



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

VIZUALIZACE OBLIČEJOVÉ MIMIKY PRO ROBOTICKOU HLAVU

FACE MASK VISUALIZATION FOR A ROBOTIC HEAD

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ POSPÍCHAL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MARTIN APPEL

BRNO 2020

Zadání bakalářské práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky
Student: **Tomáš Pospíchal**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Mechatronika
Vedoucí práce: **Ing. Martin Appel**
Akademický rok: 2019/20

Ředitel ústavu Vám v souladu se zákonem č.1111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Vizualizace obličejové mimiky pro robotickou hlavu

Stručná charakteristika problematiky úkolu:

Pro přirozenou komunikaci uživatele s humanoidním robotem je simulace mimiky nesmírně důležitá. Lidský obličej má desítky obličejových svalů, které spoluvytvářejí mimiku obličeje. Nahrazení obličejových svalů aktuátory je pro jejich počet velice obtížné a drahé. Myšlenka této práce je vizualizovat mimiku obličeje promítáním obrazu na masku obličeje. Tato maska je přirozeně nerovná a je potřeba deformovat promítaný obraz tak, aby výsledek vytvořil požadovaný obraz. Projekční vzdálenost je velmi malá a vyžádá si speciální projektor s minimální projekční vzdáleností nebo úpravu konstrukce.

Zpětná projekce potřebuje plastovou masku se speciální povrchovou úpravou. Výroba samotné projekční masky také není jasná. Existuje představa o finální podobě, ale technologie výroby není vyřešená a existuje jistě více variant, jak dosáhnout daného výsledku. Je potřeba zvolit takovou metodu, která je cenově dostupná a zároveň splňuje projekční požadavky. Složitý tvar projekční masky způsobí, že světlo z projektoru bude dopadat na některá místa pod velice ostrým úhlem. Tento úhel způsobí rozdílný kontrast promítaného obrazu. Je vhodné tuto vadu řešit buď kalibrací, nebo mechanicky pomocí soustavy zrcátek.

Cíle bakalářské práce:

- 1) Vyberte vhodný projektor a realizujte promítací masku. Sestrojte testovací aparaturu pro zpětnou projekci na polopropustnou masku.
- 2) Provedte zpracování obrazu s cílem eliminovat deformaci promítaného obrazu na zakřivenou projekční plochu. Demonstrujte splnění cíle na jedné triviální úloze, jako je mřížka, a jedné netriviální úloze.
- 3) Provedte zpracování obrazu s cílem eliminovat rozdílný kontrast promítaného obrazu na zakřivenou projekční plochu. Demonstrujte splnění cíle pomocí ukázkové úlohy.
- 4) Vytvořte uživatelský nástroj, který umožní vzít jakýkoliv obrazový zdroj a promítnout ho na masku bez hlubších znalostí problematiky.

Seznam doporučené literatury:

CORKE, Peter I. Robotics, vision and control: fundamental algorithms in MATLAB. Berlin: Springer, 2011. Springer tracts in advanced robotics, v. 73. ISBN 9783642201431.

VALÁŠEK, Michael. Mechatronika. Praha: České vysoké učení technické, 1995. ISBN 80-01-01276-X.

GREPL, Robert. Kinematika a dynamika mechatronických systémů. Brno: Akademické nakladatelství CERM, 2007. ISBN 978-80-214-3530-8.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2019/20

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Bakalářská práce se zabývá využitím zpětné projekce pro vizualizaci mimiky sociobota. Je navržena a implementována kalibrace obrazu k zamezení deformace a změny jasů promítaného obrazu způsobenou zakřivením projekční plochy. Identifikace deformace provedena pomocí promítané šachovnice. K aproximaci deformace použita metoda pohyblivých vážených nejmenších čtverců (MLS). Dále je navržena testovací soustava a úprava projektoru pro kratší promítací vzdálenost.

Summary

This bachelor thesis deals with face exposure visualisation using backward projection. Image calibration is implemented for correction of distortion and brightness change caused by curved projection plane. Identification of deformation is done by projected checkerboard pattern. For deformation method of Moving Least Squares (MLS) is used. Test tool is designed and projector upgrade for shorter throw distance is implemented.

Klíčová slova

MLS, DLP, projektor, korekce obrazu, restituace obrazu, shorthrow

Keywords

MLS, DLP, projector, image correction, image restitution, shorthrow

POSPÍCHAL, T. *Vizualizace obličejové mimiky pro robotickou hlavu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2020. 40 s. Vedoucí Ing. Martin Appel.

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci na téma *Vizualizace obličejové mimiky pro robotickou hlavu* vypracoval samostatně s požitím odborné literatury a pramenů uvedených v seznamu, který tvoří přílohu této práce.

V Brně dne 24.5.2019

.....

Tomáš Pospíchal

Obsah

1	Úvod	2
2	Eliminace deformací promítaného obrazu	3
2.1	Představení problému a popis projekční soustavy	3
2.2	Přehled a spojení jednotlivých algoritmů	5
2.3	Korekce perspektivy kamera - plátno	5
2.4	Detekce šachovnice	7
2.4.1	Detekce rohů šachovnice	9
2.4.2	Nalezení struktury šachovnice	11
2.5	Deformace obrazu	13
2.5.1	Metoda MLS - Moving least squares	13
2.5.2	Interpolace obrazových dat podle výsledku MLS	15
2.5.3	Získání cílových bodů pro před-deformaci obrazu	16
2.6	Ukázka	18
3	Úprava kontrastu promítaného obrazu	20
4	Testovací soustava	23
4.1	Popis HW	23
4.1.1	Polopropustná maska	23
4.1.2	Popis konstrukce	23
4.1.3	TI DLP Projektor + BeagleBone Black	26
4.2	Popis SW	27
4.2.1	BeagleBone Black + DLP projektor	28
4.2.2	Matlab program PC	30
4.2.3	Matlab uživatelské prostředí	31
5	Závěr	33
6	Seznam použitých zkratek a symbolů	39
7	Seznam příloh	40

1. Úvod

Cílem sociálního robota je interakce a komunikace s lidmi, v mezilidské komunikaci je mimika obličeje důležitou složkou (nonverbální) komunikace, proto bývají tito roboti vybaveni vizualizací obličeje. Pro vizualizaci bývá používán LED display, animatronika a v neposlední řadě zpětná projekce, kterou se zabývá tato práce.

Použití zpětné projekce je méně komplikované a nákladné, než animatronika jež vyžaduje desítky aktuátorů pro nahrazení obličejových svalů a oproti komerčně dostupným LED displayům, navíc zpětná projekce umožňuje vytvoření 3D plastického obličeje, při konstrukci humanoidního robota žádoucím atributem.

Promítáním obrazu na nepravidelný povrch dochází k jeho zkreslení, tohoto jevu je využíváno u structured light 3D skenování, kdy je na povrch promítnut známý vzor a následným zpracováním obrazu stanovena vzdálenost - například u Microsoft Kinect a Intel RealSense.

Při vizualizaci je však zkreslení promítaného obrazu nežádoucím efektem, dalším problémem promítání na zakřivenou plochu je lokální změna jasu způsobena rozdílnou vzdáleností plochy od projektoru, rozdílným úhlem dopadu paprsků a u zpětné projekce navíc i rozdílná tloušťka masky a povrchové úpravy masky. Pro eliminaci těchto zkreslení je nutné identifikovat deformaci promítaného obrazu a tím nalézt mapování mezi promítnutým obrazem a zdrojovým obrazem, umožňující zpracování zdrojového obrazu předdeformací a korekci intenzity jednotlivých pixelů tvořících zdrojový obraz.

2. Eliminace deformací promítaného obrazu

V této kapitole je představena problematika kalibrace deformace obrazu. Nejprve je sestaven model znázorňující jednotlivá zkreslení vyvolané soustavou, poté uveden přehled použitých kroků a algoritmů pro dosažení eliminace deformací předzpracováním promítaného obrazu. Tyto algoritmy jsou následně v jednotlivých podkapitolách představeny podrobněji.

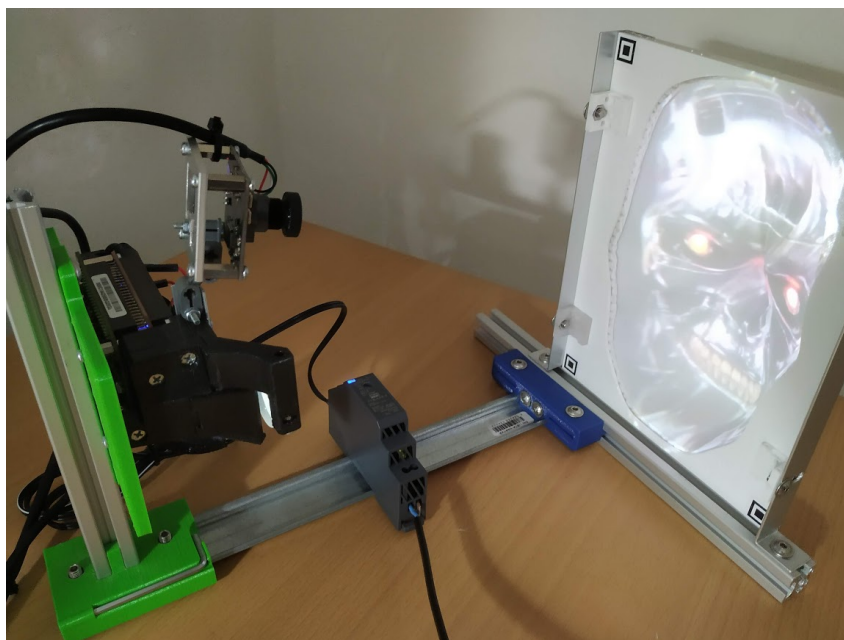
2.1. Představení problému a popis projekční soustavy

obrázek testovací soustavy

Podstatou eliminace deformace promítaného obrazu neboli "geometrická restituce je odstranění zkreslení, vzniklé nedokonalostí zobrazovacího systému, či v důsledku geometrie způsobu snímání"[2].

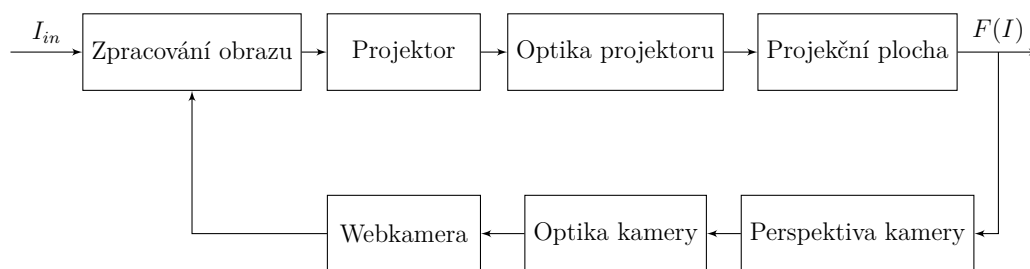
Algoritmus lze rozdělit do dvou kroků:

Cílem prvního kroku je nalezení mapování mezi body ve zdrojovém obrazu a odpovídajícími body na projekční ploše zachycených kamerou [1]. Problém komplikuje neznalost vzájemných poloh, orientace a optické vlastností soustavy kamery, projektoru a projekční plochy, na obr.2.1 je soustava zobrazena. Parametry utvářejí subsystémy optické soustavy, znázorněny v blokovém schématu obr. 2.2, působící na vstupní data - zdrojový obraz. Zkreslení způsobené jednotlivými subsystémy lze postupně identifikovat a odstranit.



Obrázek 2.1: Testovací soustava skládající se z projektoru s přídatnou optikou, webkamery a projekční plochy [autor]

2.1. PŘEDSTAVENÍ PROBLÉMU A POPIS PROJEKČNÍ SOUSTAVY



Obrázek 2.2: Znázornění subsystémů v blokovém schématu [autor]

Prvním ze zkreslení, které je nutné odstranit je zkreslení způsobené optikou kamery, to je odstraněno provedením kalibrace kamery, za použití Matlab Camera Calibrator app variantu s modelem pro fisheye zkreslení [37].

Z několika desítek snímků kalibrační šachovnice v různé orientaci a vzdálenosti zorného pole kamery je pro použitou kameru získána kalibrační funkce odstraňující zkreslení, ta je vyexportována a za použití funkce *undistortFisheyeImage* jsou všechny snímky z kamery před následným zpracováním upraveny. Bez odstranění zkreslení kamery by docházelo k překorigování promítaného obrazu.

Dalším nutným krokem je odstranění perspektivního zkreslení obrazu kamery vzhledem k rovině projekční plochy. V ideálním případě by kamera ležela v optické ose projektoru a k žádnému zkreslení by nedocházelo, toto však není fyzicky možné, jako řešení pro malé vyosení optických os se geometrická transformace jeví jako adekvátní kompromis.

K identifikaci tohoto globálního zkreslení jsou v pořízeném obrazu identifikovány QR finder markery, sloužící jako referenční body se známou pozicí a poměrem stran. Ze znalosti těchto bodů je poté nalezena transformační rovnice a obraz upraven. Bližší podrobnosti jsou uvedeny v kap. 2.3.

Konečným krokem je identifikace deformací způsobených zejména projekční plochu, ale z menší části, také optickou vadou přídatné optické soustavy projektoru.

