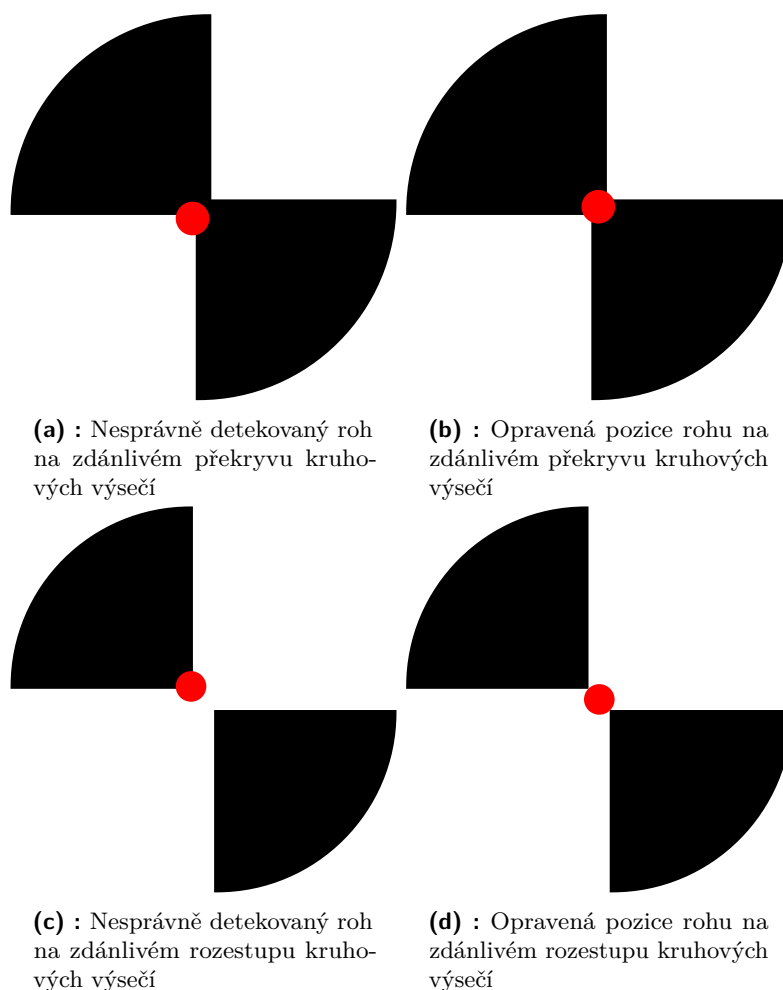
kde  $I(x, y)$  je hodnota jasové složky obrazu v bodě o souřadnicích  $(x, y)$ . Přitom platí, že body  $(x_i, y_i)$  a  $(\bar{x}_i, \bar{y}_i)$  jsou umístěny na kružnici se středem v bodě  $(s_x, s_y)$  a spojnice každé dvojice bodů  $(x_i, y_i)$  a  $(\bar{x}_i, \bar{y}_i)$  prochází středem  $(s_x, s_y)$ . Do výpočtu je zahrnuto  $n$  dvojic bodů.

Opravenou pozici rohu potom nalezneme jako

$$\arg \min_{(s_x, s_y)} \left( A(s_x, s_y) \right), \quad (4.5)$$

kde za  $s_x$  a  $s_y$  dosadíme souřadnice pixelů z daného okolí nepřesně detekovaného rohu.

Zjednodušeně řečeno, pozici každého rohu upravíme tak, abychom posílili středovou symetrii v jeho blízkosti.



Obrázek 4.2: Chyby v detekci středového rohu obrazce na rozostřených snímcích

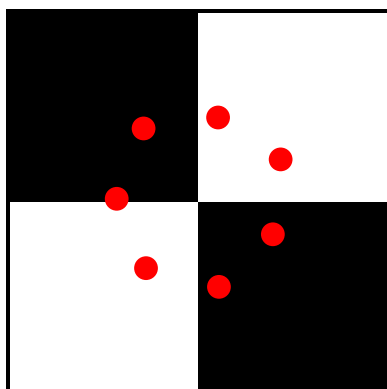
### 4.3 Korelační test na určení rohů šachovnice

Známe množinu všech rohů nalezených v obraze. Nyní iterativně ověříme, které její prvky jsou podobné rohům šachovnice. To nám umožní většinu rohů vyloučit a posílit znalost o možné poloze hledaného vzoru. Další, výpočetně náročnější testy už budeme vykonávat pouze na množině rohů šachovnice. Detekujeme-li v obraze desítky až stovky rohů, rohům šachovnice se podobají (za běžných podmínek) pouze jednotky z nich.

#### 4.3.1 Princip korelačního testu

Každý z rohů vyšetříme následujícím způsobem. Okolo rohu vytyčíme kružnici a zjistíme intenzitu jasové složky v několika bodech ležících na kružnici. Hodnoty uložíme do pole a budeme kontrolovat, zda odpovídají šachovnicové





**Obrázek 4.3:** Ilustrace náběru hodnot intenzity jasové složky v (červených) bodech na kružnici okolo detekovaného rohu

posloupnosti. Způsob náběru hodnot je znázorněn na obr. 4.3. Ve skutečnosti používáme 48 referenčních bodů.

Označíme posloupnost nabraných hodnot jako  $c$  a zavedeme vzorovou posloupnost  $p$  ve tvaru

$$p = (0, \dots, 0, 255, \dots, 255, 0, \dots, 0, 255, \dots, 255), \quad (4.6)$$

která obsahuje popořadě čtvrtinu černých pixelů, čtvrtinu bílých, černých, a zase bílých. Ideální roh šachovnice odpovídá vzorové posloupnosti  $p$  nebo jejímu posunu, který vzniká natočením obrazce v rovině.

Vzájemný posun  $s$  posloupností  $c$  a  $p$  získáme korelační funkcí. Cyklickým posunem posloupnosti  $p$  se snažíme dosáhnout co nejvyšší podobnosti obou posloupností. Hledáme tedy

$$\arg \max_s \sum_{i=0}^n (c_i \cdot p_{i+s}) \quad (4.7)$$

s tím, že přesáhne-li index  $(i + s)$  délku posloupnosti  $n$ , použije se zbytek po dělení  $(i + s) \pmod{n}$ , čímž přeteče na začátek posloupnosti.

Nyní známe orientaci potenciálního vzoru v rovině, zbývá ještě ověřit, jak přesně odpovídá ideálnímu rohu šachovnice. Za tímto účelem spočítáme *střední* hodnotu  $m_c$  posloupnosti  $c$  jako

$$m_c = \frac{1}{2} (\max(c) + \min(c)). \quad (4.8)$$

Postup byl zvolen namísto výpočetně náročnějšího mediánu. Metoda se může zdát náchylná vůči šumu (náhodné hodnotě jednoho pixelu), toto riziko je však potlačeno rozostřením obrázku ve fázi předzpracování. Podobný problém nebyl při testování zaznamenán.

*Střední hodnota*  $m_c$  nám umožní rozdělit prvky posloupnosti  $c$  na černé a bílé. Pro každý prvek je ověřena platnost logického výrazu

$$(c_i \geq m_c \wedge p_{i+s} \geq m_c) \vee (c_i < m_c \wedge p_{i+s} < m_c). \quad (4.9)$$



**Obrázek 4.4:** Deformace vzoru při šikmém pohledu

Tedy zjistíme, jestli jsou černé a bílé pixely na místech, kde by v případě ideálního rohu šachovnice měly být.

Hranice mezi černou a bílou se dynamicky mění podle světelných podmínek. Na přeexponovaných snímcích může mít černá barva hodnotu jasové složky nad naivní hranicí 128, na podexponovaných může naopak hodnota bílé barvy klesnout pod 128. Přesto nedojde ke zmatení algoritmu.

### ■ 4.3.2 Vliv vzájemné polohy kamery a obrazce

V dosavadních úvahách jsme předpokládali, že je vzor snímán kamerou kolmo shora, tedy že se nachází přímo pod dronem. To v reálném provozu samozřejmě neplatí a musíme algoritmus přizpůsobit situacím, kdy bude vzor snímán zešikma. Na obr. 4.4 můžeme vidět, že při šikmém snímání dochází k deformaci obrazce. Pravoúhlé trojúhelníky se nejeví jako pravoúhlé a přestává platit poměr rozložení bílých a černých pixelů v posloupnosti  $p$ .

Algoritmus vyžaduje, aby byla podmínka (4.9) splněna pro daný počet prvků posloupnosti  $c$ . Z této statistiky vyjmeme body ležící poblíž očekávané hranice kvadrantů. Vznikne tím tolerance, která umožňuje detekovat i rohy šachovnice snímané z šikmého pohledu.

### ■ 4.3.3 Potlačení falešných detekcí

Výše je popsána naivní implementace algoritmu, který ověří podobnost vzorové posloupnosti  $p$  s posloupností  $c$  získanou z jediné kružnice okolo vyšetřovaného rohu. Takto navržený algoritmus produkuje nepřijatelné množství falešných detekcí (*false positive*). Systém *hetect* proto provádí korelační test na několika kružnicích o různém poloměru a až poté rozhoduje, zda je daný roh rohem šachovnice.

K dosažení pozitivního výsledku testu je zapotřebí, aby roh splnil princip naivního algoritmu na daném počtu poloměrů. Neměli bychom však zapomenout, že máme k dispozici ještě údaj o pravděpodobném natočení obrazce, neboli hodnotu  $h$  z rovnice (4.7). Můžeme si tedy dovolit přidat ještě požadavek na malou změnu tohoto ukazatele mezi jednotlivými poloměry.

Poloměry vyhodnocujeme od nejmenšího po největší, v blízkosti středu obrazce lze očekávat vysokou nejistotu určování parametru  $h$ . Jakmile se s narůstajícím poloměrem jeho hodnota ustálí, můžeme začít zaznamenávat, na kolika poloměrech byl splněn naivní test. Body na kružnici navíc umísťujeme tak, aby mezi jednotlivými poloměry neležely v přímce a pokryly co největší plochu obrázku.

Zbývá odpovědět na otázku, jak volit množinu použitých poloměrů. Známe pravděpodobné natočení obrazce, nemáme však představu o jeho vzdálenosti od kamery, respektive poloměru. Zpočátku nám nezbývá než prohledávat tak blízko rohu, aby mohly být detekovány i vzdálené obrazce. Nastavení poloměrů je jedním z faktorů, které limitují minimální velikost detekovatelného obrazce.

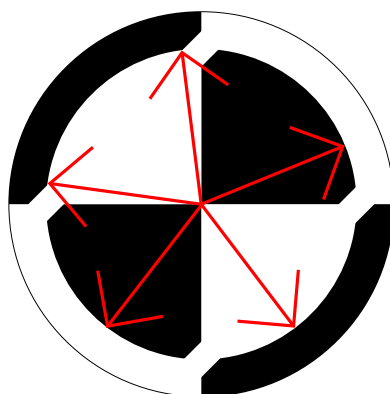
Situace se mění ve chvíli, kdy dojde k pozitivní detekci přistávacího vzoru. Informaci o jeho velikosti můžeme využít pro vyhodnocování následujících snímků. Lze předpokládat, že se mezi jednotlivými snímky změní jen minimálně. Systém *hetect* potom nastavuje použité poloměry dynamicky a eliminace falešných detekcí je ještě účinnější. Dojde-li ke ztrátě vzoru, vrací se k původní strategii.