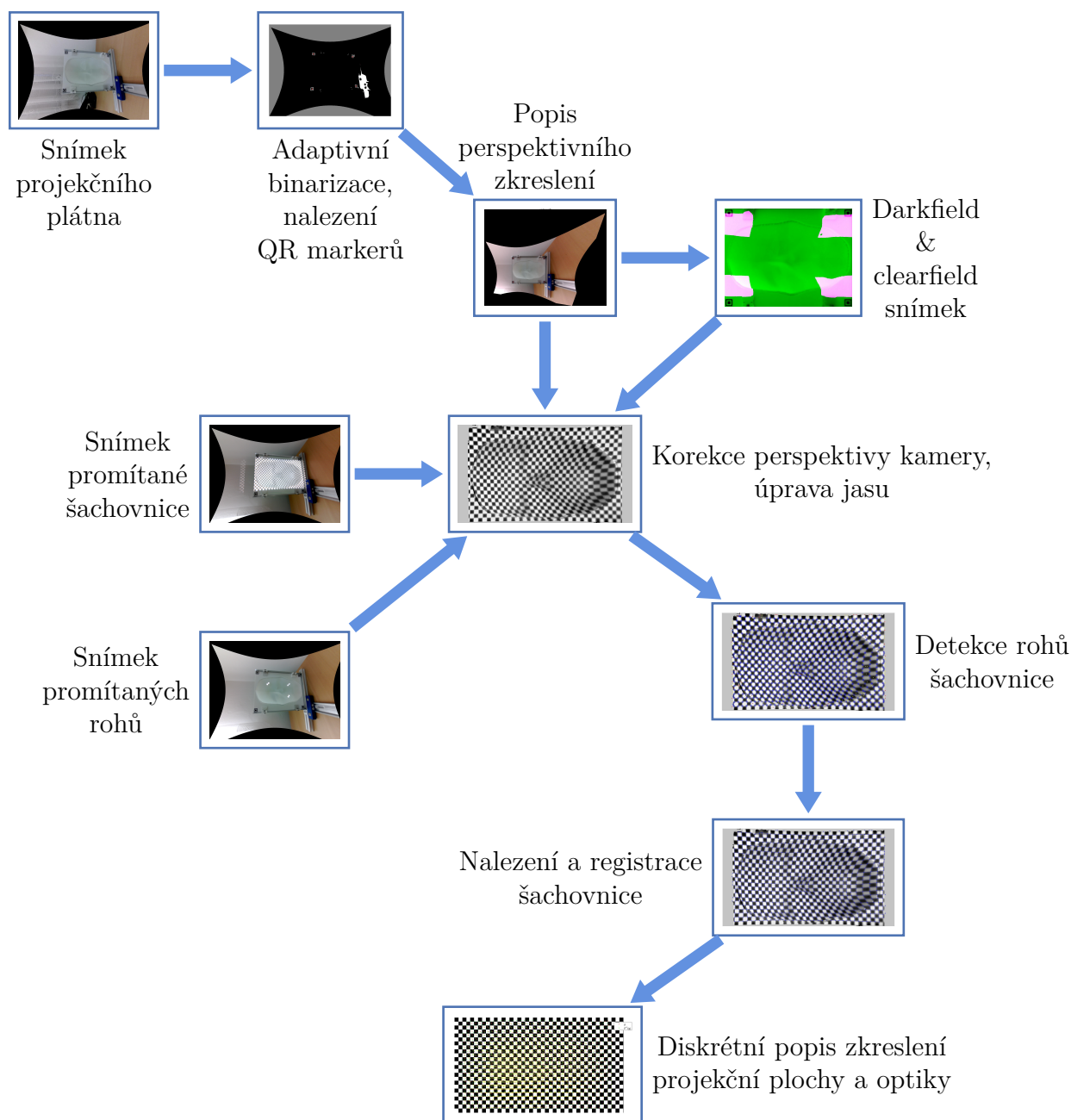
Transformační rovnice však pro složitost deformací (na rozdíl od korekce zkreslení perspektivou) není možno nalézt v uzavřeném tvaru [2].

Identifikace deformace je docílena promítnutím šachovnice a nalezením jejich rohů. S dostatečně hustou sítí bodů (rohů) je pak možno zkreslení identifikovat [2] a za použití některého z algoritmu pro deformaci obrazu ovládaného počátečními a koncovými body, popřípadě polygonové sítě tuto deformaci eliminovat.

Algoritmus pro detekci šachovnice a použitá metoda deformace obrazu je popsána v kap. 2.4 kap. 2.5.

Objasnění které body jsou použity jako počáteční a koncové pro před-deformování obrazu znázorněno na obr. 2.14 a blíže popsáno v kap. 2.5.3

2.2. Přehled a spojení jednotlivých algoritmů



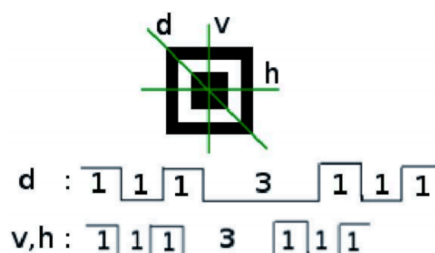
Obrázek 2.3: Spojení jednotlivých algoritmů [autor]

2.3. Korekce perspektivy kamera - plátno

Mírné perspektivní zkreslení vzniklé rozdílnou optickou osou kamery a projektoru lze korigovat, protože se však v případě obličejové masky nejedná o rovinu, míra korigovatelného zkreslení je omezena do ztráty informace překrytím ploch.

Určením čtyř bodů roviny projekční plochy je stanovena transformace perspektivy. Pro detekci těchto čtyř bodů je projekční plocha vybavena QR finder vzory, jejichž charakteristický vzor opakování bílé a černé dodržuje poměr relativních tlouštěk čar 1:1:3:1:1, ilustrováno na obr 2.4

2.3. KOREKCE PERSPEKTIVY KAMERA - PLÁTNO



Obrázek 2.4: Charakteristický poměr v QR finder [35]

Prvním krokem při hledání QR finder vzorů je adaptivní binarizace obrazu. Odhadnutí lokální průměrné intenzity v okolí pixelu, sloužící jako lokální práh binarizace, je provedeno rozmazáním obrazu (konvolucí s průměrujícím filtrem). V MatLab s použitím funkce *adaptthresh*, nejlepších výsledků bylo dosaženo s Gaussovým filtrem.

Následně jsou na základě předpokládané tloušťky čar na snímku QR finderů v binárním obraze provedeny morfologické operace otevření *imopen* a uzavření *imclose* za účelem odstranění objektů s nižší šířkou, než předpokládaná tloušťka čar, resp. zaplnění děr v objektech menších, než předpokládaná tloušťka čar.

Poté je, za účelem nalezení objektů obsahující díry, provedena filtrace objektů *bwpropfilt* podle eulerovy charakteristiky - v Matlab implementované jako počet objektů v oblasti mínus počet děr v těchto objektech.

Po nalezení těchto objektů je těžištěm děr veden řez (navzorkování) v horizontálním, vertikálním a diagonálním směru obrazu. V těchto vzorcích je podle [36] za použití funkcí *pulsewidth* a *pulsesep* stanoven poměr mezi impulzy potenciálního QR finderu, pokud se jednotlivé poměry vejdu do tolerance lze binární objekt považovat za nalezený QR finder.

Nalezené QR findery jsou seřazeny podle hodinových ručiček a ze známého poměru stran projekční plochy nalezeny cílové body obdélníku zachovávající tento poměr.

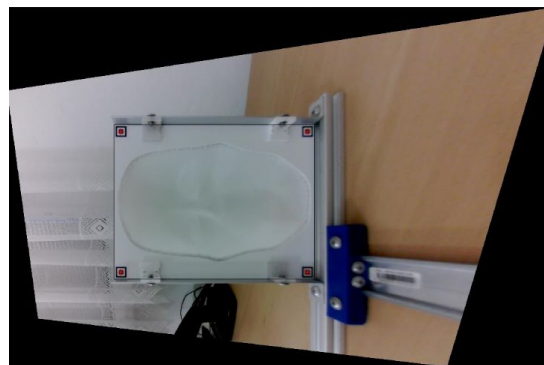
Následně je použita funkce *fitgeotrans*, která ze zadaných bodů stanoví lineární transformační matici.

K transformované poloze středů QR finderů je přičtena šířka markeru a nalezeny tak rohy projekční plochy, podle kterých je následně obraz oříznut.

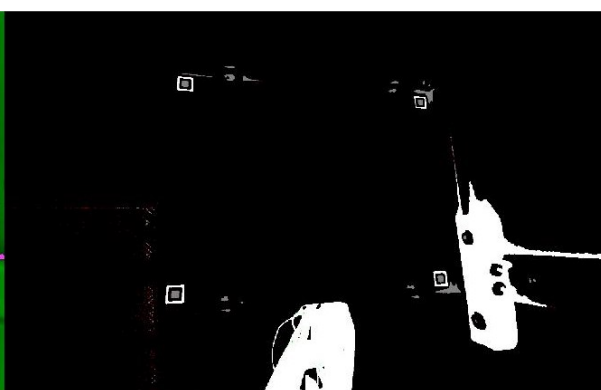
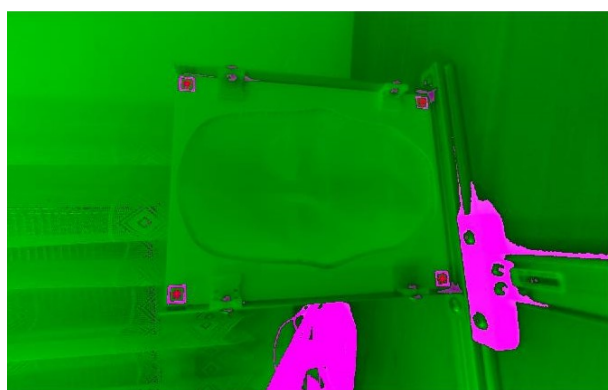
Takto nalezená transformace a oříznutí obrazu je za použití funkce *imwarp* a *imcrop* aplikována na snímky vstupující do všech následujících operací zpracování obrazu.



(a) Snímek rámečku, zkreslení kamery korigováno



(b) Snímek rámečku, zkreslení perspektivy korigováno



(c) Vizualizace adaptivní binarizace

Obrázek 2.5: Nalezení QR finder markerů [autor]

2.4. Detekce šachovnice

Jak bylo uvedeno v kapitole 2.1 pro odstranění deformace je nejprve nutné tuto deformaci identifikovat.

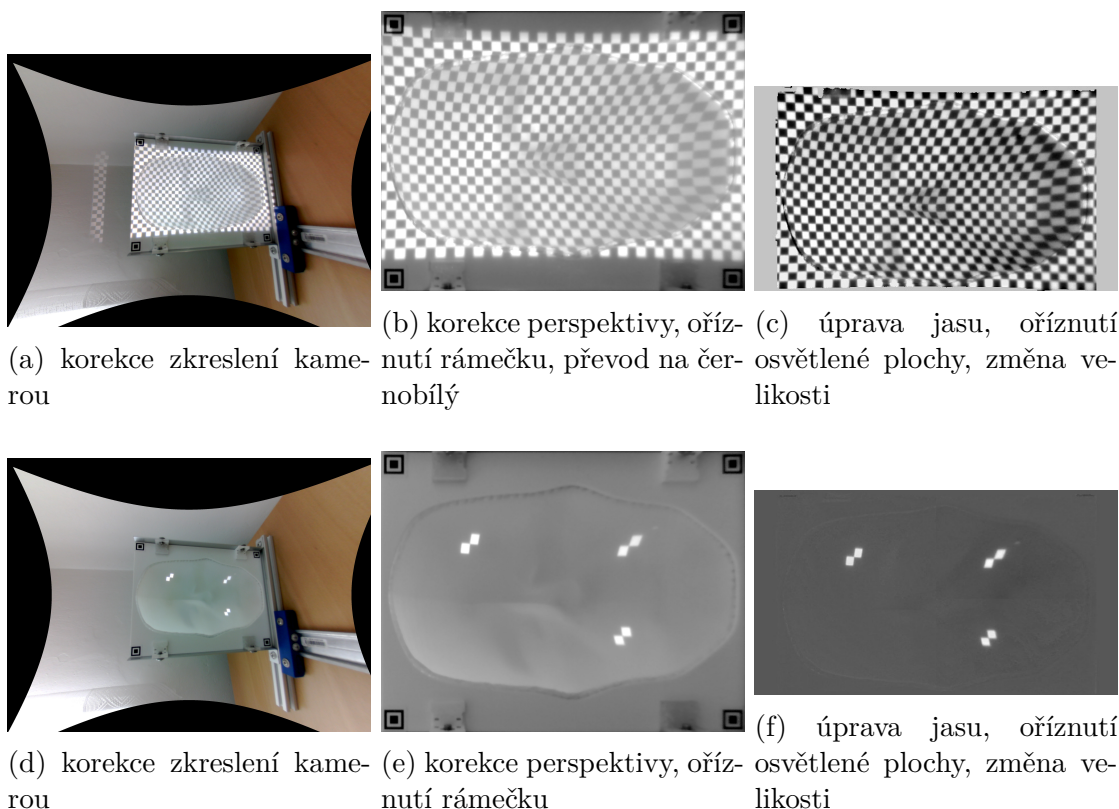
Identifikace lze dosáhnout dostatečně hustou sítí bodů [2].

Nabízí se několik kalibračních vzorů mezi dva nejpoužívanější patří tečky a šachovnice [3]. Kalibrační vzor teček umožňuje vyšší hustotu bodů, avšak protože tečka je větší objekt než pouze bod, je také ovlivněna deformací [5], která bez znalosti modelu zkreslení nelze kompenzovat [3]. Šachovnice je odolnější proti zkreslení [3], i při silném zkreslení si lokální oblast okolo rohu šachovnice zachovává rotační symetrii [7], hrany umožňují určit konektivitu bodů [4, 7]. Detekce šachovnice pro kalibraci kamery je standardní záležitostí, Matlab disponuje funkcí *detectCheckerboardPoints* [6], která dosahuje dobrých výsledků [7], avšak zřejmě následkem poměrně velkých, nepravidelných deformací a omezujících podmínek pro konektivitu (nevstupující do funkce jako parametr) detekuje pouze část šachovnice s podobnou deformací. [6].

Proto byla implementována detekce šachovnice skládající se ze dvou kroků: detekce rohů šachovnice za použití prstencového vzorku okolí pixelu, očekávaný průběh vzorku obsahující roh šachovnice jsou dvě periody funkce sin [8, 10, 11]. Následně jsou rohy spojeny do mřížky a odstraněny falešně pozitivní rohy podle [7].

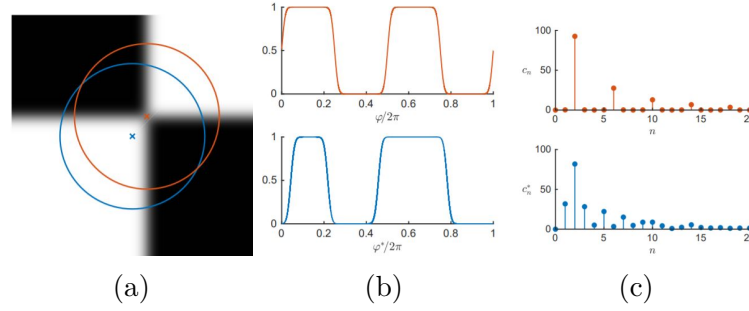
Úprava vstupních snímků pro algoritmus detekce

Získané popisy zkreslení obrazu z předchozích kroků jsou aplikovány a vstupní obrazy korigovány. Je využito znalosti darkfield a clearfield snímků pro korekci šedotónových hodnot vstupního snímku. Nejprve je z odhadnuta hodnota světlých a tmavých čtverců šachovnice, binarizací vstupního obrazu a stanovením mediánu po aplikaci masky a invertované masky. Tyto hodnoty jsou poté v histogramu přesunuty na hodnoty odpovídající tmavým částem darkfield obrazu a světlým částem v clearfield. Takto upravený vstupní snímek je váhován lokální jasovou transformací (3.1).



Obrázek 2.6: Úprava jasu vstupních snímků [autor]

2.4.1. Detekce rohů šachovnice



Obrázek 2.7: Princip detekce rohu kruhového vzorku, zastoupení dominantních složek Fourierovy transformace [9]

Autoři ChESS [10] poukazují na nevhodnost použití generalistických algoritmů pro detekci rohů (Harris and Stephens) na detekci šachovnice pro velmi vysoký počet falešně pozitivních rohů komplikují vyhodnocení, na které se většina předcházejících publikací zaměřovala.

Navrhují použití prstencového vzorkování podobně jako u FAST rohového detektoru avšak s větším poloměrem se zachováním stejného počtu vzorků. Pro rychlost detekce (až 700Hz pro VGA rozlišení obrazu na běžně dostupném HW) operace sčítání a odčítání aproximují fft pro $f=2\text{Hz}$.

Autoři v [8] používají standardní fft transformaci a navrhuje vyřazení odezev, vycházející z amplitudy, nižších než prahová hodnota stanovená na základě velikosti šumu v obraze.

Pro jednodušší implementaci změny poloměru kružnice a možnosti získání fáze určující natočení rohu šachovnice byla zvolena metoda navržená v [8].

Vzorkovací body u_i, v_i na diskretní kružnici se středem u_c, v_c jsou definovány jako

$$u_i = \left[r \cos\left(2\pi \frac{i}{M}\right) + u_c \right] \quad (2.1)$$

$$v_i = \left[r \sin\left(2\pi \frac{i}{M}\right) + v_c \right] \quad (2.2)$$

kde $i = 1, 2, \dots, M$, M je počet vzorků a r poloměr kružnice. Vzorky pak tvoří 1D vektor $x = (x_1, x_2, \dots, x_M)^T$.

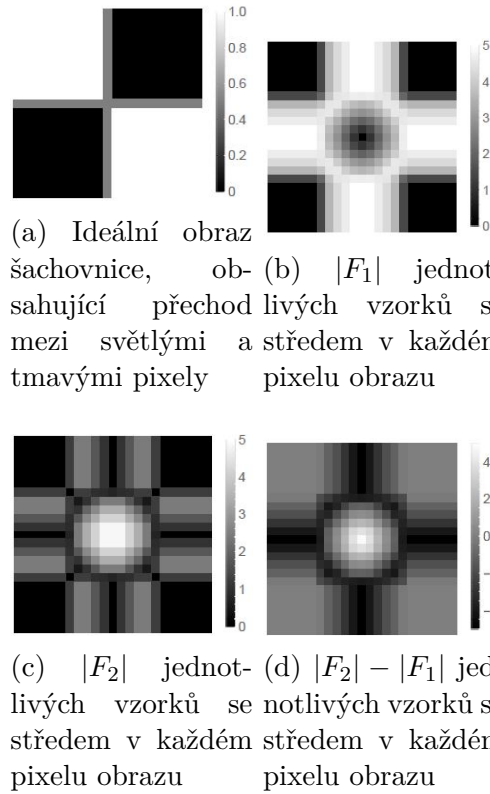
Diskretní Fourierova transformace tohoto vektoru je definována

$$F_k = \sum_{m=0}^{M-1} x_m e^{-j(2\pi k) \frac{m}{M}} \quad (2.3)$$

po aplikaci FT: $\mathbf{F}_k = \mathbf{F}\{\mathbf{x}\} = (F_0, F_1, \dots, F_{M-1})$ kde F_k je komplexní číslo, index k představuje frekvenci vztaženou k celkovému počtu vzorků M .

Jak je možno vidět na obr. 2.7 v rohu šachovnice je dominantní složka vzorku F_2 , na hraně šachovnice zase F_1 . Jejich kombinací (obr. 2.8) lze docílit lepší odezvy, která je stanovena jako:

$$R = F_2 - F_1 \quad (2.4)$$



Obrázek 2.8: Amplitudová charakteristika vzorků ideální šachovnice [8]

Za použití funkce *Fast 2D peak finder* [12], která provede vyhlazení odezvy R a nalezne lokální maxima porovnáním 3×3 okolí bodu získáváme souřadnice těchto rohů.

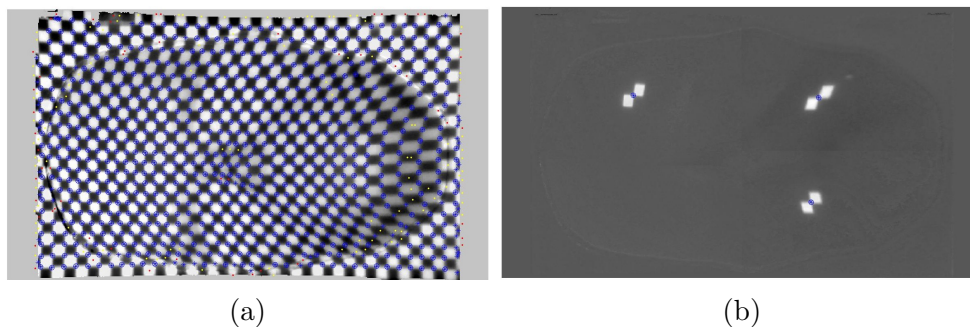
Pro ověření rohu šachovnice podobně jako v [11] natočíme obraz o úhel θ (2.7) konvolucí s přizpůsobeným filtrem (2.5) získáme skalární hodnotu popisující jistotu rohu.

$$\begin{bmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 & -1 & -1 & -1 \end{bmatrix} \quad (2.5)$$

Informaci o natočení rohu šachovnice získáme z fáze ϕ složky vzorku F_2 . Vzhledem k lineárnímu fázovému posuvu pro $f = 2Hz$ a pootočení vrcholů \cos do úhlopříčky rohu je úhel natočení stanoven θ podle (2.7)

$$\phi = \tan^{-1} \left(\frac{\text{Im}\{F_2\}}{\text{Re}\{F_2\}} \right) \quad (2.6)$$

$$\theta = \frac{\phi}{2} + \frac{\pi}{4} \quad (2.7)$$



Obrázek 2.9: Nalezené kandidátní rohy [autor]
(modré vysoká míra korelace, žluté míra korelace >0 , červené korelace ≤ 0)

2.4.2. Nalezení struktury šachovnice

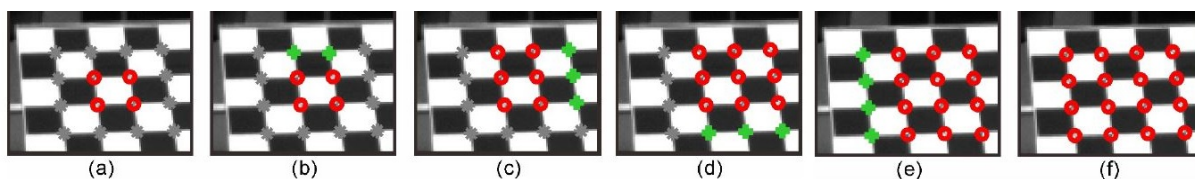
Spojení jednotlivých bodů do mřížky rohů tvořících šachovnici je provedeno podle [7], kroky pro nalezení a odstranění falešně pozitivních rohů doporučené autory [7] jsou nahrazeny algoritmem kombinující [8, 11] představených v předchozí podkapitole 2.4.1.

Struktura šachovnice je nalezena za použití omezení vycházejících z korelace okolí rohu, distribuce jasové intenzity v a na hranici černých a bílých čtvercích šachovnice a také orientace rohu, což dělá metodu více robustní proti silnému zkreslení a extrémní perspektivě [7].

Algoritmus se skládá z následujících kroků [7]:

1. nalezení počátečního čtyřhranu tvořeného vrcholy čtyř sousedících kandidátních rohů s vysokou jistotou
2. iterativní expanze čtyřhranu ve směrech nahoru, doprava, dolů, doleva, dokud nezůstávají žádné další kandidátní rohy splňující omezení

Jelikož pro tuto konkrétní aplikaci nepočítáme s hledáním více šachovnic v jednom obraze, nalezení nového počátečního rohu a následná iterativní expanze není opakována.



Obrázek 2.10: Kroky pro nalezení struktury šachovnice a) Čtyři počáteční rohy, b) expanze směrem nahoru, c) expanze směrem doprava, d) expanze směrem dolů e) expanze směrem doleva, f) expanze dokončena [7]

Nalezení počátečních rohů

”Výběr čtyř počátečních rohů je kritickým krokem v celé proceduře hledání struktury šachovnice, proto jsou použity striktní omezení pro nalezení spolehlivých počátečních rohů”[7].

Pro počáteční čtyřhran je konektivita povolena pouze s kandidátními rohy s maximální hodnotou korelace s (2.5). První bod (levý spodní c_L) je náhodně vybrán, expanze probíhá po směru hodinových ručiček levý horní c_T , pravý horní c_R , pravý spodní c_B . Hledání rohů v daném směru probíhá jako při standardní expanzi (obr. 2.11).

Kontrola je provedena následovně: součet všech úhlů je 360° a počáteční a koncový bod je identický. Pokud není splněno dojde k výběru nového počátečního bodu.

Expanze struktury šachovnice

Po nalezení počátečního čtyřhranu probíhá expanze podle 2. bodu, kandidátními body jsou všechny nalezené rohy, bez ohledu na korelaci s (2.5).

Expanze daným směrem

Při expanzi zvoleným směrem $\alpha \in [-90^\circ, 0^\circ, 90^\circ, 180^\circ]$ je tento úhel nejprve upraven o natočení počátečního rohu $\Delta\theta_i$

$$\Delta\theta_i = \min([0^\circ - \theta_i, 90^\circ - \theta_i]) \quad (2.8)$$

$$\alpha' = \alpha - \Delta\theta \quad (2.9)$$

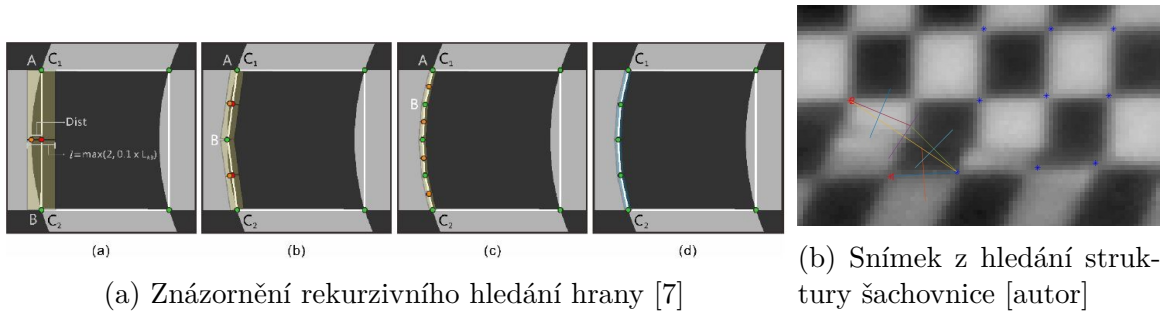
Počet kandidátních rohů je zúžen podmínkou maximální povolené odchylky $\Delta\alpha_{max}$ od expanzního úhlu α' a úhlu mezi kandidátními rohy a počátečním rohem α_N

$$\alpha_N = \tan^{-1} \left(\frac{y_N - y_i}{x_N - x_i} \right) \quad (2.10)$$

$$|\alpha_N - \alpha'| < \Delta\alpha_{max} = 45^\circ \quad (2.11)$$

Dále jsou vyřazeny rohy se stejnou orientací, minimální rozdíl mezi natočením kandidátního rohu θ_N a natočením počátečního rohu θ_i je stanoven na základě minimální separace natočení θ rohů s vysokou mírou korelace s (2.5), v ideálním případě blíží se 90°

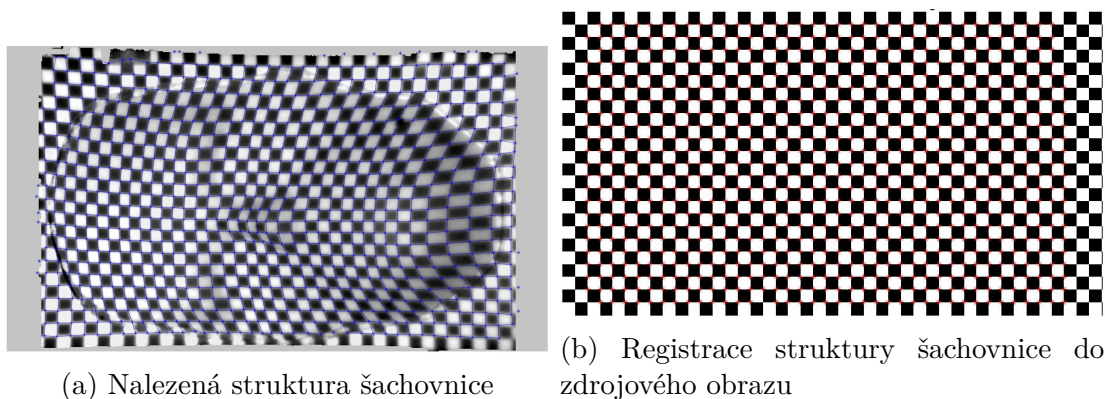
Mezi zbývajících kandidátními body a počátečním rohem je vedena úsečka, pro případ, že je hrana šachovnice zdeformována, je středem úsečky vedena kolmice, pokud je v dosahu $l = \max(2, 0, 1xL_{AB})$ (kde L_{AB} je délka úsečky), vzorků podél kolmice nalezen bod blíže přechodu 50% referenční hodnotě vzorků (*midcross*). Pak je tento bod novým vrcholem a proces se rekurzivně opakuje.