## ■ Funkce `findChessboardCorners`

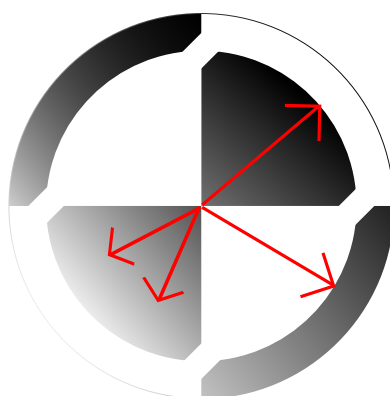
V této kapitole se sluší poznamenat, že knihovna OpenCV obsahuje funkci `findChessboardCorners` určenou přímo k detekci rohů šachovnice. Bývá hojně užívaná např. pro účely kalibrace kamer. Funkce metodou segmentace hledá jednotlivá políčka, které dává do souvislosti a skládá z nich pravděpodobnou podobu šachovnicového pole [9], dokáže ovšem pracovat s šachovnicovými poli o minimálních rozměrech  $4 \times 4$  políčka. Pro naši aplikaci proto nemohla být použita.

## ■ 4.4 Ověření přítomnosti kruhové hranice

Známe množinu bodů v obraze, které se jeví jako rohy šachovnice. Na ní si můžeme dovolit aplikovat výpočetně náročnější testy spočívající v ověření přítomnosti kruhové hranice vnitřní části vzoru 3.3. Měly by vést k finálnímu nalezení přistávacího vzoru, pokud je v obraze přítomen.



**Obrázek 4.5:** Ilustrace způsobu prohledávání metodou odstředných paprsků



**Obrázek 4.6:** Příklad pozvolné změny jasu na přistávacím vzoru s použitím metody naivní komparace jasu

#### ■ 4.4.1 Metoda odstředných paprsků

Střed domnělého obrazce je jedna z mála informací, kterou v této fázi máme. Začneme tedy prohledávat obrázek od středu obrazce k okrajům, jak znázorňuje obr. 4.5, a pokusíme se detekovat kontrastní přechod z černé barvy na bílou (nebo naopak).

##### ■ Základní myšlenka metody

Mohli bychom postupovat ve směru každé šipky z obr. 4.5, uložit si počáteční hodnotu intenzity jasové složky blízko středu a pokračovat, dokud nenalezneme pixel, jehož jasová složka je dostatečně vzdálena výchozí hodnotě. Přístup ovšem selže v situacích s pozvolnou změnou jasu, jak ilustruje obr. 4.6, kdy je prohledávání zastaveno předčasně.

Opačný přístup, sledování derivace jasu, tedy rozdílu intenzit sousedních pixelů ve směru paprsku, zase naráží na problém se stabilitou. Derivace jasu je zatížena výrazným šumem a dochází k náhodnému ukončení prohledávání.

Systém *hetect* pracuje s klouzavým průměrem  $I_a(k)$  intenzity jasové složky  $I(k)$  definovaným jako

$$I_a(k+1) = \frac{\alpha_1 I_a(k) + \alpha_2 I(k+1)}{\alpha_1 + \alpha_2}, \quad (4.10)$$

kde  $k$  je pořadí navštívení daného pixelu ve směru paprsku. Konstanty  $\alpha_1$  a  $\alpha_2$  určují *hloubku paměti* klouzavého průměru, experimentálně byly nastaveny na hodnoty  $\alpha_1 = 6$  a  $\alpha_2 = 1$ .

V průběhu postupu kontrolujeme hodnotu jasu tří následujících (ještě neprozkoumaných) pixelů a jejich rozdíl od aktuální hodnoty klouzavého průměru. Je-li výraz

$$\left| I_a(k) - \frac{1}{3} (I(k+1) + I(k+2) + I(k+3)) \right| \quad (4.11)$$

větší, než mezní hodnota, narazili jsme na hranici s ostrou změnou jasové složky a prohledávání ve směru daného paprsku je zastaveno.

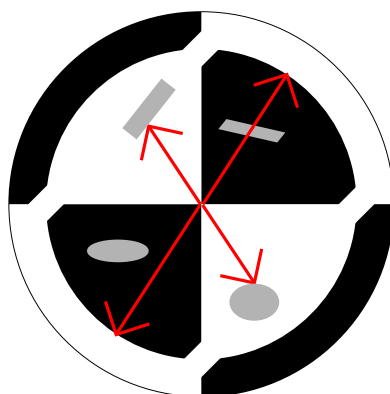
Nalezení vhodné mezní hodnoty není triviálním úkolem. Vlivem osvětlení se může rozdíl mezi černou a bílou barvou významně měnit. Využíváme proto informací získaných korelačním testem (kapitola 4.3.1). Mezní hodnota je nastavována dynamicky na základě rozsahu detekovaných jasů, odpovídá průměru hodnot  $\max c$  a  $\min c$  z rovnice (4.8).

#### ■ Potlačení vlivu ostrých stínů

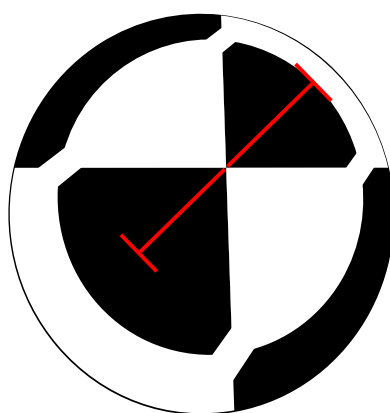
Je-li přistávací vzor rovnoměrně osvětlen, metoda funguje spolehlivě. Poradí si také s plynulou změnou jasu. Objeví-li se však v obrazci ostrý stín, algoritmus ho bez dalších úprav nedokáže odlišit od hledaného přechodu. Systém *hetect* využívá k řešení tohoto problému středové symetrie grafického vzoru. Prohledávání probíhá pro každý směr souběžně na obě strany od středu. K detekci hranice je vyžadována vysoká hodnota výrazu (4.11) na obou stranách.

Narazí-li paprsek na jedné straně na stín, prohledávání není předčasně ukončeno a pokračuje do chvíle, kdy na kontrastní přechod nedojde i druhý paprsek. Je-li obrazec zarušen stínem stejně vzdáleným od středu na obou stranách, dojde k předčasnému zastavení paprsku i s použitím této metody. Situaci ilustruje obr. 4.7.

Pokud bychom mohli předpokládat, že bude obrazec na snímcích dokonale středově symetrický, nic by nám nebránilo nastavit toleranci délky protilehlých paprsků přísně. Pravděpodobnost, že budou rušivé elementy na obou stranách ležet v totožné vzdálenosti od středu, je velmi malá. Úspěšnost potlačení vlivu stínů by tak byla vysoká. Na reálných snímcích se však projevují různé typy zkreslení. Dochází k deformaci zobrazovací soustavou kamery na okrajích zorného pole, při náklonu helikoptéry je obrazec zkreslen perspektivou.



**Obrázek 4.7:** Přistávací vzor zarušený ostrými stíny nebo jinými rušivými elementy s kontrastním přechodem



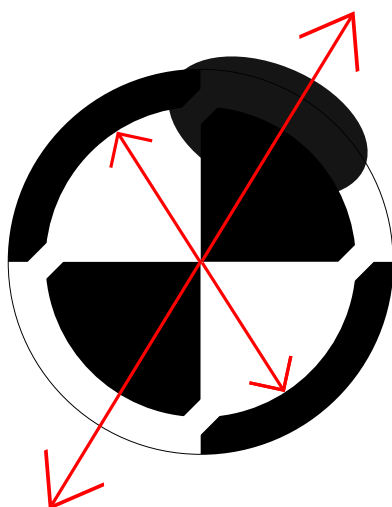
**Obrázek 4.8:** Pohled na zkreslený přistávací vzor s narušenou středovou symetrií

Důsledkem těchto jevů přestává být středová symetrie striktní a je nutno zvýšit toleranci na délku protilehlých paprsků, viz obr. 4.8. To částečně snižuje účinnost opatření a ostré stíny mohou narušit funkci systému *hetect*.

Ostré stíny nebo jiné rušivé elementy se mohou nacházet také na okraji obrazce a zakrýt hledaný přechod. Jak je vidět na obr. 4.9, v takovém případě nedojde k zastavení paprsku na správném místě. Tuto vlastnost metody jsme ochotni akceptovat za cenu zlepšení výsledků v situacích z obr. 4.7. Experimenty ukazují, že přechod zastíněný na velké části okraje by stejně ve většině případů znemožnil nalezení vzoru.

#### ■ Dvoustupňové užití metody dostředných paprsků

Máme k dispozici souřadnice bodů, které výše popsaná metoda označila jako hraniční, tedy ležící na vnitřní kružnici obrazce. Můžeme předpokládat, že část bodů není lokalizovaná správně. Buď selhaly metody potlačení vlivu stínů nebo došlo k chybnému zastavení paprsků jedním z problémů znázorněných na obr. 4.10. Ilustrace situaci zveličují, demonstrují však časté potíže zazname-



**Obrázek 4.9:** Přistávací vzor s ostrým stínem nebo jiným rušivým elementem zakrývajícím hledaný přechod z černé barvy na bílou

nané při testování. Obr. 4.10a ilustruje komplikace na hranici černého a bílého kvadrantu, obr. 4.10b předčasné zastavení prohledávání vlivem zdánlivého rozestoupení černých částí.

Systém *hetect* použije množinu délek všech nabraných paprsků k nastavení horního i dolního odhadu vzdálenosti hledaného okraje od středu. S touto znalostí spustí metodu odstředných paprsků podruhé. Vynutí přitom, aby nedošlo k zastavení prohledávání před spodním odhadem vzdálenosti a zároveň nepokračovalo nad její horní odhad. Získáme tak nové souřadnice bodů s potlačením vlivů z obr. 4.10 a menším rozptylem délek paprsků.

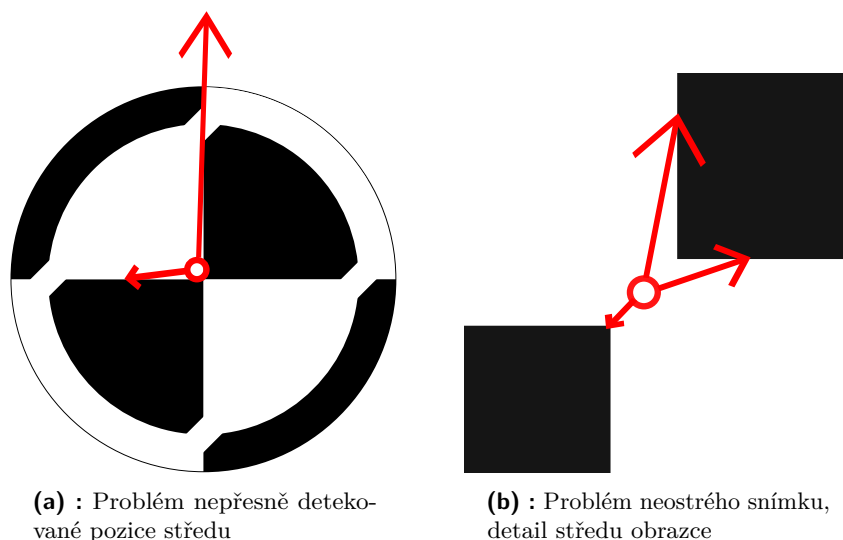
#### ■ 4.4.2 Proklad hraničních bodů elipsou

Nacházíme se uprostřed procesu vyšetřování jednoho z rohů v obraze. Víme, že se jedná o roh šachovnice. Známe souřadnice bodů tvořících kontrastní hranici kolem rohu. Zbývá ověřit, zda hraniční body získané metodou odstředných paprsků formují kružnici. Formují-li, můžeme s vysokou mírou pravděpodobnosti prohlásit, že je vyšetřovaný roh středem přistávacího obrazce.

Je důležité si připomenout, že přistávací vzor není vždy snímán přímo shora, jak ukazuje obr. 4.4. Kružnice se na snímač kamery promítá jako elipsa. V důsledku toho nebudeme provádět proklad bodů kružnicí, ale elipsou.

#### ■ Testování spojitosti délek paprsků

Před provedením samotného prokladu se snažíme hraniční body co nejefektivněji roztrždit a vyloučit extrémny. Výsledek prokladu může být silně ovlivněn



**Obrázek 4.10:** Příklady chybně zastavených paprsků v rizikových situacích

jen malým počtem bodů ležících daleko od kruhové hranice.

Víme, že se poloměr elipsy  $r$  v závislosti na úhlu  $\varphi$  mění podle předpisu

$$r = \sqrt{(a \cos(\varphi))^2 + (b \sin(\varphi))^2}, \quad (4.12)$$

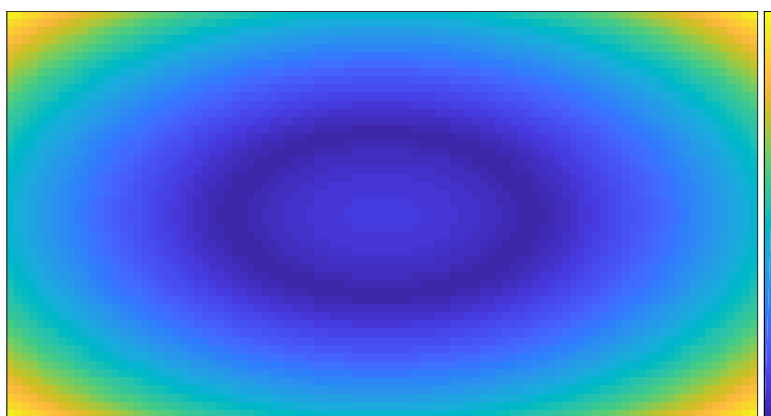
tedy spojitě. Zásadní pro nás je, že délka validních paprsků se též mění spojitě. Pro dané vzorkování (počet paprsků) a maximální předpokládané zkreslení obrazce (poměr délek poloos  $a$  a  $b$ ) lze nalézt horní odhad rozdílu délek sousedních paprsků. Mezi nabranými paprsky hledá systém *hetect* úseky, v nichž jsou všechny změny pod tímto horním odhadem. Předpokládáme, že body náležící těmto úsekům leží na hledaném kruhovém přechodu. Tím jsou z prokladu elipsou vyřazeny body, které s vysokou pravděpodobností nenesou užitečnou informaci.

### ■ Průběh vlastního prokladu

Známe množinu bodů, které podle dosavadních zjištění leží na kruhové hranici vnitřní části přistávacího vzoru. Funkcí *fitEllipse* [8] knihovny OpenCV provedeme proklad bodů elipsou. Funkce implementuje aproximaci ve smyslu nejmenších čtverců metodou *LIN* popsanou v článku [11].

Výstupem funkce *fitEllipse* je obdélník opsaný nalezené elipse. Z obdélníku vyčteme souřadnice středu  $(x_0, y_0)$ , délky hlavní a vedlejší poloosy  $a$  a  $b$ , úhel





**Obrázek 4.11:** Hodnota ukazatele  $\Delta$  v závislosti na poloze v souřadném systému.

$\Theta$  mezi poloosou  $a$  a souřadnou osou  $x$  a sestavíme rovnici elipsy ve tvaru

$$\frac{\left((x - x_0) \cos(\Theta) + (y - y_0) \sin(\Theta)\right)^2}{a^2} + \frac{\left((x - x_0) \sin(\Theta) + (y - y_0) \cos(\Theta)\right)^2}{b^2} - 1 = 0. \quad (4.13)$$

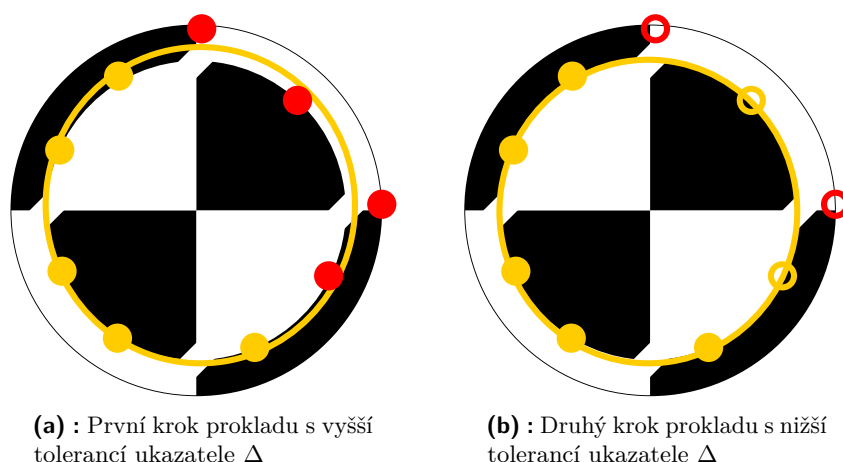
Nyní pro každý hraniční bod rozhodneme, zda leží v blízkosti získané elipsy, či nikoli. Úlohu vzdálenosti bodu od elipsy není možno řešit analyticky, metody numerických výpočtů jsou shrnuty v publikaci [10]. Z důvodu šetření výpočetních zdrojů žádnou z metod nevyužijeme, spokojíme se s dosazením bodů do levé strany rovnice (4.13).

Nepracujeme s přesnou vzdáleností bodu od křivky, získáme ukazatel  $\Delta$  s jistou vypovídající hodnotou o tomto údaji. Rozložení hodnot ukazatele v souřadném systému je znázorněno na obr. 4.11. Výpočet byl proveden na elipse se středem v počátku a délkou poloos  $a = 4$ ,  $b = 2$ .

## Metoda dvoustupňového prokladu

Nyní bychom mohli zjistit počet hraničních bodů, pro něž hodnota ukazatele  $\Delta$  nepřekročila danou mez, a na základě počtu bodů rozhodnout, zda se jedná o přistávací vzor. Přes všechna dosavadní opatření se však může stát, že je získaná elipsa ovlivněná několika odchýlenými body, a obrazec kvůli nim nebude detekován. Systém *hetect* proto provádí dvoustupňový proklad.

Metodu dvoustupňového prokladu ilustruje obr. 4.12. V prvním kroku jsou hraniční body vytříděny s vyšší tolerancí ukazatele  $\Delta$ . Červenými kolečky v obr. 4.12a značíme body, které toleranci přesáhly. Následuje druhý krok, kdy jsou k prokladu použity pouze body, jež toleranci splnily. Na novou elipsu je poté uplatněna nižší tolerance ukazatele  $\Delta$ . V obr. 4.12b jsou červeně



**Obrázek 4.12:** Princip dvoustupňového prokladu elipsou

vyznačeny body, které nespĺnily striktnější test na hodnotu ukazatele. Kolečka s prázdným středem reprezentují body nepoužité k prokladu v druhém kroku. I ty byly ale v druhém kroku testovány a některé z nich testem prošly (žluté s prázdným středem).

V příkladu z obr. 4.12 bylo 7 bodů z 9 označených za blízké elipse. Při použití jednoduchého prokladu by to bylo pouze 5 z 9, a to i za použití mírnějšího testu na ukazatel  $\Delta$ .

#### ■ Potlačení falešných detekcí

Dalším stupněm ověření schopným zabránit části falešných detekcí je kontrola souřadnic středu získané elipsy. Ten by neměl být vzdálen od souřadnic detekovaného rohu. I zde je zavedena tolerance kvůli perspektivě a jiným deformacím.

Klesne-li velikost vyšetřovaného obrazce na jednotky pixelů, vzroste prudce vliv všech odchylek a nepřesností zmiňovaných v průběhu celé práce a vzorkovací chyba znemožní přesný výpočet ukazatele  $\Delta$ . Je proto uměle zavedena minimální velikost vzoru, který je možno systémem *hetect* nalézt. Elipsy, jejichž poloosy nemají délku alespoň 8 obrazových pixelů, jsou ignorovány.

Roh, jenž projde všemi fázemi detekčního algoritmu s pozitivním výsledkem, lze nyní s velkou pravděpodobností označit za hledaný přistávací vzor.

### ■ 4.5 Využití informací mezi jednotlivými snímky

Podaří-li se v obraze přistávací vzor nalézt, získáme náhle řadu informací, které mohou usnadnit zpracování následujících snímků. Snímáme-li kamerou

scénu kontinuálně, můžeme využít znalosti pozice a rozměrů vzoru k urychlení běhu algoritmu a ke zvýšení jeho spolehlivosti.

### ■ Adaptivní počet hledaných rohů

Jedním z parametrů, který systém *hetect* nastavuje dynamicky, je maximální počet hledaných rohů v obraze. Než dojde k pozitivnímu nálezu obrazce, je limit pro počet rohů nastaven na hodnotu 250. Po úspěšné detekci je limit postupně snižován, v případě ztráty vzoru dojde ke skokovému navýšení. Na snímcích bez složitého pozadí (hrubý koberec, síťovina, koruny stromů) osciluje limit okolo hodnoty 30 rohů.

### ■ Maskování snímku a odhad polohy středu obrazce

Při kontinuálním snímání se poloha středu obrazce mezi jednotlivými snímky mění jen omezeně. Stačí proto hledat rohy pouze v okolí posledního pozitivního nálezu. Funkce *goodFeaturesToTrack* umožňuje nastavení masky pro prohledávání. Zmenšení zkoumané oblasti vede k nižšímu počtu nalezených rohů a zvyšuje rychlost algoritmu.

Vliv na velikost změny pozice má především snímkovací frekvence a kvalita regulačního systému helikoptéry. Testování systému *hetect* probíhalo na snímcích pořízených při manuálně řízeném letu. Velikosti masky  $M$  je stanovena předpisem

$$M = \alpha r + \beta. \quad (4.14)$$

Při letu ve vyšší výšce dochází k menšímu posunu obrazu vlivem translačního pohybu helikoptéry, proto je ve výpočtu zahrnut poloměr vzoru z předchozího snímku  $r$ . Konstanta  $\beta$  nese význam třesu obrazu, který není závislý na vzdálenosti od vzoru. Hodnoty  $\alpha$  a  $\beta$  byly zjištěny experimentálně.

Pokud není vzor detekován v okolí posledního kladného nálezu, následující snímek je opět prohledán celý.