Obrázek 2.11: Expanze zvoleným směrem

Na základě empiricky stanoveného kritéria: poměru střední hodnoty vzorků podél segmentů úsečky a rozptylu rozdílů pixelů nalevo a napravo od pixelů tvořící segmenty úsečky je vybrán bod s nejvyšší hodnotou tohoto poměru. Ověření nového rohu je provedeno expanzí ze sousedního kandidátního rohu směrem ke kritériem vybranému bodu.

Po nalezení struktury šachovnice je využito nesymetrického vzoru tří rohů (obr. 2.9b), a jejich poloze v nalezené mřížce, ke zjištění orientace promítaného obrazu a pokud je nutné, pak k jeho překlopení.



Obrázek 2.12: Výstup detekce šachovnice [autor]

2.5. Deformace obrazu

Na základě nalezení promítaných, optickou soustavou deformovaných, rohů šachovnice je získána diskrétní informace o deformaci obrazu způsobených projekční soustavou (optika společně s promítací plochou) v rozích šachovnice. Pro rekonstrukci těchto bodů, a tím i obrazu provedeme před-deformaci zdrojového obrazu pro promítání metodou MLS, která na základě zadaných počátečních a cílových bodech interpoluje deformaci po celé ploše obrazu. Nejprve je představena metoda samotná a poté postup volby počátečních a cílových bodů, pro docílení eliminace deformace promítaného obrazu.

2.5.1. Metoda MLS - Moving least squares

Deformace je ovládána sadou kontrolních bodů p a cílových (deformovaných) pozic těchto kontrolních bodů q , pro tyto body není vyžadováno umístění do pravidelné mřížky.[29]

Deformace zadané dvojicemi bodů p, q jsou poté aproximovány na rovnoměrné mřížce bodů v řešením malých lineárních rovnic (2×2) v každém bodě v pro všechny dvojice p, q . Pro jednotlivé řešení tohoto menšího počtu systému rovnic umožňuje velmi rychlé deformace mřížek skládajících se z desetitisíců vrcholů v reálném čase.[29]

Popis metody

Metoda MLS [30, 29] neboli pohyblivých vážených nejmenších čtverců se od obyčejné metody nejmenších čtverců liší použitím váhovací funkce w_i (2.13), s přívlastkem pohyblivé pro svou závislost na bodu vyhodnocení v_i na pravidelné mřížce zvyšující tak vliv blízkého okolí bodu [31].

Formulace a odvození výrazu reziduálního součtu čtverců převzato z [29]:
Pro každý bod v hledáme nejlepší transformaci $l_v(\mathbf{x})$ minimalizující výraz (2.12)

$$S^* = \sum_i w_i |l_v(p_i) - q_i|^2 \quad (2.12)$$

kde S^* je reziduální součet čtverců, p_i a q_i jsou řádkové vektory neuspořádaných kontrolních bodů a w_i (2.13) váhový koeficient pro daný kontrolní bod p_i v bodě vyhodnocení v

$$w_i = \frac{1}{|p_i - v|^{2\alpha}} \quad (2.13)$$

Protože je váhový koeficient (2.13) závislý na bodu vyhodnocení v dostaneme rozdílnou transformaci $l_v(\mathbf{x})$ pro každý bod v .

Deformační funkce f obrazu je poté definována

$$f(v) = l_v(v) \quad (2.14)$$

Protože transformace $l_v(\mathbf{x})$ je afinní, skládá se ze dvou částí: lineární transformační matice M a translace T .

$$l_v(\mathbf{x}) = \mathbf{x}M + T \quad (2.15)$$

Translační složka T lze dosazením (2.15) do rovnice (2.12) a parciální derivaci vzhledem k proměnným q_i, p_i vyjádřit v závislosti na M .

$$l_v(\mathbf{x}) = \mathbf{x}M + T = (x - p_*)M + q_* \quad (2.16)$$

kde $q_* = \frac{\sum_i w_i q_i}{\sum_i w_i}$ a $p_* = \frac{\sum_i w_i p_i}{\sum_i w_i}$ jsou vážená těžiště kontrolních bodů.

Minimalizovaný výraz metodou nejmenších čtverců lze tedy přepsat do tvaru

$$S^* = \sum_i w_i |\hat{p}_i M - \hat{q}_i|^2 \quad (2.17)$$

kde S^* je reziduální součet čtverců, substituce $\hat{p}_i = p_i - p_*$, $\hat{q}_i = q_i - q_*$

Autoři [29] odvozují analytické řešení $f(x)$ jak pro sadu kontrolních bodů, tak i pro sadu úseček a rozdílné podmínky omezující potenciálně nežádoucí deformace obrazu vynucením tvaru lineární transformační matice M .

Velkou výhodou MLS je možnost rozdělit řešení $f(x)$ na část závislou pouze na počáteční poloze kontrolního bodu p_i , a tak v případě animace - postupně změně koncového bodu q_i snížit počet operací a zrychlit tak výpočet. [29]

Afinní deformace

Matice M není omezena, tato úprava dovoluje všechny deformace popsané lineární transformací a translací - rotace, zrcadlení, stříh, stlačení, roztažení.

Tato metoda nemusí produkovat realistické deformace protože minimalizační funkce dovoluje nerovnoměrné stlačení a roztažení, a stříh. [29]

$$f_a(v) = \sum_j A_j \hat{q}_j + q_* \quad (2.18)$$

kde A_j je skalární hodnota, závislá pouze na p_i a v

$$A_j = (v - p_*) \left(\sum_i \hat{p}_i^T w_i \hat{p}_i \right)^{-1} \hat{p}_j^T \quad (2.19)$$

význam jednotlivých členů rovnice je objasněn v předcházejícím odstavci

Deformace se zachováním podobnosti (Similarity)

Jedná se o speciální podskupinu afinních deformací obsahující pouze translaci, rotaci a rovnoměrné stlačování s roztahováním. Tato varianta oproti předchozí metodě produkuje realističtější deformace. [29]

Deformační funkce je získána omezením transformační matice M

$$M = [M_1 \quad M_2] \quad (2.20)$$

kde M_1, M_2 jsou sloupcové vektory 2×1 .

$$M_1^T M_1 = M_2^T M_2 = \lambda^2 \text{ a } M_1^T M_2 = 0 \rightarrow M_2 = M_1^\perp \quad (2.21)$$

Deformační funkce $f_s(v)$ pro body na pravidelné mřížce

$$f_s(v) = \sum_i \hat{q}_i \left(\frac{1}{\mu_s} A_i \right) + q_* \quad (2.22)$$

kde μ_s a A_i jsou závislé pouze na p_i a v , mohou být předpočteny.

$$\mu_s = \sum_i w_i \hat{p}_i \hat{p}_i^T \quad (2.23)$$

$$A_i = w_i \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i^\perp \end{pmatrix} \begin{pmatrix} v - p_* \\ -(v - p_*)^\perp \end{pmatrix}^T \quad (2.24)$$

Tuhé deformace (Rigid)

Nedovoluje střih, roztahování, stlačování. Tato deformace je vhodná při požadavku zachování tvaru. [29].

Matice deformace M je ortogonální $M^T M = I$. Deformační funkce $f_r(v)$ pro body na pravidelné mřížce

$$f_r(v) = |v - p_*| \frac{\vec{f}_r(v)}{|\vec{f}_r(v)|} + q_* \quad (2.25)$$

kde $\vec{f}_r(v)$ lze opět předpočítat

$$\vec{f}_r(v) = \sum_i \hat{q}_i A_i \quad (2.26)$$

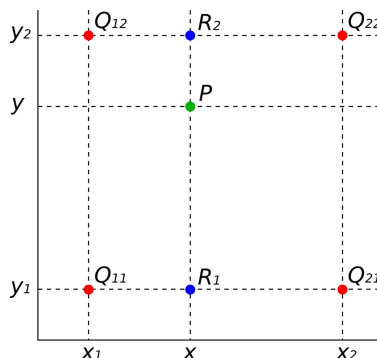
,kde A_i podle (2.24).

2.5.2. Interpolace obrazových dat podle výsledku MLS

Výstupem analytického řešení MLS $f(v)$ jsou deformované pozice bodů v , ve kterých proběhlo vyhodnocení, tyto body tvoří pravidelnou mřížku.

Deformovanou polohu bodů na zbylé ploše obrazu je možno interpolovat na jednotlivých čtyřhranech tvořících mřížku v . Po nalezení deformované polohy všech bodů obrazu je nutné provést interpolaci hodnoty bodů (pixelu). Interpolaci je tak nutno provést ve dvou krocích.

K interpolaci je použita metoda bilineární interpolace, používá váženého průměru čtyř okolních pixelů pro výpočet finální interpolované hodnoty. Výsledkem je vyhlazení obrazu. Algoritmus provádí interpolaci v horizontálním i vertikálním směru. Tato metoda dosahuje lepšího výsledku než nearest neighbour (zaokrouhlením vybírající nejbližší pixel) a výpočetní čas je nižší než bikubická interpolace (zvažující 4×4 okolí celkem tedy 16 okolních pixelů, kdy bližším pixelům je při výpočtu dána vyšší váha) [28].



Obrázek 2.13: Interpolační čtverec [26]

Rovnice bilineární interpolace ve variantě pro interpolaci na jednotkovém čtverci s počátkem v nule [26], do které lze převést obecný čtverec/obdélník úpravou $x' = x/(x_2 - x_1)$, $y' = y/(y_2 - y_1)$, význam Q_{nm} , x_n , y_n dle obr. 2.13

$$f(x, y) = [f(Q_{11}) \cdot (1 - x') + f(Q_{21}) \cdot x'] \cdot (1 - y') + [f(Q_{12}) \cdot (1 - x') + f(Q_{22}) \cdot x'] \cdot y' \quad (2.27)$$

kde pro vyšší efektivitu algoritmu je část výrazu v hranatých závorkách, závislá pouze na x , počítána pouze jednou pro všechny y s konkrétním x .

Při deformování obrazu je deformace $f(v)$ ve tvaru dopředného mapování, to přináší dva problémy: mezery a překrytí ve výsledném obrazu. Standardním řešením je převedení na inverzní transformaci v jednotlivých čtvercích/obdélnících mřížky a poté pro každý pixel výstupního obrazu nalézt pozici ve zdrojovém obraze a interpolovat [27].

Jednodušším řešením umožňující již zmíněnou výhodu předpočítání výrazu v hranatých závorkách (2.27), je zvýšení vzorkování zdrojového obrazu a tím zamezení vzniku mezer ve výstupním obraze, překrytí pak pouze způsobuje, že daný pixel může být interpolován více než jednou a sníží tak rychlost výpočtu. Pro omezení překryvu je velikost kroku ve zdrojovém obraze vždy pro každý jednotlivý čtverec/obdélník ze sklonu hodnot $f(v)$, analytického řešení deformace MLS, v jeho vrcholech stanovena následujícím způsobem:

$$i, j = \frac{1}{\frac{\max(f(Q_n)) - \min(f(Q_n))}{\min\{x_2 - x_1, y_2 - y_1\}}} \quad (2.28)$$

2.5.3. Získání cílových bodů pro před-deformaci obrazu

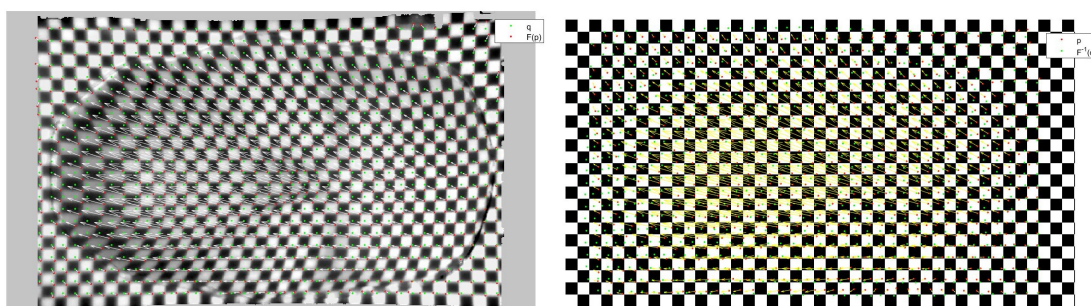
S možností identifikovat deformaci obrazu v diskrétních bodech - rozích šachovnice a jejich registraci oproti rohům ve zdrojovém obraze, společně s metodou deformace obrazu

využívající počáteční a koncové body, je zbývajícím krokem zvolení těchto kontrolních bodů.

Pokud zavedeme notaci pro souřadnice rohů ve zdrojovém obraze šachovnice jako p , pak lze detekované rohy v deformované šachovnici označit jako $F(p)$. Pro rekonstituovanou polohu bodů p v deformovaném obraze zavedeme značení q .

Poloha bodů q je stanovena translací těžiště mřížky bodů p do těžiště deformovaných bodů $F(p)$ a z průměrné pozice prvních a posledních řádků a sloupců deformované mřížky $|F(p)|$ je stanoveno měřítko bodů p .

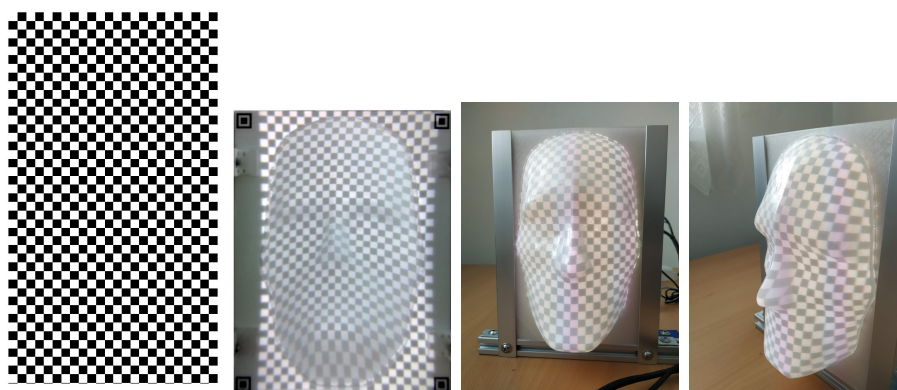
Pro nalezení bodů $F^{-1}(q)$, do kterých je třeba před-deformovat zdrojový obraz, provedeme výpočet $f(v)$ metodou MLS, kde jako počáteční poloha kontrolních bodů jsou zvoleny body $F(p)$ a cílovými body jsou p . Touto deformací je aproximována inverzní funkce přenosu optické soustavy F^{-1} . Interpolací deformace $f(x)$ v bodech q získáme body $F^{-1}(q)$, znalostí těchto bodů umožňuje před-deformování zdrojového obrazu metodou MLS $p \rightarrow F^{-1}(q)$ a tím rekonstrukci, soustavou deformovaného, promítaného obrazu.