### ■ Změna parametrů korelačního testu

Korelační test na nalezení rohu šachovnice je prováděn na několika poloměrech kolem každého rohu. Jak popisuje kapitola 4.3.3, poloměry musí být natolik malé, aby umožnily nalezení vzoru ve velké vzdálenosti od kamery. To má negativní vliv na počet falešných detekcí. Známe-li však rozměr obrazce z předchozích snímků, přizpůsobíme tomu poloměry korelačního testu. Tím zlepšíme odolnost vůči vlivům z obr. 4.2 a výrazně omezíme počet falešných rohů šachovnice, tedy snížíme výpočetní náročnost algoritmu.

## ■ Režim velkého přiblížení

Zásadní vlastností systému *hetect* je schopnost detekovat přistávací vzor i v situaci, kdy není celý v zorném poli kamery. Klesne-li helikoptéra při letu natolik, že poloměr obrazce přesáhne třetinu zorného pole, dojde k přepnutí do režimu velkého přiblížení. V tomto režimu neprobíhá ověřování přítomnosti kruhové hranice obrazce a pro detekování vzoru stačí přítomnost rohu šachovnice.

Aby bylo možné navádět helikoptéru přesně na střed, musí být detekován právě jeden roh šachovnice v každém snímku. Jsou proto zvětšeny poloměry, na kterých je prováděn korelační test. Tím je výrazně sníženo riziko, že dojde k detekci několika rohů šachovnice v jednom snímku, a zároveň je zvýšena odolnost proti rozostření (viz obr. 4.2). Velikost masky  $M$  z rovnice (4.14) je nastavena na fixní hodnotu, jelikož ztrácíme informaci o poloměru obrazce.

## ■ 4.6 Odhad relativní pozice helikoptéry vůči přistávacímu vzoru

Známe obrazové souřadnice středu detekovaného vzoru, známe i polohu hraničních bodů na vnitřní kružnici obrazce. Pro účely navádění helikoptéry a řízení přistávacího manévru bychom potřebovali získat informaci o poloze helikoptéry v prostoru.

Podle modelu dírkové kamery [2] (se zanedbáním radiálního a tangenciálního zkreslení) je bod  $(X, Y, Z)$  ve světovém souřadném systému transformován do souřadného systému kamery zobrazením

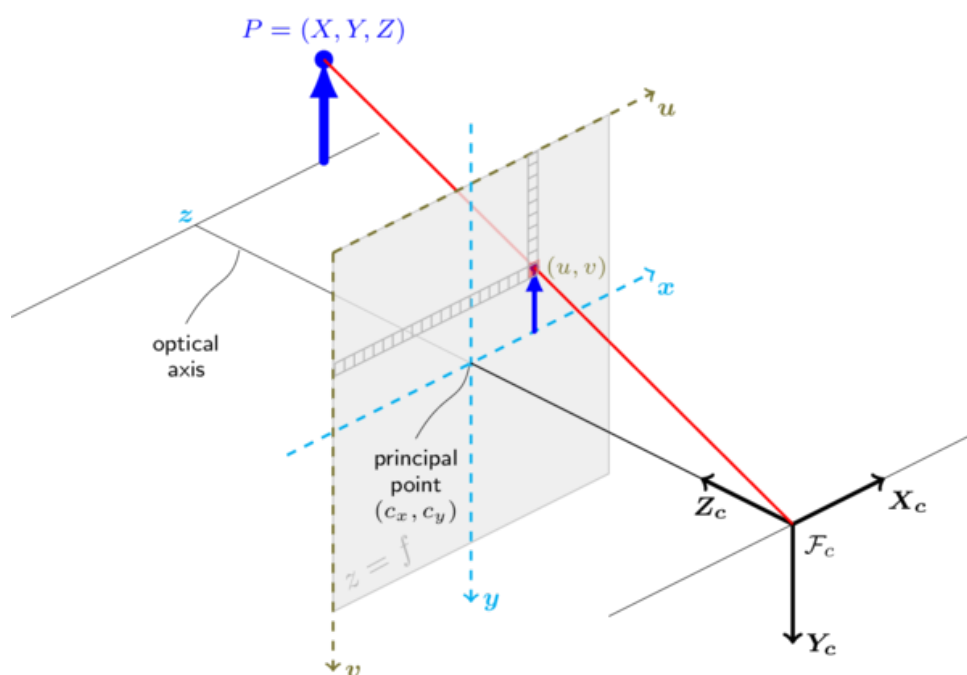
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \quad (4.15)$$

kde  $r_{ij}$  jsou prvky rotační matice a  $t_i$  prvky translačního vektoru. A následně je promítnut na plochu snímače a pixel o souřadnicích  $(u, v)$  jako

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (4.16)$$

Přitom  $f_x$  a  $f_y$  jsou ohniskové vzdálenosti čočky,  $(c_x, c_y)$  je optický střed a  $s$  je parametr závislý na vzdálenosti bodu od čočky. Způsob projekce ilustruje obr. 4.13.

Pracujeme se zkalibrovanou kamerou, u níž známe parametry  $f_x, f_y, (c_x, c_y)$ . Známe i souřadnice pixelů zmíněných bodů, hodnotu vektoru  $(u, v)$ . Umístíme-li počátek světového souřadného systému do středu přistávacího vzoru a



**Obrázek 4.13:** Zobrazení bodu v prostoru na rovinu snímače pomocí modelu dírkové kamery [2]

definujeme-li rozměry obrazce, známe i polohu bodů  $(X, Y, Z)$ . Naším cílem je získat translační vektor  $t$  a rotační matici  $R$ , které reprezentují relativní polohu kamery vůči přistávacímu vzoru.

Tento typ úlohy, kdy provádíme rekonstrukci polohy objektu v prostoru z obrazových souřadnic, se nazývá *PnP problem*. K jejímu řešení byla použita funkce *solvePnP* [2] knihovny OpenCV. Ta implementuje několik metod řešení problému, v systému *hetect* je užita metoda *CV\_ITERATIVE* založená na Levenberg-Marquardtově optimalizaci. Funkce iterativně minimalizuje druhou mocninu chyby projekce zadaných bodů.

Vstupními parametry funkce jsou (kromě dalších) matice kamery ve tvaru

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4.17)$$

a koeficienty radiálního a tangenciálního zkreslení  $k_i$  a  $p_i$ . Tyto parametry je nutno nastavit na základě použité kamery.

Jako referenční body do funkce zadáváme střed obrazce a několik bodů na obvodu vnitřní kružnice přistávacího vzoru. Obrazové souřadnice bodů na obvodu kružnice spočítáme z parametrů proložené elipsy jako

$$\begin{bmatrix} u \\ v \end{bmatrix} = S + a \cos(\varphi) \begin{bmatrix} \cos(\Theta) \\ \sin(\Theta) \end{bmatrix} + b \sin(\varphi) \begin{bmatrix} -\sin(\Theta) \\ \cos(\Theta) \end{bmatrix}, \quad (4.18)$$

kde  $S$  je střed elipsy,  $a$  a  $b$  délky poloos,  $\Theta$  rotace elipsy v souřadném systému a  $\varphi \in \langle 0; 2\pi \rangle$ . Souřadnice korespondujících bodů v prostoru jsou

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R \begin{bmatrix} \cos(\varphi) \\ \sin(\varphi) \\ 0 \end{bmatrix}, \quad (4.19)$$

kde  $R$  je poloměr vnitřní kružnice přistávacího vzoru a  $\varphi \in \langle 0; 2\pi \rangle$ . Výstupem funkce *solvePnP* je hledaný translační vektor  $t$  a rotační matice  $R$ .

Z proložené elipsy jsme tedy schopni získat odhad relativní polohy helikoptéry v prostoru. V režimu velkého přiblížení známe pouze obrazové souřadnice středu obrazce, proklad elipsou neprobíhá a ztrácíme přehled o výšce helikoptéry.

## Kapitola 5

### Ověření funkčnosti systému *hetect*

V této kapitole jsou shrnuty výsledky testů provedených na systému *hetect* za různých podmínek. Ukazují omezení systému, popisují situace, v nichž systém prokázal funkčnost, a situace, ve kterých mohou nastat komplikace. Obrazový materiál je k dispozici na přiloženém CD. Dodány jsou vstupní i zpracované snímky a soubory *log.txt*, ve kterých lze dohledat odhad polohy helikoptéry, informace o době zpracování snímku nebo použitém limitu maximálního počtu rohů. Vybrané obrázky jsou k prohlédnutí na konci práce v příloze A.

#### 5.1 Ověření jednotlivých fází detekčního algoritmu

Obr. A.1 zachycuje jednotlivé fáze vyhodnocování snímku. Systém *hetect* je do jisté míry odolný vůči narušení přistávacího vzoru stíny, proto byl vybrán obrázek, který tuto vlastnost dobře demonstruje. Zobrazeny jsou originály obrázků před rozmazáním a převodem do černobílého spektra.

V obr. A.1a jsou vyznačeny všechny rohy nalezené detektorem Shi-Tomasi po zpřesnění pozice rohů (viz kapitola 4.2.1). Limit maximálního počtu nalezených rohů byl nastaven na hodnotu 150.

Na obr. A.1b můžeme vidět průběh korelačního testu, jemuž je věnována kapitola 4.3. Bílé kružnice kolem všech rohů odpovídají poloměrům, na kterých vyšel test podobnosti s rohem šachovnice pozitivně. Dva rohy zvýrazněné zeleně byly vyhodnoceny jako rohy šachovnice, test podobnosti u nich vyšel pozitivně na potřebném počtu poloměrů. V tomto případě evidujeme jeden falešný nález rohu šachovnice. Obrázek byl vyhodnocen bez znalosti očekávaného poloměru obrazce. Pokud bychom ho zpracovali v režimu kontinuálního snímání, korelační test by byl účinnější, jak dokazuje obr. A.1d, a k falešnému nálezu by nedošlo.

Na oba domnělé rohy šachovnice je aplikována metoda odstředných paprsků, vysvětlená v kapitole 4.4.1. Z obr. A.1c je zjevné, že nedošlo k předčasnému za-

stavení prohledávání při styku paprsku se stínem. U části paprsků zafungovala metoda založená na středové symetrii obrazce (viz kapitola 4.4.1). U ostatních paprsků je hranice stínu pozvolná a nedošlo k naplnění podmínky (4.11), nebo nastal styk paprsku se stínem příliš blízko středu a nebyl překročen minimální odhad délky paprsku při aplikaci metody dostředných paprsků v druhém kroku. Kombinací zmíněných opatření se podařilo získat platnou polohu hraničních bodů.

Hraniční body jsou ve dvou krocích proloženy elipsou, postup je podrobně popsán v kapitole 4.4.2. Na obr. A.1e je zakreslena elipsa po prvním prokladu, na A.1f je výsledná elipsa. Červené body jsou elipse příliš vzdáleny a nesplnily limit parametru  $\Delta$ . Elipsa kolem falešně detekovaného rohu šachovnice nemá minimální požadovaný rozměr a byla ignorována.

Intenzita jasové složky v oblasti vrženého stínu spadala do intervalu pro černou barvu, výsledkem je přesto korektně detekovaný přistávací vzor.

## 5.2 Detekce vzoru za nepříznivého osvětlení

Cílem projektu bylo navrhnout detekční algoritmus tak, aby se byl schopen vyrovnat s řadou nepříznivých vlivů, např. s proměnlivým osvětlením. Na obr. A.2 jsou snímky pořízené kamerou dronu při přistávání. Během zlomku sekundy se skokově změnila expozice a hodnota jasové složky černé barvy se dostala do intervalu, ve kterém byla na předchozích snímcích barva bílá. Systém situaci vyhodnotil správně díky adaptivnímu nastavování hranice mezi černou a bílou. Vzor byl na obou snímcích korektně detekován.