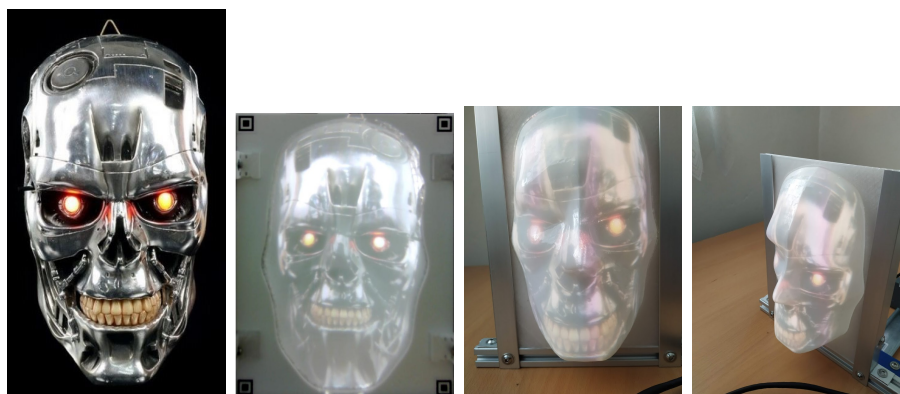
(a) Zobrazení bodů $F(p)$, q v deformovaném obraze (b) Zobrazení bodů p , $F^{-1}(q)$ ve zdrojovém obraze

Obrázek 2.14: Cílové body MLS [autor]

2.6. Ukázka

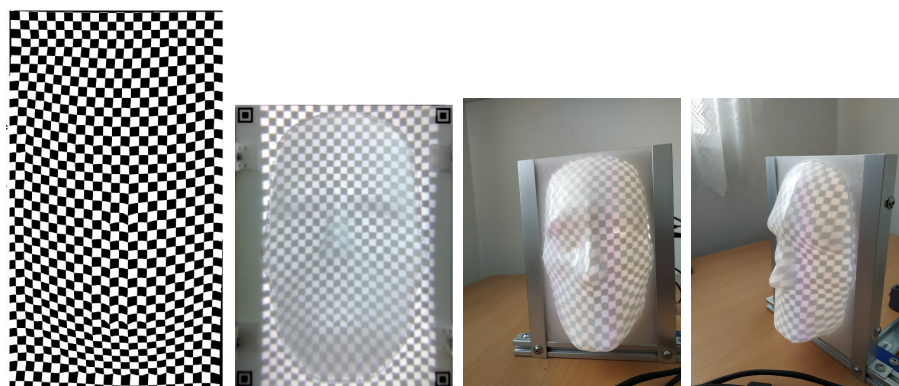


(a) Promítaný obraz (b) Snímek ze zadní strany (c) Snímek ze přední strany (d) Snímek z úhlu



(e) Promítaný obraz [43] (f) Snímek ze zadní strany (g) Snímek ze přední strany (h) Snímek z úhlu

Obrázek 2.15: Nekorigované snímky [autor]



(a) Promítaný obraz (b) Snímek ze zadní strany (c) Snímek ze přední strany (d) Snímek z úhlu



(e) Promítaný obraz (translace (f) Snímek ze zadní strany (g) Snímek ze přední strany (h) Snímek z úhlu
MLS)[43]

Obrázek 2.16: Korigované snímky před-deformací [autor]

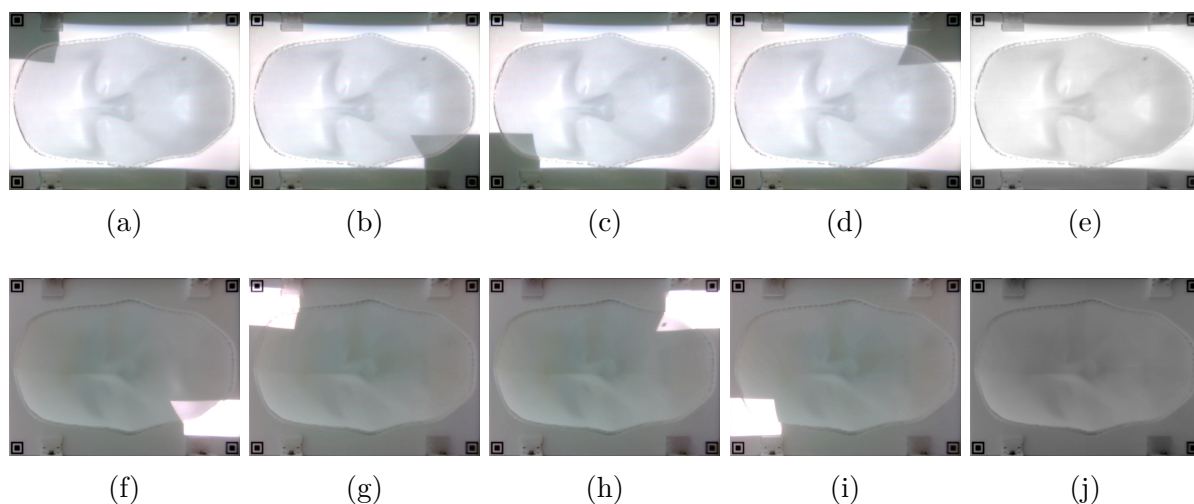
3. Úprava kontrastu promítaného obrazu

Eliminaci rozdílného kontrastu provedeme lokální jasovou transformací na základě neosvětleného obrazu (dark field) a osvětleného obrazu (clear field). Rovnice lokální jasové transformace [45]

$$I_c = (I - \text{darkfield}) / (\text{clearfield} - \text{darkfield}); \quad (3.1)$$

Vzhledem k provázání autoexpozure kamery a band suppression (interferenční pruhy) kamery nebylo vhodné provést manuální nastavení expozure doby kamery.

Do rohů promítaného obrazu jsou proto přidány referenční jasové oblasti (obr. 3.2) a obrazy jsou pořízeny kamerou v sekvenci (obr. 3.1), následně je na nich provedena transformace perspektivy podle kap. 2.4 a jejich jasové hodnoty tmavé a světlé, rozdílné následkem autoexpozure kamery, pomocí referenčních jasových oblastí v promítaném obrazu sesouhlasny. Detekce těchto oblastí je provedena binarizací snímku (obr. 3.2) od kterého je odečten snímek s opačnou hodnotu jasu v dané referenční oblasti.



Obrázek 3.1: Sesouhlasení hodnot jasu, c),j) výstupní obraz [autor]



Obrázek 3.2: Nalezené referenční jasové oblasti [autor]

Takto upravené obrazy jsou použity pro úpravu jasu vstupních snímků v kap. 2.4

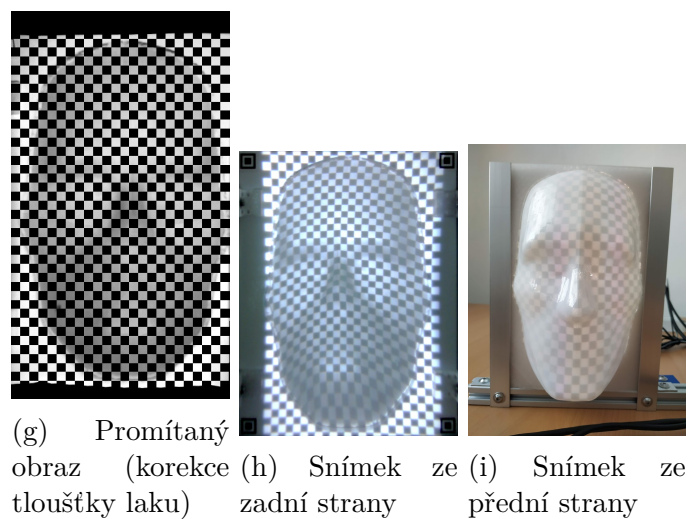
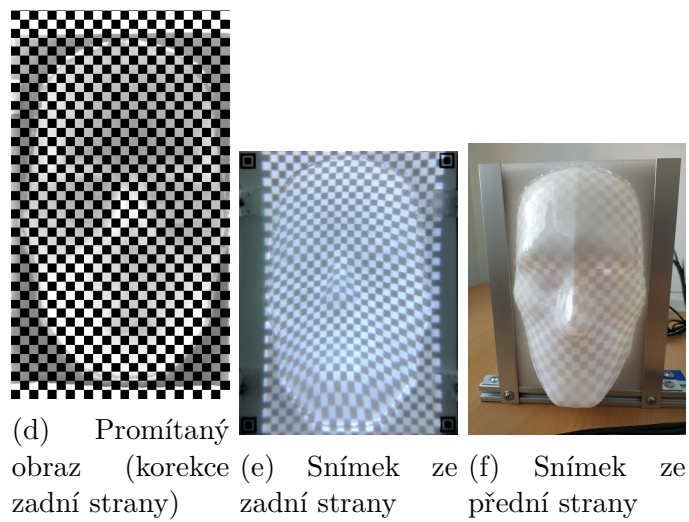
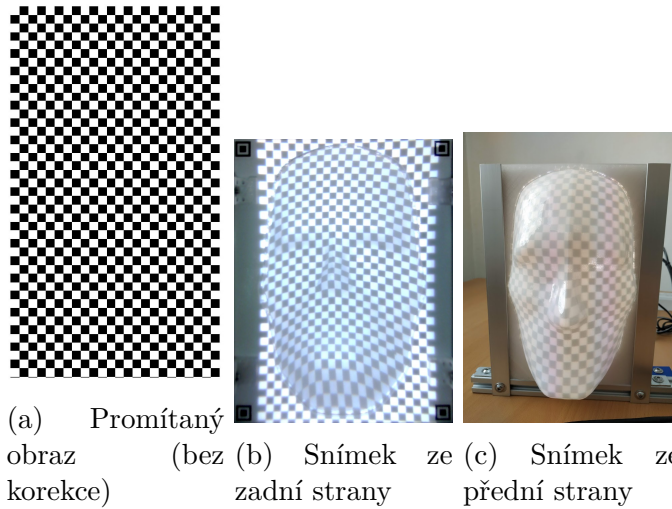
Pro korekci promítaného obrazu jsou sesouhlasené darkfield a clearfield snímky dále deformovány algoritmem MLS (kap. 2.5) na základě bodů získaných detekcí rohu šachovnice v kap. 2.5.3. Jako počáteční body (podle notace v kap. 2.5.3) jsou zvoleny $F(p)$ a cílové $p =$ aproximace inverzní funkce F^{-1} .

Rozdíl snímků darkfield a clearfield nese informaci o jasu promítaného obrazu v daném pixelu.

Negativem tohoto rozdílu snímku je stanovena kompenzace zdrojového obrazu. Tento negativ není v rozsahu od 0-1, proto je jeho maximální hodnota roztažením histogramu (násobením obrazu) a lineárním posuvem (přičtením) přesunuta blíže k 1 pro co nejmenší snížení celkového jasů promítaného obrazu.

Tato korekce provede normalizaci jasů ze zadní strany projekční plochy. Pro zpětnou projekci má však význam přední strana. Proto z hypotézy, že jasná místa zachycená kamerou jsou způsobena vyšší tloušťkou barvy masky - odrazem, je provedeno zvýšení rozdílu jasů. Rozdíl snímku darkfield a clearfield je roztažen a posunut do 1. Porovnání jasových korekcí obr. 3.3.

Ukázka promítaného obrazu před a po korekci



Obrázek 3.3: Korekce jasu [autor]

4. Testovací soustava

4.1. Popis HW

Fyzická testovací soustava obr. 2.1 pro zpětnou projekci je složena z projektoru, rámu, webkamery a projekční plochy - průsvitné masky. K projektoru byla přidána externí optika za účelem zkrácení projekční vzdálenosti. V následujících podkapitolách jsou tyto podsestavy a jejich výroba blíže popsána.

4.1.1. Polopropustná maska

Polopropustná maska byla vyrobena technologií vakuového formování za pomoci Strojlabu FSI Brno.

Pro vytvoření formy/kopyta pro vakuové formování je nutné zvolit materiál odolávající teplotám tavení termoplastů, při prvním pokusu byla, ve snaze snížení nákladů, otestována forma vytištěná z PETG na FDM 3D tiskárně, tato forma nebyla předeřhřívána a selhala již při druhém pokusu, kdy došlo k natavení a deformování oblasti nosu, oblasti s větší kontaktní plochou však ještě odolávaly.

Při druhém, již úspěšném, pokusu byla použita sádrová forma.

Model masky byl vytvořen v programu Blender, následně vyexportován do Solidworks, kde byla vymodelována forma pro sádrový odlitek. Forma pro sádrový odlitek byla vytištěna na FDM 3D tiskárně z PLA. Sádrový odlitek pak při vakuovém formování PETG 0.6mm desky sloužil jako forma/kopyto.

Čirá vytvarovaná maska byla pro možnost projekce nasprejována tenkou vrstvou bílého matného laku RAL9010M, ze zadní strany pro snížení odlesků projektoru snímaných kamerou.



Obrázek 4.1: Forma pro sádrový odlitek, sádrový odlitek, zformovaná maska [autor]

4.1.2. Popis konstrukce

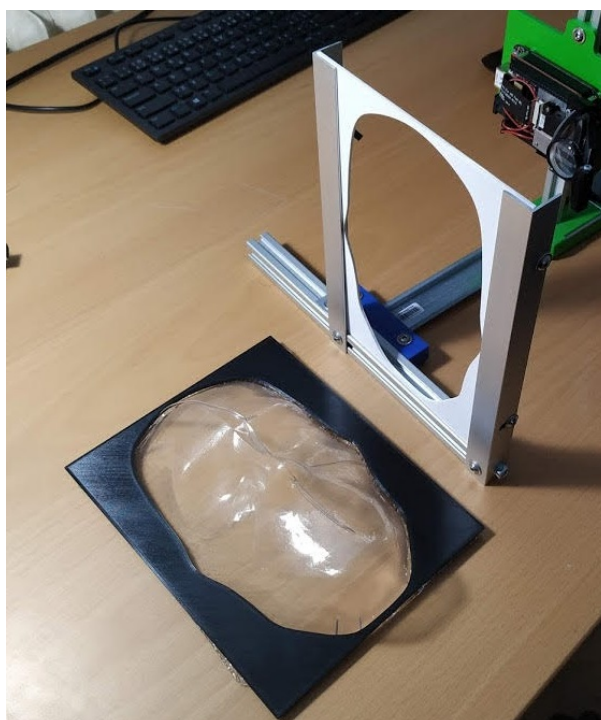
Rám je vyroben kombinací hliníkových 20x20 a L profilů, DIN lišty, spojených plastovými díly vytištěnými na FDM 3D tiskárně.