Na obr. A.3 jsou snímky záměrně poškozeného přistávacího vzoru. V případě překrytí části kruhové hranice na obr. A.3a dojde k chybné lokalizaci několika hraničních bodů. Díky testu na spojitost délek paprsků (kapitola 4.4.2) nejsou použity pro proklad elipsou a obrazec je detekován správně.

Na obr. A.3b byl překryt střed přistávacího vzoru, nemohl tak být detekován středový roh, na jehož nalezení závisí chod celého algoritmu. Podobné situace částečně řeší provádění korelačního testu (kapitola 4.3.1) na velkých poloměrech. S jejich použitím může být detekován jako roh šachovnice i jeden z rohů, který vznikl umístěním nálepky a neplatí pro něj striktně středová symetrie. Test je však spuštěn v režimu nezávislého vyhodnocování každého snímku, systém tedy nemá odhad rozměrů hledaného obrazce a korelační test probíhá pouze v malém okolí rohu.

Další příklad z části zastíněného obrazce poskytuje obr. A.4. Přistávací vzor byl správně detekován v případě, kdy se ostrá hrana stínu nenacházela ve stejné vzdálenosti od středu na protilehlých stranách obrazce. To je v souladu s principy použité metody na překonávání stínů popsané v kapitole 4.4.1.



limit počtu rohů	čas expozice	počet snímků	úspěšně detekováno	falešné detekce	průměrný čas zpracování [ms]
50	auto	19	15	0	47,18
150	auto	19	17	0	64,70
800	auto	19	17	0	220,73
150	zkrácený	19	19	0	47,03

**Tabulka 5.1:** Úspěšnost detekce vzoru na různém pozadí

## 5.3 Vliv prostředí na výsledek detekce

### 5.3.1 Lokalizace vzoru na různém pozadí

Bylo zhotoveno 22 fotografií vzoru umístěného na různých typech podlahy, venkovní dlažby, koberec a trávníku. Snímky jsou pořízeny kamerou v telefonu *Nokia 7 Plus* a zmenšeny na rozlišení  $1280 \times 720$  pixelů.

Každý snímek byl vyhodnocen samostatně, bez předchozí znalosti rozměrů nebo pozice vzoru a s pevným limitem pro maximální počet rohů. Simulujeme situaci, kdy helikoptéra vletí do neprozkoumaného prostředí. Test byl spuštěn s různým nastavením limitu pro maximální počet rohů, zpracované snímky jsou na CD v oddělených složkách. V praxi bychom provozovali systém *hetect* v režimu kontinuálního snímání, kdy systém nastavuje limit počtu rohů automaticky, aby našel vzor a zároveň optimalizoval výpočetní náročnost.

Výsledky testu shrnuje tabulka 5.1. Přistávací vzor se podařilo detekovat ve všech 22 testovaných prostředích. Příliš nízký limit maximálního počtu rohů způsobil potíže na obrázcích se složitým pozadím, kdy nebyl nalezen roh ve středu přistávacího vzoru. Vzor umístěný na mřížovou strukturu nebo řídký trávník vyžadoval snížení času expozice kamery. Na přesvětlených záběrech se vyskytovalo velké množství rohů s vyšším kontrastem, než hledaný roh šachovnice, viz obr. A.5.

### 5.3.2 Cílené umístění rohu šachovnice do záběru kamery

V prvním testu bylo na podlahu vedle přistávacího vzoru umístěno šachovnicové pole. Kamera se pohybovala tak, že do záběru nejprve vstoupil přistávací vzor, který byl úspěšně detekován. Na dalších záběrech se začíná objevovat šachovnicové pole, které nemá žádný vliv na průběh detekce. Systém prohledává snímek pomocí masky pouze v okolí poslední platné detekce a zbylé rohy šachovnice nejsou vůbec detekovány, situaci demonstruje obr. A.6.

Ve chvíli, kdy hledaný vzor opustí zorné pole, je prohledán opět celý snímek. Systém nalezne všechny šachovnicové rohy, díky testu na přítomnost kruhové

počet snímků	limit počtu rohů	falešné detekce	průměrný čas zpracování [ms]
39	150	0	52,26

**Tabulka 5.2:** Výsledky zpracování snímků bez přítomnosti přistávacího vzoru

hranice však nedojde k žádné falešné detekci.

V dalším testu je snímáno šachovnicové pole, které se postupně vzdaluje od kamery. Na prvních snímcích dojde ke stejné situaci jako u prvního testu. Šachovnicové rohy jsou nalezeny, nejsou však označeny za hledaný vzor. Při takovém oddálení, kdy velikost políček šachovnice klesne na 13 pixelů, už systém není schopen rozlišit čtvercová políčka od kružnice. Vzorkovací chyba při takto nízkém rozlišení vede k nesprávnému výsledku testu na přítomnost kruhové hranice a jsou zaznamenány falešné detekce přistávacího vzoru.

Použité fotografie i vyhodnocené snímky jsou umístěny na CD ve složce *sachovnice*.

### 5.3.3 Zpracování snímků bez přítomnosti vzoru

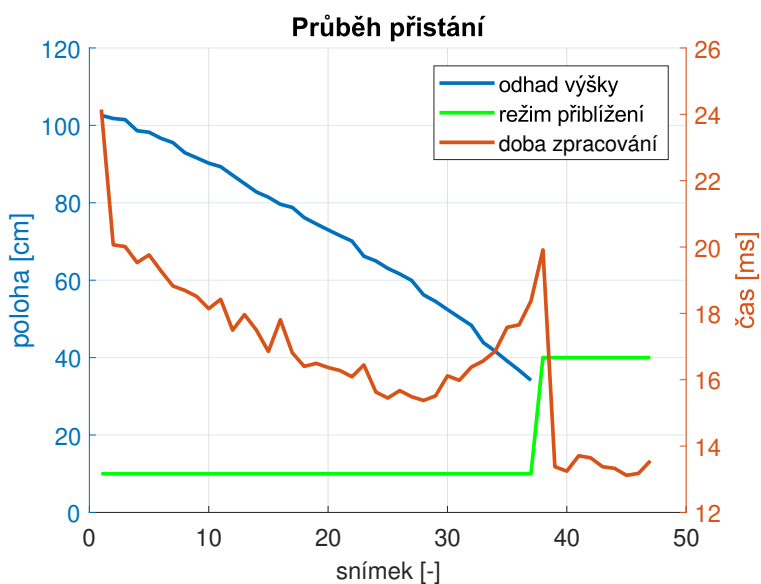
Náchylnost systému k výskytu falešných detekcí byla testována na sadě 39 stažebých obrázků [5] o rozlišení přibližně  $1280 \times 728$  zmenšených před zpracováním na 0,7 násobek původních rozměrů. Sada obsahuje letecké snímky a záběry z dronů z rozmanitého prostředí, příklad na obr. A.7. Nebyla zaznamenána žádná falešná detekce, výsledky testu shrnuje tabulka 5.2. Testovací sada i zpracované snímky jsou k dispozici na přiloženém CD ve složce *bez\_vzoru*.

## 5.4 Vyhodnocení snímků pořízených za letu

Následující test ověřuje schopnost systému detekovat přistávací vzor na snímcích pořízených kamerou připevněnou na helikoptěře. Helikoptéra byla řízena manuálně. Obrázky o rozlišení  $752 \times 480$  pixelů byly snímány s průměrnou frekvencí 25 snímků za sekundu, ukládány na paměťové úložiště a následně zpracovány na počítači s procesorem *Intel Core i5-8250U*. Vyhodnocování dat přímo na výpočetní jednotce dronu a současné ukládání obrázků by snímací frekvenci snížilo.

### 5.4.1 Dosednutí helikoptéry na přistávací vzor

V rámci testovacích letů bylo simulováno 6 dosednutí helikoptéry na přistávací vzor. Manuálně řízený dron provedl přiblížení k vyznačenému místu a následně



**Obrázek 5.1:** Graf průběhu dosednutí helikoptéry

dosedl na zem. Cílem bylo otestovat přechod systému do režimu velkého přiblížení a lokalizaci středu obrazce v tomto režimu.

Ve všech 6 pokusech došlo ke korektnímu přepnutí do režimu přiblížení. Na obr. A.11a je snímek, na němž se obrazec v jednom z pokusů přiblížil na kritickou vzdálenost potřebnou pro přepnutí. Na následujících snímcích algoritmus vyhledával pouze roh šachovnice bez ověřování přítomnosti kruhové hranice. Obr. A.11b zachycuje střed přistávacího vzoru po dosednutí helikoptéry na zem.

V grafu 5.1 můžeme pozorovat vyhodnocený záznam průběhu jednoho z přistání helikoptéry. Na počátku sekvence se čas zpracování snímku pohybuje v intervalu 20 až 24 ms. Díky využití maskování a adaptivnímu limitu pro maximální počet rohů dochází postupně k urychlování procesu. S přibližujícím se vzorem však začne narůstat plocha, kterou je třeba prozkoumat metodou odstředných paprsků, z toho důvodu po 20. snímku sekvence začne čas výpočtu opět narůstat. Ke skokovému urychlení dojde s přechodem systému do režimu přiblížení, obraz je zpracováván s periodou mezi 12 a 14 ms. Ve stejnou chvíli systém ztrácí informaci o výšce helikoptéry.

Tabulka 5.3 obsahuje statistiku úspěšnosti detekce. V 6 pokusech bylo nasbíráno 206 snímků, na 201 z nich byl přistávací vzor nalezen. Při prvním pokusu došlo ke chvilkové ztrátě kontroly nad helikoptérou, která vyústila v prudký manévr. Přistávací vzor se mezi jednotlivými snímky posunul o větší vzdálenost, než předpokládá velikost masky pro detekci rohů. Na dalších záběrech byl obraz z kamery rozmazaný. Z těchto důvodů nebyl vzor na 4 snímcích nalezen. V pátém pokusu byl detekován falešný roh šachovnice v režimu přiblížení, proto nebyl na jednom snímku identifikován střed vzoru.

pokus	počet snímků	úspěšně detekováno	falešné detekce	průměrný čas zpracování [ms]
1	35	31	0	19,37
2	37	37	0	18,05
3	19	19	0	19,90
4	33	33	0	19,52
5	35	34	1	18,22
6	47	47	0	18,08
celkem	206	201	1	-

**Tabulka 5.3:** Úspěšnost detekce vzoru při přistání

### 5.4.2 Snímání vzoru z větší výšky

Abychom ověřili funkčnost systému ve větší vzdálenosti než těsně před dosednutím, byl proveden komplexní testovací let. V průběhu letu bylo pořízeno 1626 snímků. Dron vzlétal z přistávacího místa. Jakmile nabral výšku a přistávací vzor se dostal do záběru kamery, byl úspěšně detekován. Systém přešel do režimu maskování, tedy využíval polohy a rozměrů posledního detekovaného obrazce pro vyhodnocování dalších snímků. V obrázcích po zpracování je prohledávaná oblast (maska) znázorněna červeným čtvercem. Originální záběry z helikoptéry i zpracované snímky jsou k dispozici na přiloženém CD ve složce *zkusebni\_let*.

Část nasbíraných snímků je silně rozmazaná z důvodu prudkých manévrů a dlouhého času expozice kamery. To nám umožňuje ukázat limity detekčního algoritmu. Na některých obrázcích není přistávací vzor vůbec přítomen. Můžeme proto pozorovat přechod systému mezi prohledáváním celého snímku, dojde-li ke ztrátě vzoru, a režimem maskování. V průběhu letu se helikoptéra vzdálila od přistávacího vzoru natolik, že délky poloos proložené elipsy klesly pod 8 pixelů tvořících limit rozlišovací schopnost systému *hetect*.

Výsledky testu jsou zaznamenány v tabulce 5.4. Příčiny nenalezení obrazce dále upřesňuje tabulka 5.5, ze které vyplývá, že většina selhání byla způsobena dlouhou dobou expozice a rozmazáním obrazu. Vzorky jsou na základě subjektivního dojmu roztrženy na *rozmazané*, u kterých dochází ke ztrátě kontrastu, ale obrazec je v nich okem jasně rozpoznatelný, a *extrémně rozmazané*, u nichž je přistávací vzor natolik znehodnocený, že nemůžeme očekávat jeho nalezení. Schopnost algoritmu vyhodnotit rozmazaný obraz demonstruje obr. A.9.

Sloupec *nad limitem vzdálenosti* v tabulce 5.4 značí stav, kdy velikost vzoru klesla pod hranici rozlišovací schopnosti systému. Na obr. A.8 je první snímek, na kterém došlo ke ztrátě detekce vlivem oddálení. Průběh polohy helikoptéry odhadnutý rekonstrukcí projekce přistávacího vzoru na snímač kamery je vyneseno do grafu 5.2. Z grafu můžeme vyčíst maximální výšku

-	celkem snímků	úspěšně detekováno	nad limitem vzdálenosti	nepřítomnost vzoru	nenalezen
počet	1626	1030	311	235	50
podíl	100%	63%	19%	14%	3%

**Tabulka 5.4:** Úspěšnost detekce vzoru v rámci komplexního testovacího letu

-	celkem nenalezeno	extrémně rozmazané	rozmazané	ostatní selhání	falešné detekce
počet	50	28	12	8	2
podíl z nenalezených	100%	56%	24%	16%	4%
podíl celkem	3%	2%	0,7%	0,5%	0,1%

**Tabulka 5.5:** Příčiny nenalezení vzoru v rámci komplexního testovacího letu

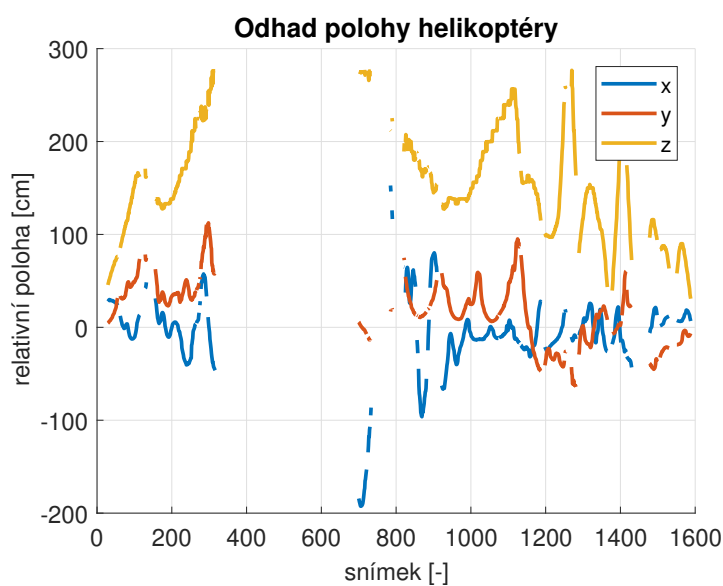
helikoptéry 277 cm, ve které byl ještě obrazec detekován. Poloměr vnitřní kružnice použitého vzoru je 14,7 cm.

Rozmazáním obrazu, tvarem listu papíru a dílem náhody došlo v průběhu testovacího letu k jedné falešné detekci. Situaci zachycuje obr. A.10. Nesprávně detekovaný vzor značí tučná fialová elipsa.

Z výsledků testu je možné vyčíst také informace o rychlosti zpracování obrazových dat. Čas potřebný pro vyhodnocení jednotlivých snímků je vynesena v grafu 5.3. Na snímcích, kde je detekován přistávací vzor, osciluje doba zpracování mezi 10 ms a 20 ms. Na snímcích bez přítomnosti vzoru (nebo na snímcích, kde nebyl detekován) narůstá doba zpracování ke 40 ms a výše. Statistická data jsou obsažena v tabulce 5.6. Srovnání výpočetní náročnosti detekce rohů v obraze s ostatními kroky algoritmu poskytuje graf 5.4. Z grafu vyplývá, že čas spotřebovaný detektorem rohů je relativně konstantní. Doba potřebná pro vykonání ostatních kroků algoritmu závisí na úspěšnosti detekce a hodnotě limitu maximálního počtu rohů. V ideálních případech je přistávací vzor mezi všemi rohy identifikován již za 2 ms.

## 5.5 Odhad vzdálenosti helikoptéry od vzoru

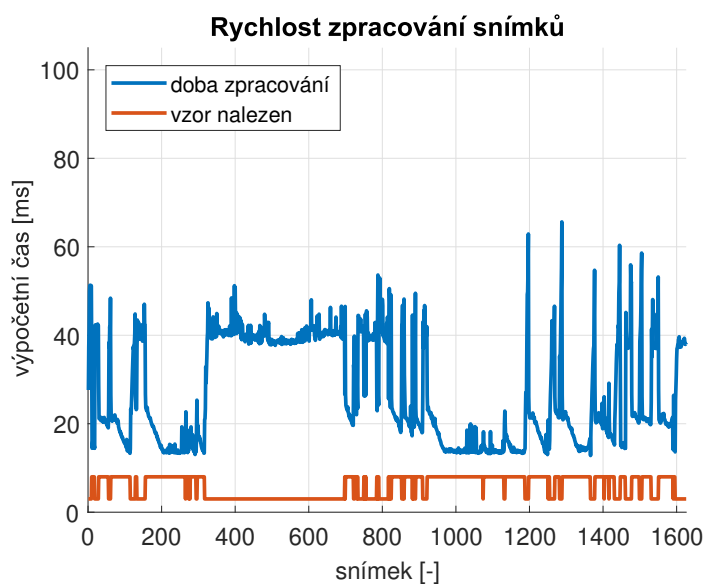
Součástí systému *hetect* je odhad relativní polohy helikoptéry vůči přistávacímu místu. Provedli jsme test, jehož cílem bylo ověření přesnosti odhadu vzdálenosti od lokalizovaného vzoru. Kamera helikoptéry byla umístěna do bodu o známých souřadnicích. S vizuálním vzorem jsme se pohybovali v zorném poli kamery a současně se snímky zaznamenávali polohu vzoru zjištěnou externím zaměřovacím systémem. Obrázky byly následně zpracovány systémem *hetect*, výsledky odhadu prezentuje graf 5.5.



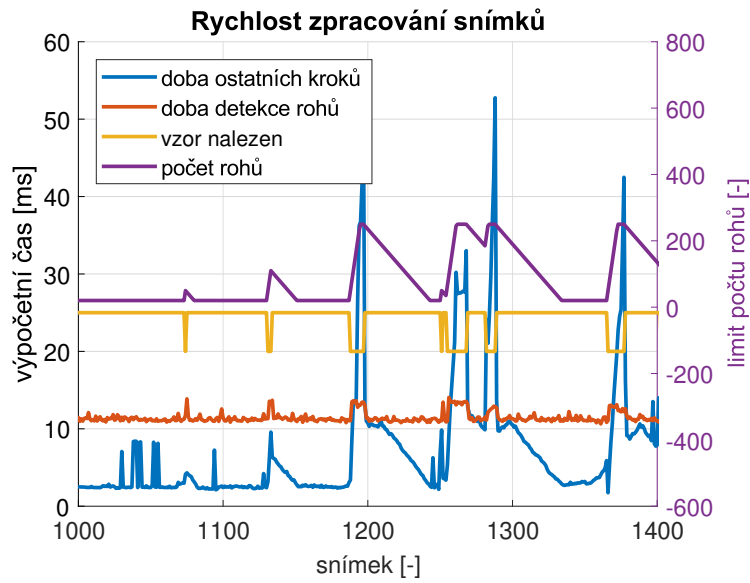
**Obrázek 5.2:** Odhad relativní polohy dronu v průběhu komplexního testovacího letu

průměrný čas zpracování [ms]	vzor detekován	vzor nenalezen
detekce rohů	11,32	13,05
ostatní části algoritmu	6,44	26,33

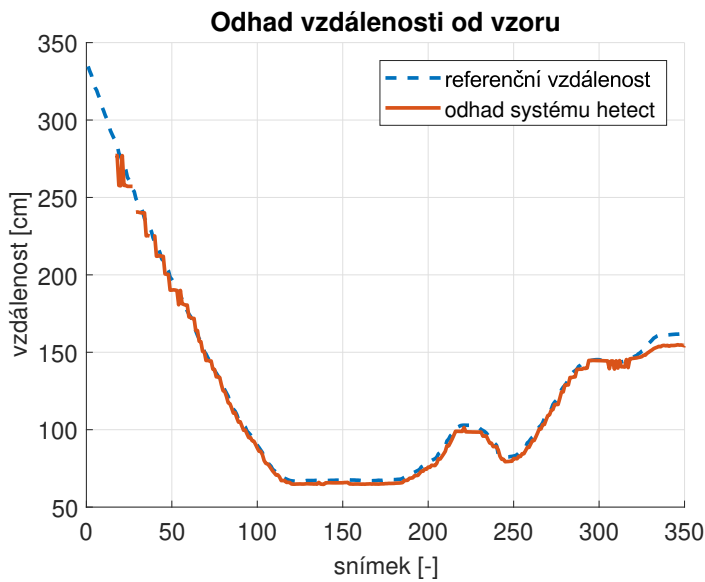
**Tabulka 5.6:** Výpočetní čas spotřebovaný jednotlivými částmi detekčního algoritmu v průběhu komplexního testovacího letu



**Obrázek 5.3:** Doba zpracování snímků v průběhu testovacího letu



Obrázek 5.4: Poměr výpočetní náročnosti jednotlivých fází detekce



Obrázek 5.5: Odhad vzdálenosti kamery od přistávacího vzoru





## Kapitola 6

### Závěr

Cílem práce bylo navrhnout vizuální vzor vhodný pro značení přistávacího místa helikoptéry a vyvinout algoritmus schopný detekovat tento vzor v obraze. Důraz byl kladen na schopnost algoritmu zaměřit střed vzoru i v situaci, kdy není celý v zorném poli kamery. Zároveň bylo snahou umožnit nalezení částečně poškozeného přistávacího vzoru, zejména vlivem stínů.

Po prostudování příbuzných projektů a různých přístupů k problému jsme navrhli několik prototypů vizuálního vzoru s šachovnicovým rohem uprostřed. Provedli jsme řadu praktických experimentů a vybrali nejvhodnější vzor. Výsledkem práce je lokalizační systém *hetect* schopný nalézt vzor v obrazových datech a odhadnout relativní polohu helikoptéry vůči přistávacímu místu.