K uchycení masky byly vytištěny rámečky (obr. 4.2) s dírou odpovídající odsazenému obrysu masky. Tyto rámečky přiložené z obou stran masky jsou následně uchyceny v hliníkových L profilech s drážkou pro vedení přítlačných držáků. Na rámečku masky ze strany projektoru jsou nalepeny QR finder vzory umožňující korekci perspektivy představené v kap. 2.3.

Sestava masky, rámečků a hliníkových L profilů je šrouby uchycena k posuvným maticím uvnitř drážky 20x20 hliníkového profilu. Tento 20x20 profil je následně spojen pomocí vytištěného adaptéru [38] s DIN lištou, adaptér po uvolnění šroubů umožňuje posun 20x20 profilu a tím i masky v ose X.

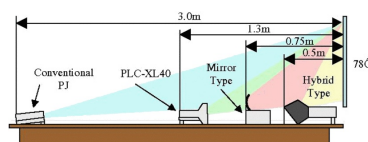
DIN lišta tvoří posuv osy Y pro adaptér Z osy. Adaptér Z osy je pomocí stavěcích šroubů možno zaaretovat v žádané poloze.

Pro osu Z byl opět použit 20x20 profil, sestavě projektoru je umožněn posuv v drážce 20x20 profilu, aretace je prováděna šrouby s posuvnou maticí.



Obrázek 4.2: Systém uchycení masky za pomocí rámečku [autor]

Za účelem zkrácení projekční vzdálenosti a vytvoření tzv. short throw projektoru, byla k TI DLP Projektoru (kap. 4.1.3) přidána soustava čoček a zrcadla. Z uvažovaných typů konstrukcí projektoru v [39] vychází jako nejefektivnější soustava hybridní kombinující jak čočky tak (konkávní) zrcadlo, obr.4.3 se srovnáním jednotlivých typů konstrukce.



Obrázek 4.3: Porovnání jednotlivých konstrukcí projektorů a jejich projekčních vzdáleností [39]

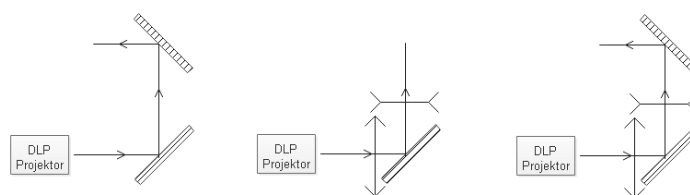
Při volbě čoček je z požadavku jednoduchosti konstrukce a nákladů nepřevyšující cenu samotného projektoru omezen počet čoček na dvě, konkávní zrcadlo nahrazeno plochým,

kteře zalomením optické dráhy odečte z celkové vzdálenosti od projekční plochy rozměry BeagleBone.

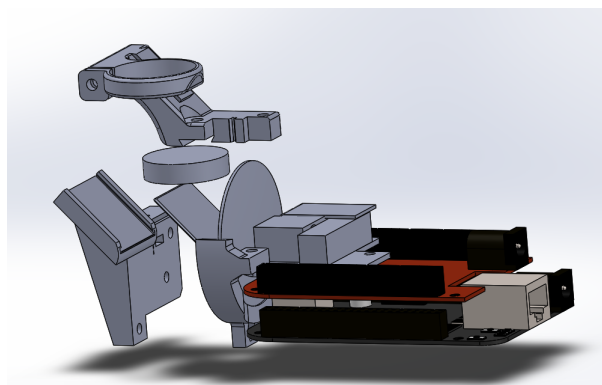
Při výběru čoček byla použita spojka získána z optiky otočného reflektoru, ze které vznikal první prototyp mechanické platformy robotické hlavy, odpovídající rozptylka pak byla zvolena pro dosažení požadované ohniskové vzdálenosti a tím zaostřeného obrazu. Projektor samotný disponuje manuálním doostřením (obr. 4.8c).

Mechanická konstrukce držáku optiky (obr. 4.5) byla provedena 3D tiskem, má modulární charakter pro akomodaci všech uvažovaných řešení, vzdálenější zrcadlo je možno upevnit externě, pro testovací soustavu nebylo použito.

Pro získání obrazu je k soustavě připevněna Waveshare IMX179 8MPx USB kamera, která svoji optikou, při dané projekční vzdálenosti, umožňuje oproti webkamerám záběr celé projekční plochy.



Obrázek 4.4: Schematické řešení konstrukce [autor]



Obrázek 4.5: Exploze modelu přídavné optické soustavy [autor]

Dle specifikace DLP EVM (obr. 4.7) je throw ratio (poměr vzdálenosti od projektoru ku šířce obrazu) 1,6. Při kombinaci čoček a zrcadla dosahuje při projekční vzdálenosti 325 mm obraz šířky A4 (297 mm), throw ratio je přibližně 1,1. Při kombinaci dvou čoček a dvou zrcadel dosahuje této šířky obrazu při projekční vzdálenosti 260 mm, poměr je tedy 0,87, čímž spadá do kategorie short throw projektorů (0,38-1,4 dle [42]).



(a) 2 čočky, 2 zrcadla

(b) 2 čočky, 1 zrcadlo

Obrázek 4.6: Měření projekční vzdálenosti pro jednotlivé konstrukce přídavné optické soustavy [autor]

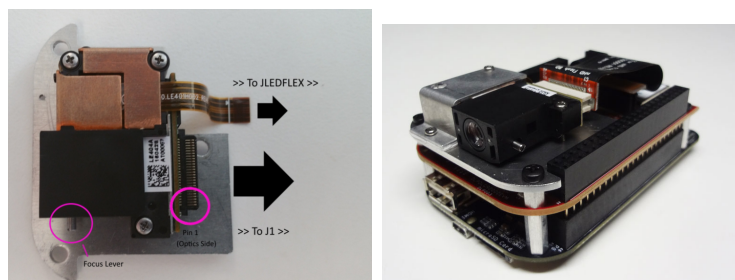
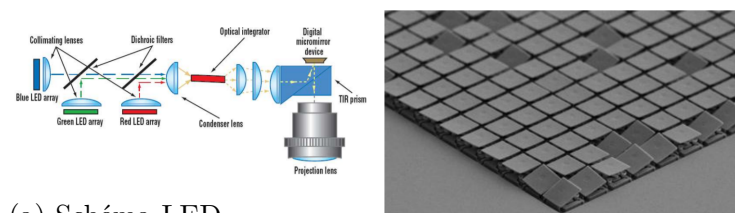
4.1.3. TI DLP Projektor + BeagleBone Black

Při výběru cenově dostupných projektorů a rozhodování mezi variantou koupě některého z LED projektorů z eBay, který by byl následně kompletně rozmontován, zvítězil vývojový modul DLP® LightCrafter™ Display 2000 EVM od Texas Instruments (obr. 4.8d), cenově nejdostupnější platforma demonstrující TI DLP micro-electro-mechanical systems (MEMS) technologii, která k modulaci světla používá Digital Micromirror Device (DMD) - detail zobrazený v obr. 4.8b

DLP DMD, společně s přidruženou elektronikou, zdrojem světla, optikou a nutnými mechanickými komponenty, jsou kombinovány do kompaktního optického modulu neboli optical engine (obr. 4.8a, 4.8c). Vývojový modul dále obsahuje LED driver a DLP kontrolér (obr. 4.10, specifikace obr.4.7) přijímající obrazová data a příkazy pro konfiguraci od externího procesoru (BeagleBone Black, Raspberry Pi, převaděč HDMI do RGB888). Jako externí procesor byl použit BeagleBone Black (BBB), pro který je DLP vývojový modul koncipován, zasílání obrazových dat z počítače prostřednictvím BBB a ovládání projektoru je popsáno v kap. 4.2.1.

PARAMETER	MIN	TYP	MAX	UNIT
Brightness		20	30 ⁽¹⁾	lm
LED Current	100	320	500	mA
Brightness Uniformity	75%			
Throw Ratio		1.6		
Offset		100%		
Focus Range	10		60	inch
Image Diagonal Size	10		60	inch

Obrázek 4.7: Specifikace DLP LightCrafter Display 2000 EVM [14]



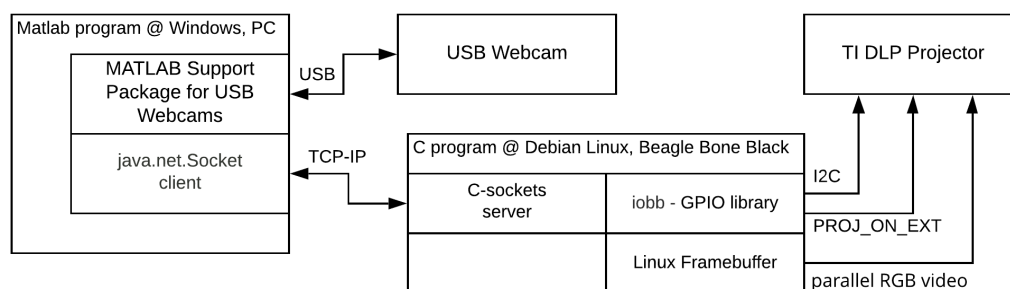
Obrázek 4.8: TI DLP Projektor EVM, TI BeagleBone Black (BBB)

4.2. Popis SW

Soustava se z softwarové strany skládá z Matlab programu běžícího na Windows PC, který provádí zpracování obrazu, a C programu běžícího na Linux BBB, který řídí TI DLP projektor. Fyzická vrstva pro komunikaci z webkamerou je USB, pro komunikaci s BeagleBone Black (BBB) je 100BASE-T Ethernet.

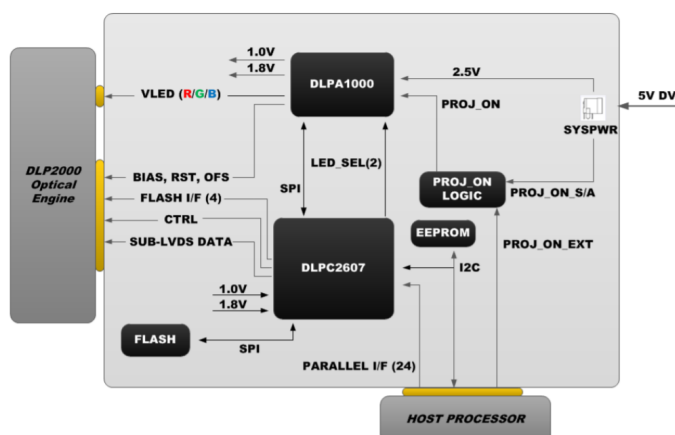
Matlab a C program spolu komunikují přes TCP, C program běžící na BBB sloužící jako server a Matlab program jako klient.

Byl vytvořen komunikační protokol specifikující prvními 4 byty velikost zprávy, 5 byte obsahuje stavové informace/příkazy a umožňuje tak ovládání projektoru, volitelně následovanými obrazovými daty (maximálně 2^{32}). C server je blíže popsán v podkapitole 4.2.1, Matlab program v podkapitolách 4.2.2, 4.2.3



Obrázek 4.9: Schéma komunikace [autor]

4.2.1. BeagleBone Black + DLP projektor



Obrázek 4.10: Blokový diagram DLP LightCrafter Display 2000 EVM[14]

Po správném nastavení operačního systému (Texas Instruments vydal předkonfigurovaný Debian Jessie (verze 8.9) [14]), BeagleBone Black (BBB) při bootování komunikuje s EEPROM DLP (znázorněna na 4.10) a provede načtení konfigurace GPIO (cape overlay)[14].

BeagleBone poté může komunikovat s DLP projektorem pomocí [14]:

- Paralelní I/F video data (RGB888)
- I²C příkazy
- Zapnutí/vypnutí DLP (PROJ_ON_EXT)

Automatické spuštění C programu, sloužícího jako server při komunikaci s Matlab, po startu operačního systému Linux Debian je provedeno pomocí démonu pro správu systému *systemd* [32] podle příkladu v [33].

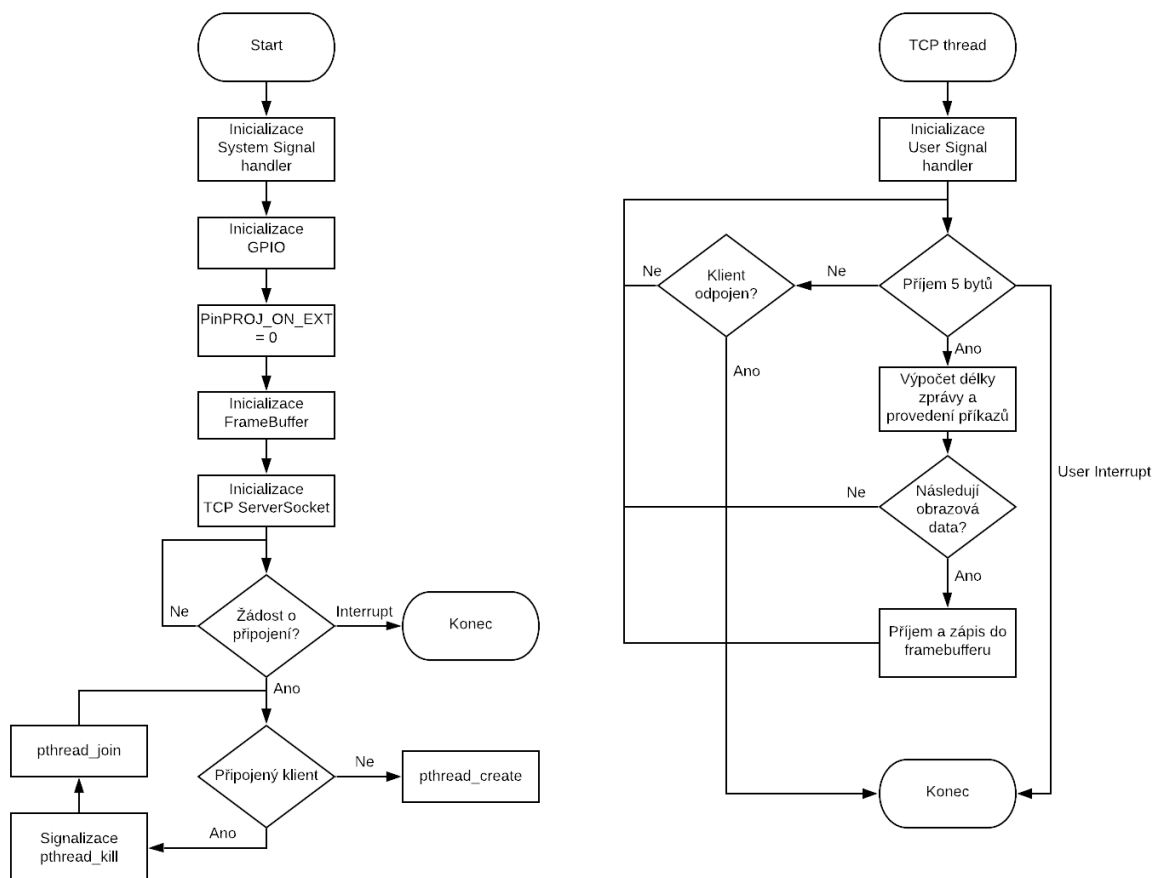
Popis funkcionality C programu běžícího na BBB

Pro ovládání projektoru je implementováno ovládání GPIO PROJ_ON_EXT s využitím GPIO knihovny *iobb* [15] a pro zasílání I²C příkazů použita C funkce *system()* [34] pro konfiguraci akceptování obrazových dat DLP projektorem, projektor může přijímat další příkazy [14], ty však nejsou implementovány.

TCP server (implementován pomocí příkladů z [22, 23, 24, 25]) pro zaručení dostupnosti využívá knihovny *pthread.h* - hlavní vlákno sleduje žádosti o nová připojení a v případě, že došlo k novému připojení a předchozí klient, jehož komunikace běží na jiném vlákně, předem řádně neukončil komunikaci (deadlock), vynutí ukončení již připojeného klienta pomocí signalizace *pthread_kill* [19, 20, 21]

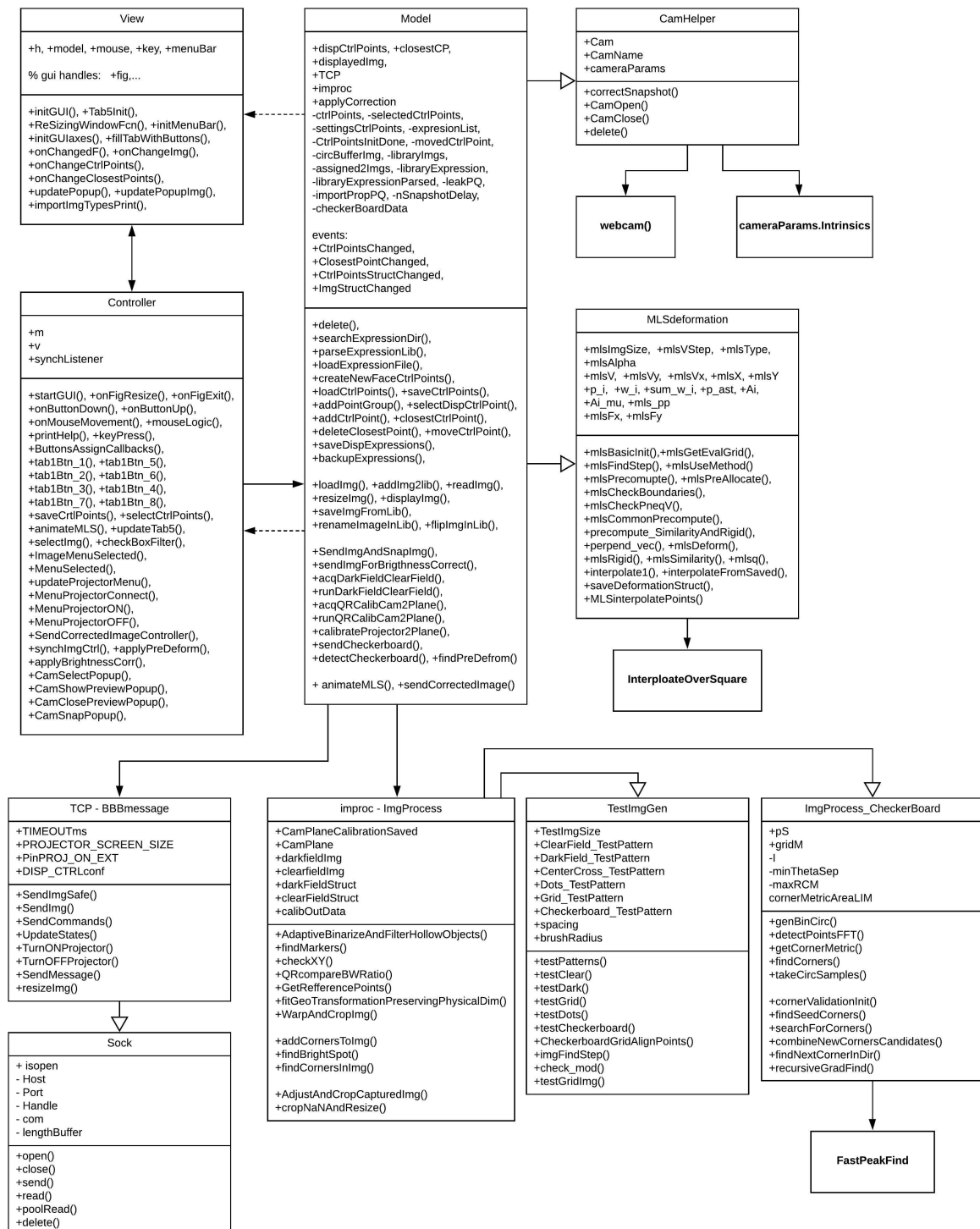
Spojení s klientem *clientSocket* je provedeno v novém vlákně *pthread_create* a je ponecháváno otevřené podobně jako HTTP2 pro snížení zpoždění způsobené navazováním komunikace.

Obdržená obrazová data, z TCP komunikace z Matlab, jsou v BGRA formátu zapisována do Linux Framebufferu [16], implementováno podle příkladů v [17, 18].



Obrázek 4.11: Znázornění C programu běžícího na BeagleBone Black [autor]

4.2.2. Matlab program PC



Obrázek 4.12: UML Class diagram[autor]

Program v Matlab následuje designový postup model, controller, view, rozděluje tak logiku grafického rozhraní (view, controller) a algoritmů (model).

Pro TCP komunikaci s BeagleBoneBlack jsou využity javaSockets [44], rozhraní pro zasílání zpráv je rozděleno do class *TCP-BBBmessage.m* - implementující protokol a *Sock* (zdrojový kód pro *Sock.m* získán od MechLab) abstrakce pro javaSockets.

Class *Model.m* implementuje struktury pro ukládání a načítání obrazových dat a kontrolních bodů pro deformaci MLS. Dále dědí z class *Improc.m*, *CamHelper*, *MLSDeformation.m*.

Class *Improc.m* implementující algoritmy detekce šachovnice a korekci jasu uvedené v kap. 2.4 a 3 podle zdrojů v těchto kapitolách uvedených, jako externí kód je použita funkce *FastPeakFind.m* [12].

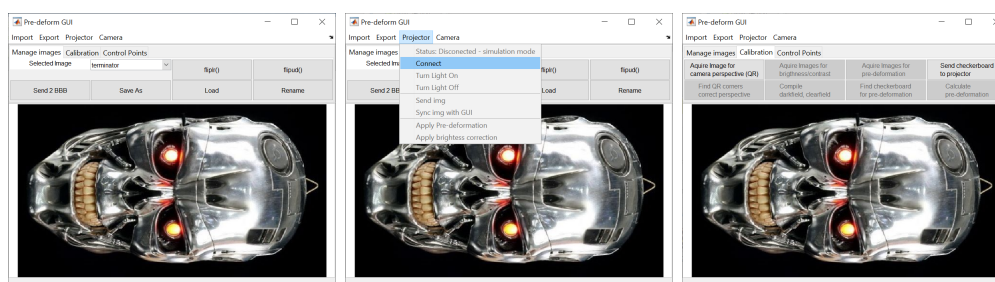
Class *MLSDeformation.m* implementující algoritmus pro deformaci obrazu (kap. 2.5) a bilineární interpolaci (kap. 2.5.2) podle zdrojů uvedených v těchto kapitolách.

Class *CamHelper.m* ovládá webkameru a eliminuje zkreslení optiky webkamery z uložených parametrů *cameraParams*.

Metody *Model.m* dále koordinují získávání obrazu z webkamery a jednotlivé operace zpracování obrazu.

Funkcionality Class *Controler.m* a *View.m* jsou popsány v následující podkapitole.

4.2.3. Matlab uživatelské prostředí



Obrázek 4.13: GUI [autor]

Uživatelské prostředí *Pre-deform GUI* implementované v class *Controler.m* a *View.m* umožňuje

Importování a exportování libovolných, MatLabem podporovaných obrazových dat. V kartě *Manage images* je možno obraz překlopit pro získání správné orientace promítaného obrazu. Dropdown menu umožňuje vybrat z již ve struktuře načtených obrazů, jména obrazů ve struktuře je možno editovat.

Karta *Control points* umožňuje uživatelem zadat kontrolní body (popsáno v manuálu v příloze) pro funkci MLS, ta poté provede deformaci obrazu, deformovaný obraz je poté přidán do struktury a je možno ho vyexportovat nebo promítnout projektorem. Jednotlivým kontrolním bodům je možno přiřadit název a exportovat. Exportované body je poté možno načíst.

Karta *Calibration* umožňuje zadat parametry do funkcí pro zpracování obrazu (kalibrace perspektivy kamery, jasu a deformace). Jednotlivé kroky jsou postupně odemykány.

Dropdown menu *Projektor* ovládá TCP připojení k projektoru, zobrazuje stav a umožňuje zasílat příkaz zapnutí a vypnutí projektoru. Duplikuje možnost zaslání obrazu z panelu *Manage images*, navíc umožňuje přidání automatické synchronizace zobrazeného a promítaného obrazu. Obsahuje také volbu zda provést předdeformaci a jasovou kalibraci.

Dropdown menu *Camera* umožňuje volbu webkamery, zobrazení náhledu a pořízení snímku.

5. Závěr

Byl vybrán cenově dostupný projektor TI DLP vývojový modul a zprovozněn jeho kontrolér BeagleBone Black. Pro ovládání projektoru a zasílání obrazových dat byl vytvořen na straně BeagleBone Black TCP server v jazyce C využívající knihovny pthreads.h pro zamezení deadlocku v případě neukončení komunikace.

Byla zkonstruována testovací soustava skládající se z projekční masky, vyrobené technologií vakuového formování za podpory StrojLab FSI VUT Brno a rámu umožňující posuv ve třech osách, za účelem zkrácení projekční vzdálenosti byla k projektoru přidána externí optika.

Byl vytvořen program v MatLab umožňující načtení a editaci obrazových dat v grafickém uživatelském rozhraní (GUI) a jejich následnému promítnutí.

Za účelem eliminace zkreslení obrazu deformací byl navržen a implementován postup skládající se z kalibrace zkreslení optiky kamery, kompenzace zkreslení vyvolané perspektivou kamery, identifikace zkreslení, způsobené projekční plochou a přídatnou optikou, postup je zakončený předdeformací zdrojových obrazových dat za účelem rekonstrukce promítaného, soustavou zkresleného, obrazu.

Kalibrace kamery byla provedena s využitím Matlab Camera Calibrator app, získané parametry pak uloženy.

Pro odstranění perspektivního zkreslení byly na rámeček projekční masky umístěny QR finer vzory, jejich detekce implementována v Matlab dosahuje v příznivých světelných podmínkách dobrých výsledků, při ostrém osvětlení odlesky inkoustu QR finder vzorů komplikují binarizaci obrazu.

Pro identifikaci zkreslení byl zvolen promítací vzor šachovnice, detekce rohů šachovnice je provedena Fourierovou transformací prstencového vzorku okolí pixelu a na něm charakteristického průběhu o dvou periodách v rozích šachovnice. Za účelem získání struktury šachovnice byl implementován algoritmus postupného růstu kandidátních bodů ve všech směrech, s výběrem následujícího rohu s využitím hrany šachovnice. Kombinace těchto algoritmů umožňuje spolehlivou detekci struktury šachovnice i v silně deformovaných místech šachovnice.

Po identifikaci deformace je pro rekonstrukci bodů v obraze implementován algoritmus pohyblivých vážených čtverců (MLS), umožňující na základě zadání počátečních a koncových bodů, získaných z rohů šachovnice, aproximovat deformaci pro celý obraz. Při velkém počtu kontrolních bodů dochází ke značnému zpomalení výpočtu, to je možno řešit rozšířením algoritmu o funkci upravující počet uvažovaných kontrolních bodů pouze na body v dynamicky definovaném okolí vyhodnocovaného bodu.

Nalezené deformace s výpočtem 1-5 min jsou však po výpočtu uloženy a jejich aplikace na zdrojový obraz je pak omezena pouze rychlostí bilineární interpolace, která je v řádu 0.1 s, oproti výpočtu s vyšším počtem bodů zanedbatelná. Při zadání menšího počtu bodů (2-15) jsou deformace obrazu nalezeny řadově v 0.1 - 1 s.

Jasová kalibrace je provedena po nalezení deformace, snímky osvětlené a neosvětlené projekční plochy jsou deformovány za účelem rekonstrukce do zdrojového obrazu. Po deformaci je informace o jasu promítaného pixelu převedena do zdrojového obrazu. Následným vynásobením obrazu je jas před promítnutím lokálně korigován. Při změně osvětlení plochy je možno, z uložené deformace získané z rohů šachovnice, tuto kalibraci rychle opakovat.

Literatura

- [1] SUKTHANKAR, Rahul a Matthew D. MULLIN. Automatic Keystone Correction for Camera-Assisted Presentation Interfaces. TAN, Tieniu, Yuanchun SHI a Wen GAO, ed. *Advances in Multimodal Interfaces — ICMI 2000* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, 2000-10-26, s. 607-614 [cit. 2020-06-11]. *Lecture Notes in Computer Science*. DOI: 10.1007/3-540-40063-X_79. ISBN 978-3-540-41180-2. Dostupné z: <http://link.springer.com/10.1007/3-540-40063-X9>
- [2] JAN, Jiří. Číslicová filtrace, analýza a restaurace signálů. 2. upr. a rozš. vyd. Brno: VUTIUM, 2002. ISBN 8021415584.
- [3] MALLON, John a Paul F. WHELAN. Which pattern? Biasing aspects of planar calibration patterns and detection methods. *Pattern Recognition Letters* [online]. 2007, 28(8), 921-930 [cit. 2020-06-12]. DOI: 10.1016/j.patrec.2006.12.008. ISSN 01678655. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0167865506003114>
- [4] Which pattern (circle pattern or checkerboard pattern) should be used for automotive camera calibration (fisheye/wide webcam)?. *ResearchGate | Find and share research* [online]. Copyright (c) 2008 [cit. 12.06.2020]. Dostupné z: https://www.researchgate.net/post/Which_pattern_circle_pattern_or_checkerboard_pattern_should_be_used_for_automotive_camera_calibration_fisheye_wide_webcam
- [5] computer vision - camera calibration: why chessboards? - *Signal Processing Stack Exchange*. *Signal Processing Stack Exchange* [online]. Dostupné z: <https://dsp.stackexchange.com/questions/24734/camera-calibration-why-chessboards>
- [6] Detect checkerboard pattern in image - *MATLAB detectCheckerboardPoints*. *MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink* [online]. Copyright (c) 1994 [cit. 12.06.2020]. Dostupné z: <https://www.mathworks.com/help/vision/ref/detectcheckerboardpoints.html#d120e133366>
- [7] YAN, Yizhen, Peng YANG, Lei YAN, Jie WAN, Yanbiao SUN a Kevin TANSEY. Automatic checkerboard detection for camera calibration using self-correlation. *Journal of Electronic Imaging* [online]. 2018, 27(03) [cit. 2020-06-12]. DOI: 10.1117/1.JEI.27.3.033014. ISSN 1017-9909. Dostupné z: <https://www.spiedigitallibrary.org/journals/journal-of-electronic-imaging/volume-27/issue-03/033014/Automatic-checkerboard-detection-for-camera-calibration-using-self-correlation/10.1117/1.JEI.27.3.033014.full>
- [8] ZHANG, Qi a Caihua XIONG. A New Chessboard Corner Detection Algorithm with Simple Thresholding. HUANG, YongAn, Hao WU, Honghai LIU a Zhouping YIN, ed. *Intelligent Robotics and Applications* [online]. Cham: Springer International Publishing, 2017, 2017-08-06, s. 532-542 [cit. 2020-06-12]. *Lecture Notes in Computer Science*. DOI: 10.1007/978-3-319-65292-4_46. ISBN 978-3-319-65291-7. Dostupné z: http://link.springer.com/10.1007/978-3-319-65292-4_46

- [9] SPITSCHAN, Benjamin a Jörn OSTERMANN. Robust Fourier-Based Checkerboard Corner Detection for Camera Calibration. VERA-RODRIGUEZ, Ruben, Julian FIERREZ a Aythami MORALES, ed. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications [online]. Cham: Springer International Publishing, 2019, 2019-03-03, s. 538-546 [cit. 2020-06-12]. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-13469-3_63. ISBN 978-3-030-13468-6. Dostupné z: http://link.springer.com/10.1007/978-3-030-13469-3_63
- [10] BENNETT, Stuart a Joan LASENBY. ChESS – Quick and robust detection of chess-board features. Computer Vision and Image Understanding [online]. 2014, 118, 197-210 [cit. 2020-06-12]. DOI: 10.1016/j.cviu.2013.10.008. ISSN 10773142. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S1077314213001999>
- [11] DAO, Vinh Ninh a Masanori SUGIMOTO. A Robust Recognition Technique for Dense Checkerboard Patterns. In: 2010 20th International Conference on Pattern Recognition [online]. IEEE, 2010, 2010, s. 3081-3084 [cit. 2020-06-12]. DOI: 10.1109/ICPR.2010.755. ISBN 978-1-4244-7542-1. Dostupné z: <http://ieeexplore.ieee.org/document/5597291/>
- [12] Natan (2020). Fast 2D peak finder (<https://www.mathworks.com/matlabcentral/fileexchange/37388-fast-2d-peak-finder>), MATLAB Central File Exchange. Retrieved June 13, 2020.
- [13] MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink [online]. Copyright (c) 1994 [cit. 17.06.2020]. Dostupné z: https://www.mathworks.com/help/supportpkg/usbwebcams/index.html?s_tid=CRUX_lftnav
- [14] TI DLP(R) LightCrafter(TM) Display 2000 EVM User's Guide (Rev. C) [online]. [cit. 2020-06-18]. Dostupné z: <https://www.ti.com/lit/pdf/dlpu049>
- [15] SHABAZ. BeagleBone Black (BBB) and PocketBeagle I/O (GPIO), SPI and I2C Library for C – 2019 Edition [online]. [cit. 2020-06-18]. Dostupné z: https://www.element14.com/community/community/designcenter/single-board-computers/next-genbeaglebone/blog/2019/08/15/beaglebone-black-bbb-io-gpio-spi-and-i2c-library-for-c-2019-edition#jive_content_id_How_is_I2C_used
- [16] The Frame Buffer Device API — The Linux Kernel documentation. The Linux Kernel Archives [online]. Copyright (c) Copyright The kernel development community [cit. 18.06.2020]. Dostupné z: <https://www.kernel.org/doc/html/latest/fb/api.html>
- [17] AB1JX. Re: Writing to the framebuffer is slow [příspěvek v diskuzním fóru]. In: Raspberry Pi Forums [online]. [cit. 2020-06-18]. Dostupné z: <https://www.raspberrypi.org/forums/viewtopic.php?t=213964>
- [18] Raspberry Compote: Low-level Graphics on Raspberry Pi (part four) . Raspberry Compote [online]. Dostupné z: <http://raspberrypcompote.blogspot.com/2013/03/low-level-graphics-on-raspberry-pi-part.html>

- [19] MCCCCS. Catch Ctrl-C in C [příspěvek v diskuzním fóru]. In: Stack Overflow - Where Developers Learn, Share, & Build Careers [online]. Dostupné z: <https://stackoverflow.com/a/54267342>
- [20] Steven D, Igor Liferenko. How sigaction() differs from signal()? [příspěvek v diskuzním fóru]. In: Stack Overflow - Where Developers Learn, Share, & Build Careers [online]. Dostupné z: <https://stackoverflow.com/questions/45477254/how-sigaction-differs-from-signal>
- [21] Beej's Guide to Unix IPC. Beej's Web Page [online]. Dostupné z: <https://beej.us/guide/bgipc/>
- [22] DONAHOO, Michael J. a Kenneth L. CALVERT. TCP/IP sockets in C: practical guide for programmers. 2nd ed. Burlington: Morgan Kaufmann, c2009. Morgan Kaufmann practical guides series. ISBN 978-0123745408.
- [23] HALL, Brian "Beej Jorgensen". Beej's Guide to Network Programming: Using Internet Sockets 2002. ISBN 1705309909.
- [24] Server and client example with C sockets on Linux – BinaryTides. BinaryTides – Coding, Software, Tech and Reviews [online]. Copyright (c) 2020 [cit. 2020-06-18]. Dostupné z: <https://www.binarytides.com/server-client-example-c-sockets-linux/>
- [25] EDDIEA. tcp-nonblocking-server.c [online]. [cit. 2020-06-18]. Dostupné z: <http://www.cs.tau.ac.il/~eddiea/samples/Non-Blocking/tcp-nonblocking-server.c.html>
- [26] Bilinear interpolation. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-06-19]. Dostupné z: https://en.wikipedia.org/wiki/Bilinear_interpolation
- [27] Spatial transformations: Forward mapping » Steve on Image Processing and MATLAB - MATLAB & Simulink. MATLAB Central Blogs | Get the inside view on MATLAB & Simulink! [online]. Copyright © 1994 [cit. 2020-06-19]. Dostupné z: <https://blogs.mathworks.com/steve/2006/04/28/spatial-transforms-forward-mapping/>
- [28] PARSANIA, Mr. Pankaj S. a Dr. Paresh V. VIRPARIA. *A Comparative Analysis of Image Interpolation Algorithms*. IJARCCCE [online]. 2016, 5(1), 29-34 [cit. 2020-06-20]. DOI: 10.17148/IJARCCCE.2016.5107. ISSN 22781021. Dostupné z: <http://ijarcce.com/upload/2016/january-16/IJARCCCE%207.pdf>
- [29] SCHAEFER, Scott, Travis MCPHAIL a Joe WARREN. *Image deformation using moving least squares*. In: ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06 [online]. New York, New York, USA: ACM Press, 2006, 2006, s. 533- [cit. 2020-06-20]. DOI: 10.1145/1179352.1141920. ISBN 1595933646. Dostupné z: <https://www.cs.rice.edu/~jwarren/research/mls.pdf>

- [30] LEVIN, David. *The approximation power of moving least-squares*. *Mathematics of Computation* [online]. 1998, 67(224), 1517-1532 [cit. 2020-06-20]. DOI: 10.1090/S0025-5718-98-00974-0. ISSN 0025-5718. Dostupné z: <http://www.cs.jhu.edu/~misha/Fall05/Papers/levin98.pdf>
- [31] Moving least squares. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-06-21]. Dostupné z: https://en.wikipedia.org/wiki/Moving_least_squares
- [32] Systemd. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-06-21]. Dostupné z: <https://cs.wikipedia.org/wiki/Systemd>
- [33] FAIRHEAD, Harry. Applying C - Running Programs With Systemd. I-programmer.info [online]. 14 October 2019 [cit. 2020-06-21]. Dostupné z: <https://www.i-programmer.info/programming/cc/13148-applying-c-running-programs-with-systemd.html>
- [34] System(3) — Linux manual page. Linux Programmer's Manual [online]. [cit. 2020-06-21]. Dostupné z: <https://man7.org/linux/man-pages/man3/system.3.html>
- [35] HURBAN, M. QR code detection under ROS implemented on the GPU. Brno:Brno University of Technology, Faculty of Mechanical Engineering, 2017. 58 pp. [cit. 2020-06-23]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=149981. Supervisor doc. Ing. Jiří Krejsa, Ph.D..
- [36] Kei Otsuka (2020). QR Code reader example (<https://www.mathworks.com/matlabcentral/fileexchange/71090-qr-code-reader-example>), MATLAB Central File Exchange. Retrieved June 23, 2020.
- [37] Fisheye Calibration Basics. MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink [online]. Copyright (c) 1994 [cit. 2020-06-24]. Dostupné z: <https://www.mathworks.com/help/vision/ug/fisheye-calibration-basics.html>
- [38] Celt-X FRC Din Rail Mount Collection by CeltXRobotics - Thingiverse. Thingiverse - Digital Designs for Physical Objects [online]. [cit. 2020-06-24]. Dostupné z: <https://www.thingiverse.com/thing:3579883>
- [39] MATSUMOTO, Shinya, Ryuhei AMANO, Michihiro OKUDA, Takaharu ADACHI a Shinichi OKUNO. Ultra-short Throw Distance Front Projector with Mirror-Lens Hybrid Projection Optical System. In: 2008 Digest of Technical Papers - International Conference on Consumer Electronics [online]. IEEE, 2008, 2008, s. 1-2 [cit. 2020-06-24]. DOI: 10.1109/ICCE.2008.4587974. ISBN 978-1-4244-1458-1. Dostupné z: <http://ieeexplore.ieee.org/document/4587974/>
- [40] WONG, William G. Building an Embedded DLP Projector for \$99 and a BeagleBone. ElectronicDesign [online]. 2017 [cit. 2020-06-25]. Dostupné z: <https://www.electronicdesign.com/technologies/embedded-revolution/article/21805540/building-an-embedded-dlp-projector-for-99-and-a-beaglebone>

- [41] Application Report: Getting Started With TI DLP (R) Display Technology [online]. Texas Instruments Incorporated, 2015 [cit. 2020-06-25]. Dostupné z: https://www.ti.com/lit/an/dlpa059d/dlpa059d.pdf?ts=1593079607452&ref_url=https%253A%252F%252Fwww.ti.com%252Fdlp-chip%252Fdisplay-and-projection%252Fpico-chipsets%252Fgetting-started.html
- [42] Short Throw Projectors for Large Images in Tight Spaces. Projector People. Projectors for Home Theater - House of Worship and More - ProjectorPeople.com [online]. Copyright (c) 1999 [cit. 2020-06-25]. Dostupné z: <https://www.projectorpeople.com/resources/short-throw.asp>
- [43] PNG images: Terminator [online]. [cit. 2020-06-25]. Dostupné z: <http://pngimg.com/imgs/heroes/terminator/>
- [44] Socket (Java Platform SE 7). Moved [online]. [cit. 2020-06-25] Dostupné z: <https://docs.oracle.com/javase/7/docs/api/java/net/Socket.html>
- [45] HORÁK, Karel. *Zpracování vícerozměrných signálů: Jasové transformace* [online]. [cit. 2020-06-25]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/zvs_cz.php

6. Seznam použitých zkratek a symbolů

QR	quick response (code)
DLP	digital light processing
ChESS	Checkerboard extraction by subtraction and sumation
FAST	Features from accelerated segment test
VGA	Video Graphics Array
HW	Hardware
FT	Fourierova transformace
MLS	moving least squares
GUI	grafical user interface
PLA	termoplast
A-PET	termoplast
PETG	termoplast
FDM	fused deposition modeling
TI	Texas Instruments
EVM	Evaluation module
MEMS	Micro electro mechanical syste
DMD	Digital Micromirror device
BBB	BeagleBone Black
TCP	Transmission Control Protocol
GPIO	general-purpose input/output
UML	Unified Modeling Language