Systém *hetect* je navržen pro vyhodnocování dat v reálném čase. Interní parametry algoritmu jsou nastavovány dynamicky, aby zajistily spolehlivost systému a zároveň optimalizovaly rychlost běhu. V situacích, které to umožňují, je prohledáváno pouze okolí předpokládané polohy vzoru.

Experimenty v laboratorních podmínkách i testy na datech pořízených za letu prokázaly schopnost systému detekovat přistávací vzor a odhadnout relativní polohu helikoptéry v různém prostředí a za různých podmínek osvětlení. Zaznamenané problémy s detekcí způsobovalo především nastavení času expozice kamery. V některých případech se podařilo detekovat i vzor narušený stíny a dalšími vlivy. Algoritmus se ukázal být odolný vůči falešným detekcím v testu na leteckých fotografiích rozmanitého terénu. Problémy způsobilo šachovnicové pole, které systém při velkém oddálení není schopen rozlišit od hledaného vzoru. Situaci by řešilo navýšení minimálních rozměrů detekovatelného vzoru, zároveň bychom tím ale omezili pracovní rozsah systému.

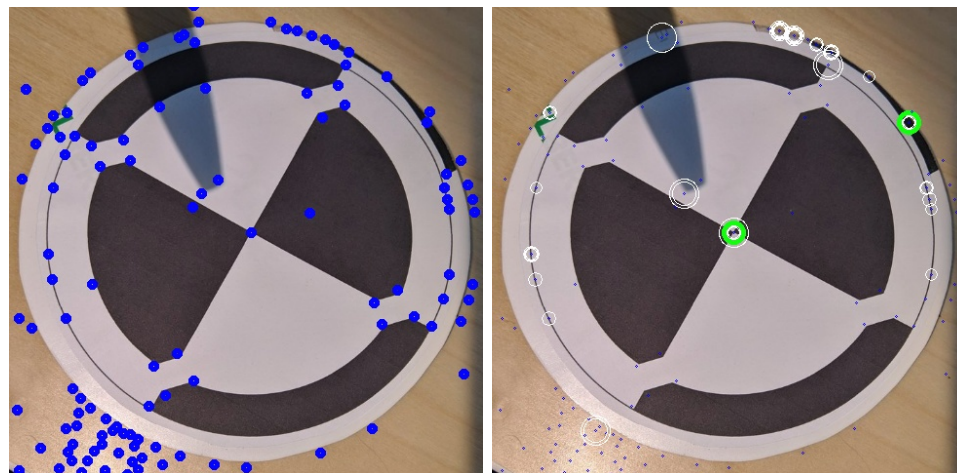
Důsledně ověřeno bylo chování algoritmu při dosedání helikoptéry na vyznačené místo. V okamžiku přiblížení k vzoru na určenou vzdálenost přešel systém do režimu přiblížení a úspěšně detekoval vzor i v situaci, kdy nebyl celý v zorném poli kamery. V ojedinělých případech detekce selhala z důvodu

rozmazání obrazu.



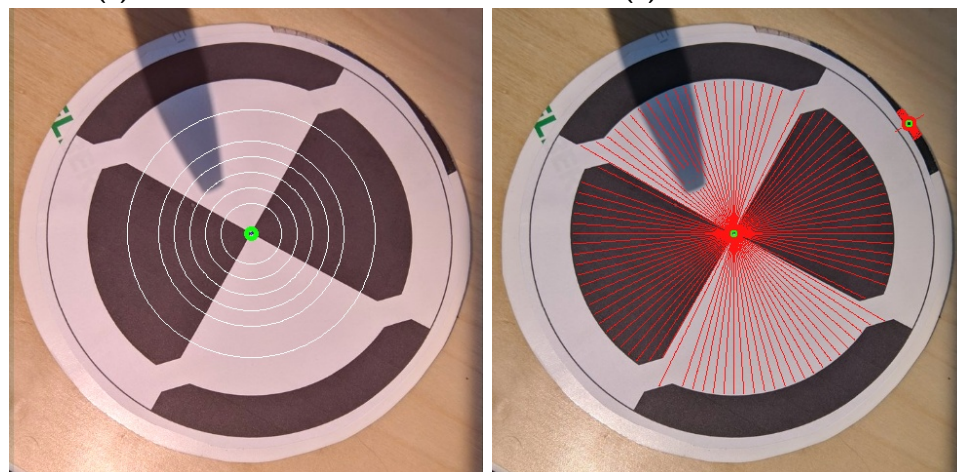
## **Příloha A**

### **Obrazový materiál**



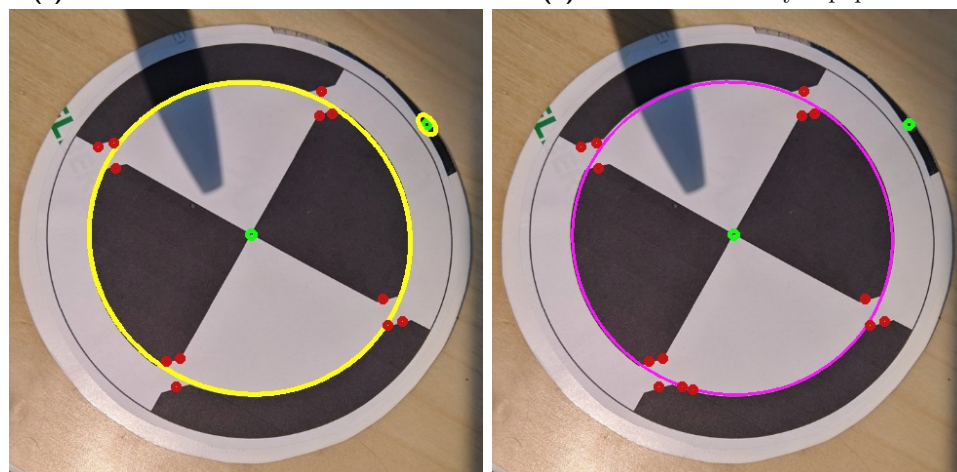
(a) : Detekce rohů v obraze

(b) : Korelační test



(c) : Korelační test v režimu maskování

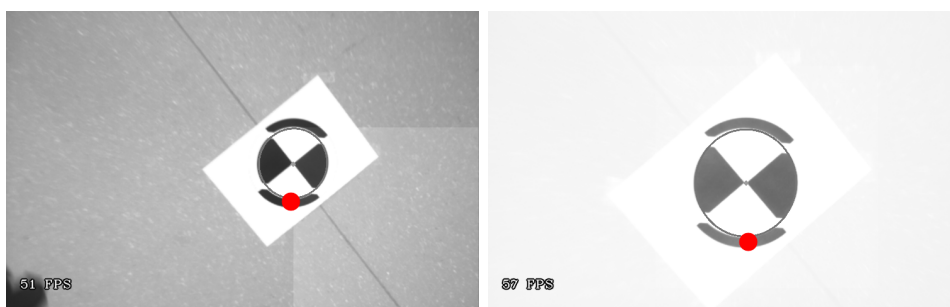
(d) : Metoda odstředných paprsků



(e) : První krok prokladu elipsou

(f) : Druhý krok prokladu elipsou

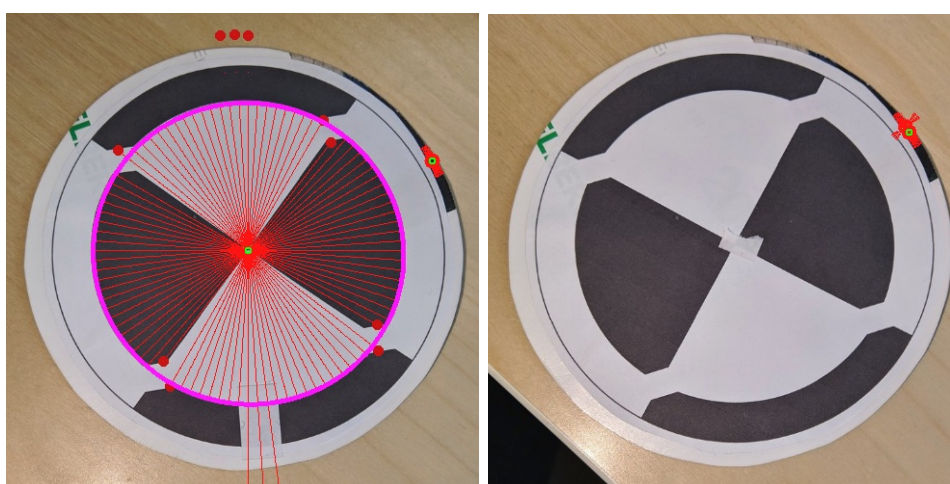
**Obrázek A.1:** Jednotlivé fáze detekce zarušeného přistávacího vzoru (2. část)



(a) : Černá barva ve vyznačené oblasti s hodnotou jasové složky  $I = 50$

(b) : Černá barva ve vyznačené oblasti s hodnotou jasové složky  $I = 140$

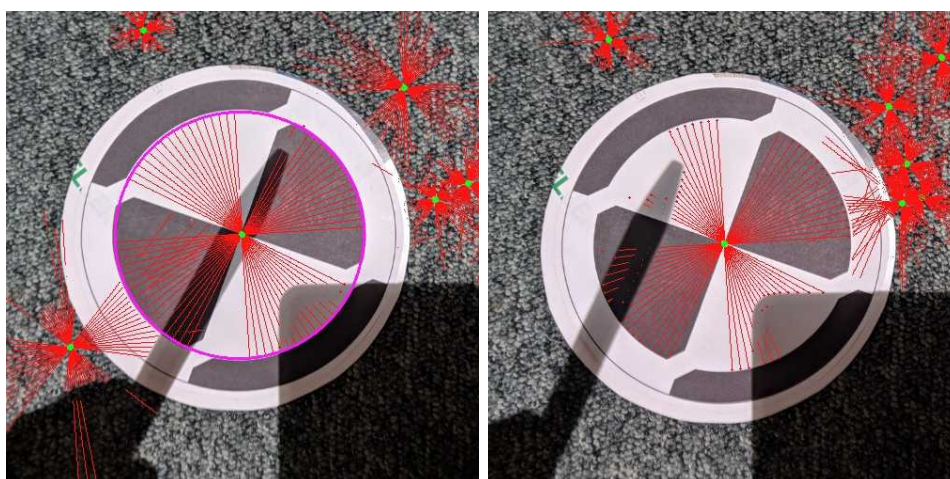
**Obrázek A.2:** Náhlá změna jasu při kontinuálním snímání



(a) : Porušení kruhové hranice

(b) : Porušení středu obrazce

**Obrázek A.3:** Porušení přistávacího voru na různých místech



(a) : Stíny rozložené nesymetricky kolem středu obrazce

(b) : Zastínění na obou stranách ve stejné vzdálenosti od středu

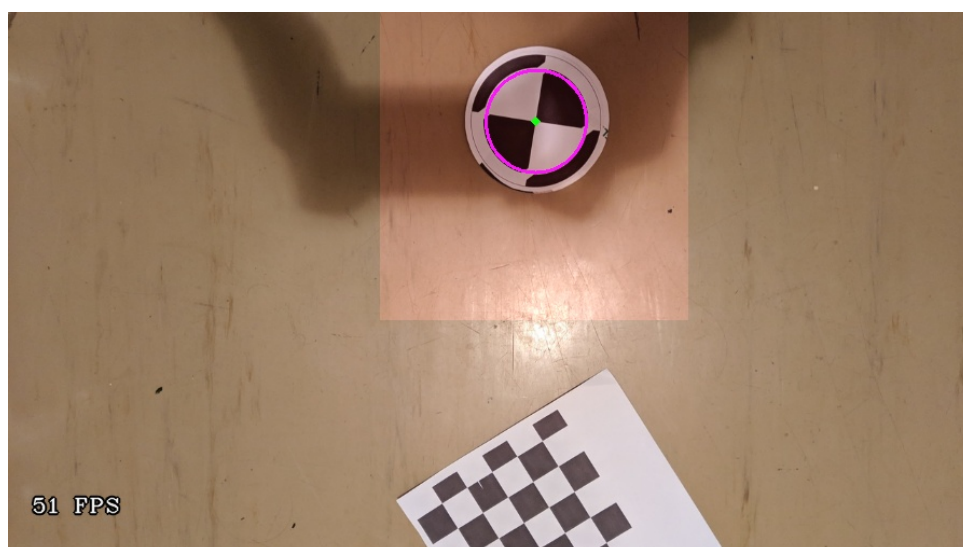
**Obrázek A.4:** Vliv stínů v různých místech přistávacího obrazce



(a) : Přesvětlený snímek, vzor nebyl nalezen

(b) : Snímek se sníženým časem expozice, vzor byl úspěšně nalezen

**Obrázek A.5:** Vzor na řídkém trávníku s vysokým počtem kontrastních rohů



**Obrázek A.6:** Šachovnicové pole mimo prohledávanou oblast snímku



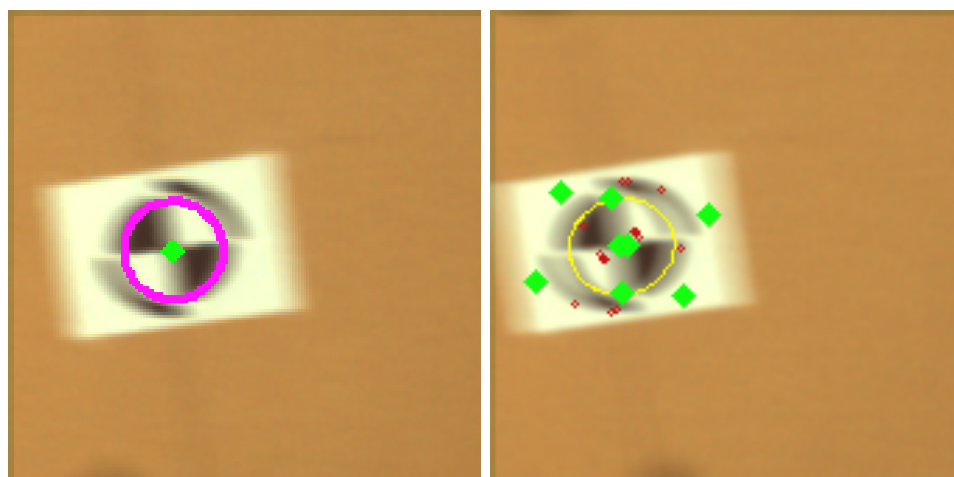
(a) : Vyhodnocen korektně

(b) : Vyhodnocen korektně

**Obrázek A.7:** Letecké snímky bez přítomnosti přistávacího vzoru [5]



**Obrázek A.8:** Kritická vzdálenost, ve které přestává být obrazec detekován



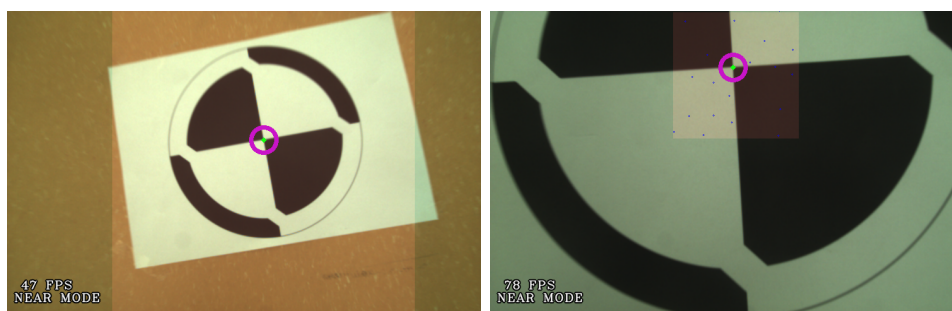
**(a)** : Vzor byl detekován

**(b)** : Vzor nebyl detekován

**Obrázek A.9:** Výřez snímku s rozmazaným přistávacím vzorem



Obrázek A.10: Výřez snímku s falešnou detekcí přistávacího vzoru



(a) : Vzdálenost, ve které dojde k přepnutí systému do režimu přiblížení

(b) : Detekovaný střed obrazce při dosednutí helikoptéry

Obrázek A.11: Přiblížení k přistávacímu vzoru





## Příloha B

### DVD

K práci je přiloženo DVD, které obsahuje text ve formátu .pdf, zdrojové kódy systému *hetect*, originální snímky použité pro testování, vyhodnocené snímky a logovací soubory programu. DVD má následující adresářovou strukturu:

- prace\_pdf
- testovaci\_snimky
  - bez\_vzoru
  - priblizeni
  - ruzna\_pozadi
  - sachovnice
  - stiny
  - vzdalenost
  - zkusebni\_let
- zdrojove\_kody

Zdrojové kódy jsou v současné době dostupné též na webové adrese <https://gitlab.fel.cvut.cz/gartnjan/hetect>.





## Příloha C

### Literatura

- [1] Introduction to computer vision: Tutorial 2: Image matching. <http://ai.stanford.edu/~syyeung/cvweb/tutorial2.html>. navštíveno 22. 4. 2019.
- [2] Opencv 2.4.13.7 documentation: Camera calibration and 3d reconstruction. [https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html), 2014. navštíveno 10. 4. 2019.
- [3] Opencv 3.0.0-dev documentation: Harris corner detection. [https://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_feature2d/py\\_features\\_harris/py\\_features\\_harris.html](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html), listopad 2014. navštíveno 22. 4. 2019.
- [4] Opencv 3.0.0-dev documentation: Shi-tomasi corner detector and good features to track. [https://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_feature2d/py\\_shi\\_tomasi/py\\_shi\\_tomasi.html](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_shi_tomasi/py_shi_tomasi.html), listopad 2014. navštíveno 23. 4. 2019.
- [5] The best free stock photos & videos shared by talented creators. <https://www.pexels.com/>, 2019. navštíveno 4. 5. 2019.
- [6] Opencv 2.4.13.7 documentation: Miscellaneous image transformations. [https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous\\_transformations.html](https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html), 2019. navštíveno 16. 4. 2019.
- [7] Opencv 2.4.13.7 documentation: Smoothing images. [https://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian\\_median\\_blur\\_bilateral\\_filter/gaussian\\_median\\_blur\\_bilateral\\_filter.html](https://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html), 2019. navštíveno 16. 4. 2019.
- [8] Opencv 2.4.13.7 documentation: Structural analysis and shape descriptors. [https://docs.opencv.org/2.4/modules/imgproc/doc/structural\\_analysis\\_and\\_shape\\_descriptors.html?highlight=fitellipse#fitellipse](https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=fitellipse#fitellipse), 2019. navštíveno 16. 4. 2019.

- [9] OpenCV 3.4.3 documentation: Camera calibration and 3d reconstruction. [https://docs.opencv.org/3.4.3/d9/d0c/group\\_\\_calib3d.html#ga93efa9b0aa890de240ca32b11253dd4a](https://docs.opencv.org/3.4.3/d9/d0c/group__calib3d.html#ga93efa9b0aa890de240ca32b11253dd4a), 2019. navštíveno 16. 4. 2019.
- [10] D. Eberly. *Distance from a Point to an Ellipse, an Ellipsoid, or a Hyperellipsoid*. geometrictools.com, 2013.
- [11] A. W. Fitzgibbon and R. B. Fisher. A buyer's guide to conic fitting. *5th British Machine Vision Conference*, 1995.
- [12] C. Harris and M. Stephens. A combined corner and edge detector. *Fourth Alvey Vision Conference*, 1988.
- [13] D. Kerr, S. Coleman, and B. Scotney. Comparing cornerness measures for interest point detection. *International Machine Vision and Image Processing Conference*, 2008.
- [14] T. Krajník, M. Nitsche, et al. A practical multirobot localization system. *Journal of Intelligent and Robotic Systems (JINT)*, 2014.
- [15] K. Kuglerová. Automatický vzlet a přistání robotické helikoptéry, bakalářská práce. *ČVUT FEL, Katedra kybernetiky*, 05 2017.
- [16] S. Lange, N. Sünderhauf, and P. Protzel. Autonomous landing for a multirotor uav using vision. *International Conference on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS*, 2008.
- [17] S. Lee, J. Jang, and K. Baek. Implementation of vision-based real time helipad detection system. *International Conference on Control, Automation and Systems*, pages 191–194, 01 2012.
- [18] S. Saripalli, J. F. Montgomery, and G. Sukhatme. Visually-guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, 19:371 – 380, 03 2002.
- [19] J. Shi and C. Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Gärtner** Jméno: **Jan** Osobní číslo: **466304**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra kybernetiky**  
Studijní program: **Kybernetika a robotika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Rozpoznávání vizuálního vzoru pro automatické přistávání robotické helikoptéry**

Název bakalářské práce anglicky:

**Visual Pattern Recognition for Autonomous Landing of Robotic Helicopter**

Pokyny pro vypracování:

Cílem práce je návrh metody pro relativní lokalizaci robotické helikoptéry nad přistávacím místem na základě rozpoznávání grafického vzoru umístěného na místě pro přistávání.

1. Seznamte se se základními metodami zpracování obrazu a metodami detekce a popisu objektů v obraze.
2. Proveďte rešerši prací, které řeší podobný problém.
3. Na základě získaných informací a konzultace s vedoucím práce navrhnete vhodný vzor a metodu pro jeho vyhledání. Návrh by měl být proveden s ohledem na maximální rozsah vzdáleností pro lokalizaci vzoru, přičemž zásadní je možnost detekce při velkém přiblížení.
4. Navrženou metodu implementujte a ověřte její funkci s reálnou kamerou v laboratorních podmínkách, nebo na datech pořízených v reálných podmínkách. Pokuste se při experimentech postihnout možné nepříznivé vlivy prostředí, zejména vliv osvětlení.
5. Proveďte vyhodnocení dosažitelné přesnosti, robustnosti detekce a rychlosti zpracování obrazu a vyhodnoťte, kde mohou být meze správné funkce metody.

Seznam doporučené literatury:

- [1] T. Krajník, M. Nitsche et al.: A Practical Multirobot Localization System. Journal of Intelligent and Robotic Systems (JINT), 2014  
[2] Šonka, Milan a Václav Hlaváč. Počítačové vidění. Praha: Grada, 1992  
[3] K. Kuglerová, Automatický vzlet a přistání robotické helikoptéry, bakalářská práce, katedra kybernetiky, ČVUT v Praze, FEL, 2017

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Jan Chudoba, inteligentní a mobilní robotika CIIRC**

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **24.01.2019**

Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce: **20.09.2020**

Ing. Jan Chudoba  
podpis vedoucí(ho) práce

doc. Ing. Tomáš Svoboda, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta