

UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA EXPERIMENTÁLNÍ FYZIKY

DIPLOMOVÁ PRÁCE

Analýza detekce vybraných dekorativních vad skel automobilových světlometů optickou nekoherentní metodou



Vypracoval:	Bc. Martin Učík
Studijní program:	N1701 Fyzika
Studijní obor:	Aplikovaná fyzika
Forma studia:	Prezenční
Vedoucí diplomové práce:	RNDr. Petr Šmíd, Ph.D.
Termín odevzdání práce:	leden 2022

Prohlášení

Prohlašuji, že jsem předloženou diplomovou práci vypracoval samostatně pod vedením RNDr. Petra Šmída, Ph.D. a že jsem použil zdrojů, které cituji a uvádím v seznamu použitých pramenů.

V Olomouci dne 18. ledna 2022

.....
Bc. Martin Učík

Poděkování

Děkuji za podporu projektu IGA_PrF_2020_007 Univerzity Palackého v Olomouci a projektu CZ.02.1.01/0.0/0.0/17_049/0008422 Ministerstva školství, mládeže a tělovýchovy České republiky.

Dále děkuji doc. Pavlu Tučkovi, Ph.D. za konzultace. Především bych však velmi rád poděkoval vedoucímu práce RNDr. Petru Šmídovi, Ph.D. za odborné vedení, cenné rady, vstřícnost při konzultacích a v neposlední řadě velkou trpělivost během přípravy této diplomové práce.

Bibliografická identifikace

Jméno a příjmení autora	Bc. Martin Učík
Název práce	Analýza detekce vybraných dekorativních vad skel automobilových světlometů optickou nekoherentní metodou
Typ práce	Diplomová
Pracoviště	Katedra experimentální fyziky
Vedoucí práce	RNDr. Petr Šmíd, Ph.D.
Rok obhajoby práce	2022
Abstrakt	Práce se zabývá senzorem, který je schopný detekovat dekorativní vady na průhledných částech skel světlometů. Využívá se vysoce kontrastního binárního obrazce vykresleném na LCD panelu, který je pozorován transmisní metodou přes sklo CMOS kamerou. Původní světelná vlna se při průchodu sklem deformuje, přičemž je cílem senzoru odlišit změny způsobené sklem dané samotným tvarem a změny způsobené dekorativními vadami na tomto skle. Senzor pracuje se šesti metodami, z nichž každá jiným způsobem zpracovává vstupní snímky pořízené CMOS kamerou. Pro každou metodu se během pořizování vstupních snímků na LCD panelu vykresluje binární obrazce optimalizované pro danou metodu. Senzor je testován na pěti různých dekorativních vadách a každá z metod má jinou úspěšnost jejich detekce. S použitím vyvinutých metod jsou všechny studované dekorativní vady úspěšně lokalizovány. Výsledkem senzoru jsou výstupní snímky pro každou metodu, z nichž lze sestavit kompozitní snímek podávající komplexní informaci o analyzovaném skle.
Klíčová slova	Detekce vad, sklo světlometu, optická nekoherentní metoda, obrazová analýza
Počet stran	98
Počet příloh	3
Jazyk	český

Bibliographical identification

Autor's first name and surname	Bc. Martin Učík
Title	Analysis of detection of selected defects of vehicle headlamp cover lenses by an optical non-coherent method
Type of thesis	Master
Department	Department of Experimental Physics
Supervisor	RNDr. Petr Šmíd, Ph.D.
The year of presentation	2022
Abstract	<p>The thesis studies a sensor which is able to detect decorative defects on transparent glass parts of headlight. It uses a high-contrast binary pattern displayed on an LCD panel, which is observed through a glass by a CMOS camera using the transmission method. The original light wave deforms as it passes through the glass, and the aim of the sensor is to distinguish between changes caused by the glass given by its shape and changes caused by decorative defects on the glass. The sensor uses six methods, each of which processes the input photos taken by the CMOS camera in a different way. For each method, binary patterns optimized for that method are displayed on the LCD panel while taking input photos. The sensor is tested on five different decorative defects and each of the methods has a different detection success. Using the developed methods, all studied decorative defects are successfully localized. Output images for each method are the final result of the sensor, from which a composite image can be assembled giving complex information about the analyzed glass.</p>
Keywords	Defect detection, headlamp lens cover, optical non-coherent method, image analysis
Number of pages	98
Number of appendices	3
Language	czech

Obsah

Úvod	8
1 Teorie	9
1.1 Sobelův operátor	9
1.2 Kontrast	10
1.3 Jevy moaré a aliasing	10
1.3.1 Jev moaré	10
1.3.2 Jev aliasing	11
1.4 Zobrazovací rovnice	11
1.5 Clonové číslo a hloubka ostrosti	12
1.6 Princip činnosti LCD panelu typu IPS	13
1.7 Princip činnosti CMOS snímače s posunutými mikročočkami	14
1.8 Korekce šumu snímače kamery s objektivem	14
1.8.1 Metoda korekce snímkem při rovnoměrném osvětlení	15
1.8.2 Vinětace	17
1.9 Nejistota měření	18
1.9.1 Zápis výsledku měření a zaokrouhlování	19
2 Experiment	20
2.1 Myšlenka experimentu	20
2.1.1 Kryt světlometu a dekorativní vady	20
2.2 Popis experimentálního senzoru	22
2.2.1 Komponenty	23
2.2.2 Programy	24
2.3 Příprava senzoru pro měření	26
2.3.1 Zobrazovací rovnice objektivu	26
2.3.2 Experimentální ověření zobrazovací rovnice senzoru	33
2.3.3 Optimalizace zobrazení binárního obrazce na CMOS snímači	34
2.3.4 Určení clonového čísla	34
2.3.5 Spektrální optimalizace kontrastu	36
2.4 Nastavení senzoru	39
2.4.1 Zaostření na LCD panel	40
2.4.2 Minimalizace jevu moaré	41
2.4.3 Zvýšení kontrastu	43
2.5 Měření FFC	45
2.5.1 Určení velikosti souboru N a M pro metodu FFC	45
2.5.2 Měření FFC pro všechny předpokládané expozice	46
2.6 Popis metod	47
2.6.1 Okolí pixelu	47

2.6.2	Citlivost metody	48
2.6.3	Binární obrazce metod	48
2.6.4	Metody Maximum a Aliasing	49
2.6.5	Metody Variance a Grey	53
2.6.6	Metoda Saturace	57
2.6.7	Metoda Rotace	60
2.6.8	Určení parametrů metod	63
3	Výsledky a diskuze	66
3.1	Analýza dekorativních vad	66
3.1.1	Voda na skle	66
3.1.2	Matný	68
3.1.3	Zástřík	69
3.1.4	Cizí tělísko	70
3.1.5	Šlína v číré	71
3.1.6	Shrnutí a kompozitní výstupní snímky	72
3.2	Diskuze	74
	Závěr	76
	Seznam použitých zdrojů	77
	Seznam použitých symbolů a zkratk	81
	A Metody	83
	B Kompozitní výstupní snímky	96
	C Ostatní	98

Úvod

Tradiční manuální způsob kontroly kvality výrobků je náročný na čas i lidskou práci. Nese s sebou také jistou chybovost. Automatizací tohoto procesu je snaha postup urychlit a zvýšit úspěšnost detekce vad^{[1][2]}.

V automobilovém průmyslu se dekorativní vady analyzují na transparentním^[3] i odrazném povrchu^[4]. Hlavními komponenty senzoru, který se o automatizaci stará, jsou kamera, počítač a emitor. Emitorem bývá LCD panel, případně LED panel, který je zdrojem difúzního světla^[5]. Na rozdíl jiných povrchů je v těchto případech nutné zajistit kontrolované světelné podmínky. Máme na mysli především obraz na emitoru, který kamera snímá odrazem nebo průchodem světla přes analyzovaný výrobek. Často se jedná o komplexní tvary, které se snímají z mnoha úhlů. Tímto způsobem se alespoň část snímaného povrchu blíží rovinnému tvaru. Ne vždy je to ovšem možné. K vzájemné změně polohy se používá robotická paže^{[6][4]}.

K problematice vyhodnocení snímků pořízených kamerou se přistupuje především aplikací filtrů a detektorů hran^[3]. Tento způsob si vede poměrně dobře s vadami výrazně odlišitelnými od pozadí. Pokud se vady projevují na snímku pořízeném kamerou velmi slabě, mohou být aplikací filtrů odstraněny.

Máme-li k dispozici velkou sadu snímků výrobků s defekty i bez nich, je možné senzor vyvíjet cestou strojového učení, jak ukazují příklady v textilním průmyslu^{[1][2][7]}. Tato metoda je s časem stále levnější a dostupnější^[8]. V případě, že takovou sadu k dispozici nemáme, nebo se nutná velikost takovéto sady může rovnat celkovému počtu výrobků, je třeba k řešení přistoupit tradičnější cestou zmíněných filtrů.

Schopnost detekovat vady je úzce svázána s velikostmi pixelů emitoru, snímače kamery a s vlastnostmi objektivu kamery. Na emitoru jsou zobrazeny vzory, jejichž deformaci po průchodu výrobkem analyzuje počítač. Vzorem může být rovnoměrná intenzita^{[3][6]}, pozvolný sinusový průběh^[4] nebo obrazec s ostrými hranami^[6].

Tato práce se věnuje detekci dekorativních vad průchodem světla průhlednými částmi krytů světlometů. V rámci vývoje senzoru analyzuje možnosti zobrazení vzoru na LCD panelu na snímač kamery. Cílem je dosáhnout velmi přesného zobrazení a v některých případech i vysokého kontrastu. Úkolem práce je zachytit pět různých typů dekorativních vad včetně těch nejmenších.

Kapitola 1

Teorie

V první kapitole jsou vysvětleny základní teoretické pojmy z oblastí zpracování obrazu a optiky, s nimiž text v dalších kapitolách pracuje. Dále jsou vysvětleny principy činnosti použitých přístrojů. Závěr kapitoly pojednává o nejistotě měření a jejím zápisu v této práci.

1.1 Sobelův operátor

Sobelův operátor se používá ve zpracování obrazu, kdy výstupem operace je obraz se zvýrazněnými hranami. Jedná se o diskrétní diferenční operátor, který pracuje s rozšířeným okolím (3×3) px² centrálního pixelu e (obr. 1.1)^[9].

a	b	c
d	e	f
g	h	i

Obrázek 1.1: Rozšířené okolí centrálního pixelu e.

Je definován směrový vektor derivace mezi centrálním a sousedním pixelem jako podíl rozdílu jejich intenzit ku jejich vzdálenosti. Směr tohoto vektoru udává jednotkový vektor od centrálního k sousednímu pixelu. Součtem, neboli sumou těchto celkem osmi vektorů je^[9]

$$\vec{G} = \left[\frac{c - g - a + i}{4} + \frac{f - d}{2}; \frac{c - g + a - i}{4} + \frac{b - h}{2} \right]. \quad (1.1)$$

S ohledem na práci v obrazové analýze s nízkými hodnotami celých čísel pracujeme s upravenou sumou $\vec{G}' = 4\vec{G}$. Tím předcházíme možné ztrátě nejméně významných bitů během výpočtu^[9].

$$\vec{G}' = [c - g - a + i + 2(f - d); c - g + a - i + 2(b - h)], \quad (1.2)$$

což lze přepsat do Sobelových masek

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, H_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad (1.3)$$

Výstupem jednoho pixelu obrazu na vstupu je vektor, s jehož amplitudou a směrem lze dále pracovat.

1.2 Kontrast

Pro výpočet kontrastu K využijme obecného výrazu pro kontrast neboli viditelnost

$$K = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}}, \quad (1.4)$$

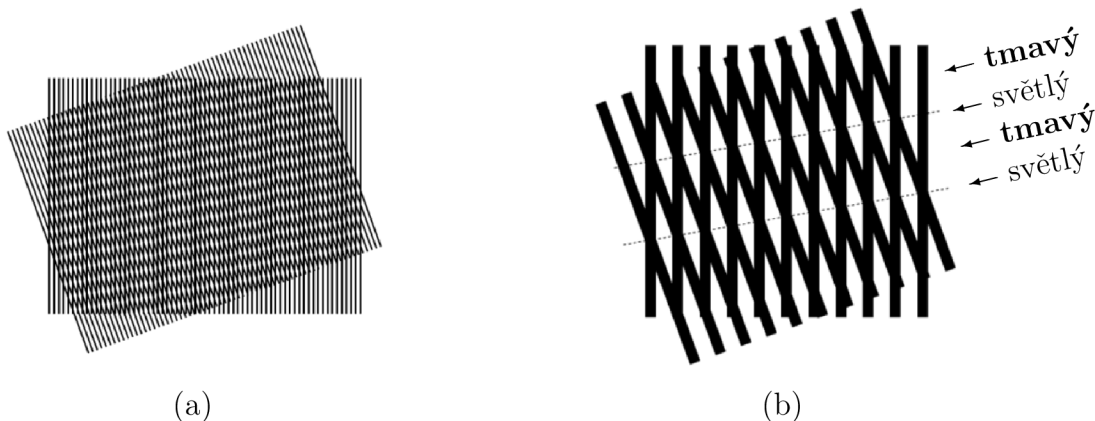
kde $K \in \langle 0, 1 \rangle$, I_{\min} a I_{\max} představují nejmenší a největší hodnoty intenzity v obraze. Ve speciálním případě můžeme kontrast K počítat z výřezu snímku (2×1) px², tedy pouze ze dvou pixelů.

1.3 Jev moaré a aliasing

1.3.1 Jev moaré

Všude tam, kde dochází ke vzájemnému překryvu periodicky se opakujících struktur, jakými jsou například mřížky, můžeme pozorovat jev moaré. Dochází ke vzniku nového vzoru se světlými a tmavými oblastmi [obr. 1.2(a)]. Tento vzor ale neexistuje samostatně ani v jedné původní struktuře. Jev moaré je důsledkem interakce mezi nimi.

Vyplňují-li tmavé oblasti jedné struktury světlé oblasti struktury druhé, pak je výsledkem superpozice tmavá oblast. Pokud se ale tmavé oblasti původních struktur vzájemně překrývají, vzniká jejich superpozicí v novém vzoru naopak světlejší oblast než v předchozím případě^[10], jak je naznačeno na detailu na obr. 1.2(b).



Obrázek 1.2: (a) Překryv dvou mřížek, na nichž můžeme pozorovat viditelný jev moaré a (b) detail, kde v důsledku superpozice proužků vznikají světlé a tmavé oblasti. Převzato a upraveno z Amidror^[10].

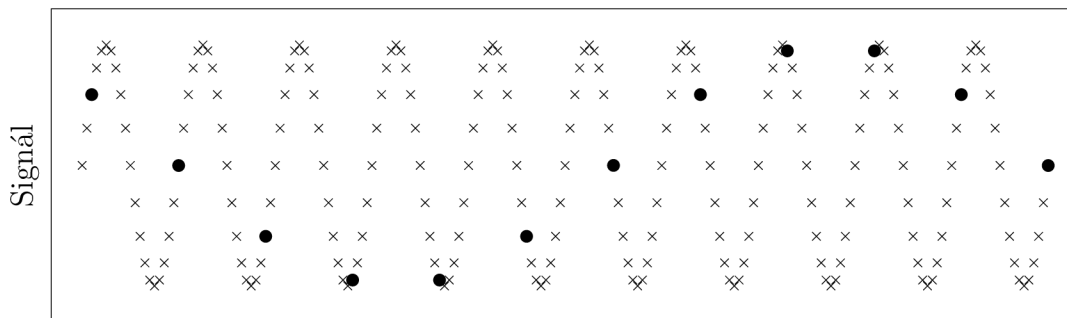
Nový vzor je velmi citlivý na jakoukoli změnu vzájemné pozice (posunutí) či změnu tvaru struktur (deformace). Díky tomu našel jev moaré využití v řadě aplikací. Uveďme např. metrologii^[11], fotolitografii^[12] a studium deformace^{[13][14]}. Na druhou stranu se může projevovat paraziticky např. v digitální fotografii^[15] a existuje snaha jej odstranit nebo alespoň potlačit^[16].

1.3.2 Jev aliasing

Vzorkováním vstupního signálu nižší frekvencí f_v , než je dvojnásobek nejvyšší frekvence vstupního signálu f_{\max} , tj. není splněn Shannonův teorém^[17]

$$f_v > 2f_{\max}, \quad (1.5)$$

může dojít k aliasingu (obr. 1.3).



Obrázek 1.3: Ukázka aliasingu: původní signál (křížky) s vyšší frekvencí je vzorkován (černé tečky) nižší frekvencí, čímž vzniká frekvence, která se v původním signálu nevyskytuje.

Pro jeho potlačení je postačující před samotným vzorkováním odfiltrovat pomocí dolní propusti ze vstupního signálu všechny vyšší frekvence, které nesplňují Shannonův teorém [vztah (1.5)].

V praxi se s aliasingem setkáváme například ve videozáznamu rychle se otáčejících kol, kdy je frekvence snímání obrazu jen mírně vyšší (nižší) než frekvence otáčení kola. V tom případě pozorujeme zdánlivě pomalejší rychlost (nebo dokonce opačný směr) rotace kola, než se děje ve skutečnosti.

1.4 Zobrazovací rovnice

Nejprve si zavedme znaménkovou konvenci. Předmětovou vzdálenost a měříme od optické soustavy proti směru šíření světla a obrazovou vzdálenost a' ve směru šíření světla. Příčnou polohu předmětu y měříme od optické osy ve svislém směru nahoru a příčnou polohu obrazu y' v opačném směru. Příčné zvětšení m , které je dáno jako poměr příčné polohy obrazu a předmětu

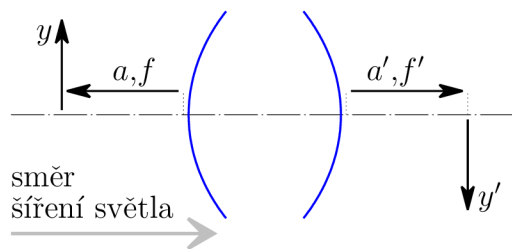
$$m = \frac{y'}{y}, \quad (1.6)$$

je v případě převráceného obrazu kladné číslo. Situace je znázorněna na obr. 1.4.

Dále se omezme na paraxiální neboli Gaussův prostor^[18], kde je zobrazení téměř ideální a příčná velikost předmětu je v porovnání s podélnými parametry optické soustavy zanedbatelná.

Odvodme zobrazovací rovnici pro výpočet předmětové vzdálenosti a v situaci, při níž je předmět zobrazován optickou soustavou na detektor ležící na optické ose, který je tvořen maticí pixelů, u nichž předpokládáme čtvercový tvar a ekvidistantní vzdálenosti. Vyjdeme z Gaussova tvaru zobrazovací rovnice^[18]

$$\frac{f}{a} + \frac{f'}{a'} = 1, \quad (1.7)$$



Obrázek 1.4: Zavedení znaménkové konvence.

kde f , f' jsou předměťová a obrazová ohnisková vzdálenost optického systému, a , a' jsou vzdálenosti předmětu od předmětového hlavního bodu a obrazu od obrazového hlavního bodu. Z rovnice (1.7) si vyjádříme podíl $\frac{a'}{a}$, který z podobnosti trojúhelníků odpovídá příčnému zvětšení m . Odtud již jednoduše získáváme rovnici pro předmětovou vzdálenost

$$a = \frac{f'}{m} + f. \quad (1.8)$$

Příčný rozměr předmětu je y . V obrazové rovině detektoru vzniká jeho obraz y' , jehož velikost je obecně neceločíselným násobkem $n_{\mathbb{R}}$ příčného rozměru pixelu detektoru y_d , tj. $y' = n_{\mathbb{R}} \cdot y_d$. Příčné zvětšení je tedy podle vztahu (1.6)

$$m = \frac{n_{\mathbb{R}} \cdot y_d}{y}. \quad (1.9)$$

Tento výsledek dosadíme do rovnice (1.8) a získáváme vztah pro vzdálenost předmětu od předmětové hlavní roviny optické soustavy

$$a = \frac{f'}{n_{\mathbb{R}}} \cdot \frac{y}{y_d} + f. \quad (1.10)$$

1.5 Clonové číslo a hloubka ostrosti

Kamera se skládá z objektivu a těla kamery. Tělo obsahuje CMOS snímač zaznamenávající obraz zmenšený objektivem.

Ohnisková vzdálenost f' a clonové číslo c jsou hlavními parametry objektivu. Clonové číslo

$$c = \frac{f'}{D} \quad (1.11)$$

udává, kolikrát je ohnisková vzdálenost f' větší než průměr vstupní pupily D ^{[18][19]}.

Zavádíme hyperfokální vzdálenost G jako vzdálenost předmětového bodu, jehož rozptýlený kroužek při zaostření objektivem na nekonečno je právě tak velký, jako je tolerovaná hodnota $\Delta y'_0$. Tedy je-li velikost neostrosti obrazu menší jak $\Delta y'_0$, potom považujeme tento obraz za ostrý^[18].

$$G = \frac{f'D}{\Delta y'_0} \frac{1}{1 - \frac{D}{D'} \frac{f'}{g}}, \quad (1.12)$$

kde D' je průměr výstupní pupily. Jeli předmětová vzdálenost vztažená ke vstupní pupile g mnohem větší jak ohnisková vzdálenost f' , tj. $g \gg f'$, pak se výraz (1.12) zjednoduší^[18]

$$G \approx \frac{f'^2}{c\Delta y'_0}. \quad (1.13)$$

Obraz je ostrý pouze pro ty předměty, jejichž předmětová vzdálenost g leží v intervalu (g_1, g_2) , kde

$$g_1 = \frac{gG}{G+g}, \quad g_2 = \frac{gG}{G-g}. \quad (1.14)$$

Rozdíl g_2 a g_1 definujeme jako hloubku ostrosti^[18]

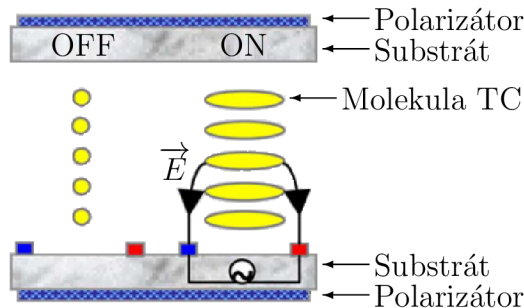
$$\Delta_g = g_2 - g_1 = \frac{2g^2G}{G^2 - g^2}. \quad (1.15)$$

Můžeme si povšimnout při pohledu na vztah (1.13), že hloubka ostrosti je značně závislá na clonovém čísle objektivu.

1.6 Princip činnosti LCD panelu typu IPS

Panel z tekutých krystalů LCD (angl. *liquid-crystal display*) v módu IPS (angl. *In-Plane Switching*) se skládá ze dvou rovnoběžných souběžně orientovaných polarizačních filtrů a dvou substrátů mezi nimi (obr. 1.5). Mezi substráty se nachází tekuté krystaly LC (angl. *liquid crystal*). Na vnitřních stěnách substrátů jsou tenké vrstvy polymeru, které tvoří drážky ve směru polarizačních filtrů. Tekuté krystaly se v blízkosti polymeru orientují souhlasně s drážkami^[20]. LCD panel je v tuto chvíli pro světlo propustný. Aplikuje-li se elektrické pole elektrodami oběma umístěnými na jednom substrátu, pak se tekuté krystaly stočí ve směru elektrické intenzity \vec{E} , čímž se LCD panel stane pro světlo nepropustným.

Tekuté krystaly se tímto způsobem stáčí v rovině panelu. Na rozdíl od dalších módů (VA a TN módy), kde se tekuté krystaly stáčí mimo rovinu^[21], poskytuje IPS lepší pozorovací úhel^[22].

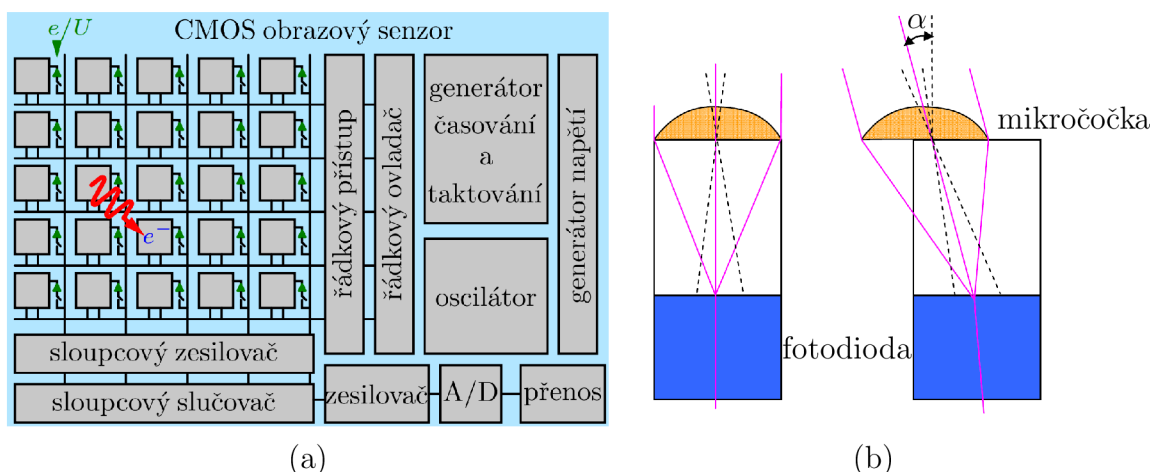


Obrázek 1.5: Schématický diagram IPS módu LCD panelu. Převzato a upraveno z Hong^[23].

1.7 Princip činnosti CMOS snímače s posunutými mikročočkami

CMOS snímač je matice fotorezistorů, které nazýváme pixely. Ty jsou od sebe fyzicky odděleny a každý z nich má vlastní převodník náboje na napětí^[24]. Během expozice dopadající záření generuje fotoelektrony ve sběrné oblasti polovodiče, které jsou akumulovány v potenciálové jámě. Následně jsou tyto fotoelektrony vyčítány a převáděny na napětí na každém pixelu zvlášť^[24]. Tato napětí jsou přivedena jednotlivě řádkovými a sloupcovými zesilovači k A/D převodníku. Zpracování signálu z fotorezistorů je na stejném čipu, na jakém je vlastní snímač^[24] [obr. 1.6(a)].

Pro zvýšení pravděpodobnosti dopadu záření na světlocitlivou plochu pixelu jsou na matici pixelů umístěny mikročočky, které fokusují do sběrné oblasti i to záření, které by bez použití mikročoček dopadlo mimo tuto oblast a snímač by jej nedetekoval [obr. 1.6(b)]. Paprsky tohoto záření dopadající na snímač nejsou rovnoběžné, ale vycházejí z ohniska. Na pixely na okrajích snímače proto dopadají tyto paprsky pod úhlem α . To je motivací posunutí mikročoček tak, aby jimi fokusované paprsky dopadaly opět do sběrné oblasti pixelu^[25].



Obrázek 1.6: (a) Schéma CMOS snímače, převzato z Černoch^[24]. (b) Schéma k vysvětlení motivace posunu mikročoček na pixelu matice CMOS snímače, převzato a upraveno z datového listu výrobce^[25].

1.8 Korekce šumu snímače kamery s objektivem

Snímač kamery je tvořen maticí pixelů, které při rovnoměrném osvětlení (angl. *uniform illumination*) nevykazují stejné vlastnosti zejména vlivem teploty (termální šum v důsledku termálních excitací) a nedokonalostí výroby (se zmenšující se velikostí snímače je obtížnější zajistit stejné rozměry pro všechny jeho pixely – různé světlocitlivé plochy).

Obecně existuje celá řada šumů¹. Je vhodné zmínit, že i optická soustava umístěná

¹Například šum světelného signálu, výstřelový šum, fotonový výstřelový šum, přeslech a temný proud. U každého z těchto šumů lze předpokládat normální rozdělení, i když v některých případech je vhodnější Poissonovo rozdělení^[26]. Podrobněji lze k šumům přistoupit s ohledem na elektronický obvod snímače, kde lze modelovat řádkové, sloupcové šумы a odečítací šum vznikající během zesílení a konverze elektronů na napětí AD převodníkem a s tím související kvantovací šum^[26].

před snímačem vnáší jisté zkreslení dané optickými aberacemi a dalšími vlastnostmi spojenými s průchodem světla látkou. My se však omezíme na popis pouze pomocí šumu s pevným vzorem FPN (angl. *Fixed Pattern Noise*), který v sobě zahrnuje obecně šumy invariantní v čase.

Šum FPN zahrnuje dva typy šumu. Šum DSNU (angl. *dark signal non-uniformity*) pozorujeme se zakrytým objektivem. Při nulovém osvětlení by měl snímač detekovat teoreticky nulovou intenzitu světla. Ve skutečnosti ale jistou intenzitu detekuje. Ta je způsobena tzv. temným proudem, který je tvořen termálními elektrony a je závislý na teplotě^[27] a expoziční době^{[28][29]}.

Druhým šumem je šum PRNU (angl. *photo response non-uniformity*). Zorné pole objektivu je rovnoměrně osvětleno a vlivem nepřesného zobrazení a odlišností jednotlivých prvků matice snímače detekujeme nerovnoměrný zisk fotoelektronů. Tento šum zahrnuje i vinětaci, která se projevuje poklesem detekované intenzity světla směrem k okrajům snímku (zejména rohy snímku). Příčinou je absorpce dopadajícího světla v optické soustavě před snímačem a také geometrie snímače, kdy na světlocitlivé plochy pixelů snímače vzdálenějších od osy objektivu dopadá méně světla. Je snaha oba šumy potlačit do té míry, abychom při rovnoměrném osvětlení zorného pole objektivu pozorovali ideálně nulové rozdíly intenzity světla mezi jednotlivými pixely snímače. K tomu se využívá metoda korekce pomocí snímku při rovnoměrném osvětlení^[30].

1.8.1 Metoda korekce snímkem při rovnoměrném osvětlení

Touto metodou míníme metodu FFC (angl. *Flat-Field Correction*), kterou jsme schopni teoreticky odstranit šum FPN. Reálně jsou však hodnoty intenzit snímku po korekci stále zatíženy chybou.

Pro účely odvození využijme maticového formalismu, kdy si pod pojmem snímek představme matici hodnot. Podle Williamse a kol.^[30] nejprve pořídíme soubor temných snímků se zakrytým objektivem, aby na snímač nedopadalo žádné světlo. Přitom neměníme nastavení kamery resp. objektivu a to i během pořizování všech následujících snímků. Každý snímek reprezentuje záznam temného proudu (šum DSNU) a výsledkem je soubor N snímků $D_i^{\text{raw}}[x,y]$, kde i značí i -tý snímek souboru a x, y jsou souřadnice prvku matice neboli pixelu snímku. Z těchto snímků vypočítáme střední hodnotu

$$\bar{D}[x,y] = \frac{1}{N} \sum_{i=1}^N D_i^{\text{raw}}[x,y]. \quad (1.16)$$

Poté před objektiv umístíme plošný zdroj světla, který vytváří rovnoměrné osvětlení v celém zorném poli objektivu (angl. *flat-field*, odtud název metody). Opět pořídíme sérii M snímků $F_j^{\text{raw}}[x,y]$, kde j značí j -tý snímek. Snímek by měl být poměrně jasný, ale žádný pixel snímače nesmí být v saturaci. Tímto získáváme informaci o šumu PRNU zahrnujícím v sobě zkreslení objektivu vlivem aberací a vinětaci. Snímek rovněž obsahuje šum DSNU. Střední hodnotu určujeme z M hodnot korigovaných o střední hodnotu šumu DSNU^[30]

$$\bar{F}[x,y] = \frac{1}{M} \sum_{j=1}^M \left(F_j^{\text{raw}}[x,y] - \bar{D}[x,y] \right), \quad (1.17)$$

což můžeme také napsat jako

$$\bar{F}[x,y] = \bar{F}^{\text{raw}}[x,y] - \bar{D}[x,y]. \quad (1.18)$$

Tento výsledek dále dělíme střední hodnotou intenzity^[30]

$$\mu = \frac{1}{w \cdot h} \sum_{x=1}^w \sum_{y=1}^h \bar{F}[x,y], \quad (1.19)$$

kde w je šířka snímku a h výška snímku v jednotkách pixelů. Tím získáváme normalizovanou hodnotu

$$\bar{f}[x,y] = \frac{\bar{F}[x,y]}{\mu}. \quad (1.20)$$

Pořízený snímek R , u kterého chceme odstranit šum FPN, jednoduše dosadíme do vztahu^[30]

$$C[x,y] = \frac{R[x,y] - \bar{D}[x,y]}{\bar{f}[x,y]}, \quad (1.21)$$

kde C je výsledný korigovaný snímek (angl. *corrected image*) metodou FFC. Na člen $-\bar{D}[x,y]$ v čitateli můžeme z pohledu metrologických pojmů pohlížet jako na korekci systematické chyby a na jmenovatel $\bar{f}[x,y]$ jako na převrácenou hodnotu korekčního součinitele^[34].

Názorná ukázka metody FFC

Simulujme matici snímače o ploše (5×5) px² a ukažme si, jakým způsobem výše odvozené rovnice fungují. Mějme kameru s objektivem, jejíž snímač má rozsah intenzity od 0 do 10 a obraz je ovlivněn šumem FPN, který se skládá ze dvou složek - šumů DSNU a PRNU.

Nejprve je potřeba modelovat dva snímky \bar{D} a \bar{F}^{raw} nutné pro metodu FFC. Pro jednoduchost předpokládejme, že velikosti obou souborů snímků M , N jsou rovny jedné², tj. $\bar{D} \equiv D^{\text{raw}}$ a $\bar{F}^{\text{raw}} \equiv F^{\text{raw}}$.

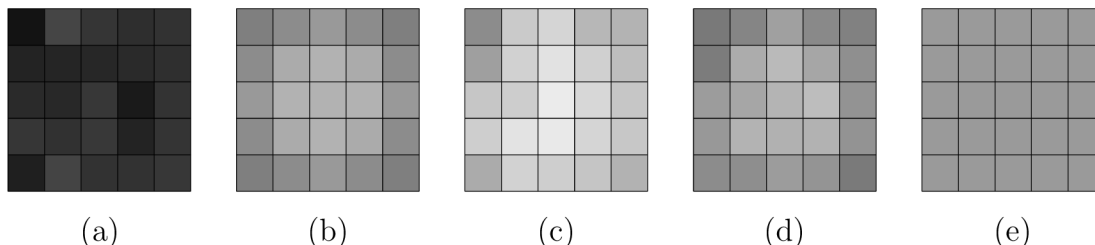
První snímek simuluje pořízení při zakrytém objektivu a obsahuje šum DSNU. Druhý snímek naopak simuluje pořízení při rovnoměrném osvětlení zorného pole objektivu a obsahuje šum PRNU včetně vinětace a šum DSNU. Pro vytvoření šumů DSNU a PRNU bez vinětace použijme generátor náhodných čísel s normálním rozdělením³ (střední hodnota μ , směrodatná odchylka σ) `random.normal` z knihovny `numpy` v programu `Python3`. Nejprve modelujme matici s náhodným šumem DSNU ($\mu = 2$, $\sigma = 0,5$) [obr. 1.7(a)]. Tím vzniká první snímek \bar{D} . Potom si ručně vytvoříme vinětaci [obr. 1.7(b)], kterou přenásobíme simulovaným šumem PRNU bez vinětace ($\mu = 0$, $\sigma = 0,03$) a získáváme šum PRNU s vinětací. K výsledku součinu přičtíme šum DSNU, čímž získáváme druhý snímek \bar{F}^{raw} [obr. 1.7(c)].

S takto předchystanými snímky nyní přejdeme k ukázce, jak samotná metoda FFC funguje. Pořídíme soubor snímků s temným proudem a z nich vytvoříme průměrný snímek \bar{D} podle vztahu (1.16) [obr. 1.7(a)]. Dále pořídíme druhou sérii snímků s rovnoměrným osvětlením a rovněž vytvoříme průměrný snímek \bar{F}^{raw} [obr. 1.7(c)]. Odečtíme od něj snímek \bar{D} podle vztahu (1.18), čímž dostáváme snímek \bar{F} [obr. 1.7(d)]. Pro tento snímek určíme střední hodnotu intenzity μ ze vztahu (1.19) a normováním podle

²V praxi je toto nežádoucí kvůli tomu, že oba snímky D^{raw} a F^{raw} jsou zatíženy šumem, který se korekcí snímku R jistou mírou přenáší na korigovaný snímek C . Podmínkou $M > 1$, $N > 1$ se tento vliv minimalizuje^[29].

³Oba šumy DSNU a PRNU mají obecně neznámá rozdělení. V našem případě volíme apriorní normální rozdělení. Na funkci metody FFC a tedy účelu názorné ukázky tato volba nemá vliv.

vztahu (1.20) získáváme normalizovaný snímek \bar{f} . Pokud je metoda korekce správná, měl by po dosazení snímku \bar{F}^{raw} (snímek pořízený při rovnoměrném osvětlení zorného pole objektivu) za snímek R ve vztahu (1.21) vyjít snímek se stejnou intenzitou v každém pixelu simulovaného snímače. A skutečně nám takto vychází snímek s rovnoměrnou intenzitou [obr. 1.7(e)].



Obrázek 1.7: Simulace metody FFC: (a) modelovaný snímek \bar{D} s tmavým šumem DSNU, (b) modelovaná vinětace projevující se ztmavením rohů, (c) modelovaný snímek \bar{F}^{raw} při rovnoměrném osvětlení obsahující šумы DSNU a PRNU včetně vinětace, (d) snímek \bar{F} při rovnoměrném osvětlení zbavený šumu DSNU a (e) korigovaný snímek při rovnoměrném osvětlení.

1.8.2 Vinětace

Pokles intenzity světla při rovnoměrném osvětlení optické soustavy projevující se ztmavením obrazu směrem od jeho středu k okrajům označujeme jako vinětaci. Podle Ray^[31] rozlišujeme pro digitální zobrazení čtyři druhy vinětace. Těmi jsou mechanická, optická, přirozená a pixelová vinětace.

Příčinou mechanické vinětace je částečná blokáce mimoosových paprsků například neprůhlednou částí optického filtru nebo nesprávně upevněnou čočkou. Následkem je změna tvaru vstupní pupily. Pokud je mimoosový paprsek zcela zablokovaný, detekujeme v místě myšleného průsečíku mimoosového paprsku s obrazovou rovinou snímače nulovou intenzitu a dochází k omezení zorného pole^[31].

Optická vinětace se projevuje u optických soustav s více čočkami. Zadní čočky blíže ke snímači jsou stíněny předními čočkami, což snižuje efektivitu průchodu čočkami pro mimoosové paprsky a v obrazové rovině pozorujeme pokles intenzity směrem k okrajům snímače. Vliv optické vinětace lze snížit zvýšením clonového čísla^[31].

Podstatou přirozené vinětace není překážka ve dráze mimoosového paprsku, ale přirozený pokles dopadajícího světelného toku na snímač v závislosti na úhlu α mezi mimoosovým paprskem a optickou osou. Tento pokles se v detekované intenzitě řídí přibližně čtvrtou mocninou kosinu úhlu α . Tomuto typu vinětace jsou více náchylné širokouhlé objektivy ve srovnání například s telecentrickými objektivy^[31].

Pixelová vinětace v digitálních kamerách je důsledkem úhlové závislosti detekce světla jednotlivými pixely, protože na pixely blíže optické ose dopadá větší světelný tok než na ty vzdálenější. Tento jev je možné redukovat vhodným umístěním mikročoček na snímači tak, aby maximalizovaly možný záchyt světelného toku na světlocitlivou plochu pixelu^[31].

1.9 Nejistota měření

Pokud neznáme pravou hodnotu veličiny Y , můžeme provést její odhad y daný předpisem $y = f(x_1, x_2, \dots, x_n)$, kde x_i je odhad vstupní veličiny X_i ^[32]. Aby byl výsledek měření úplný, musí být jeho součástí i nejistota, která vyjadřuje míru pochybnosti o správnosti výsledku měření. Nejistotu nazýváme kombinovanou standardní nejistotou $u_C(y)$ a počítá se ze standardních nejistot $u(x_i)$ odhadů vstupních veličin x_i ^[33]^[34].

Jestliže je $u_C(y)$ určena nejvýše třemi standardními nejistotami, pak používáme lineární tvar výpočtu kombinované standardní nejistoty

$$u_C(y) = \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right| u(x_i), \quad (1.22)$$

kde $n \leq 3$ a $\frac{\partial f}{\partial x_i}$ je parciální derivace funkce f podle veličiny x_i . Pokud je $n > 3$, počítáme $u_C(y)$ v kvadratickém tvaru

$$u_C(y) = \sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \right)^2 u^2(x_i)}. \quad (1.23)$$

Při použití lineárního tvaru pro případ $n > 3$ by kombinovaná standardní nejistota byla nadhodnocená, protože je málo pravděpodobné, že by se všechny pravé hodnoty vstupních veličin x nacházely na okraji intervalů určených standardními nejistotami $u(x_i)$ jejich odhadů x_i . A opačně kvadratickým tvarem pro $n \leq 3$ bychom mohli kombinovanou standardní nejistotu podhodnotit. Podotkněme, že oba vztahy (1.22), (1.23) platí pro nekorelované vstupní veličiny X_i ^[32].

Podle vzniku dělíme standardní nejistotu $u(x_i)$ na typ A, $u_A(x_i)$, a na typ B, $u_B(x_i)$. Je-li odhad \bar{x}_i vstupní veličiny X_i vypočítán z m nezávislých měření $x_{i,k}$

$$\bar{x}_i = \frac{1}{m} \sum_{k=1}^m x_{i,k}, \quad (1.24)$$

kde $k \in \{1, 2, \dots, m\}$, potom

$$u_A(x_i) = \sqrt{\frac{1}{m(m-1)} \sum_{k=1}^m (x_{i,k} - \bar{x}_i)^2}. \quad (1.25)$$

nazýváme standardní nejistotou typu A. Standardní nejistota $u_A(x_i)$ tedy vychází z rozdělení četnosti. Na rozdíl od toho standardní nejistota typu B se určuje z apriorního rozdělení. Přitom vycházíme z dostupných znalostí^[32].

Můžeme provést odhad horní a dolní meze a_+ a a_- , mezi kterými se vstupní veličina X_i nachází s pravděpodobností 1. Na intervalu určeném oběma mezemi $\langle a_-; a_+ \rangle$ stanovujeme apriorní rozdělení. Může-li se pravá hodnota nacházet kdekoli na tomto intervalu se stejnou pravděpodobností, volíme rovnoměrné rozdělení.

Je vhodné zvolit odhad vstupní veličiny x_i tak, aby se nacházel uprostřed těchto mezí, neboli $a_+ = a_- = a$. Tím se interval, kde se pravá hodnota nachází s pravděpodobností 1, zjednoduší na tvar $\langle x_i - a; x_i + a \rangle$. Z tohoto předpokladu můžeme určit standardní nejistotu typu B odhadu vstupní veličiny x_i jako

$$u_B(x_i) = \frac{a}{\sqrt{3}}. \quad (1.26)$$

Poznamenejme, že při výpočtu $u_C(y)$ standardní nejistoty nikdy nezaokrouhlujeme. Zaokrouhlování se provádí až po určení kombinované standardní nejistoty.

1.9.1 Zápís výsledku měření a zaokrouhlování

Výsledek měření veličiny X , jejíž odhad je jmenovitě např. $x = 100,0$ Jednotka a její standardní nejistota $u(x) = 0,1$ Jednotka, zapisujeme podle následujícího tvaru^[32]

$$X = (100,0 \pm 0,1) \text{ Jednotka.}$$

Také můžeme psát, že výsledek měření je $x = 100,0$ Jednotka se standardní nejistotou $u(x) = 0,1$ Jednotka^[32].

Vypočítaná nejistota je uváděna na pět nenulových číslic. O takové nejistotě se v tomto textu vyjadřujeme jako o nezaokrouhlené. Jestliže nejistotu dosazujeme ve výpočtu kombinované standardní nejistoty, pak se dosazuje nejistota právě v tomto tvaru pro zachování přesnosti. K zaokrouhlení v souladu s doporučeními^[35] přistupujeme, je-li nejistota součástí konečného výsledku.

Je-li číslice, která je zaokrouhlením odstraněna, větší jak pět, pak předcházející číslici zvětšíme o jedničku. Pokud je menší jak pět, pak předcházející číslice zůstává nezměněna. Pokud je číslice rovna pěti a za ní je nula, pak je-li předcházející číslice lichá, zvýšíme ji o jedničku a naopak je-li sudá, zůstává nezměněna. Pokud je ale za odstraňovanou číslicí pět nenulová číslice, pak k předcházející číslici přičteme jedničku bez ohledu na její lichost či sudost^[35].

Nejistotu zaokrouhlujeme na dvě platné číslice směrem nahoru v případě, že první číslice je 1 nebo 2^[36] (dohodněme se, že ve druhém případě, je-li druhá číslice v pořadí větší jak 7, zaokrouhlujeme na jednu platnou číslici směrem nahoru). Pokud po takovém zaokrouhlení je první číslice 3 a druhá tedy 0, pak uvádíme pouze první číslici. V ostatních případech zaokrouhlujeme nejistotu na jednu platnou nenulovou číslici.

Kapitola 2

Experiment

Druhá kapitola se věnuje teoretické přípravě senzoru i jeho realizaci a následné optimalizaci. Obsahuje přehled jeho komponent spolu s programy vzniklými za účelem zjednodušení práce se senzorem. Dále popisuje postup nastavení senzoru před měřením a věnuje se teoretickému vysvětlení metod, které senzor používá k analýze pořízených snímků dekorativních vad na průhledných částech krytů světlometů.

2.1 Myšlenka experimentu

Práce se zabývá detekcí dekorativních vad na krytech světlometů, resp. na jejich transparentní části, kterou budeme dále označovat jako sklo. Využívá se vysoce kontrastního binárního obrazce na LCD panelu, který je pozorován transmisí metodou přes sklo. Původní světelná vlna se při průchodu sklem deformuje, přičemž je cílem senzoru odlišit změny způsobené samotným tvarem skla a změny způsobené vadami na něm. Tato analýza probíhá pomocí několika k tomuto účelu vytvořených programů, které označujeme jako metody.

2.1.1 Kryt světlometu a dekorativní vady

Světlomet automobilu se v jednoduchém přiblížení skládá ze světelného zdroje a reflektoru, které jsou chráněny krytem světlometu (obr. 2.1). Ten se vyrábí z bílého a černého plastového granulátu, který je vstříkovan pod tlakem do formy. Z bílého granulátu vzniká průhledná část krytu světlometu, kterou dále nazýváme sklo, a z černého granulátu je vytvořen jeho okraj. Na sklo je nastříkán ochranný lak, který zvyšuje odolnost povrchu proti poškrábání a rovněž obsahuje UV filtr, který chrání sklo před žloutnutím^[37].



Obrázek 2.1: Kryt světlometu bez dekorativních vad.

Plocha testovaného skla je velice komplikovaná. Různé části plochy mají odlišnou křivost a mohou být spojeny plynulým přechodem nebo šíbrem, což je jakýsi zlom mezi nimi. Součástí skla je také hrana - ostrý spoj mezi dvěma plochami, jak je znázorněno na obr. 2.1.

Během samotného procesu výroby krytu světloometu může dojít ke vzniku dekorativních vad (nebo jen vad)^[5]. Uvedme si pět z nich, které budou z pohledu jejich detekce dále analyzovány.

1 Voda na skle

Před nanesením ochranného a zároveň funkčního laku na sklo se na povrchu skla nachází drobné kapénky vody. Ty jsou na něm uvězněny nanesením laku. Tyto kapky vody se při průchodu světla sklem chovají jako asymetrické čočky [obr. 2.2(a)], jejichž průměry jsou v jednotkách milimetrů.

2 Matný

Během výroby skla dochází k nedokonalému vstřikování plastu daném nehomogenními podmínkami ve vstřikovací formě. To se projevuje na povrchu skla, který má vyšší drsnost a působí matněji než okolní hladké sklo [obr. 2.2(b)]. Matný je plošná vada v jednotkách cm^2 .

3 Zástřík

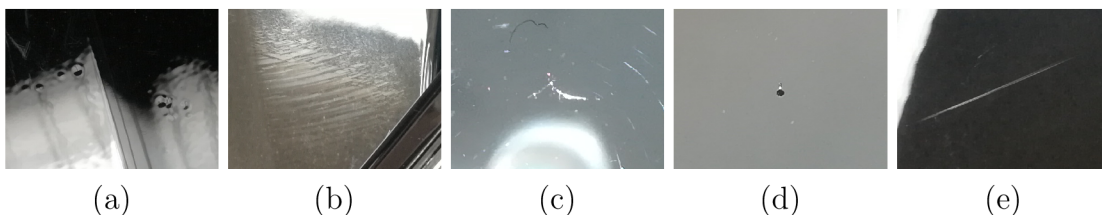
Při lakování se na povrch skla nanese nehomogenita, kolem které vznikne ostré rozhraní vůči zbytku laku. Lak tedy v tomto místě není homogenní. Zátřík je drobná vada v jednotkách milimetrů [obr. 2.2(c)].

4 Cizí tělísko

V průběhu vstřikování i lakování může být v krytu světloometu zachycena částice, která tam nepatří. Většinou jde o světlo absorbující částici [obr. 2.2(d)]. Vada má velikost do jednoho milimetru.

5 Šlíra v čiré

Tento defekt má podélný tvar o délce až několik centimetrů [obr. 2.2(e)].



Obrázek 2.2: Přehled studovaných dekorativních vad na skle: (a) voda na skle s linií šíbru, (b) matný (nerovný povrch výlisku pod lakem), (c) zástřík, (d) cizí tělísko, (e) šlíra v čiré.

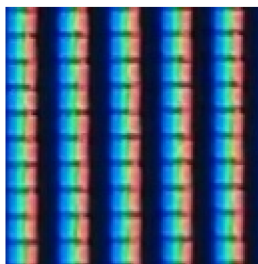
2.2 Popis experimentálního senzoru

Po sestavení senzoru a jeho uvedení do provozu se objevuje několik komplikací. Především jde o jev moaré, který výrazně deformuje podobu obrazu binárního obrazce na LCD panelu zaznamenaném kamerou. Je proto důležité, aby obě roviny LCD panelu a CMOS snímače byly rovnoběžné. Také úhel svíraný oběma mřížkami (mřížky pixelů LCD panelu a CMOS snímače) musí být nulový.

Nicméně i přes tato opatření jev moaré může kolem středu snímku stále existovat. Příčinou je necelónásobný zobrazovací poměr mezi velikostí obrazu pixelu LCD panelu a pixelu CMOS snímače. Na snímku bychom kolem středu pozorovali soustředné kružnice. Zajistíme-li, aby zobrazovací poměr byl přirozené číslo, potom lze jev moaré v podstatné části snímku minimalizovat. Avšak dále od středu jej vlivem soudkovitého zkreslení potlačit nemůžeme. Je zjevné, že právě proto se držák kamery skládá s tolika posuvů a náklonů.

Při pohledu na strukturu vykresleného binárního obrazce na LCD panelu v barevném režimu RGB ¹ (obr. 2.3) je zřejmý vertikální směr všech tří subpixelů v rámci jednoho pixelu. Pokud bychom binární obrazec otočili o 90 stupňů, vypadal by snímek velmi podobně.

Představme si stejnou situaci, ale v barevném režimu G. Tj. svítily by pouze zelené subpixely. Potom by se oba případy velice lišily. V prvním případě by byl obrazec tvořen souvislou čarou zelených subpixelů, ve druhém případě by byl ale přerušovaný a jednalo by se spíše o přerušovaný řetězec. Tento důsledek se uplatňuje při volbě vhodného binárního obrazce.



Obrázek 2.3: Detailní pohled na strukturu pixelů LCD panelu v barevném režimu RGB, na němž je vykreslen binární obrazec s vertikálními proužky o šířce proužků 1 px a periodě 2 px.

Vzhledem ke spektrální závislosti CMOS snímače je třeba posoudit vhodný barevný režim LCD panelu, jichž je celkem sedm (R, G, B, RG, RB, GB a RGB). Zároveň jsou subpixely určitým způsobem uspořádány. Tyto vlivy je nutné zvážit, aby byl binární obrazec co nejvíce kontrastní.

¹Pixel LCD panelu se skládá ze tří subpixelů třech různých barev - červený (R), zelený (G) a modrý (B) subpixel. Barevným režimem např. RG míníme ten stav, při kterém mohou na LCD panelu svítit pouze subpixely R a G a to současně v rámci jednoho pixelu. Celkem máme sedm barevných režimů (R, G, B, RG, RB, GB a RGB).

2.2.1 Komponenty

Senzor sestává ze tří částí. První částí je LCD panel, na kterém se vykreslují binární obrazce.

Binární obrazec

Binární obrazec je definovaná matice $1\,920 \times 1\,200$ bodů, jejíž prvky mají hodnotu 1 nebo 0. Pokud má prvek hodnotu 1, pak pixel na LCD panelu na jemu odpovídající pozici svítí. Ve druhém případě je vypnutý. Binární obrazec je vykreslen v jednom se sedmi barevných režimů LCD panelu. Definujeme elementární buňku binárního obrazce reprezentovanou elementární maticí B , jejíž řádky a sloupce odpovídají horizontálnímu a vertikálnímu směru na LCD panelu. Periodickým opakováním elementární matice vzniká binární obrazec.

Před obrazovkou LCD panelu se nachází druhá část a to prostor pro umístění krytu světlometu se zajišťovacími tyčemi, na které je kryt světlometu uložen. Poslední část představuje detektor neboli CMOS kamera s objektivem upevněná v pojízdném nastavitelném držáku na kolejnici. Všechny tři části jsou v jedné ose, kterou nazýváme osa senzoru. Ta je definovaná osou objektivu, u které je předpoklad, že je rovnoběžná s kolejnicí a normálou LCD panelu [obr. 2.4(a)]. Součástí senzoru je také LED panel a šedé filtry používané při měření FFC. Celý senzor je uzavřen v krytu, který zamezuje působení vnějšího rušivého světla během činnosti senzoru.

Senzor je řízen dvěma počítači. Na hlavním počítači ovládáme programy, spouštíme metody pro detekci dekorativních vad a pořizujeme snímky kamerou připojenou ethernetovým kabelem. Tento počítač zároveň komunikuje sériovým portem s druhým počítačem, pomocí něhož vykreslujeme na obrazovce LCD panelu binární obrazec.

LCD panel

LCD panel i2460Pxqu od firmy AOC je typu IPS. Oblast displeje má rozměry $518\text{ cm} \times 324\text{ cm}$ s úhlopříčkou 61 cm . Jas displeje je $300\text{ cd} \cdot \text{m}^{-2}$. Rozměry jednoho pixelu jsou $269\text{ }\mu\text{m} \times 269\text{ }\mu\text{m}$ a celkem jich je na displeji $1\,920 \times 1\,200$ ^[38].

LED panel

LED panel FR-BL89X155 od firmy Edmund Optics poskytuje rovnoměrné osvětlení s nominální teplotou chromatičnosti $6\,000\text{ K}$. Aktivní svítící oblast má rozměry $89\text{ mm} \times 155\text{ mm}$. Výrobce udává uniformitu osvětlení $\pm 5\%$ ^[39].

Šedé filtry

Optické šedé filtry redukují množství světla, které jím prochází. K dispozici máme několik šedých filtrů od firmy Thorlabs s různou optickou hustotou OD (tabulka 2.1).

CMOS kamera

Pro pořizování snímků používáme kameru acA1600-60gm od firmy Basler s monochromatickým CMOS senzorem. CMOS senzor nese označení EV76C570 a jeho rozlišení je $1\,600\text{ px} \times 1\,200\text{ px}$ ^[40]. Matici snímače tvoří pixely o velikost $4,5\text{ }\mu\text{m} \times 4,5\text{ }\mu\text{m}$, na nichž se nachází posunuté mikročočky. Diagonála aktivní plochy je 9 mm ^[25].

Tabulka 2.1: Optické šedé filtry od firmy Thorlabs s různou optickou hustotou OD.

OD	označení
0,1	NE201B
0,2	NE202B
0,3	NE203B
0,4	NE204B
1,0	NE210B
2,0	NE220B

Kameru lze ethernetovým kabelem připojit k počítači, kterým můžeme skrze program `pylon Viewer`^[41] kameru ovládat. Rovněž je k dispozici knihovna `pypylon`^[42], kterou je možné implementovat do vlastního kódu v jazyce `python`.

Objektiv

Objektiv Xenoplan 1.4/23 od firmy Schneider Kreuznach s numerickou aperturou $NA = 0,36$ má ohniskovou vzdálenost 23 mm. Clonové číslo je nastavitelné od 1,4 do 11 a ostření objektivem se provádí manuálně. Objektiv je vhodný pro senzory s úhlopříčkou 11 mm. Výrobce uvádí minimální rozlišení objektivu $3,65 \mu\text{m}$ ^[43].

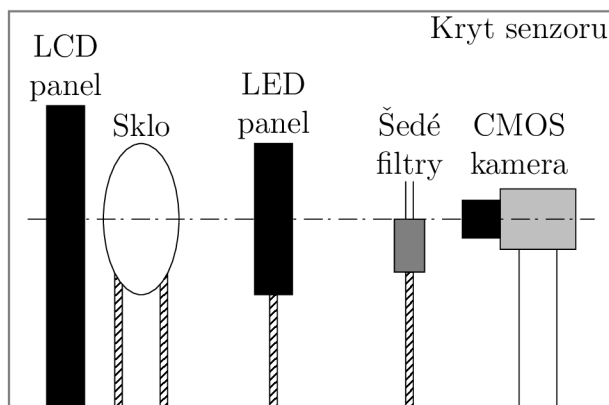
Držák kamery

Držák kamery disponuje mnoha šrouby, kterými kamerou manipulujeme. Pro další popis si zavedeme osový ortogonální souřadnicový systém v souladu s obr. 2.4(b). Nechť osa z je rovnoběžná s osou kolejnice a normálou LCD panelu. Dále nechť vertikální osa držáku kamery je rovnoběžná s osou x a osa y je kolmá na osu x a z .

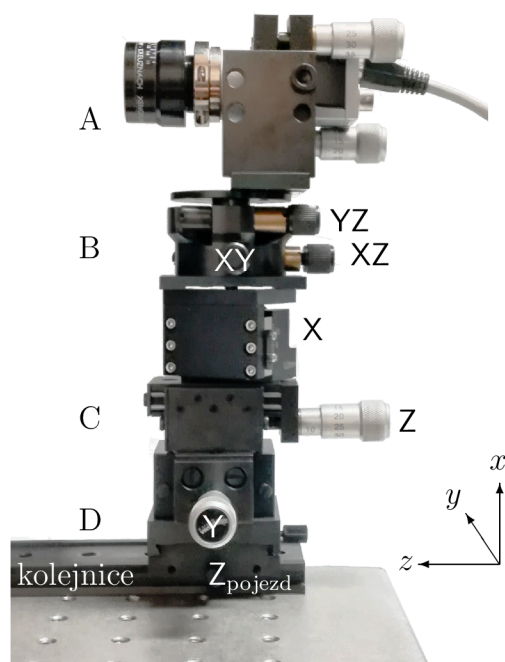
Nyní popíšeme držák kamery jako sestavu čtyř bloků A, B, C a D. V prvním bloku A je uchycena kamera. Přestože jsou v tomto bloku dva šrouby, kterými lze kamerou otáčet v rovinách rovnoběžných na dvě hlavní roviny xz a yz , tyto šrouby nepoužíváme. Druhý blok B kamerou otáčí třemi šrouby ve všech třech hlavních rovinách xy , xz a yz . Označme si tyto šrouby po řadě XY, XZ a YZ. Pomocí bloku C měníme jemným posuvem tří šroubů X, Y, Z polohu kamery ve směru hlavních os x , y , z . Poslední blok D manipuluje kamerou ve směru os z a y . Jedná se o kolejnicové uchycení a posuv je hrubý. Šroub pro posuv ve směru osy y nepoužíváme, pouze posuv ve směru osy z . Změna polohy se technicky neprovádí rotační šroubu, ale mechanickým pojezdem po kolejnici. Tento posuv si označme jako Z_{pojezd} .

2.2.2 Programy

Pro zjednodušení optimalizace, nastavování senzoru a obecně práce se senzorem slouží několik, pro účely této práce vzniklých programů (`pythonMatice`, `pythonOstrost`, `pythonMoare` a `pythonKontrast`). S nimi lze obsluhovat vykreslování binárního obrazce na LCD panelu, pořizovat snímky kamerou, ostřit objektivem na vybranou rovinu v jeho zorném poli, minimalizovat jev moaré a maximalizovat kontrast binárního obrazce. Některé programy pracují s vlastními binárními obrazci. Zdrojové kódy programů jsou v příloze C Ostatní na příloženém CD.



(a)



(b)

Obrázek 2.4: (a) Schéma senzoru a (b) popis bloků a manipulačních šroubů držáku kamery.

pythonMaticce

Pro práci se senzorem používáme program `pythonMaticce` s grafickým uživatelským rozhraním, ve kterém lze vytvořit, uložit nebo načíst matici vzoru binárního obrazce. Pro úpravu matice slouží vestavěný editor. Dále je možné využít funkci vytvoření matice proužků s periodou T a úhlem sklonu θ svíraným mezi horizontální rovinou a osou proužků.

Hlavní funkcí programu je odeslání elementární matice na druhý počítač svému podprogramu `pythonMaticceScreen`. Ten z přijaté elementární matice vytváří binární obrazec, který poté vykresluje na LCD panel. Program `pythonMaticce` také komunikuje s kamerou a pořizuje snímky se zadanou expoziční τ . Ukládá snímky jednotlivě i v sérii a pořizuje rovněž výřezy snímků.

Obsahuje funkci pro pořízení sérií snímků nutných pro metodu FFC, kdy zaznamená nejprve N snímků při zakrytém objektivu, poté čeká 20 s pro nastavení LED panelu a optických šedých filtrů a pokračuje pořízením dalších $M = N$ snímků.

pythonOstrost

K zaostření obrazu máme k dispozici program `pythonOstrost`. Pomocí něj jsme schopni poměrně spolehlivě a objektivně zaostřit na požadovanou rovinu v zorném poli objektivu. Program běží ve smyčce, kdy načítá z kamery snímky s expoziční $\tau = 60$ ms. Z nich ihned vybírá oblast (6×6) px² ze středu snímku. V tomto poli nachází minimální a maximální intenzitu pixelů, z nichž počítá kontrast podle vztahu (1.4). Na konci cyklu v grafickém uživatelském rozhraní vykresluje výřez snímku a vypisuje nejmenší hodnotu intenzity a také kontrast.

pythonMoare

Program `pythonMoare` je velice jednoduchý a svůj výstup vypisuje do příkazového řádku. Ve smyčce pořizuje kamerou s expozicí $\tau = 20$ ms snímek, z něhož vybere dva řádky. Ty se ve svých středech protínají pod pravým úhlem. Vertikální řádek představuje výšku a horizontální šířku snímku, který reprezentuje oblast, ve které chceme minimalizovat jev moaré.

Program na obou souborech vyhodnocuje minimální a maximální intenzitu. Rozdíly těchto intenzit obou souborů poté vypisuje do příkazového řádku.

pythonKontrast

Pro maximalizaci kontrastu používáme program `pythonKontrast`. Program ve smyčce ze snímku zaznamenaném kamerou při expozici $\tau = 40$ ms vybírá středový kříž s vertikální a horizontální úsečkou o délkách 100 px. Na těchto úsečkách vyhodnocuje kontrasty podle vztahu (1.4) z minimálních a maximálních intenzit.

V grafickém uživatelském rozhraní tyto kontrasty vypisuje a také vykresluje oba průběhy intenzity.

2.3 Příprava senzoru pro měření

2.3.1 Zobrazovací rovnice objektivu

Maticice LCD panelu v předmětové rovině objektivu je tvořena pixely. Předpokládejme, že mají čtvercový tvar. Příčnou velikost jednoho pixelu LCD panelu (strana čtverce) označme y_{LCD} . V obrazové rovině objektivu se nachází CMOS snímač, u něž také předpokládáme čtvercový tvar jeho pixelů. Příčná velikost jednoho pixelu je y_{CMOS} . Nyní dosadíme tyto veličiny do zobrazovací rovnice (1.10) s tím, že f a f' představují předmětovou a obrazovou ohniskovou vzdálenost objektivu. Pak dostáváme zobrazovací rovnici určující předmětovou vzdálenost a od předmětové hlavní roviny objektivu k předmětové rovině LCD panelu

$$a = \frac{f'}{n_{\mathbb{R}}} \cdot \frac{y_{\text{LCD}}}{y_{\text{CMOS}}} + f. \quad (2.1)$$

Obě příčné velikosti pixelů i ohniskové vzdálenosti lze snadno zjistit z datových listů výrobců. Ti však neuvádí k těmto údajům nejistotu, kterou může být obtížné stanovit pro každý výrobek zvlášť. Z tohoto důvodu nejprve zjistíme, a případně ověříme, některé z těchto parametrů.

Příčné velikosti pixelů LCD panelu a snímače kamery

Pixel snímače CMOS kamery má podle datového listu vertikální i horizontální rozměr $y_{\text{CMOS}} = 4,5 \mu\text{m}$ ^[40]. Považujeme tuto hodnotu za skutečnou, protože není snadné ji ověřit. Z datového listu LCD panelu čteme, že vertikální i horizontální rozměr pixelu je $y_{\text{LCD}} = 269 \mu\text{m}$ ^[38]. Protože jde o relativně velkou hodnotu, můžeme ji ověřit. K tomuto účelu využijeme kameru s telecentrickým objektivem S5LPJ1199 firmy Sill Optics, který potlačuje perspektivu tak, že jím prochází pouze paprsky rovnoběžné (resp. paprsky téměř rovnoběžné) s optickou osou objektivu. Telecentrický objektiv s pracovní

vzdáleností 92,0 mm a s relativní chybou 2 % má konstantní příčné zvětšení $\Gamma = 0,2$ s relativní chybou 5 %. Maximální distorze měřená pro vlnovou délku 586 nm v případě použitého CMOS snímače je 0,1 %^[44].

Telecentrický objektiv

Telecentrický objektiv eliminuje perspektivu a zvětšení předmětu zůstává konstantní. Přestože také potlačuje distorzi, která se definuje jako procentuální odchýlení polohy bodu v obrazové rovině od předpokládané polohy měřeno od optické osy

$$\text{Distorze [\%]} = \frac{\text{Měřená poloha [px]} - \text{Předpokládaná poloha [px]}}{\text{Předpokládaná poloha [px]}} \cdot 100 \%, \quad (2.2)$$

tak pro maximální přesnost je nutné dodržet pracovní vzdálenost kvůli závislosti distorze na vzdálenosti předmětu od objektivu. Pro dosažení ještě vyšší míry přesnosti je nutná kalibrace použitím kalibrační mřížky umístěné do středu hloubky ostrosti.

Telecentrickým objektivem nelze přeastřovat způsobem jako u běžných objektivů, ale polohu obrazové roviny měníme vkládáním kovových kroužků definované tloušťky do prostoru, kde je kamera našroubovaná na objektiv.

Telecentrický objektiv má hloubku ostrosti a mimo ni je předmět rozostřený. Avšak jeho velikost obrazu zůstává stejná, pomíneme-li rozmazání okrajů obrazu.

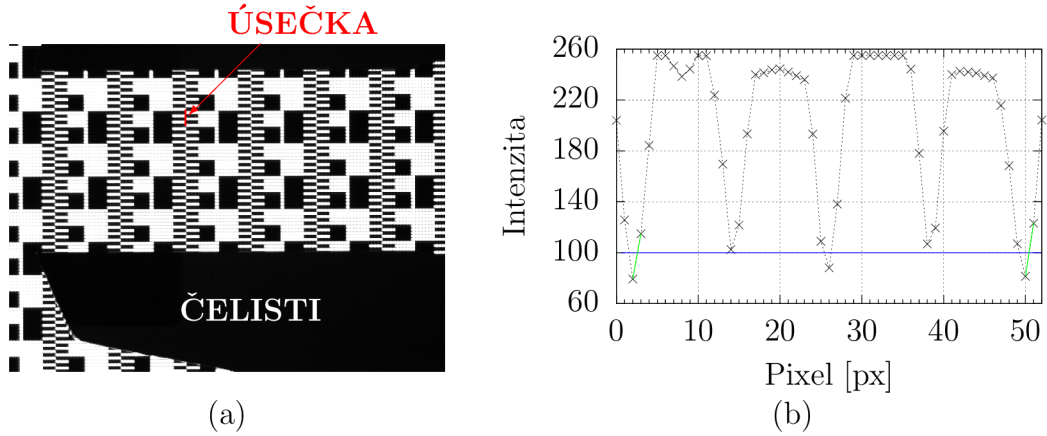
Rozměr pixelu LCD panelu y_{LCD} určíme dvěma metodami. V první metodě měříme příčnou velikost obrazu pixelu LCD panelu v jednotkách pixelů CMOS snímače. Tuto veličinu označme jako periodu T_{px} (dolní index označuje jednotku pixel, v jejíž násobcích je veličina vyjádřena), a z ní poté určíme jí odpovídající příčný rozměr pixelu LCD panelu v jednotkách mikrometrů.

Ve druhé metodě umísťujeme do roviny LCD panelu posuvné měřítko s danou roztečí čelistí tak, aby se čelisti nacházely v zorném poli objektivu. Na základě pořizovaného snímku poté sečteme periody T_{px} mezi oběma čelistmi. Tuto sumu označme PP (poznamenejme, že počet period PP nemusí být celočíselný) a jednoduše získáváme velikost pixelu LCD panelu v mikrometrech. Pro obě metody postačuje pořídit jeden společný snímek kamerou s telecentrickým objektivem.

Kameru s objektivem umísťujeme tak, aby osa objektivu byla rovnoběžná s normálou LCD panelu, na němž je vyobrazen obrazec v barevném režimu RGB. Na panel přikládáme posuvné měřítko pro kontrolní měření. Na posuvném měřítku je nastavena taková rozteč čelistí, aby v zorném poli objektivu bylo dostatečné množství pixelů LCD panelu mezi těmito čelistmi. Proto nastavujeme rozteč 15 mm a poté odečítáme skutečnou hodnotu $\Delta_{\text{pm}} = 14,95$ mm. S přiloženým měřidlem na LCD panelu pořizujeme snímek s expoziční dobou $\tau = 20$ ms [obr. 2.5(a)].

Metoda 1

Vhodně vybíráme úsečku pro měření periody T_{px} [na obr. 2.5(a) označená červeně] a její průběh intenzity vykresluje v závislosti na relativní souřadnici v jednotkách pixelů [obr. 2.5(b)]. Tato úsečka zahrnuje celkem čtyři periody T_{px} . Za krajní hodnoty vyhodnocovaného intervalu považujeme průsečíky dvou zeleně vyznačených úseček



Obrázek 2.5: (a) Snímek binárního obrazce na LCD panelu spolu s čelistmi posuvného měřítka pořízený kamerou vybavenou telecentrickým objektivem. Červeně je vyznačena vybraná úsečka pro určení periody T_{px} . (b) Intenzitní profil červené úsečky s vyhodnocovaným intervalem určeným dvěma průsečíky ($p_{\text{levý}}$, $p_{\text{pravý}}$) zelených úseček s modrou přímkou značící intenzitu o konstantní hodnotě 100.

(mezi dvěma naměřenými intenzitami) s konstantní hodnotou intenzity o velikosti 100 označenou modrou horizontální přímkou. Průměty průsečíků na horizontální ose jsou $p_{\text{levý}} = 2,5809 \text{ px}$ a $p_{\text{pravý}} = 50,4444 \text{ px}$.

Hledanou periodu T_{px} si vyjádříme jako jejich rozdíl dělený čtyřmi

$$T_{\text{px}} = \frac{p_{\text{pravý}} - p_{\text{levý}}}{4}. \quad (2.3)$$

Mezi periodou T_{px} a příčnou velikostí rozměru pixelu LCD panelu y_{LCD} platí vztah $\Gamma \cdot y_{\text{LCD}} = T_{\text{px}} \cdot y_{\text{CMOS}}$. Odtud si vyjádříme y_{LCD} s využitím vztahu (2.3)

$$y_{\text{LCD}} = \frac{p_{\text{pravý}} - p_{\text{levý}}}{4\Gamma} y_{\text{CMOS}}. \quad (2.4)$$

Po dosazení získáváme výsledek měření první metodou pro kontrolu rozměru pixelu LCD panelu

$$y_{\text{LCD}} = (269 \pm 12) \mu\text{m}, \quad (2.5)$$

kde $u_C(y_{\text{LCD}}) = 12 \mu\text{m}$ je standardní kombinovaná nejistota².

Metoda 2

Ve druhém měření na obr. 2.5(a) zvolme svislou žebříkovitou strukturu třetí ze šesti zleva. Spočítejme celistvé periody T_{px} a toto číslo označme PP_c . Počet celistvých period zahrnuje ty periody, které mají zřejmý horní i spodní okraj. Takovýchto period T_{px} napočítáme celkem $PP_c = 55$ a celková délka napočítaných period je $d_c = 657 \text{ px}$. Podílem celkové délky d_c počtem period PP_c si vyjádříme délku jedné periody

$$T_{\text{px}} = \frac{d_c}{PP_c}. \quad (2.6)$$

Odhadněme vzdálenost $\Delta x_{\text{horní}}$ od horní meze intervalu 55 period T_{px} k hraně horní čelisti posuvného měřítka na 1 px. Totéž provedme pro dolní mez, kde vzdálenost

²Určení nejistoty je vysvětleno v sekci Nejistoty měření - Metoda 1.

$\Delta x_{\text{dolní}}$ odhadujeme na 4 px. Součet obou vzdáleností $\Delta x_{\text{horní}} + \Delta x_{\text{dolní}} = 5$ px a počet period PP ohraničený oběma čelistmi je proto

$$PP = \frac{\Delta x_{\text{horní}} + \Delta x_{\text{dolní}}}{T_{\text{px}}} + PP_c. \quad (2.7)$$

Hodnota počtu period PP pochopitelně není přirozeným číslem.

Odtud již můžeme jednoduše spočítat příčnou velikost pixelu LCD panelu y_{LCD} jako podíl rozteče čelistí posuvného měřítka Δ_{pm} a počtu period PP jimi svíranými, $y_{\text{LCD}} = \Delta_{\text{pm}}/PP$. Pokud využijeme vztahů (2.6), (2.7), můžeme vyjádřit

$$y_{\text{LCD}} = \Delta_{\text{pm}} \left[PP_c \left(1 + \frac{\Delta x_{\text{horní}} + \Delta x_{\text{dolní}}}{d_c} \right) \right]^{-1}. \quad (2.8)$$

Po dosazení nám vychází

$$y_{\text{LCD}} = (270 \pm 15) \mu\text{m}, \quad (2.9)$$

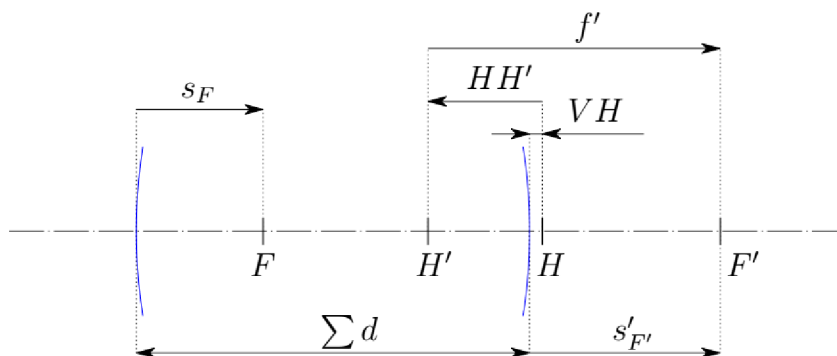
kde $u_C(y_{\text{LCD}}) = 15 \mu\text{m}$ je standardní kombinovaná nejistota³.

Srovnáme-li nyní výsledky měření oběma metodami, docházíme k závěru, že výsledek metody 1 má menší nejistotu, a proto dále počítejme s touto hodnotou velikosti pixelu LCD panelu.

Předmětová a obrazová ohnisková vzdálenost objektivu kamery

Z datového listu objektivu Xenoplan 1.4/23 od výrobce Schneider-Kreuznach^[43] přímo zjišťujeme obrazovou ohniskovou vzdálenost $f' = 22,52$ mm. Předmětovou ohniskovou vzdálenost odvodíme z dalších uvedených parametrů, kterými jsou poloha předmětového ohniskového bodu $s_F = 10,23$ mm, poloha obrazového ohniskového bodu $s'_{F'} = 14,95$ mm, vzdálenost mezi předmětovým a obrazovým hlavním bodem $HH' = -9,41$ mm a celková optická délka $\sum d = 30,93$ mm, což je vzdálenost mezi předním vrcholem první čočky a zadním vrcholem poslední čočky. Výrobce se přitom řídí znaménkovou konvencí podle normy DIN 1335.

Tyto parametry vynesme do obr. 2.6 v souladu s normou a na jeho základě jsme schopni zjistit $f = 22,54$ mm.



Obrázek 2.6: Optická soustava objektivu.

³Určení nejistoty je vysvětleno v sekci Nejistoty měření - Metoda 2.

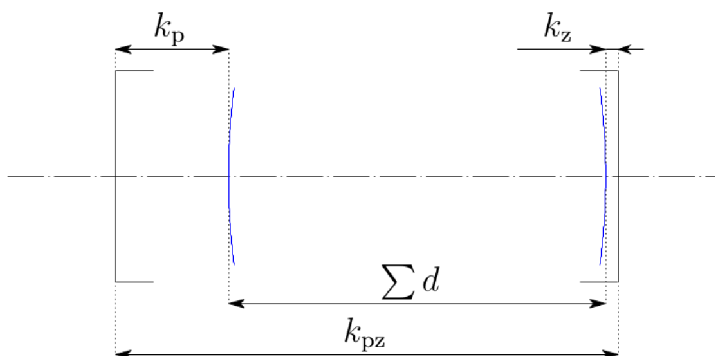
Nejistoty zobrazovací rovnice

Zdálo by se, že máme vše potřebné pro výpočet předmětové vzdálenosti a z rovnice (2.1). Nicméně, protože je snadnější určit vzdálenost předmětu od hrany objímky objektivu než od předmětové hlavní roviny, je třeba zjistit vzdálenost této roviny k hraně objímky. Vzdálenost předmětu od hrany objímky označme l a vzdálenost od hrany objímky k předmětové hlavní rovině l' . Platí tedy

$$a = l + l'. \quad (2.10)$$

Ze schématu na obr. 2.6 můžeme vypočítat vzdálenost předmětové hlavní roviny a vrcholu první optické plochy jako součet $\sum d$ a vzdálenosti označené VH , která je rovna 1,84 mm.

Přední optická plocha je zasazena v objímce. Je tedy nutné určit vzdálenost od roviny hrany objímky k vrcholu optické plochy, kterou označme k_p . Výrobce v datovém listu objektivu uvádí hodnotu tohoto parametru $k_p = 8,2$ mm spolu s dalšími rozměry objektivu $k_z = 1,3$ mm a $k_{pz} = 40,7$ mm^[43], které jsou vyneseny do obr. 2.7. Tyto



Obrázek 2.7: Vybrané rozměry objektivu podrobené měření (vlevo je přední část objektivu).

rozměry jsme schopni ověřit posuvným měřítkem. Při měření rozměrů k_p a k_z přiložíme mezi nechráněnou optickou plochu a dolní konec posuvného měřítka optický papír, jehož tloušťka změřená posuvným měřítkem je $d_{\text{papír}} = (69 \pm 20) \mu\text{m}$, kde $u_C(d_{\text{papír}}) = 20 \mu\text{m}$ je standardní kombinovaná nejistota⁴. Naměřené rozměry k_p , k_z v tabulce 2.2 jsou součtem tloušťky papíru $d_{\text{papír}}$ a naměřených hloubek h_p , h_z posuvným měřítkem, tedy $k_p = h_p + d_{\text{papír}}$ a $k_z = h_z + d_{\text{papír}}$ ⁵. V tabulce 2.2 je dále uveden rozměr k_{pz} ⁶. Zjištěné rozměry jsou srovnány s těmi udávanými výrobcem^[43]. V posledním sloupci je pro porovnání $\sum d = k_{pz} - k_p - k_z$ ⁷.

Je patrné, že rozdíly parametrů nejsou zanedbatelné, což platí především pro $\sum d$, jejíž výrobcem udávaná hodnota by měla být přesná. Její relativní chyba δ je sice

⁴Standardní kombinovanou nejistotu $u_C(d_{\text{papír}})$ počítáme ze dvou nejistot. První nejistotou je standardní nejistota typu A, kdy z celkem 21 měření naměříme $13 \times$ tloušťku optického papírku $0,05$ mm a $8 \times 0,1$ mm. Podle vztahu (1.25) získáváme $u_A(d_{\text{papír}}) = 0,0054294$ mm. Druhá standardní nejistota je typu B a vychází z nejistoty stupnice posuvného měřítka a již dříve jsme ji určili jako $u_B(\Delta_{\text{pm}}) = 0,014434$ mm. Obě nezaokrouhlené nejistoty dosaďme do vztahu (1.22) a vychází nám $u_C(d_{\text{papír}}) = 0,019863$ mm $\doteq 0,020$ mm.

⁵Nejistoty rozměrů k_p , k_z spočítáme jako lineární kombinovanou standardní nejistotu tloušťky papíru a posuvného měřítka $u_C(k_p) = u_C(k_z) = u_C(d_{\text{papír}}) + u_B(\Delta_{\text{pm}}) = 0,034297$ mm $\doteq 0,04$ mm.

⁶Nejistota $u_C(k_{pz})$ je rovna nejistotě posuvného měřítka $u_B(\Delta_{\text{pm}}) \doteq 0,015$.

⁷Nejistota $u_C(\sum d)$ je rovna $u_C(k_p) + u_C(k_z) + u_C(k_{pz}) = 0,083028 \doteq 0,09$ mm.

Tabulka 2.2: Srovnání naměřených rozměrů objektivu s hodnotami uváděnými výrobcem.

-	k_p (mm)	k_z (mm)	k_{pz} (mm)	$\sum d$ (mm)
měření	$8,47 \pm 0,04$	$0,72 \pm 0,04$	$39,719 \pm 0,015$	$30,53 \pm 0,09$
výrobce	8,2	1,3	40,7	30,93

1,29 %⁸, my bychom ji ale chtěli v řádu promile vzhledem k přesnosti optických parametrů. Tuto nepřesnost zohledněme tak, že hodnotám f a f' přiřadíme nejistoty $u_C(f)$ a $u_C(f')$, které počítáme jako $u_C(f) = \delta \cdot f$ a $u_C(f') = \delta \cdot f'$ ⁹. Totéž provedeme pro parametr $\sum d$ udávaný výrobcem a vzdálenost VH ¹⁰.

Jsme tedy schopni zjistit vzdálenost předmětové hlavní roviny a hrany objímky $l' = k_p + \sum d + VH = (41,2 \pm 0,5)$ mm¹¹, kde k_p je změřená hodnota a $\sum d$ a VH jsou od výrobce^[43]. Shrňme si všechny potřebné parametry:

$$\begin{aligned}
 y_{\text{CMOS}} &= 4,5 \mu\text{m}, \\
 y_{\text{LCD}} &= (269 \pm 9) \mu\text{m}, \\
 f &= (22,5 \pm 0,3) \text{ mm}, \\
 f' &= (22,5 \pm 0,3) \text{ mm}, \\
 l' &= (41,2 \pm 0,5) \text{ mm}.
 \end{aligned} \tag{2.11}$$

Protože ohniskové vzdálenosti f a f' jsou spolu s nejistotami shodné, můžeme položit $f = f'$. Nyní z rovnic (2.1) a (2.10) vyjádříme l

$$l = f' \left(\frac{1}{n_{\mathbb{R}}} \cdot \frac{y_{\text{LCD}}}{y_{\text{CMOS}}} + 1 \right) - l'. \tag{2.12}$$

Tento vztah pro daný $n_{\mathbb{R}}$ násobek¹² určuje vzdálenost l , což je vzdálenost předmětu v rovině LCD panelu od hrany objímky objektivu. Při této vzdálenosti l je příčná velikost obrazu pixelu LCD panelu $n_{\mathbb{R}}$ násobkem příčné velikosti pixelu snímače kamery.

Do vztahu (2.12) dosadíme $n_{\mathbb{R}} = 1$ a spočítáme l_1 . Totéž provedeme pro $n_{\mathbb{R}} = 2$ a $n_{\mathbb{R}} = 3$ ¹³.

$$\begin{aligned}
 l_1 &= (1,33 \pm 0,08) \text{ m}, \\
 l_2 &= (0,66 \pm 0,04) \text{ m}, \\
 l_3 &= (0,431 \pm 0,027) \text{ m}.
 \end{aligned} \tag{2.13}$$

Poznamenejme, že zobrazovací rovnice v paraxiální aproximaci platí v oblasti blízko optické osy. Pro větší vzdálenosti od optické osy již zobrazení není přesné a jsou zjevné aberace optického zobrazení.

Nejistoty měření velikosti pixelu LCD panelu a vzdálenosti předmětu od hrany objímky objektivu

Metoda 1

⁸ Relativní chybu určujeme jako $\left| \frac{30,531 - 30,93}{30,93} \right| \cdot 100 \% \doteq 1,29 \%$.

⁹ $u_C(f) = \delta \cdot f = 0,29077 \text{ mm} \doteq 0,3 \text{ mm}$, $u_C(f') = \delta \cdot f' = 0,29051 \text{ mm} \doteq 0,3 \text{ mm}$.

¹⁰ $u_C(\sum d) = \delta \cdot \sum d = 0,39900 \text{ mm}$, $u_C(VH) = \delta \cdot VH = 0,023736 \text{ mm}$.

¹¹ $u_C(l') = u_C(k_p) + u_C(\sum d) + u_C(VH) = 0,45703 \text{ mm} \doteq 0,5 \text{ mm}$.

¹² Připomeňme si, že v obrazové rovině detektoru vzniká obraz předmětu pixelu LCD panelu y_{LCD} a tento obraz je obecně neceločíselným $n_{\mathbb{R}}$ násobkem velikosti pixelu CMOS snímače y_{CMOS} .

¹³ Určení nejistot je vysvětleno v sekci Nejistoty měření - Vzdálenost předmětu od hrany objímky objektivu.

Telecentrický objektiv má nenulovou disperzi rovnu 0,1 %. Uvažujme bod na snímku o souřadnicích $[x, y]$, jehož vzdálenost od středu snímku značíme jako předpokládanou polohu v souladu s popisem ve vztahu (2.2). Rozměry snímku jsou $1\,600\text{ px} \times 1\,200\text{ px}$. Potom

$$\text{Předpokládaná poloha [px]} = \sqrt{\left(\frac{1600}{2} - x\right)^2 + \left(\frac{1200}{2} - y\right)^2}. \quad (2.14)$$

Rozdíl Δ mezi měřenou a předpokládanou polohou je

$$\Delta[\text{px}] = \text{Předpokládaná poloha [px]} \cdot \frac{\text{Distorze [\%]}}{100\%}, \quad (2.15)$$

což je pološířka apriorního rovnoměrného rozdělení, z níž vycházíme při určení standardní nejistoty typu B polohy pixelu p o souřadnicích $[x, y]$ vlivem distorze

$$u_B(\text{distorze}) = \frac{1}{\sqrt{3}} \cdot \sqrt{\left(\frac{1600}{2} - x\right)^2 + \left(\frac{1200}{2} - y\right)^2} \cdot \frac{\text{Distorze [\%]}}{100\%}. \quad (2.16)$$

Uveďme si vztah pro kombinovanou standardní nejistotu polohy pixelu p . Ta se skládá z nejistoty vlivem distorze a nejistoty odečtu polohy ze snímku. Nejmenší dílek snímku je 1 pixel. Vynásobíme-li poloviční šířku intervalu rovnoměrného rozdělení koeficientem $\frac{1}{\sqrt{3}}$, získáváme standardní nejistotu typu B $u_B(\text{dílek}) = \frac{0,5}{\sqrt{3}}\text{ px} = 0,288\,68\text{ px}$. Nyní můžeme vypočítat kombinovanou standardní nejistotu polohy pixelu p o souřadnicích $[x, y]$

$$u_C(p[x,y]) = u_B(\text{distorze}) + u_B(\text{dílek}). \quad (2.17)$$

Spočítejme nejistoty průmětů průsečíků $p_{\text{levý}}$ a $p_{\text{pravý}}$ na obr. 2.5(b). Pixelům na pozicích 2 px a 50 px na obr. 2.5(b) odpovídají souřadnice $[648, 244]$ a $[648, 291]$ na snímku pořízeném kamerou. Dosadíme souřadnice do vztahu (2.17). Potom kombinované standardní nejistoty obou průmětů průsečíků jsou $u_C(p_{\text{levý}}) = 0,365\,08\text{ px}$ a $u_C(p_{\text{pravý}}) = 0,350\,52\text{ px}$.

Výpočet příčné velikosti pixelu LCD panelu je dán vztahem (2.4). Zvětšení Γ telecentrického objektivu je 0,2 a jeho relativní chyba je podle výrobce 5 %^[44]. Předpokládejme, že hodnota zvětšení jistě leží v uzavřeném intervalu $\langle(1 - 0,05)\Gamma; (1 + 0,05)\Gamma\rangle = \langle 0,19; 0,21\rangle$ a rozdělení na tomto intervalu je rovnoměrné. Poloviční šířku tohoto intervalu 0,01 násobíme koeficientem $\frac{1}{\sqrt{3}}$ a získáváme standardní nejistotu typu B $u_B(\Gamma) = 0,01/\sqrt{3} = 0,005\,773\,5$. Nyní máme vše pro výpočet kombinované standardní nejistoty velikosti pixelu LCD panelu¹⁴

$$u_C(y_{\text{LCD}}) = \left| \frac{\partial y_{\text{LCD}}}{\partial p_{\text{levý}}} \right| u_C(p_{\text{levý}}) + \left| \frac{\partial y_{\text{LCD}}}{\partial p_{\text{pravý}}} \right| u_C(p_{\text{pravý}}) + \left| \frac{\partial y_{\text{LCD}}}{\partial \Gamma} \right| u_B(\Gamma) \doteq 12\ \mu\text{m}.$$

¹⁴Nejistota s pěti platnými číslicemi má hodnotu $u_C(y_{\text{LCD}}) = 11,797\ \mu\text{m}$.

Metoda 2

Pro výpočet nejistoty $u_C(y_{\text{LCD}})$ berme v úvahu pouze nejistotu odečtu měřidla. Podíváme-li se totiž na vztah (2.8), je zřejmé, že nejistoty dělence i dělitele ve jmenovateli výrazu jsou silně korelované a jakákoli nejistota vzniklá u dělence je kompenzována téměř totožnou nejistotou vzniklou u dělitele. Navíc vzhledem k jejich velikostem s ohledem na nejistotu odečtu posuvného měřidla bychom je navíc mohli stejně zanedbat. Tedy $u_C(y_{\text{LCD}}) = u_B(\Delta_{\text{pm}}) \doteq 0,015 \text{ mm}$.

Vzdálenost předmětu od hrany objímky objektivu

Nejistotu vzdálenosti l předmětu v rovině LCD panelu od hrany objímky objektivu pro dané $n_{\mathbb{R}}$ určíme následovně. Vyjděme přitom ze vztahu (2.12), který dosadíme do lineárního tvaru kombinované standardní nejistoty (1.22)

$$u_C(l_{1,2,3}) = \left| \frac{\partial l}{\partial f'} \right| u_C(f') + \left| \frac{\partial l}{\partial y_{\text{LCD}}} \right| u_C(y_{\text{LCD}}) + \left| \frac{\partial l}{\partial l'} \right| u_C(l') \quad (2.18)$$

Potom pro $n_{\mathbb{R}} = 1, 2$ a 3 získáváme kombinované standardní nejistoty

$$\begin{aligned} n_{\mathbb{R}} = 1, u_C(l_1) &= 7,7099 \cdot 10^{-2} \text{ m} \doteq 8 \cdot 10^{-2} \text{ m}, \\ n_{\mathbb{R}} = 2, u_C(l_2) &= 3,8923 \cdot 10^{-2} \text{ m} \doteq 4 \cdot 10^{-2} \text{ m}, \\ n_{\mathbb{R}} = 3, u_C(l_3) &= 2,6198 \cdot 10^{-2} \text{ m} \doteq 2,7 \cdot 10^{-2} \text{ m}. \end{aligned} \quad (2.19)$$

2.3.2 Experimentální ověření zobrazovací rovnice senzoru

Ověřme odvozené teoretické vzdálenosti l_1, l_2 a l_3 od hrany objímky objektivu k LCD panelu. K tomu potřebujeme svinovací metr a program `pylon Viewer`^[41].

Začátek svinovacího metru uchyťme do držáku tak, aby jeho kovový konec přiléhal k LCD panelu a současně byla jeho rozvinutá část v ose senzoru. Pro $n_{\mathbb{R}} = 3$ je vzdálenost $l_3 = (431 \pm 27) \text{ mm}$. Posuvem Z_{pojezd} hrubě dostáváme kameru s ohledem na hranu objímky do této polohy. Po zajištění posuvu pozici jemně doladíme šroubem Z [obr. 2.4(b)].

Odstraňme držák spolu se svinovacím metrem ze zorného pole objektivu. Na objektivu kamery je přitom fixně zvolené clonové číslo 5,6. Jediné proměnné parametry kamery a objektivu jsou expoziční doba τ a rovina zaostření.

Těsně před LCD panelem v rovině xz (obr. 2.4) umísťujeme svisle kontrolní tyč, přičemž ve vodorovné rovině definované jejím horním okrajem leží osa objektivu. V programu `pylon Viewer` si najdeme středový pixel kamery o souřadnicích $[800, 600]_{\text{px}}$ a pomocí šroubů XZ, YZ [obr. 2.4(b)] dostaňme horní okraj tyče do středu obrazu kamery. Nyní máme osu objektivu rovnoběžnou s normálou LCD panelu.

Odstraňme kontrolní tyč před LCD panelem a sledujme celý vykreslený snímek programem `pylon Viewer`. Současně otáčejme šroubem XY a minimalizujme jev moaré ve snímku, který vzniká v důsledku zobrazení mřížky pixelů LCD panelu na mřížku pixelů CMOS snímače. Snažme se jev moaré potlačit především v centrální oblasti snímku. Cílem je, aby se poloměry křivosti čar vzoru jevu moaré blížily nekonečnu.

V programu `pylon Viewer` si opět přiblížme středový pixel kamery na souřadnicích $[800, 600]_{\text{px}}$ a šrouby X a Y doladíme zobrazení optickou soustavou tak, aby se pixel LCD panelu, který kamera snímá ve středu snímku, zobrazoval současně na tři pixely CMOS snímače. Pokud se pixel LCD panelu zobrazuje na více nebo méně jak tři pixely

CMOS snímače, potom šroubem Z otáčejme tak, aby se obrazová rovina, v níž je obraz pixelu LCD panelu třikrát větší jak velikost pixelu CMOS snímače, shodovala s rovinou CMOS snímače.

Přiložme držák se svinovacím metrem opět k LCD panelu a změřme vzdálenost l_3 k hraně objímky objektivu kamery. Vzdálenost je $l_3 = 430,5$ mm. Mezní hodnoty intervalu, kde se l_3 jistě nachází, jsou 430 mm a 431 mm. Stanovme kombinovanou standardní nejistotu měření $u_C(l_3) = u_B(l_3) = \frac{0,5}{\sqrt{3}}$ mm = 0,288 67 mm \doteq 0,3 mm.

Totéž provedme pro vzdálenosti l_1 a l_2 ¹⁵ Naměřené vzdálenosti porovnejme s teoretickými hodnotami [vztah (2.13)] v tabulce 2.3. Z ní plyne shoda obou postupů.

Tabulka 2.3: Porovnání naměřených a teoretických vzdáleností l od LCD panelu k hraně objímky objektivu pro $n_{\mathbb{R}} = 1, 2$ a 3.

-	l_1 (m)	l_2 (m)	l_3 (m)
teorie	$1,33 \pm 0,08$	$0,66 \pm 0,04$	$0,431 \pm 0,027$
měření	$1,332\ 0 \pm 0,000\ 3$	$0,657\ 5 \pm 0,000\ 3$	$0,430\ 5 \pm 0,000\ 3$

2.3.3 Optimalizace zobrazení binárního obrazce na CMOS snímači

Umístíme kameru postupně do všech poloh l_1 , l_2 a l_3 , jejichž hodnoty jsou v tabulce 2.3. Pořídíme pro každou polohu snímek s binárním obrazcem vertikálních proužků s elementární maticí $B = [0, 1]$. V každém ze snímků existuje místo, kde je jev moaré potlačen a obraz pixelu LCD panelu se zobrazuje právě na jeden pixel CMOS snímače pro l_1 , dva pixely pro l_2 a tři pixely pro l_3 [obr. 2.8(a) až (c)]. Zároveň existují místa, kde toto splněno není a jev moaré se zde silně projevuje [obr. 2.8(d) až (f)].

Pokud bychom počítali kontrast K , byla by jeho nejvyšší hodnota v případě vzdálenosti l_3 . Při tomto zobrazení ale vzniká parazitická struktura, kterou považujeme za nežádoucí. Přestože je binární obrazec více odolný vůči jevu moaré, tak je náročnější odlišit změny intenzity ve snímku na úrovni jednoho pixelu.

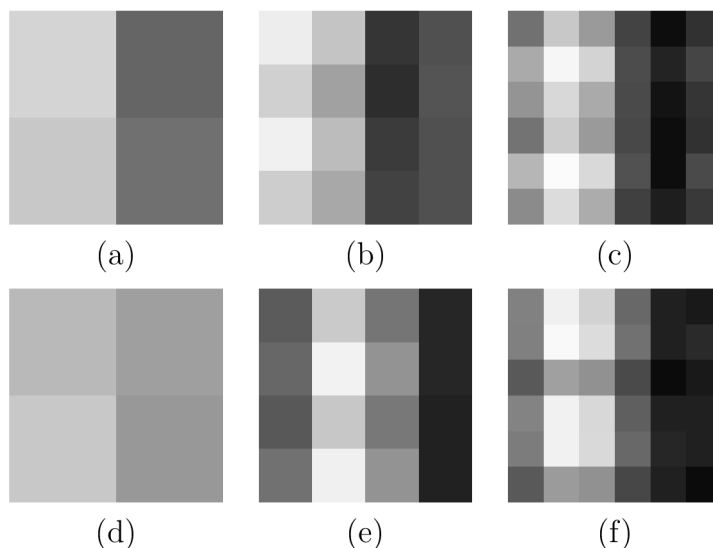
Naproti tomu pro vzdálenost l_1 by taková změna byla lépe identifikovatelná. Zřejmě jsou ale nejjemnější binární obrazce náchylnější k deformaci jevem moaré [obr. 2.8(a) a (d)]. Představíme-li si, že periodickými strukturami davající vznik jevu moaré jsou mřížky LCD panelu a CMOS snímače, pak bychom jejich vzájemným jemným posunem o velikosti zlomků periody struktury dosáhli také posunutí tmavé a světlé oblasti vzoru jevu moaré. Tím by se na části snímku [obr. 2.8(d)] zvýšil kontrast [obr. 2.8(a)]. Takto bychom důsledky parazitického jevu moaré mohli pro nejjemnější binární obrazce omezit.

S ohledem na požadavek co nejvěrnějšího zobrazení binárního obrazce na snímku pořízeném kamerou zvolme parametr l senzoru roven l_1 , kdy je naším cílem zobrazit pixel LCD panelu na pixel CMOS snímače.

2.3.4 Určení clonového čísla

Abychom zajistili ostrý obraz binárního obrazce na LCD panelu i samotného skla (průhledná část krytu světlometu), požadujeme dostatečnou hloubku ostrosti Δ_g . Z toho

¹⁵Nejistoty $u_C(l_1)$ a $u_C(l_2)$ jsou shodné s nejistotou $u_C(l_3)$.



Obrázek 2.8: Výřezy snímků s binárním obrazcem vertikálních proužků s elementární maticí $B = [0,1]$ při přesném zobrazení pixelu LCD panelu na jeden či více pixelů CMOS snímáče pro vzdálenost l rovna (a) l_1 , (b) l_2 a (c) l_3 a při nepřesném zobrazení pro (d) l_1 , (e) l_2 a (f) l_3 .

důvodu umístíme kryt světlometu vždy těsně před LCD panel, protože tím je potřeba nižší hloubka ostrosti, což úzce souvisí s clonovým číslem a vyšší světelností objektivu.

Umístíme-li kryt světlometu takto těsně před LCD panel, pak měřením posuvným metrem zjišťujeme, že vzdálenost mezi LCD panelem a nejvzdálenějším bodem na skle od LCD panelu nepřesahuje 16 cm, tj. hloubka ostrosti $\Delta_g > 16$ cm.

Pokud se LCD panel nachází na vnějším okraji hloubky ostrosti směrem od kamery, je jeho vzdálenost rovna předmětové vzdálenosti g_2 vztahované ke vstupní pupile. V souladu se značením ve vztahu (1.15) představuje naměřená vzdálenost 16 cm minimální požadovanou hloubku ostrosti Δ'_g .

Upravme rovnici (1.15)

$$\Delta'_g < \frac{2g^2G}{G^2 - g^2} \quad (2.20)$$

a dosadíme za g vzdálenost g_2 z rovnice (1.14) a poté za hyperfokální vzdálenost G podle (1.13). Po několika úpravách dostáváme vztah

$$\Delta'_g < \frac{2g_2^2yc}{f'^2 + 2g_2yc}. \quad (2.21)$$

Odtud vyjádříme clonové číslo

$$c > \frac{f'^2}{2g_2\Delta y'_0 \left(\frac{g_2}{\Delta'_g} - 1 \right)}. \quad (2.22)$$

Obrazová ohnisková vzdálenost f' je $(22,5 \pm 0,3)$ mm. Veličina $\Delta y'_0$ je maximální tolerovaný průměr obrazu bodu na snímáči kamery, který je roven velikosti pixelu snímáče $y_{\text{CMOS}} = 4,5 \mu\text{m}$. Předmětová vzdálenost vztahovaná ke vstupní pupile g_2 odpovídá vzdálenosti $l_1 = 1332$ mm LCD panelu od hrany objímky objektivu [vztah (2.13)], neboť poloha vstupní pupily odpovídá poloze hrany objímky objektivu. Dosazením

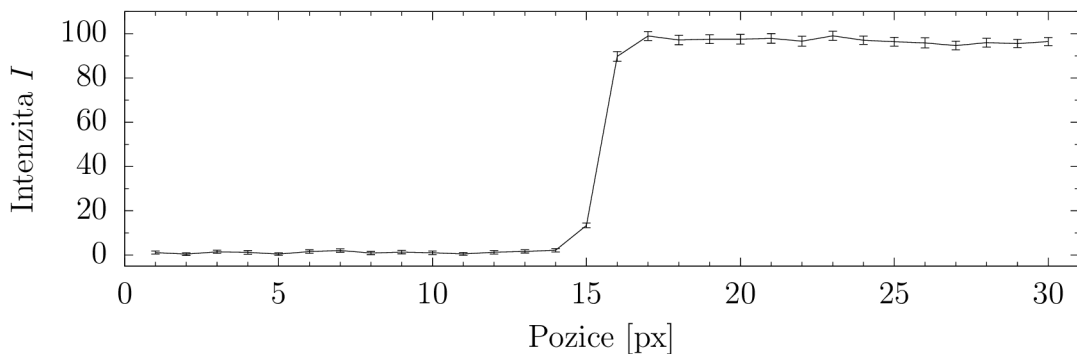
konkrétních hodnot do vztahu (2.22) zjišťujeme $c > (5,4 \pm 0,4)$ ¹⁶. Nejbližší clonové číslo na objektivu splňující tuto podmínku bez ohledu na nejistotu je $c = 5,6$, kterému podle (1.15) odpovídá hloubka ostrosti $\Delta_g = 189 \text{ mm}$ ¹⁷. Je tedy splněna podmínka o minimální hodnotě Δ_g . Neboli vše do vzdálenosti 189 mm od LCD panelu směrem ke kameře bychom měli na kameře pozorovat ostře.

2.3.5 Spektrální optimalizace kontrastu

Protože na LCD panelu vykreslujeme binární obrazec, jehož pozměněný obraz na snímku z kamery následně analyzujeme, je důležité dosáhnout jeho vysokého kontrastu. Binární obrazec může být na LCD panelu vykreslen při sedmi různých barevných režimech. Chceme rozhodnout, při kterém barevném režimu pozorujeme na CMOS snímači kamery nejvyšší kontrast.

Kamera je umístěná v pozici l_1 , tj. když se jeden pixel LCD panelu zobrazuje na jeden pixel CMOS snímače, a je zaostřena na plochu LCD panelu. Nastavme expozici kamery na $\tau = 5 \text{ ms}$. Při této hodnotě nemůže dojít k saturaci pixelů neboli při nejintenzivnějším barevném režimu RGB mají pixely hodnotu intenzity menší jak 255. Na LCD panelu nechme zobrazit vertikální jednotkový skok postupně ve všech sedmi barevných režimech tak, abychom svislé rozhraní světlé a tmavé oblasti pozorovali ve středu snímku kamery. Pro každý barevný režim je nutné sestavu opětovně seřadit šroubem Y v bloku C [obr. 2.4(b)] kvůli horizontálnímu posunu rozhraní vlivem vertikálního uspořádání subpixelů.

Pořídíme pro každý barevný režim 100 snímků. Z těchto snímků si vyberme řádek o délce 30 px, jehož dva prostřední pixely (15. a 16. pixel) odpovídají pixelům o souřadnicích [799, 600] a [800, 600] ve středu CMOS snímače. Vypočítejme pro každý pixel řádku jeho průměrnou hodnotu ze všech snímků. Na obrázku 2.9 je vykreslená intenzita v barevném režimu G spolu se směrodatnými odchylkami.



Obrázek 2.9: Průměrný průběh intenzity I jednotkového skoku na řádku CMOS snímače v barevném režimu G. Chybové úsečky odpovídají směrodatným odchylkám.

Na průběhu intenzity z průměrovaného řádku vyhodnoťme její strmost. Abychom ale mohli barevné režimy mezi sebou porovnat, musíme je normovat. K normování intenzitního průběhu každého barevného režimu použijme krajních deset hodnot, které

¹⁶Kombinovanou standardní nejistotu $u_C(c)$ určujeme podle vztahu (1.22), kam za funkci f dosazujeme pravou stranu rovnice (2.22), kterou derivujeme podle vstupních veličin f' a g_2 . Veličinu Δ'_g považujeme za přesnou, protože je určena s dostatečnou rezervou.

¹⁷V skutečnosti se dekorativní vady vyskytují na skle tak, že jej lze přiklopit k LCD panelu a tím ještě snížit požadovanou hloubku ostrosti.

již nejsou ovlivněny přechodem. Z krajních hodnot vlevo spočítáme průměrnou nízkou intenzitu \bar{I}_{\min} a z hodnot vpravo průměrnou vysokou intenzitu \bar{I}_{\max} , jak je uvedeno v tabulce 2.4¹⁸.

Tabulka 2.4: Průměrné hodnoty 10 krajních nízkých a vysokých intenzit \bar{I}_{\min} a \bar{I}_{\max} pro sedm barevných režimů LCD panelu. Ve třetím sloupci je jejich rozdíl.

-	\bar{I}_{\min}	\bar{I}_{\max}	$\Delta\bar{I}$
R	$1,24 \pm 0,18$	$79,97 \pm 0,26$	$78,7 \pm 0,5$
G	$1,15 \pm 0,16$	$96,5 \pm 0,4$	$95,4 \pm 0,6$
B	$0,95 \pm 0,15$	$53,42 \pm 0,12$	$52,47 \pm 0,26$
RG	$1,69 \pm 0,18$	$172,6 \pm 0,3$	$170,9 \pm 0,5$
RB	$1,58 \pm 0,18$	$126,91 \pm 0,18$	$125,3 \pm 0,4$
GB	$1,53 \pm 0,18$	$136,75 \pm 0,22$	$135,2 \pm 0,4$
RGB	$2,09 \pm 0,19$	$216,7 \pm 0,4$	$214,6 \pm 0,6$

Ze znalosti \bar{I}_{\min} a \bar{I}_{\max} dostáváme z průběhu intenzity $I(x)$ normovaný průběh

$$I_n(x) = \frac{x - \bar{I}_{\min}}{\bar{I}_{\max} - \bar{I}_{\min}}. \quad (2.23)$$

Spočítáme-li průměrnou nízkou a vysokou intenzitu z tohoto průběhu, dostáváme $\bar{I}_{\min} = 0$ a $\bar{I}_{\max} = 1$.

Uveďme si funkce, kterými fitujeme normalizované průběhy. Všechny funkce jsou prosté, mají vlastní limity $\lim_{x \rightarrow -\infty} = 0$ a $\lim_{x \rightarrow \infty} = 1$ a obsahují parametry p , q .

$$f_\sigma(x) = \frac{1}{1 + e^{-q\sigma(x-p_\sigma)}}, \quad (2.24)$$

$$f_E(x) = \frac{1}{2} \left[\frac{q_E(x - p_E)}{1 + |q_E(x - p_E)|} + 1 \right], \quad (2.25)$$

$$f_a(x) = \frac{1}{\pi} \arctan [q_a(x - p_a)] + \frac{1}{2}. \quad (2.26)$$

Funkce f_σ je obdobou sigmoidální přenosové funkce (angl. *Sigmoid Activation Function*) a f_E Elliotovy přenosové funkce (angl. *Elliot Activation Function*) z oblasti umělých neuronových sítí^[45]. Funkce f_a vychází z funkce arkus tangens.

Parametr p mající význam inflexního bodu také znamená posunutí, které z podstaty nemůže být pro všechny barevné režimy stejné. Proto jej musíme v nelineární regresi zahrnout. Pro rozhodnutí, který barevný režim LCD panelu ze sedmi možných poskytuje nejvyšší kontrast, není tento parametr podstatný a ve výsledcích regrese jej neuvádíme. Na druhou stranu se podívejme na koeficient determinace R^2 a parametr q (tab. 2.5 a 2.6).

Koeficient determinace R^2 udává, do jaké míry se regresní křivka shoduje s daty^[46] a můžeme jej přirovnat k procentuální správnosti našeho modelu^[47]. Pro všechny barevné režimy má nejvyšší hodnotu koeficientu determinace funkce f_σ a její hodnoty R^2 jsou ve všech případech větší jak 0,9, což považujeme za dostatečnou hodnotu míry podobnosti modelu s daty^[47]. Parametr q značí strmost přechodu, tedy od vyššího q očekáváme i vyšší ostrost zobrazení [obr. 2.10(a)]. Z dat v tabulkách 2.5 a 2.6 jsou

¹⁸Nejistoty $u_A(\bar{I}_{\min})$ a $u_A(\bar{I}_{\max})$ v tabulce 2.4 jsou standardní nejistoty typu A. Nejistota $u_C(\Delta\bar{I})$ je jejich zaokrouhleným součtem.

Tabulka 2.5: Koeficient determinace R^2 a parametr q tří regresních křivek aplikovaných na normalizovaný jednotkový přechod intenzity pro barevné režimy R, G a B.

barevný režim	R	G	B
R_σ^2	0,999 71	0,999 62	0,999 60
R_s^2	0,998 83	0,999 12	0,998 76
R_a^2	0,998 96	0,999 14	0,998 95
q_σ	$3,41 \pm 0,10$	$4,49 \pm 0,17$	$3,72 \pm 0,13$
q_E	$9,2 \pm 1,8$	20 ± 7	$7,9 \pm 1,1$
q_a	$5,9 \pm 0,8$	12 ± 4	$5,6 \pm 0,6$

Tabulka 2.6: Koeficient determinace R^2 a parametr q tří regresních křivek aplikovaných na normalizovaný jednotkový přechod intenzity pro barevné režimy RG, RB, GB a RGB.

barevný režim	RG	RB	GB	RGB
R_σ^2	0,999 83	0,999 52	0,998 77	0,999 75
R_s^2	0,999 16	0,997 69	0,997 75	0,998 12
R_a^2	0,999 27	0,998 33	0,998 04	0,998 69
q_σ	$3,43 \pm 0,08$	$2,10 \pm 0,07$	$1,89 \pm 0,10$	$2,76 \pm 0,07$
q_E	$9,3 \pm 1,6$	$4,2 \pm 0,7$	$4,5 \pm 0,7$	$4,8 \pm 0,6$
q_a	$6,0 \pm 0,8$	$2,9 \pm 0,3$	$3,1 \pm 0,4$	$3,6 \pm 0,3$

patrné nejvyšší hodnoty q_σ , q_E a q_a pro barevný režim G. Z toho plyne výběr tohoto barevného režimu pro další postup.

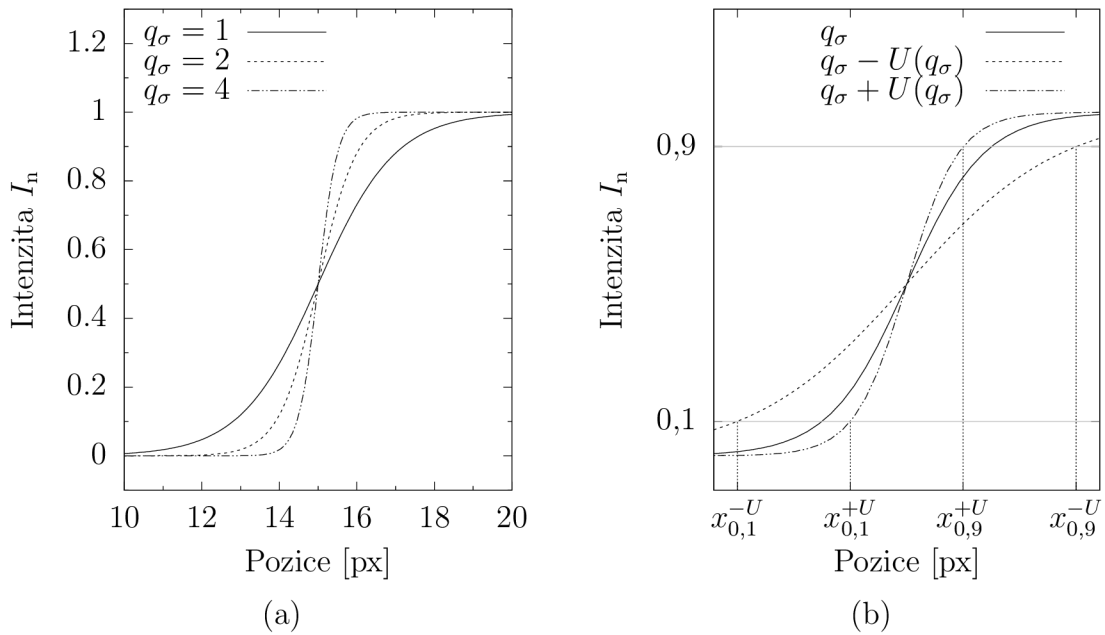
Přestože je hodnota parametru q největší pro barevný režim G, tak nevíme, jestli je tato hodnota dostatečně velká. Ukažme si proto, jak se nám různé hodnoty parametru q projevují na horizontální ose, která je v jednotkách pixelů [obr. 2.10(a)]. Využijme znalost parametru q_σ a jeho standardní nejistoty $u(q_\sigma)$, resp. koeficientem $k = 2$ rozšířené standardní nejistoty $U(q_\sigma) = k \cdot u(q_\sigma)$, a spočítejme interval délek ID na horizontální ose, kdy definujeme délku jako vzdálenost dvou bodů, které jsou průměty do horizontální osy. Promítanými body jsou průsečíky regresní funkce f_σ s normovanými intenzitami I_n rovnými 0,1 a 0,9. Na takové délce vzroste nebo poklesne normovaná intenzita o 80 %. Princip je znázorněn na obr. 2.10(b) pro tři různé q funkce f_σ . Pro tento výpočet si upravme vztah (2.24) a položíme jej roven I_n ve tvaru

$$I_n = \frac{1}{1 + e^{-[q_\sigma \pm U(q_\sigma)] \cdot x}}. \quad (2.27)$$

Všimněme si, že ve vztahu není zahrnut parametr p a naopak k parametru q je přičtena, nebo odečtena rozšířená standardní nejistota $U(q_\sigma)$. Ze vztahu (2.27) si vyjádříme

$$x = -\frac{\ln\left(\frac{1}{I_n} - 1\right)}{q_\sigma \pm U(q_\sigma)}. \quad (2.28)$$

Délku na horizontální ose pro $+U(q_\sigma)$ dostáváme rozdílem $x_{0,9}^{+U} - x_{0,1}^{+U}$, jak je znázorněno na obr. 2.10(b). Obdobně počítáme druhou délku pro $-U(q_\sigma)$. Obě délky nám vymezují interval délek ID , kde se s pravděpodobností přibližně 95 % nachází námi definovaná délka.



Obrázek 2.10: (a) Ukázka strmosti přechodu funkce f_σ pro různé hodnoty parametru q , (b) grafické znázornění průsečíků funkce (2.27) s konstantními hodnotami intenzity $I = 0,1$ a $I = 0,9$.

Tabulka 2.7: Horní a dolní meze intervalů délek ID pro sedm barevných režimů LCD panelu.

barevný režim	R	G	B	RG	RB	GB	RGB
horní mez ID	1,37	1,06	1,26	1,34	2,24	2,6	1,67
dolní mez ID	1,22	0,91	1,11	1,23	1,96	2,1	1,51

V tabulce 2.7 si všimněme, že ID pro barevný režim G má nejnižší hodnoty a nemá průnik s žádným jiným intervalem. Tento závěr potvrzuje správnost výběru barevného režimu G panelu LCD.

2.4 Nastavení senzoru

Senzor je před měřením potřeba správně nastavit. Požadujeme přesné zobrazení mřížky pixelů LCD panelu na mřížku pixelů CMOS snímače tak, aby se velikost obrazu pixelu první mřížky rovnala velikosti pixelu mřížky druhé. Je důležité mít zaostřeno na LCD panel a prostor těsně před ním, kam umisťujeme kryt světlometu [obr. 2.4(a)]. Stejně tak je třeba minimalizovat jev moaré a maximalizovat kontrast binárního obrazce v barevném režimu G panelu LCD. Rozdělme si nastavení senzoru na tři části, kdy v první zaostříme na LCD panel s využitím programu `pythonOstrost`, ve druhé minimalizujeme pomocí programu `pythonMoare` jev moaré s ohledem na střed snímku pořízeném kamerou a ve třetí části maximalizujeme kontrast pomocí `pythonKontrast`.

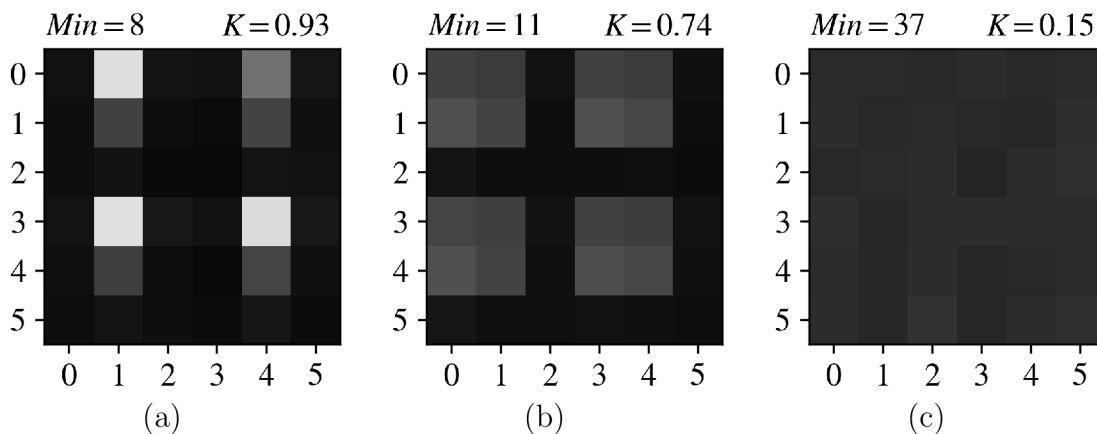
2.4.1 Zaostření na LCD panel

Šrouby na stojanu kamery [obr. 2.4(b)] jsou v polovině mezi svými krajními polohami a osa kamery je rovnoběžná s osou z . Kamera je od LCD panelu v takové vzdálenosti, aby hrana objímky objektivu byla od LCD panelu vzdálená $l_1 = 1\,332$ mm. Objektiv se clonou $f/5.6$ je třeba zaostřit na LCD panel tak, aby se nacházel na vnějším okraji hloubky ostroty směrem od kamery. Tím vznikne před LCD panelem vymezený prostor, v němž umístěný kryt světlometu pozorujeme ostře. K tomu využijeme program `pythonOstrost`, který pracuje s binárním obrazcem na LCD panelu, jehož elementární matice

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.29)$$

vytváří hustou mříž světlých pixelů. Program `pythonOstrost` opakovaně pořizuje kamerou snímek (expozice $\tau = 60$ ms) a vykresluje výřez ze středu snímku o rozměrech (6×6) px². Současně vypisuje nejmenší intenzitu I_{\min} a kontrast K .

Uveďme si dvě situace, ke kterým může dojít v rovině snímáče kamery při ostrém zobrazení. V ideálním případě se obraz světlého pixelu zcela překrývá s pixelem snímáče [reálná situace na obr. 2.11(a)]. Jestliže se obraz zcela nepřekrývá s pixelem snímáče, potom je intenzita světlého pixelu teoreticky rozložena mezi čtyři pixely snímáče [reálná situace na obr. 2.11(b)]. Mezi těmito čtveřicemi zůstává tmavý prostor o šířce 1 px, kde není intenzita světlých pixelů prakticky detekována. Při rozostřování objektivu roste intenzita těchto tmavých pixelů s tím, jak roste plocha neostrého obrazu světlého pixelu [reálná situace na obr. 2.11(c)]. Proto ostrost zvyšujeme minimalizací nejmenší intenzity I_{\min} .



Obrázek 2.11: Výstupy programu `pythonOstrost` s minimální intenzitou I_{\min} a kontrastem K . (a) Na plochu LCD panelu je zaostřeno, (b) i přes zaostřený obraz se pixel LCD panelu zobrazuje na čtyři pixely CMOS snímáče. (c) Plocha LCD panelu je mimo hloubku ostroty.

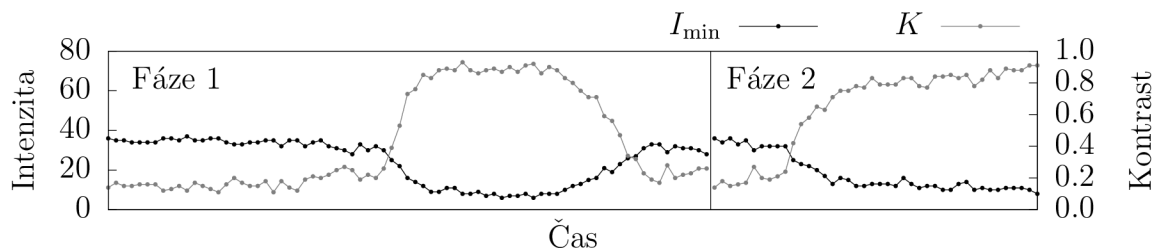
Se zvýšenou hustotou světlých pixelů tato vlastnost tmavých pixelů zaniká, protože mezi čtveřicemi nezůstává žádný prostor. Naopak snížená hustota světlých pixelů nepřináší další výhodu.

Při minimalizaci nejmenší intenzity I_{\min} postupujeme ve dvou fázích, ve kterých skenujeme, jak se mění nejmenší intenzita I_{\min} spolu s kontrastem K ve výřezu snímku (obr. 2.12). V první fázi mějme nejprve objektiv zaostřený na výchozí rovinu v prostoru

před LCD panelem směrem ke kameře, abychom pozorovali situaci na obr. 2.11(c), kde je LCD panel zřejmě mimo hloubku ostrosti. Kontrast K je velmi nízký, protože intenzita I je rovnoměrně rozložená do všech pixelů. Poté přeastřujeme na rovinu, která je ve větší vzdálenosti od objektivu a nachází se za LCD panelem, a to otáčením objektivu ve směru proti hodinovým ručičkám z pohledu kamery k LCD panelu. Během tohoto kroku se LCD panel dostává do prostoru vymezeném hloubkou ostrosti, kde jej pozorujeme ostře. V takové situaci kontrast K narůstá a nejmenší intenzita I_{\min} klesá na své minimum v souladu se situacemi na obr. 2.11(a), (b). Dalším přeastřováním je LCD panel opět mimo hloubku ostrosti a kamera snímá situaci podobnou té na obr. 2.11(c). Z dat zjišťujeme, že se minimum intenzity pohybuje v rozmezí 6 až 11 (obr. 2.12).

Následně přeastřujeme objektiv zpět na výchozí rovinu před LCD panelem. Pak přecházíme k fázi 2. V ní objektivem opatrně přeastřujeme směrem k LCD panelu, dokud nezaznameneáme hodnotu intenzity I_{\min} , která je menší nebo rovna 10. V momentě, kdy je podmínka splněna, se LCD panel nachází na vnějším okraji hloubky ostrosti, což je naším cílem.

Podotkněme, že kontrast K není pro zaostření vhodný, jelikož je závislý na míře překryvu obrazu světlého pixelu a pixelu snímače [kontrasty $K = 0,93$, $K = 0,74$ při ostrém zobrazení na obr. 2.11(a) a (b)].



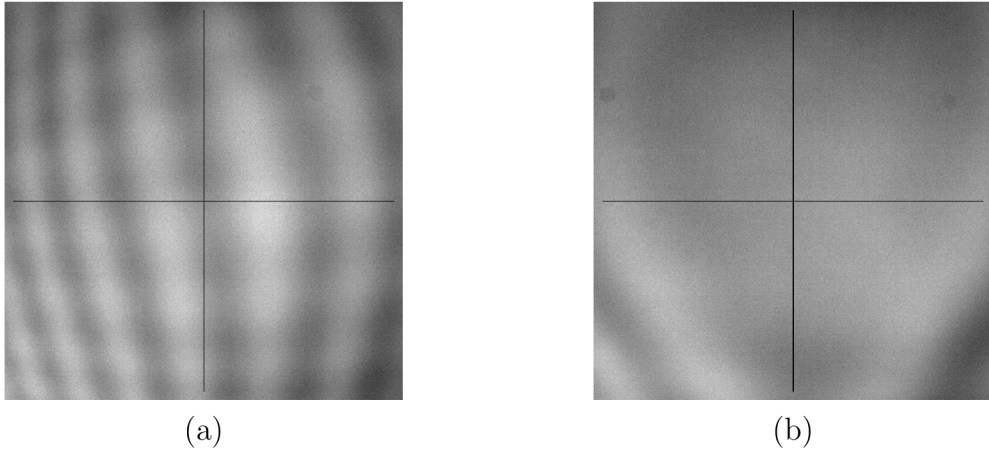
Obrázek 2.12: Záznam nejmenší intenzity I_{\min} (černě) a kontrastu K (šedě) ve výřezu snímku počítaných programem `pythonOstrost` ve fázi 1 a 2.

2.4.2 Minimalizace jevu moaré

Mějme na LCD panelu vykreslenou rovnoměrnou intenzitu. Přesto je jeho obraz pořízený kamerou výrazně ovlivněný jevem moaré. Obsahuje totiž navíc mřížkovitý vzor, který je zakřivený vlivem soudkovitého zobrazení objektivu. Intenzita ve snímku se mění periodicky [obr. 2.13(a)]. Tento vzor chceme odstranit. V ideálním případě požadujeme, aby intenzita byla stejná na celém snímku. Histogram intenzity (spektrum) takového ideálního snímku by měl všechny body rovny nule vyjma jednoho, jehož poloha na horizontální ose by odpovídala intenzitě všech pixelů ve snímku a jeho hodnota na vertikální ose počtu pixelů snímku. Ve skutečnosti vlivem šumu a moaré by byl tento pík nižší a měl nenulový rozptyl. Právě pomocí rozptylu lze popsat vliv moaré avšak za předpokladu, že na snímač kamery dopadá rovnoměrně rozložená intenzita, což splňuje binární obrazec na LCD panelu realizovaný elementární maticí $B = [1]$. V opačném případě je změna rozptylu podstatně hůře čitelná.

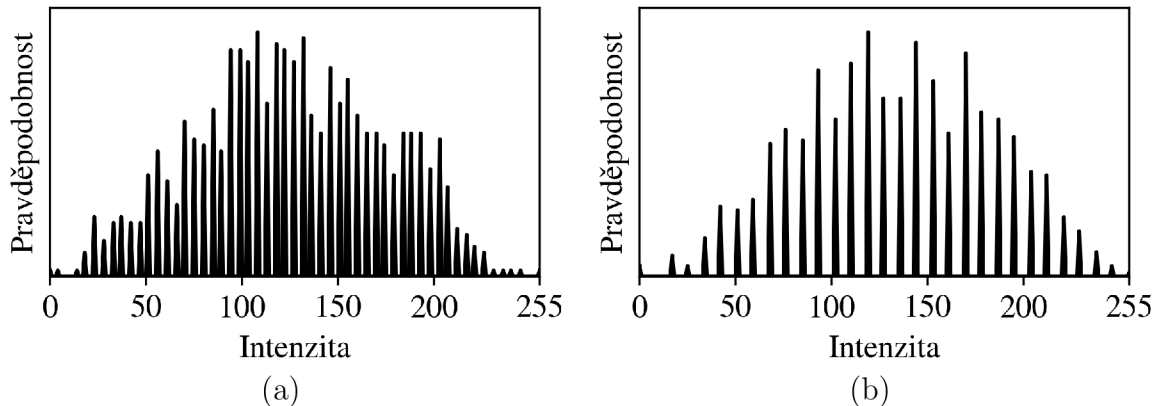
Zmenšováním rozptylu histogramu se ve snímku stává jedna intenzita dominantnější, což odpovídá snímku na obr. 2.13(b). Výpočet rozptylu histogramu celého snímku je však časově náročný a není vhodný pro rychlou analýzu vlivu jevu moaré.

Ukazuje se, že postačuje analyzovat pouze řezy snímku, kdy ideální variantou jsou dvě na sebe kolmé úsečky protínající se ve středu snímku a jejichž délka vymezuje zkoumaný prostor [černé úsečky na obr. 2.13(a), (b)].



Obrázek 2.13: (a) Snímek s výrazným projevem jevu moaré, (b) snímek s optimálním snížením vlivu jevu moaré. Z intenzit vyznačených vertikálních (horizontálních) úseček počítá program `pythonMoare` rozdíly ΔI_{ver} (ΔI_{hor}).

Analýzou histogramů intenzit v řezech při různém vlivu jevu moaré zjišťujeme, že se v nich nevyskytují osamocené píky a histogramy jsou „kompaktní“, jak vidíme na ekvalizovaných histogramech na obr. 2.14(a), (b). První ekvalizovaný histogram je počítán z intenzit horizontální úsečky na obr. 2.13(a) a druhý z horizontální úsečky obr. 2.13(b). Ekvalizaci histogramu provádíme pro snadnější pozorování. Během ma-



Obrázek 2.14: Ekvalizované histogramy intenzity horizontálních úseček na snímcích (a) s výrazným, (b) s minimalizovaným jevem moaré.

nipulace s kamerou se spektrum intenzity v neekvalizovaném histogramu prudce mění a malá změna šířky spektra se obtížně pozoruje. Ekvalizací histogramu s ukotvením nejmenší detekované intenzity I_{min} na 0 a největší intenzity I_{max} na 255 jednoduchým vztahem

$$I_{0-255} = \frac{I - I_{\text{min}}}{I_{\text{max}} - I_{\text{min}}} \cdot 255, \quad (2.30)$$

kde I a I_{0-255} jsou intenzity před ukotvením a po ukotvení, je spektrum stabilizované a změna šířky spektra se projevuje změnou počtu mezer mezi píky.

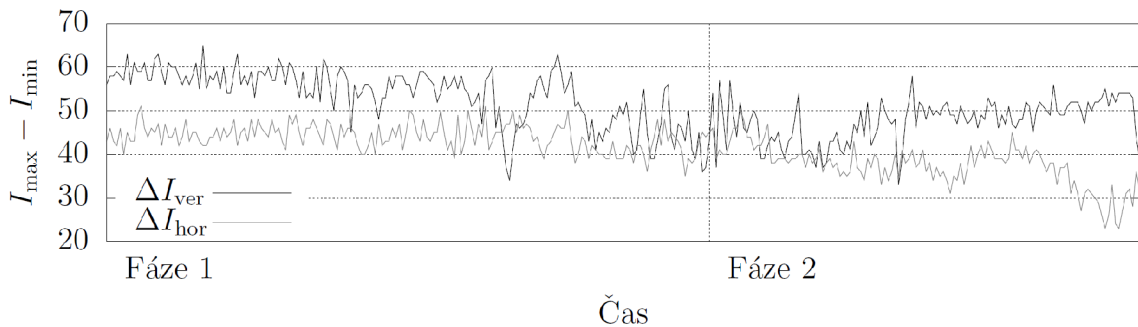
Protože je histogram v čase „kompaktní“, tj. neobjevují se v něm osamocené píky mimo hlavní rozdělení, počítá program `pythonMoare` namísto rozptylu histogramu pouze rozdíly největší a nejmenší intenzity v obou řezech. Rozdíl intenzit v horizontále označme ΔI_{ver} a ve vertikále ΔI_{hor} .

Jak uvádíme výše, program `pythonMoare` vyžaduje binární obrazec s elementární maticí $B = [1]$ na LCD panelu. Program snímá binární obrazec kamerou s expozicí $\tau = 20$ ms, při které se střední hodnota intenzity snímku nachází přibližně v polovině měřícího rozsahu snímače. Výstupem programu je opakované vypisování hodnot ΔI_{ver} a ΔI_{hor} .

Následující postup pro minimalizaci jevu moaré odpovídá konkrétním použitým zařízením a pro jiná se může postup mírně lišit. Celkem má dvě fáze. Ve fázi 1 manipulujeme kamerou šroubem XY v bloku B [obr. 2.4(b)] s cílem vynulovat úhel pootočení mřížky LCD panelu vůči mřížce snímače. Ve fázi 2 používáme šroub Z s jemným posuvem v bloku C [obr. 2.4(b)], kdy minimalizujeme počet proužků vzoru jevu moaré ve snímku.

V našem případě ve snímku na obr. 2.13(a) převládají vertikální proužky (dále jen proužky), proto je postup zaměřený na ně. Je-li úhel mezi oběma mřížkami nulový, pak vertikální úsečkou prochází pouze jeden proužek. Spektrum intenzity úsečky je úzké a rozdíl ΔI_{ver} minimální. Ve fázi 1 proto hledáme minimum ΔI_{ver} .

Ve fázi 2 maximalizujeme šířku proužků, aby jich horizontální úsečkou procházelo nejmenší možné množství. Rozdíl ΔI_{ver} se v této fázi příliš nemění na rozdíl od ΔI_{hor} , který se s rostoucí šířkou proužků zmenšuje. Jeho nejmenší hodnota spolu s nejmenší hodnotou ΔI_{ver} odráží skutečnost, kdy je jev moaré ve vybrané části snímku minimalizovaný [obr. 2.13(b)].



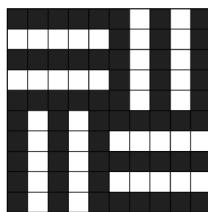
Obrázek 2.15: Záznam průběhu minimalizace moaré pomocí snižování hodnot ΔI_{ver} (černě) a ΔI_{hor} (šedě) počítaných programem `pythonMoare`.

Na obrázku 2.15 jsou vykresleny průběhy ΔI_{ver} a ΔI_{hor} . V principu tedy ve fázi 1 rotačním pohybem kamery kolem její vlastní osy shodné s osou objektivu hledáme minimum ΔI_{ver} , čímž se vůči sobě vyrovnávají mřížky pixelů LCD panelu a CMOS snímače. Ve fázi 2 pohybujeme kamerou v ose z a optimalizujeme zobrazení jedné mřížky na druhou. S ohledem na střed snímku se snažíme, aby se obraz pixelu LCD panelu překrýval s pixelem CMOS snímače kamery.

2.4.3 Zvýšení kontrastu

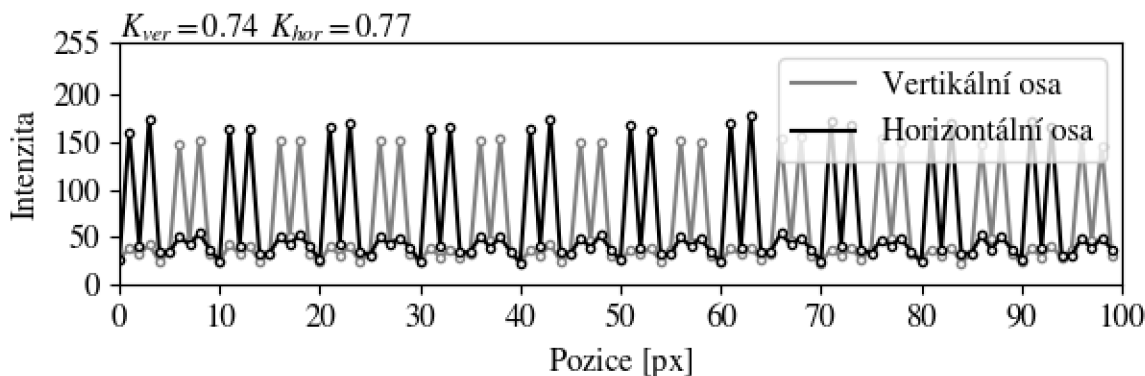
Mřížky snímače kamery a LCD panelu se vůči sobě snažíme srovnat do takové pozice, ve které se obraz mřížky LCD panelu překrývá s mřížkou snímače kamery za účelem zvýšení pravděpodobnosti získat ze snímku užitečnou informaci. Jinak řečeno požadujeme, aby se obraz pixelu LCD panelu překrýval s pixelem snímače kamery. Poznamenejme, že i když tohoto dosahujeme do jisté míry i během minimalizace jevu moaré, program `pythonKontrast` je zaměřen na užší oblast a je na zvýšení kontrastu K této oblasti více optimalizovaný. Cílem je dosáhnout nejvyššího možného kontrastu

intenzity s důrazem na střed snímku. Na LCD panelu je vykreslen binární obrazec s elementární maticí uvedenou na obr. 2.16.



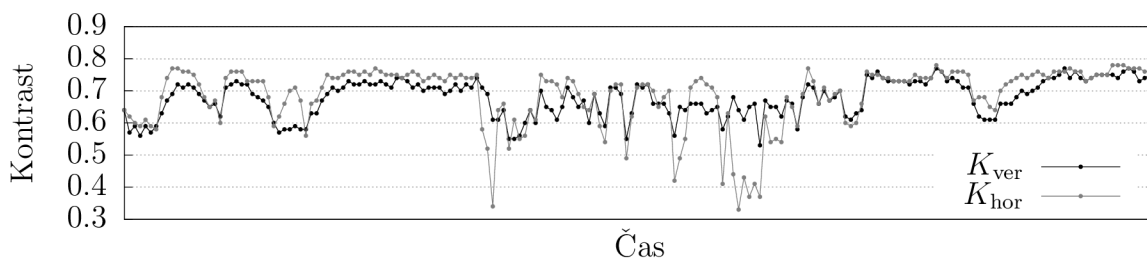
Obrázek 2.16: Elementární matice binárního obrazce programu `pythonKontrast`.

Kvůli vysoké citlivosti, nebo přesněji nestabilitě držáku kamery, je nutné manipulovat s kamerou v obou osách x , y pomocí šroubů X , Y v bloku C opakovaně [obr. 2.4(b)]. Pohyb ve směru osy x ovlivňuje pozici kamery ve směru osy y a naopak. Proto program současně vykresluje průběhy intenzity centrálního kříže s vertikální a horizontální úsečkou o délkách 100 px ve středu snímku a současně počítá kontrasty K_{ver} , K_{hor} obou úseček (obr. 2.17).



Obrázek 2.17: Ukázka programu `pythonKontrast`.

Jak moc je proces citlivý na manipulaci se šrouby s jemným posunem můžeme vyčíst z obr. 2.18, který ukazuje průběh procesu v čase, jehož výsledkem je maximalizace obou kontrastů K_{ver} a K_{hor} .

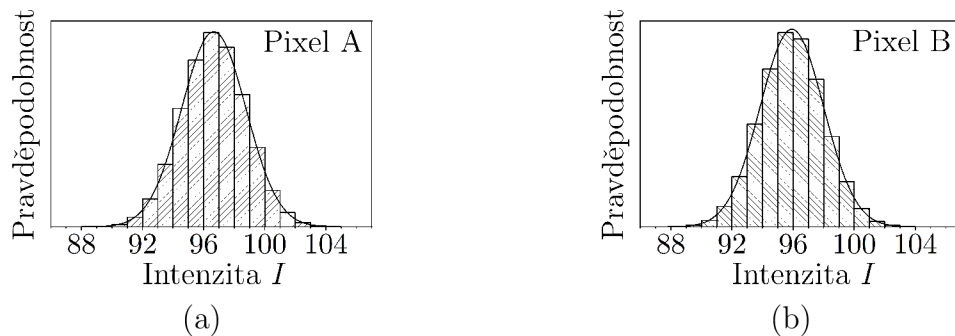


Obrázek 2.18: Záznam průběhu maximalizace kontrastu pomocí zvyšování hodnot veličin K_{ver} (černě) a K_{hor} (šedě) počítaných programem `pythonKontrast`.

Po maximalizaci kontrastu je senzor nastaven a připravený pro měření. Důležité při pořizování snímků je dodržet, aby v zorném poli objektivu nebyli jiné světelné zdroje než svítící pixely LCD panelu. Rovněž musí být zabráněno jakýmkoli možným odleskům vyjma těch od testovaného skla.

2.5 Měření FFC

Po nastavení senzoru přistupujeme k proměření snímků pro metodu FFC, která vstupní snímky metod koriguje o šum s pevným vzorem FPN. Pro zajímavost si ukažme histogramy intenzity dvou pixelů A a B , které vychází ze souboru celkem 10 000 snímků [obr. 2.19(a), (b)]. Jejich hodnoty intenzity spolu s nejistotou jsou $I_A = (96,7 \pm 2,1)$ a $I_B = (95,9 \pm 2,1)$. Provedeme-li nelineární regresey obou histogramů normálním rozdělením, pak nám vychází koeficienty determinace $R_A^2 = 0,999\ 87$, $R_B^2 = 0,999\ 9$, které nám udávají, do jaké míry se regresní křivka shoduje s daty.



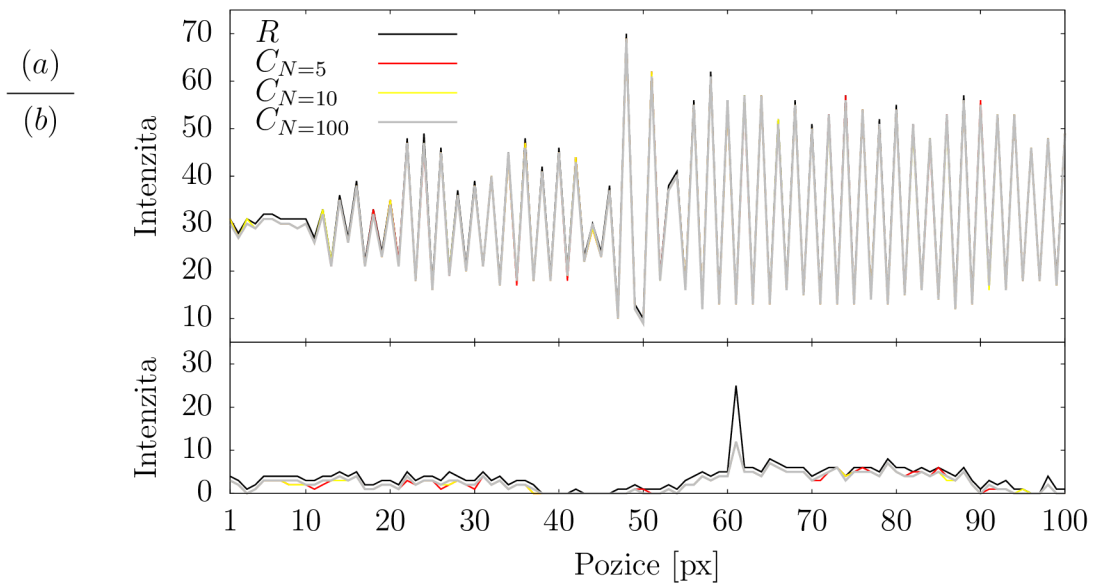
Obrázek 2.19: Histogramy intenzit pixelů (a) A, (b) B, které jsou proloženy regresní křivkou normálního rozdělení.

2.5.1 Určení velikosti souboru N a M pro metodu FFC

Pro měření FFC potřebujeme dvě série snímků D^{raw} , F^{raw} . Minimální velikosti obou souborů, které značíme N a M , zatím neznáme. Pro jednoduchost zvolme velikost souboru M rovnu velikosti souboru N .

Proveďme měření FFC na jednom ze skel, které máme připravené před LCD panelem, na němž je nastaven binární obrazec vertikálních proužků s elementární maticí $B = [1, 0]$. V programu `pythonMattice` nastavme expozici $\tau = 300$ ms a spusťme měření FFC pro $N = 100$. Program nejprve uloží 100 snímků D^{raw} se zakrytým objektivem kamery. Poté následuje doba 20 s, během které před kameru umísťujeme plošný zdroj LED panel a mezi něj a kameru dva optické šedé filtry s optickými hustotami $OD = 2,0$ a $OD = 0,3$. Je pořízeno dalších 100 snímků F^{raw} . Nakonec jsou všechny prvky mezi kamerou a připraveným sklem před LCD panelem odstraněny a je uložen snímek R . Snímek R zpracujeme metodou FFC, přičemž postupně používáme různou velikost souboru $N = 5, 10$ a 100 .

Ukažme si, jak se liší výstupní snímek C pro různá N . Vykresleme si průběhy intenzity dvou řádků ze dvou míst na snímku R a několika snímcích C pro různá N . Na obr. 2.20(a) je řádek z oblasti skla s vadou, která se nachází kolem pozice 50 px, a na obr. 2.20(b) je řádek z oblasti mimo sklo v místě stínu. V oblasti skla se průběhy pro různá N příliš neliší. Mohli bychom proto zvolit minimální hodnotu $N = 5$. V oblasti stínu, kde je dominantní složkou intenzity šum, se průběh intenzity pro $N = 5$ vícekrát odlišuje od průběhu pro $N = 100$ ve srovnání s $N = 10$, kde je shoda dostatečná. Důležité je, že na všech výstupních snímcích C byl vrchol na pozici 61 px ponížen na stejnou hodnotu intenzity pro všechna testovaná N . Protože se výstupní



Obrázek 2.20: Závislost metody FFC na velikosti souboru N . Vykreslen je průběh ze vstupního snímku R a výstupního snímku C pro tři různá n . (a) Oblast skla s binárním obrazcem na pozadí. Nízké změny intenzity na pozici 1 px až 20 px jsou důsledkem jevu moaré, na pozici 45 px až 55 px se projevuje vada s charakterem čočky a na pozici 60 px až 100 px je průběh intenzity binárního obrazce bez výrazného vlivu jevu moaré. (b) Oblast, kam teoreticky nedopadají žádné paprsky z LCD panelu.

snímky C pro $N = 10$ a $N = 100$ liší jen nepatrně, pracujme dále při měření FFC s velikostí souboru $N = M = 10$.

2.5.2 Měření FFC pro všechny předpokládané expozice

Pořídíme sérii snímků metody FFC pro takové expozice kamery, se kterými metody pro hledání dekorativních vad pracují (přehled expozic τ je v tab. 2.8). Před kameru umístíme LED panel pro rovnoměrné osvětlení snímače kamery a stojan pro uchycení optických šedých filtrů pro korekci jasů LED panelu. Jas LED panelu je konstantní a bez použití filtrů je snímek při daných expozicích τ přeexponovaný. Známe celkem čtyři různé expozice τ , se kterými metody pracují.

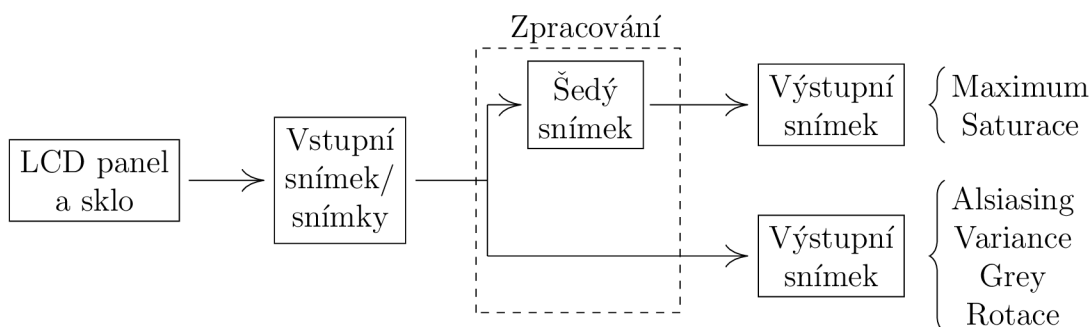
Tabulka 2.8: Doby expozice τ a jim odpovídající optické hustoty OD použitých optických šedých filtrů.

τ (ms)	20	35	200	300
OD	1,0 a 0,2	1,0 a 0,4	2,0 a 0,1	2,0 a 0,3

V programu `pythonMattice` spustíme proces měření FFC postupně pro různé expozice τ a do stojanu umístíme vhodně nakombinované šedé filtry o dané optické hustotě OD, jak uvádí tabulka 2.8. Při pořizování 10 snímků D^{raw} nejprve odstiňme zapnutý LED panel a zakryjme objektiv jeho vlastním ochranným krytem. Poté kryt sejme, před objektiv umístíme šedé filtry a pořídíme 10 snímků F^{raw} . S těmito sériemi snímků jsme schopni korigovat vstupní snímky metod značené v rámci FFC R na korigované vstupní snímky C , které jsou pak metodám postoupeny.

2.6 Popis metod

Celkově je vytvořeno šest metod založených na analýze vstupních obrazových dat. Na začátku každé metody je zobrazen binární obrazec na LCD panelu, před nímž je umístěno sklo. Kamerou je pořízen jeden vstupní snímek (metody Variance, Grey) nebo více vstupních snímků (ostatní metody), které jsou dále postoupeny zpracování jednou ze šesti metod. Během analýzy, některé metody vytváří jako vedlejší produkt šedý snímek (metody Maximum, Saturace), na němž mohou být vady okem lépe pozorovatelné ve srovnání se vstupními snímky. Na konci zpracování je černobílý výstupní snímek N_{BW} , kde N představuje první písmeno názvu metody, který je odvozen od jejího principu. Schématicky je proces zpracování na obr. 2.21. Ve výstupním snímku černé pixely představují lokalizovanou vadu na bílém pozadí.



Obrázek 2.21: Schématické znázornění zpracování vstupních dat. Na začátku procesu je binární obrazec na LCD panelu, před nímž je umístěno sklo. Kamerou je pořízen vstupní snímek, který je dále postoupen zpracování jednou ze šesti metod. Během analýzy mohou některé metody jako vedlejší produkt vytvářet šedý snímek, na němž mohou být vady okem lépe pozorovány ve srovnání se vstupním snímekem. Na konci procesu je výstupní snímek, který je černobílý.

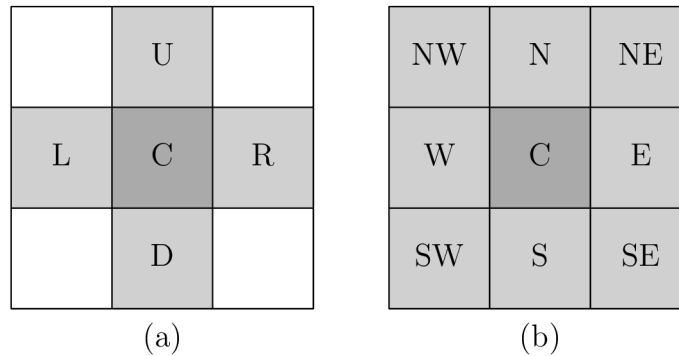
Metody potřebují kromě vstupních snímků také určit vlastní parametry. Zjišťování většiny parametrů je automatizované (D_{Max} , D_{Ali} , D_{Var} , D_{Gre} , L_{Sat} , L_{Rot} a U_{Rot}), nicméně každý parametr je úzce svázán s citlivostí metody.

Vytvořené metody jsou součástí přílohy A Metody.

2.6.1 Okolí pixelu

Vstupní snímek obsahuje pixely o souřadnicích $[x, y]$ a intenzitě I . Na pixelu vstupního snímku se v rámci metody provádí operace buď na blízkém, nebo rozšířeném okolí. V tom případě označujeme tento pixel jako centrální. Blízké okolí centrálního pixelu C je označeno písmeny L (levý pixel), R (pravý pixel), U (horní pixel) a D (dolní pixel) [obr. 2.22(a)]. Rozšířené okolí je označené jako NW (vlevo nahoře), N (nahore), NE (vpravo nahoře), E (vpravo), SE (vpravo dole), S (dole), SW (vlevo dole) a W (vlevo) [obr. 2.22(b)].

Pokud hovoříme o všech centrálních pixelech snímku S , máme na mysli všechny pixely snímku vyjma okrajových. Pro snímek S o celkovém počtu pixelů $w \cdot h$, kde w je šířka snímku a h je jeho výška v jednotkách px, existuje právě $(w - 2) \cdot (h - 2)$ centrálních pixelů.



Obrázek 2.22: (a) Blízké okolí centrálního pixelu C označené písmeny L, R, U a D. (b) Rozšířené okolí centrálního pixelu C označené písmeny NW, N, NE, E, SE, S, SW a W.

2.6.2 Citlivost metody

Rozlišujeme rozměrovou citlivost a intenzitní citlivost metod pro detekci dekorativních vad. Pod pojmem rozměrová citlivost míníme, jak název napovídá, rozlišovací schopnost metody vzhledem k minimální velikosti detekovatelné vady. Velikost vady měříme v jednotkách pixelů snímače kamery, což lze snadno převést na skutečnou velikost vztahem

$$\text{Skutečná velikost } [\mu\text{m}] = \text{Velikost vady } [\text{px}] \cdot 269 [\mu\text{m} \cdot \text{px}^{-1}].$$

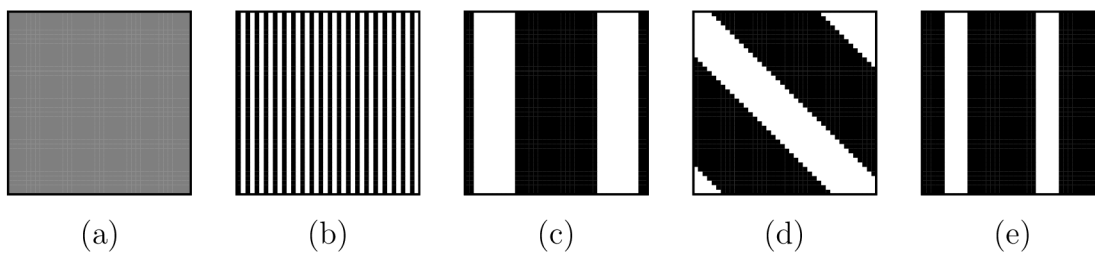
Například, pokud jedna metoda je schopna detekovat vadu o minimální velikosti 1 px a druhá metoda o velikosti 5 px, potom první metoda má 5krát lepší rozměrovou citlivost než druhá.

Druhý typ citlivosti – intenzitní citlivost – naproti tomu charakterizuje schopnost metody nalézt vadu na základě odlišnosti např. intenzit dvou sousedních pixelů. Detekuje-li metoda s jedním nastavením parametrů vadu o minimálním rozdílu 10 intenzit sousedních pixelů a s jiným nastavením detekuje vadu o minimálním rozdílu 50, potom má metoda s prvním nastavením parametrů 5krát lepší intenzitní citlivost než s druhým nastavením.

Uvědomme si, že v oblasti se silným projevem jevu moaré či s výraznými fluktuacemi tloušťky vrstvy laku na skle je intenzitní citlivost nízká v porovnání s oblastí, kde je zobrazení pixelu LCD panelu na pixel snímače přesné.

2.6.3 Binární obrazce metod

Všechny metody pracují s vlastními binárními obrazci. Některé z těchto binárních obrazců jsou sdílené více metodami. Metody Variance a Grey požadují mít na LCD panelu svítící všechny pixely. Na obrázku 2.23(a) je výřez s šedou barvou (správně by měl být bílý). Metody Maximum a Aliasing využívají dvou snímků s tenkými vertikálními proužky posunutými horizontálně o půl periody [obr. 2.23(b)]. Metoda rotace pracuje s rotujícími širokými proužky [obr. 2.23(c), (d)] a metoda Saturace s vertikálními nerotujícími ale posunovanými širokými proužky [obr. 2.23(e)].



Obrázek 2.23: Výřezy binárních obrazců metod (a) Variance a Grey, (b) Maximum a Aliasing, (c), (d) Rotace a (e) Saturace.

2.6.4 Metody Maximum a Aliasing

Obě metody Maximum a Aliasing sdílí na svém vstupu dva snímky. Binárním obrazcem prvního snímku jsou proužky o šířce 1 px a periodě 2 px. Na výřezu snímku pořízeném při maximalizaci kontrastu (obr. 2.24) vidíme, že rozdíl mezi intenzitami světlých a tmavých proužků je větší v případě vertikálních než horizontálních. Proto volíme binární obrazec s vertikálními proužky. Příčinou je struktura matice pixelů (obr. 2.3). Obrazec je realizován elementární maticí $B = [1, 0]$. Vzor druhého snímku obsahuje stejnou vertikální mřížku posunutou o polovinu periody a realizovanou elementární maticí $B = [0, 1]$.



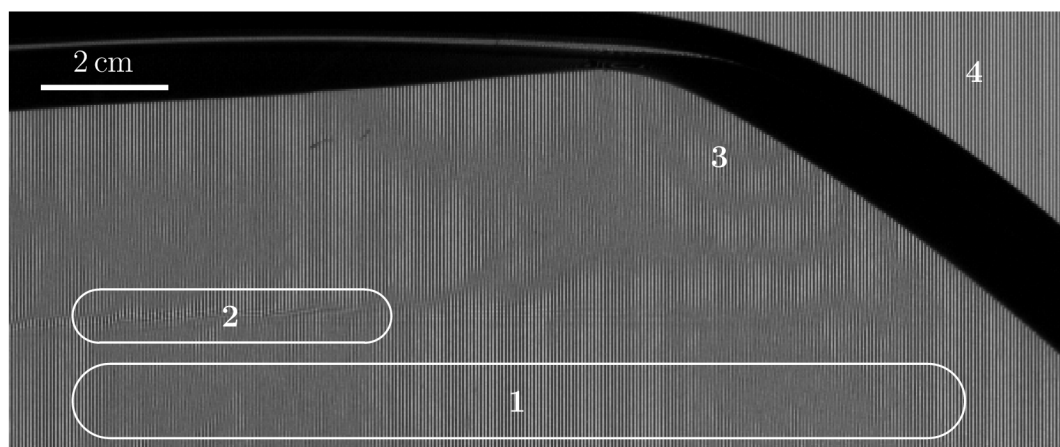
Obrázek 2.24: Výřez snímku binárního obrazce pro vysvětlení volby vertikálních proužků namísto horizontálních.

Expozici kamery $\tau = 35$ ms volíme tak, abychom využili téměř celý intenzitní rozsah CMOS snímače. Intenzita světlých proužků by se měla blížit hodnotě 255 a intenzita tmavých proužků by měla být rovna přibližně rozdílu intenzity světlých proužků od 255.

V důsledku jevu moaré pozorujeme na snímku s binárním obrazcem vertikálních proužků nejen jejich vlastní vysokou frekvenci $\frac{1}{2} \text{ px}^{-1}$, ale také rušivou nižší frekvenci. Vliv této rušivé frekvence je podstatný. Jde totiž o prolnutí světlých a tmavých proužků binárního obrazce a snížení jejich kontrastu vlivem jevu moaré. Různé projevy jsou vidět na snímku skla (obr. 2.25). Ohraničená oblast pod číslem 1 vymezuje řez vertikálními proužky rušivé nízké frekvence (zleva doprava tmavý, světlý a opět tmavý pruh). V oblasti 2 je viditelný vliv šířky. Číslo 3 představuje okraj skla s výraznou křivostí plochy, kde pozorujeme oblouky vzoru jevu moaré. A konečně pod číslem 4 nacházející se mimo sklo nepozorujeme žádnou zjevnou deformaci a kontrast proužků frekvence $\frac{1}{2} \text{ px}^{-1}$ je zde vysoký. Uvědomme si však, že pokud bychom viděli větší část snímku bez skla, stále bychom pozorovali vliv jevu moaré. Podíváme-li se na proužky nebo oblouky rušivé nízké frekvence, všimáme si jejich mírné deformace. Ta je příčinou různé tloušťky laku na vnějším povrchu skla.

Binární obrazec vstupních snímků metod Maximum a Aliasing je tvořen proužky s periodou 2 px. Popsané jevy při této periodě snižují intenzitní citlivost. Pokud bychom periodu zvětšili na dvojnásobek, potom by došlo ke snížení prolínání světlých a tmavých proužků v oblasti 1 na obr. 2.25. Nicméně, pokud bychom chtěli odstranit prolínání

i v oblastech 2 a 3, museli bychom zvětšit periodu proužků na trojnásobek a více. Tím bychom však snížili rozměrovou citlivost. Protože je prioritou rozměrová citlivost a schopnost detekovat vady o velikosti minimálně 1 px, volíme periodu proužků nejmenší možnou.

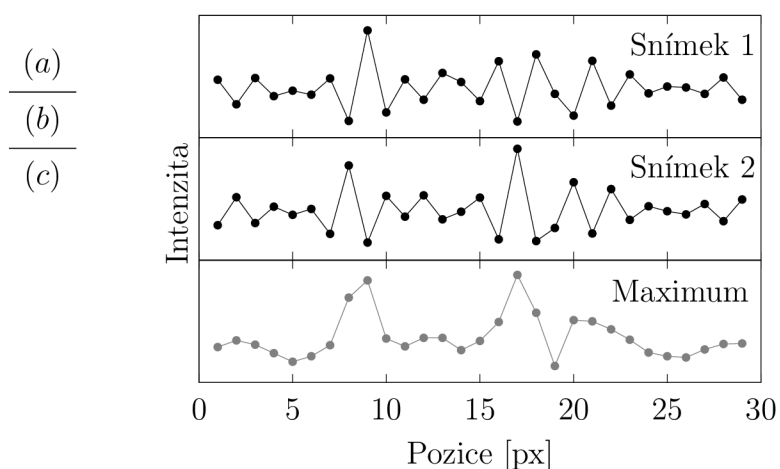


Obrázek 2.25: Vliv jevu moaré na snímek s binárním obrazem vertikálních proužků s periodou 2 px.

Metoda Maximum

Princip

Povrch skla je pokryt tenkou vrstvou laku, jejíž tloušťka je proměnná. Toto zvlnění povrchu si představme jako vrstvu mikročoček, které náhodně mění úhel lomu paprsků proužků pocházejících ze zdroje tvořícího svislé proužky. Na řádku snímače kamery pozorujeme pilovitý průběh hodnoty intenzity odpovídající střídání světlých a tmavých proužků. Vliv laku je na průběh snímačem detekované intenzity nepatrný. Mějme ale vadu, která se chová jako čočka a má ve srovnání s mikročočkami menší ohniskovou vzdálenost. Taková vada mění výrazněji úhel lomu než fluktuace tloušťky laku. Stejně se projevuje i ve změně intenzity, jak vidíme na pozicích kolem 9 px a 18 px na obr. 2.26(a).



Obrázek 2.26: Názorné vysvětlení principu metody Maximum. (a) Data ze vstupního snímku 1, (b) data ze vstupního snímku 2 a (c) zpracování dat metodou Maximum.

Pokud binární obrazec promítaný na LCD panelu posuneme o půl periody v horizontálním směru, průběh intenzity detekované snímačem se vůči horizontální ose zrcadlově prohodí [obr. 2.26(b)]. Metoda z těchto dvou profilů intenzit (před a po posunutí binárního obrazce) vytvoří nový profil [obr. 2.26(c)] vybráním větší hodnoty intenzity ze stejné pozice obou průběhů. Pokud by na skle nebyly žádné čokující objekty včetně mikročoček, byl by výsledný profil lineární. Jelikož to není náš případ, vidíme na průběhu intenzity lokální minima a maxima odpovídající konkávním a konvexním čokám v případě vady a mikročočkám v případě fluktuací tloušťky laku.

Pokud by metoda vybírala menší hodnoty ze dvou intenzit ze stejné pozice, výsledný profil by byl komplementární s profilem z vybraných větších hodnot.

Postup

Vstupní snímky S_1, S_2 mají šířku w a výšku h . Vzniká nový šedý snímek M o stejných rozměrech. Intenzita pixelu na souřadnicích $[x, y]$ je

$$M[x,y] = \begin{cases} S_1[x, y], & \text{pokud } S_1[x, y] \geq S_2[x, y], \\ S_2[x, y], & \text{pokud } S_1[x, y] < S_2[x, y], \end{cases} \quad (2.31)$$

kde $x, y \in \mathbb{N}$, $x \in \langle 0, w \rangle$ a $y \in \langle 0, h \rangle$. Snímek M obsahuje vždy vyšší intenzitu ze snímků S_1, S_2 na pozici $[x, y]$.

Označme absolutní hodnotu rozdílu intenzit dvou pixelů P a Q na snímku M jako $I^{PQ} = |I^P - I^Q|$ a zavedme parametr D_{Max} ¹⁹, abychom mohli odlišit změnu intenzity způsobenou vrstvou laku na skle a vadou. Dále nechť výrok v^{PQ} je roven pravdivostní hodnotě výrazu $I^{PQ} > D_{\text{Max}}$. Potom intenzita každého centrálního pixelu C výstupního snímku M_{BW} je dána předpisem

$$M_{\text{BW}} = \begin{cases} 0, & \text{pokud } v^{\text{CL}} \vee v^{\text{CU}} \vee v^{\text{CR}} \vee v^{\text{CD}}, \\ 255, & \text{jinak.} \end{cases} \quad (2.32)$$

Metoda Aliasing

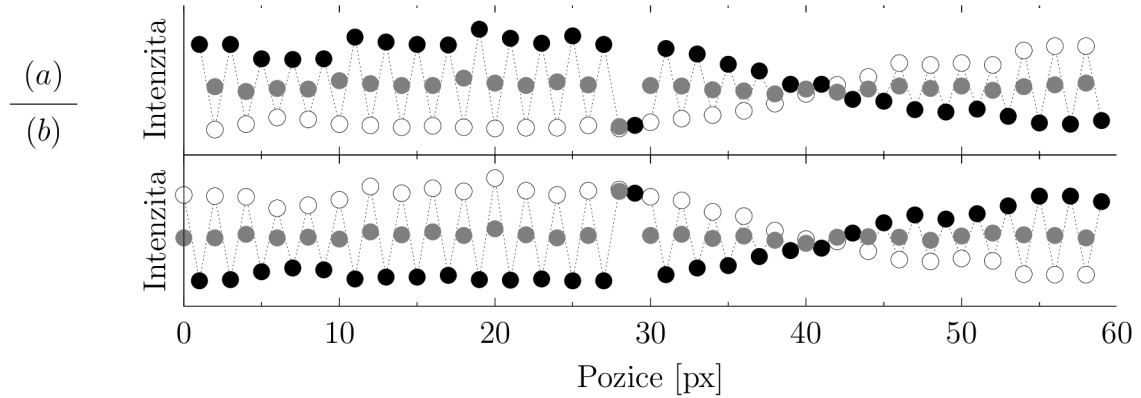
Princip

Metoda využívá nesplnění podmínky Shannonova teorému. Vysvětleme si tento princip na obr. 2.27, na němž jsou zobrazeny dva průběhy intenzity vstupních snímků 1 a 2 ze stejného řádku.

Z celého řádku vstupního snímku 1 [obr. 2.27(a)] si metoda vybírá pouze liché pixely (černé tečky). Díky podvzorkování se intenzita v tomto souboru lichých pixelů mění pozvolna. Na pozici 29 px se lichý pixel nachází v místě vady, která se projevuje skokovou změnou v intenzitě lichých pixelů. Protože změny intenzity nejsou vždy pozvolné v oblasti bez vad, jak vidíme například mezi 9. a 11. pixelem, není tento způsob detekce dostatečně robustní vůči šumu signálu. Šumem signálu je v tomto případě jakákoli změna intenzity mezi lichými nebo sudými pixely, kterou nepovažujeme za vadu. Jedná se například o změny způsobené jevem moaré nebo fluktuacemi tloušťky laku na skle. Proto na témže řádku snímku 1 metoda vybírá také všechny sudé pixely (bílé tečky). Poté se z intenzit párových (sousedních) pixelů obou souborů počítá střední hodnota

¹⁹Hodnota parametru D_{Max} metody Maximum je určena v sekci Určení parametrů metod.

intenzity (šedé tečky), což je odolnější veličina vůči šumu signálu a spolehlivěji detekuje vady díky menším změnám sousedních středních hodnot v oblasti bez vad. Pomineme-li tyto menší změny, pak je střední hodnota v oblastech bez vad téměř konstantní a to i v místě silného projevu jevu moaré kolem pozice 40 px na obr. 2.27 nebo v oblasti šíbrů.



Obrázek 2.27: Názorné vysvětlení principu metody Aliasing. (a) Data ze vstupního snímku 1 a (b) vstupního snímku 2. Bílé a černé tečky představují sudé a liché pixely. Šedé tečky představují zpracování těchto pixelů metodou Aliasing.

Na snímku 2 s posunutou vertikální mřížkou o půl periody se vada projevuje na stejném místě [obr. 2.27(b)]. Metoda na něm provádí stejnou analýzu jako na snímku 1.

Postup

Vstupní snímky S_1 , S_2 mají šířku w a výšku h . Vznikají nové snímky A_1 , A_2 o šířce $\frac{w}{2}$ a výšce h a jejich intenzity pixelu na souřadnicích $[x, y]$ jsou

$$\begin{aligned} A_1[x, y] &= \frac{1}{2} (S_1[2x, y] + S_1[2x + 1, y]), \\ A_2[x, y] &= \frac{1}{2} (S_2[2x, y] + S_2[2x + 1, y]), \end{aligned} \quad (2.33)$$

kde $x, y \in \mathbb{N}$, $x \in \langle 0, \frac{w}{2} \rangle$ a $y \in \langle 0, h \rangle$. Intenzity snímku A_1 (A_2) jsou středními hodnotami dvou sousedních intenzit snímku S_1 (S_2) v horizontálním směru.

Označme absolutní hodnotu rozdílu intenzit dvou pixelů P a Q na snímku A_n , kde $n \in \{1, 2\}$, jako $I_n^{PQ} = |I_n^P - I_n^Q|$ a zaveďme parametr D_{Ali} ²⁰. S těmito veličinami můžeme rozeznat změny intenzity způsobené vrstvou laku na skle a vadou. Dále nechť výrok v_n^{PQ} je roven pravdivostní hodnotě výrazu $I_n^{PQ} > D_{\text{Ali}}$. Potom intenzita každého centrálního pixelu C výstupního snímku A_{BW} je dána předpisem

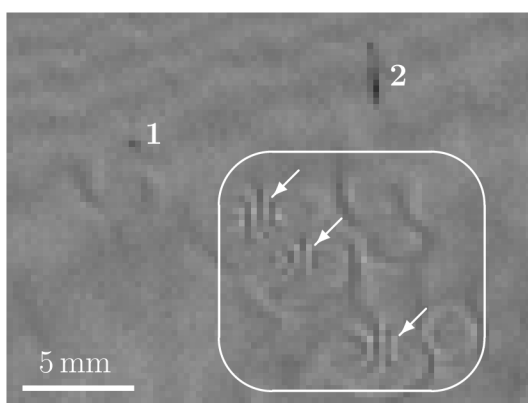
$$A_{\text{BW}} = \begin{cases} 0, & \text{pokud } (v_1^{\text{CL}} \vee v_1^{\text{CU}} \vee v_1^{\text{CR}} \vee v_1^{\text{CD}}) \wedge (v_2^{\text{CL}} \vee v_2^{\text{CU}} \vee v_2^{\text{CR}} \vee v_2^{\text{CD}}), \\ 255, & \text{jinak.} \end{cases} \quad (2.34)$$

²⁰Hodnota parametru D_{Ali} metody Aliasing je určena v sekci Určení parametrů metod.

2.6.5 Metody Variance a Grey

Metody Variance a Grey mají na vstupu shodný snímek. Binární obrazec snímku je realizován elementární maticí $B = [1]$, která na LCD panelu tvoří jednobarevnou plochu. Expozice kamery τ je nastavena na 20 ms, aby střední hodnota intenzity vzoru snímaného kamerou byla přibližně v polovině rozsahu snímače.

Před umístěním posuzovaného skla před LCD panel pozorujeme kamerou šedou plochu pouze s proužky jevu moaré. Umístěním skla před LCD panel ihned pozorujeme vliv vrstvy laku a vad, které se chovají jako čočky (obr. 2.28). U některých vad označených šipkami si všimneme, jak je šedá plocha v místě vady transformována na pomyslnou šachovnici, která odráží strukturu LCD panelu vlivem jevu moaré²¹. Rovněž si povšimneme prachu označeného čísly 1 a 2, který paprsky mířící od LCD panelu rozptyluje mimo snímač kamery (obr. 2.28).



Obrázek 2.28: Výřez vstupního snímku pro metody Variance a Grey. Šipky v rámečku ukazují na tři vady typu voda na skle, které se chovají jako čočky. Čísla 1 a 2 označují místa, kde se nachází prach na skle.

Metoda Variance

Princip

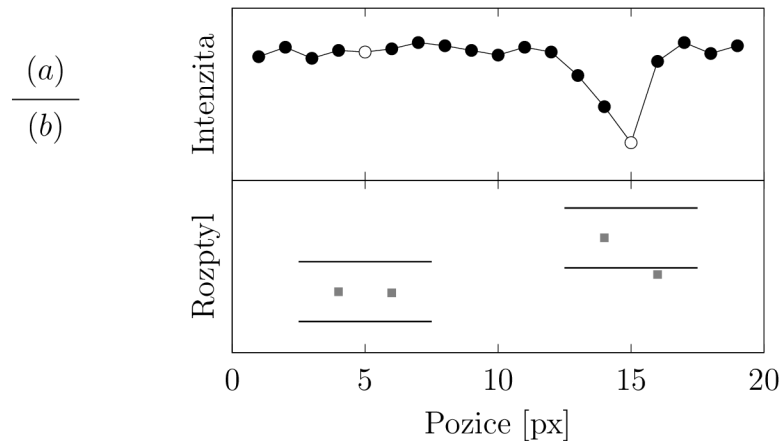
Předpokládáme, že v oblasti bez vad se intenzita mezi sousedními pixely mění plynule. V místě vady však dochází k narušení tohoto trendu a intenzita se v krajním pixelu vady výrazně mění. Z pohledu tohoto pixelu má soubor pixelů na jedné jeho straně odlišnou hodnotu rozptylu než soubor na straně druhé. Na obrázku 2.29 máme na mysli pixely na pozicích 12 px a 17 px. K takovýmto případům dochází poměrně často a nejedná se přitom nutně o vady. Avšak v těchto případech, kdy nejde o vady, převažuje stav, že se rozptyly liší buď ve vertikálním, nebo v horizontálním směru ve snímku. Z toho plyne podmínka pro přítomnost vady, aby se pro pixel, u něhož se tedy liší rozptyly v jednom směru, lišily i ve směru kolmém. Tuto podmínku pixely 12 px a 17 px nesplňují, ale pixel na pozici 15 px ano.

Jak moc se musí rozptyly na obou stranách lišit určuje konstanta D . Ukažme si způsob vyhodnocení. Na obrázku 2.29(a) jsou zvýrazněné dva pixely bílé, které označme jako centrální pixely. Pro oba určíme jejich sousední levé a pravé rozptyly, které jsou

²¹Tak usuzujeme převážně ze světlých vertikálních proužků, které které přisuzujeme k zeleným subpixelům LCD panelu.

reprezentovány šedými čtverečky [obr. 2.29(b)]. Nejprve k levému rozptylu centrálního pixelu na pozici 5 px přičteme a odečteme parametr D . Těmito získanými výslednými hodnotami znázorněnými dvěma úsečkami definujeme interval rozptylu. Pokud se uvnitř něj nachází i pravý rozptyl, pak nás tento centrální pixel dále nezajímá. Totéž provedeme i pro druhý zvýrazněný centrální pixel na pozici 15 px. Uvnitř vzniklého intervalu se pravý rozptyl zřejmě nenachází. Ke stejnému závěru docházíme i v kolmém směru, což je nutná podmínka, abychom tento pixel identifikovali jako místo vady.

Výhodou rozptylu je potlačení vlivu šumu v místech silnější a více zvlněné vrstvy laku na skle, kde dochází k nežádoucímu vzniku mřížky.



Obrázek 2.29: Názorné vysvětlení principu metody Variance. (a) Data ze vstupního snímku se dvěma bíle označenými vyšetřovanými pixely, (b) vypočítané hodnoty levostranných a pravostranných rozptylů pro dva vyšetřované pixely. K oběma levým rozptylům je přičtena a odečtena konstanta D , což je znázorněno úsečkami.

					U_2		
					U_1		
	L_n	\dots	L_2	L_1	C	R_1	R_2
					D_1		
					D_2		

Obrázek 2.30: Upravené označení pixelů v okolí centrálního pixelu C . Původní blízké okolí je rozšířeno o dalších $4(n - 1)$ pixelů. Pixely nejsou vyznačeny všechny, celá je naznačena pouze řada L .

V případě metody Variance je počet centrálních pixelů snímku S o šířce w a výšce h celkem $(w - 2n) \cdot (h - 2n)$, kde n je velikost souboru pixelů pro výpočet rozptylu.

Postup

Pro každý centrální pixel vstupního snímku S definujeme čtyři množiny intenzit

$$\begin{aligned}M_L &= \{I_i^L; i = 1, 2, \dots, n\}, \\M_R &= \{I_i^R; i = 1, 2, \dots, n\}, \\M_U &= \{I_i^U; i = 1, 2, \dots, n\}, \\M_D &= \{I_i^D; i = 1, 2, \dots, n\},\end{aligned}\tag{2.35}$$

kde n je počet prvků množiny. Množiny intenzit jsou naznačeny na obr. 2.30.

Střední hodnota intenzity prvků množiny M_L je

$$\bar{I}^L = \frac{1}{n} \sum_{i=1}^n I_i^L.\tag{2.36}$$

Zavedme veličinu pseudorozptyl $\tilde{s}_L^2 = (n-1)s_L^2$, kde s_L^2 je výběrový rozptyl prvků množiny M_L . Zřejmě

$$\tilde{s}_L^2 = \sum_{j=1}^n (I_j^L - \bar{I}^L)^2.\tag{2.37}$$

Jedná se o zjednodušení výpočtu s cílem snížit výpočetní čas. Se stejně vypočítanými pseudorozptyly prvků množin M_R , M_U a M_D zavedme parametr D_{Var} ²². Potom intenzita každého centrálního pixelu C výstupního snímku V_{BW} je dána předpisem

$$V_{\text{BW}} = \begin{cases} 0, & \text{pokud } |\tilde{s}_L^2 - \tilde{s}_R^2| > D_{\text{Var}} \wedge |\tilde{s}_U^2 - \tilde{s}_D^2| > D_{\text{Var}}, \\ 255, & \text{jinak.} \end{cases}\tag{2.38}$$

Otázkou je velikost n souboru pixelů pro výpočet pseudorozptylu. Pokud implicitně nastavíme jistou hodnotu n a spustíme určení parametru D_{Var} metody Variance, potom dostáváme konkrétní hodnotu D_{Var} . Když tento postup opakujeme pro $n = 1, 2, \dots, 10$, můžeme vykreslit závislost parametru D_{Var} na počtu pixelů n (obr. 2.31). Pozorujeme lineární závislost, která vychází z definice pseudorozptylu. Nicméně, ve výstupním snímku se změna počtu pixelů n projevuje pouze výraznějším označením vady, tj. je-li detekována vada, pak např. pro $n = 2$ je označena jedním černým pixelem a pro $n = 10$ šesti černými pixely. Z hlediska výpočetní náročnosti metody volíme velikost souboru $n = 2$, protože lokalizace vady jedním pixelem je dostatečná.

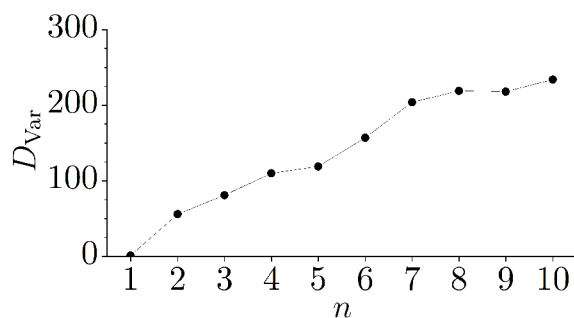
Metoda Grey

Princip

Vady absorbující případně rozptylující paprsky z LCD panelu mimo snímač kamery se na vstupním snímku projevují poklesem intenzity. Většinou se jedná o drobný prach a plocha s poklesem intenzity je v řádu jednotek px^2 .

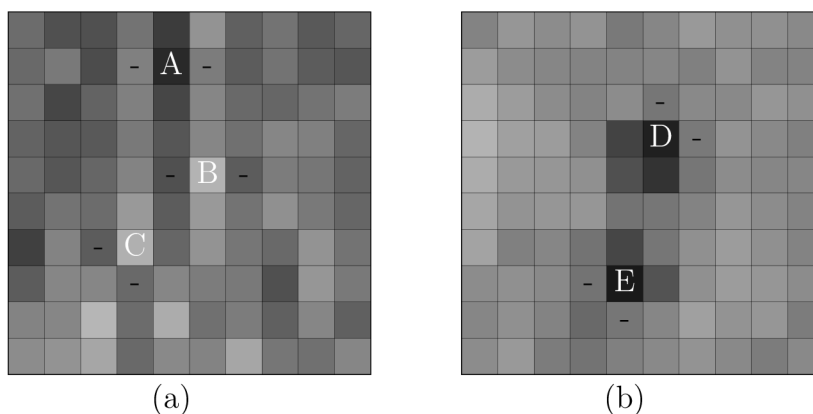
Tloušťka tenké vrstvy laku na skle je proměnlivá. V některých oblastech jsou její fluktuační natolik výrazné, že se mohou na vstupním snímku jevit jako vada. Na obrázku 2.32(a) je ukázka takové oblasti v porovnání s oblastí bez výrazných fluktuačních

²²Hodnota parametru D_{Var} metody Variance je určena v sekci Určení parametrů metod.



Obrázek 2.31: Závislost parametru D_{Var} metody Variance na velikosti souboru sousedních pixelů n pro výpočet pseudorozptylu.

tloušťky laku avšak obsahující vady [obr. 2.32(b)]. Na obou výřezech je dohromady zvýrazněno pět centrálních pixelů velkými písmeny, z nichž A, B, a C jsou v oblasti bez vad a D, E jsou součástí dvou vad.



Obrázek 2.32: Dva výřezy vstupního snímku metody Grey s pěti označenými centrálními pixely A, B, C, D, E a jejich příslušné dvojice pixelů z blízkého okolí označené „-“: (a) výřez snímku z oblasti bez vad, ale s výraznými fluktuacemi tloušťky laku na skle, (b) výřez se dvěma vadami.

Analyzujeme těchto pět pixelů s cílem zavést takovou podmínku, kterou by splňovaly pouze pixely D, E. Nejprve zavedme parametr D . Ten vybírá z obou výřezů na obr. 2.32 ty pixely, jejichž velikost změny intenzity oproti libovolnému pixelu P z jejich blízkého okolí je větší jak D . Tomu odpovídá podmínka

$$|I_L - I_C| > D \vee |I_R - I_C| > D \vee |I_U - I_C| > D \vee |I_D - I_C| > D, \quad (2.39)$$

které vyhovuje více pixelů včetně A, B, C, D a E, protože postačuje splnění pouze pro jeden pixel P z blízkého okolí. Požadujeme proto alespoň dva takové pixely P a Q, pro které platí

$$|I_P - I_C| > D \wedge |I_Q - I_C| > D. \quad (2.40)$$

Tomu vyhovují označené pixely A, B, C, D a E. Výrazné fluktuace tloušťky laku na skle se projevují podélnými vertikálními strukturami, jejichž součástí jsou i pixely A a B. Upravme proto podmínku tak, aby pixely P a Q neležely na jedné horizontální, nebo vertikální ose, tj. aby byly přilehlé.

$$(|I_L - I_C| > D \vee |I_R - I_C| > D) \wedge (|I_U - I_C| > D \vee |I_D - I_C| > D). \quad (2.41)$$

Podmínka je splněna pro pixely C, D a E. Omezme schopnost detekce pouze na ty vady, které se projevují poklesem intenzity. Odstraňme absolutní hodnoty a zvolme za menšenec pixel P blízkého okolí a za menšitele centrální pixel C. Takovou podmínku

$$(I_L - I_C > D \vee I_R - I_C > D) \wedge (I_U - I_C > D \vee I_D - I_C > D) \quad (2.42)$$

nyň splňují pouze pixely D a E.

Postup

Označme rozdíl intenzit dvou pixelů P a Q na vstupním snímku S jako $I^{PQ} = I^P - I^Q$. Zdůrazněme, že se nejedná o absolutní hodnotu a záleží na pořadí. Dále zavedme parametr D_{Gre} ²³. Nechť výrok v^{PQ} je roven pravdivostní hodnotě výrazu $I^{PQ} > D_{\text{Gre}}$. Potom intenzita každého centrálního pixelu C výstupního snímku G_{BW} je dána předpisem

$$G_{\text{BW}} = \begin{cases} 0, & \text{pokud } (v^{\text{LC}} \vee v^{\text{RC}}) \wedge (v^{\text{UC}} \vee v^{\text{DC}}), \\ 255, & \text{jinak.} \end{cases} \quad (2.43)$$

2.6.6 Metoda Saturace

Pro zvýraznění vad, které rozptylují paprsky jdoucí od LCD panelu natolik slabě, že zanikají v okolním šumu, musíme prodloužit expoziční dobu τ . Zároveň by expozice neměla být natolik dlouhá, aby se tyto vady nebo hůře samotný šum neocitly v saturaci. Tomu odpovídá expozice dlouhá přibližně 300 ms.

Metoda má na vstupu celkem osm snímků s binárním obrazcem vertikálních světlých proužků o šířce 10 px a periodě 40 px. Mřížka je postupně posunována v horizontálním směru o osminu periody rovnající se 5 pixelům.

Princip

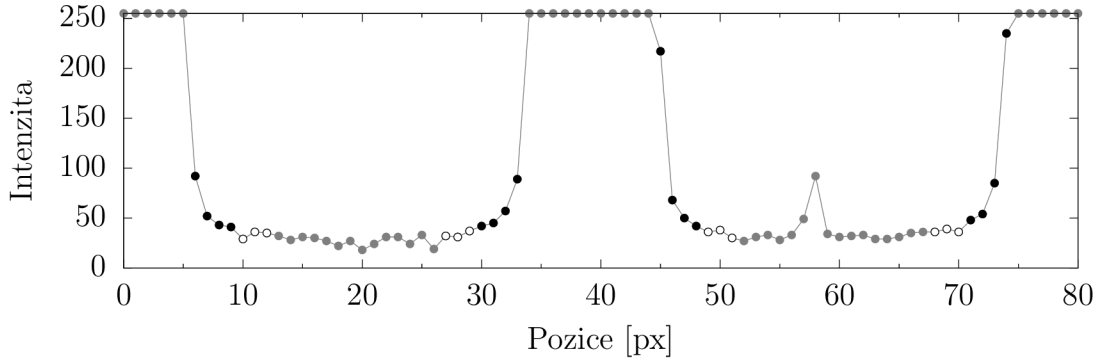
Vertikální světlé proužky slouží jako zdroje svazků paprsků, které vady rozptylují. Pokud je vada v tmavé oblasti, můžeme ji pozorovat kamerou v kontrastu s okolím, které paprsky nerozptyluje a jeví se nám jako tmavé. Viditelnost vady je nepřímo úměrná její vzdálenosti od svazku paprsků světlého proužku. S klesající vzdáleností vady od svazku paprsků světlého proužku z LCD panelu roste také množství světla, které může vada rozptýlit.

Metoda pracuje s osmi vstupními snímky, z nichž vytváří kompozici. Nejprve z každého snímku přepisuje na nulu hodnoty intenzity saturovaných pixelů a stejně tak hodnoty intenzit jejich n sousedních pixelů v horizontálním směru. Z takto upravených snímků metoda skládá kompozitní šedý snímek, kde každý pixel se souřadnicemi $[x, y]$ má hodnotu intenzity rovnu nejvyšší hodnotě intenzity z osmi upravených snímků se stejnými souřadnicemi.

Sousední pixely o počtu n se přepisují rovněž, protože oblasti tmavých proužků se na svých okrajích prolínají se světlými proužky. Názorně to ukazuje obr. 2.33. Černou

²³Hodnota parametru D_{Gre} metody Grey je určena v sekci Určení parametrů metod.

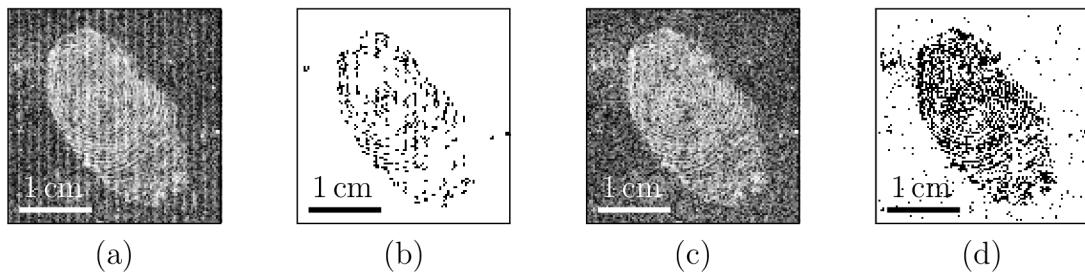
barvou jsou označeny body intenzity, které by byly vynulovány pro $n = 4$ ²⁴. Jde o nejnižší bezpečnou hodnotu n , kdy pixely vzdálenější od n sousedních pixelů dále nenabývají tak vysoké intenzity, která by byla na úrovni intenzity vady. Nejnižší hodnota n , při které zůstávají zachovány pouze body na úrovni šumu, je $n = 7$. Tomu na obr. 2.33 odpovídá odstranění černých a bílých bodů.



Obrázek 2.33: Ukázka k vysvětlení principu metody Saturace. Saturované pixely spolu s černými a bílými body jsou nulovány, aby body náležící vadám měly nejvyšší hodnoty intenzity.

Zdálo by se, že čím více zachovaných bodů ze snímku je na úrovni šumu, tím větší intenzitní citlivosti metoda dosahuje. Není to ale úplně pravda, protože některé vady jsou lépe viditelné, pokud se nachází v blízkosti světlého proužku. Vyšší hodnota n pro ně může být eliminující. Protože za cenu potlačení vlivu rozmazaného rozhraní mezi světlým a tmavým proužkem ve skutečnosti ztrácíme intenzitní citlivost kvůli nižší hodnotě poměru signálu k šumu. Prakticky tento rozdíl ukazuje srovnání detekované vady otisku prstu pro dvě různá n . Na obrázcích 2.34(a) a (c) jsou šedé snímky pro $n = 4$ a $n = 7$. Struktura vady více odpovídá realitě pro $n = 7$. Nicméně přestože výstupní snímek jasně ukazuje strukturu vady [obr. 2.34(d)], nese s sebou silný šum oproti výstupnímu snímku pro $n = 4$ [obr. 2.34(b)]. Struktura je sice silně ovlivněna rozhraním světlého a tmavého proužku, nicméně hodnota poměru signálu k šumu je na šedém snímku vyšší.

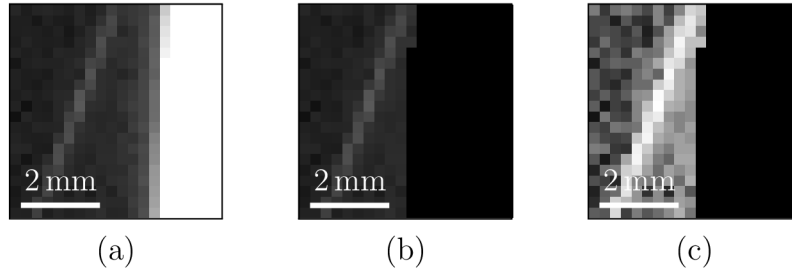
Metoda může pracovat v obou extrémech. Lepších šedých snímků dosahuje pro $n = 7$ a pro $n = 4$ si vede lépe naopak ve výstupních snímcích. My si vybíráme druhou možnost, protože nás zajímá především detekce vady.



Obrázek 2.34: Vliv čísla n na šedý a výstupní snímek. Šedý (a) a výstupní (b) snímek pro $n = 4$. Šedý (c) a výstupní (d) snímek pro $n = 7$. Šedé snímky (a) a (b) jsou s ekvalizovaným histogramem.

²⁴Blíže je hodnota n popsána v sekci Určení parametrů - Určení parametru n metody Saturace.

Na obrázku 2.35(a) vidíme výřez jednoho ze vstupních snímků, na kterém je část saturovaného proužku a podélná vada. Tyto saturované pixely a jejich okolí v horizontálním směru do vzdálenosti $n = 4$ pixelů nahrazeny pixely s nulovou intenzitou [obr. 2.35(b)]. Na obrázku 2.35(c) vidíme tentýž výřez s ekvalizovaným histogramem. Pro $n = 4$ zřejmě v obraze zůstávají pixely v oblasti bez vady s vyšší hodnotou intenzity, protože na ně dopadají paprsky světlých proužků z LCD panelu. Nicméně jejich intenzita je bezpečně nižší než intenzita pixelů podélné vady a můžeme je od sebe odlišit.



Obrázek 2.35: (a) Výřez vstupního snímku, (b) výřez snímku po odstranění saturovaných pixelů a jejich okolí v horizontálním směru do vzdálenosti $n = 4$, (c) tentýž výřez s ekvalizovaným histogramem.

Postup

Mějme vstupní snímky S_1, S_2, \dots, S_8 s šířkou w a výškou h . Dále označme k -tý vstupní snímek S_k , kde $k \in \{1, 2, \dots, 8\}$. Nechť snímek S'_k je kopií snímku S_k a intenzita $I'_k[x, y]$ snímku S'_k se rovná intenzitě $I_k[x, y]$ snímku S_k .

Snímek S_k obsahuje pixely s intenzitou $I_k[x, y] = 255$, které spolu tvoří vertikální saturované proužky. Pro každý takový pixel na snímku S_k jsou na snímku S'_k intenzity

$$I'_k[x - i, y] = 0 \quad \forall i \in \{-n, -n + 1, \dots, n\}, \quad (2.44)$$

kde $x, y \in \mathbb{N}$, $x \in \langle n, w - n \rangle$ a $y \in \langle 0, h \rangle$. Postup se saturovaným pixellem na snímku S_k znázorňuje výřez snímku na obr. 2.36, kde hodnota intenzity každého pixelu P_i je podle vztahu (2.44) ve snímku S'_k nahrazena nulou.

			P_{-n}	\dots	P_{-2}	P_{-1}	P_0	P_{+1}	P_{+2}	\dots	P_{+n}			

Obrázek 2.36: Výřez snímku S'_k o ploše $(3 \times 15) \text{ px}^2$, na kterém je část saturovaného proužku. Intenzita pixelu P_0 na snímku S_k je rovna saturované hodnotě intenzity 255, proto jsou intenzity pixelů P'_i na snímku S'_k položeny rovny nule. Obrázek je upraven zvýšením jasu, tj. intenzita $I = 40$ odpovídá černému pixelu.

Ve snímku S'_k jsou zřejmě saturované proužky vynulovány společně s jejich okolím v horizontálním směru do vzdálenosti n .

Zavedme množinu $M_{I'_k[x,y]} = \{I'_k[x,y]; k = 1, 2, \dots, 8\}$ a na jejím základě vytvořme šedý snímek

$$S[x, y] = \max(M_{I'_k[x,y]}), \quad (2.45)$$

kde $\max(M_{I'_k[x,y]})$ je největším prvkem množiny $M_{I'_k[x,y]}$. Dále zavedme parametr L_{Sat} ²⁵. Potom vzniká výstupní snímek

$$S_{\text{BW}}[x, y] = \begin{cases} 0, & \text{pokud } S[x, y] > L_{\text{Sat}} \\ 255, & \text{jinak.} \end{cases} \quad (2.46)$$

Ve snímcích S'_1, S'_2, \dots, S'_8 je na pozici $[x, y]$ nalezena největší intenzita a uložena na stejnou pozici do šedého snímku S [vztah (2.45)]. Následně je porovnána s parametrem L_{Sat} . Pokud je hodnota na souřadnici $[x, y]$ na šedém snímku S větší jak L_{Sat} , pak je na výstupním snímku S_{BW} na téže pozici zaznamenána vada [vztah (2.46)].

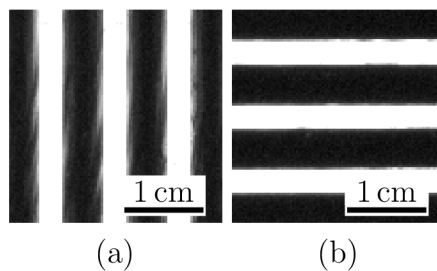
2.6.7 Metoda Rotace

Metoda dostává na vstupu celkem osm snímků. Společným jmenovatelem prvních čtyř snímků je binární obrazec světlých proužků o šířce 10 px a periodě 30 px, který je otočen vůči horizontální ose postupně o $0^\circ, 45^\circ, 90^\circ$ a 135° . Druhá čtveřice snímků obsahuje binární obrazec téže mřížky posunutý o polovinu periody 15 px, který je rotován o stejné úhly.

Expozice kamery musí být minimálně taková, aby docházelo k saturaci pixelů snímače, kterým odpovídají obrazy pixelů světlých proužků na LCD panelu. A zároveň ne příliš dlouhá, aby zůstal zachován kontrast binárního obrazce. Expozici kamery zvolme $\tau = 200$ ms.

Ze saturovaných pixelů zřejmě nedostáváme žádnou užitečnou informaci. Jinak je tomu v tmavé oblasti. Pokud se v ní nevyskytuje vada, zaznamenává snímač kamery pouze delší expozicí zesílený šum. V případě výskytu vady, která rozptyluje paprsky jdoucí od LCD panelu, ji můžeme na pozadí šumu spatřit. Protože vada v oblasti světlého proužku je neviditelná, je nutné mřížku posunout o půl periody, abychom i tuto vadu mohli detekovat.

Některé vady podlouhlého tvaru rozptylují paprsky více, pokud s proužky svírají úhel menší než 45° [obr. 2.37(a)]. Vada viditelně snižuje gradient přechodu z tmavého do světlého proužku. Na obrázku 2.37(b) je totéž místo na skle s proužky binárního obrazce otočenými o 90° . Přestože lze místy pokles gradientu detekovat, je projev vady výrazně nižší. Z tohoto důvodu jsou proužky na vstupních snímcích postupně rotovány o 45° .



Obrázek 2.37: Vliv rotace mřížky na podélné vady.

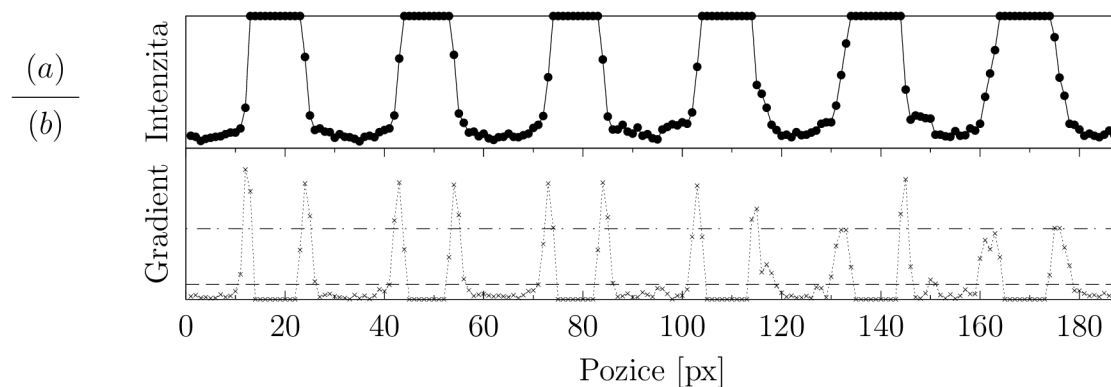
²⁵Hodnota parametru L_{Sat} metody Saturace je určena v sekci Určení parametrů metod.

Princip

Paprsky rozptylující vady snižují kontrast binárního obrazce, neboli velikost gradientu je na tomto přechodu ponížena. Rozeberme si tento jev na vadě. Na obr. 2.38(a) je průběh intenzity šesti světlých proužků, které jsou v saturaci a na obr. 2.38(b) je vypočítaný gradient pro pixel intenzity na stejné pozici. Gradient počítáme Sobelovým operátorem, který potřebuje znát rozšířené okolí centrálního pixelu (toto okolí je při výpočtu známo, avšak kvůli přehlednosti jsou vykresleny pouze centrální pixely). V průběhu gradientu pozorujeme ostré píky odpovídající prudkým změnám intenzity a tupé odpovídající pozvolným změnám. V místě saturované intenzity je gradient nulový a v oblasti šumu tmavých proužků jsou hodnoty gradientu nízké.

V průběhu gradientu pozorujeme ostré píky odpovídající prudkým změnám intenzity a tupé odpovídající pozvolným změnám. V místě saturované intenzity je gradient nulový a v oblasti šumu tmavých proužků jsou hodnoty gradientu nízké.

Levá strana obr. 2.38 odpovídá oblasti skla bez vad, na pravé straně vidíme projev vady matný. Detekce spočívá v ořezání hodnot gradientů odpovídajícím šumu a také těch, které odpovídají místům ostré změny intenzity v oblasti bez vad. K tomu slouží interval, jehož nižší mezní hodnota daná parametrem L_{Rot} je vykreslena čárkovaně a vyšší mezní hodnota daná parametrem U_{Rot} je vykreslena čerchovaně ²⁶.



Obrázek 2.38: Názorné vysvětlení principu metody Rotace. (a) Data z jednoho ze vstupních snímků, (b) vypočítané gradienty pomocí Sobelova operátoru. Čárkovaně je zvýrazněna hodnota parametru L_{Rot} a čerchovaně parametru U_{Rot} .

Nicméně po tomto kroku stále máme v souboru gradienty i z oblasti bez vad. Ukazuje se, že centrální pixely odpovídající těmto gradientům mají v rozšířeném okolí dalších maximálně pět sousedních pixelů, jejichž gradienty se nachází uvnitř intervalu. V případě centrálních pixelů v oblasti vady je těchto sousedních pixelů zpravidla šest a více. Postačuje proto zavést podmínku na počet sousedních pixelů splňujících podmínku na velikost gradientu.

Postup

Vstupní snímky S_1, S_2, \dots, S_8 , mají každý šířku w a výšku h . Označme k -tý vstupní snímek S_k , kde $k \in \{1, 2, \dots, 8\}$. Potom konvoluce snímku S_k s oběma Sobelovými maskami H_x a H_y (1.3) jsou

²⁶Hodnoty parametrů L_{Rot} a U_{Rot} metody Rotace jsou určeny v sekci Určení parametrů metod.

$$G_x(S_k)[x, y] = \sum_{i=-1}^1 \sum_{j=-1}^1 I_k[x-i, y-j] H_x[i, j], \quad (2.47)$$

$$G_y(S_k)[x, y] = \sum_{i=-1}^1 \sum_{j=-1}^1 I_k[x-i, y-j] H_y[i, j],$$

kde $x, y \in \mathbb{N}$, $x \in \langle 1, w-1 \rangle$ a $y \in \langle 1, h-1 \rangle$. Velikost gradientu na pozici $[x, y]$ snímku S_k je

$$G(S_k)[x, y] = \sqrt{G_x^2(S_k)[x, y] + G_y^2(S_k)[x, y]}. \quad (2.48)$$

Vzniká nový snímek

$$G_k[x, y] = \begin{cases} 1, & \text{pokud } \{G(S_k)[x, y] > L_{\text{Rot}}\} \wedge \{G(S_k)[x, y] < U_{\text{Rot}}\}, \\ 0, & \text{jinak.} \end{cases} \quad (2.49)$$

Ve snímku G_k jsou hodnoty $G(S_k)$, které se nenalézají v intervalu $(L_{\text{Rot}}, U_{\text{Rot}})$, nahrazeny nulou.

Uvedli jsme, že pixely o hodnotě 1 detekující vadu mají ve svém rozšířeném okolí více jak 5 pixelů rovněž o hodnotě 1. Označme $F_{\text{Rot}} = 5$ jako sumu pixelů s hodnotou 1 v rozšířeném okolí vyšetřovaného pixelu, od které je vyšetřovaný pixel považován za vadu. Aby se určil počet pixelů s hodnotou 1, tak následuje konvoluce G_k s jedničkovou maskou

$$J = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.50)$$

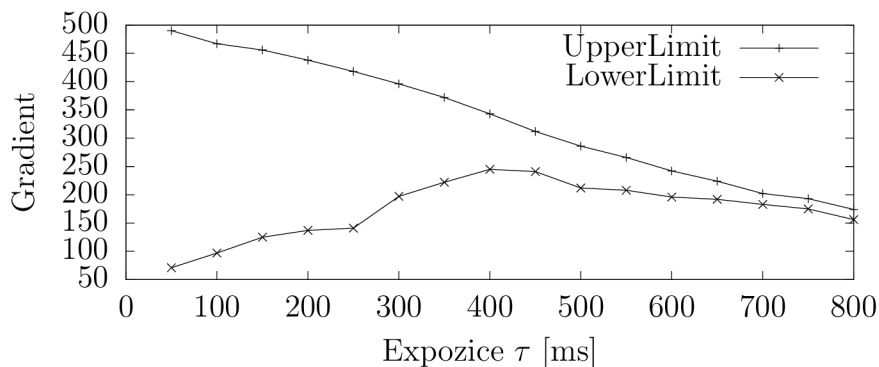
pro všechna $G_k[x, y] = 1$. V opačném případě je snímku R_k na dané pozici přiřazena 0, $R_k[x, y] \equiv 0$.

$$R_k[x, y] = (G_k * J)[x, y] = \sum_{i=-1}^1 \sum_{j=-1}^1 G_k[x-i, y-j] J[i, j]. \quad (2.51)$$

Hodnota $R_k[x, y]$ odpovídá počtu pixelů rozšířeného okolí centrálního pixelu na pozici $[x, y]$, jejichž hodnota je 1. Za předpokladu znalosti $R_k[x, y]$ pro všechna k je intenzita výstupního snímku na pozici $[x, y]$

$$R_{\text{BW}}[x, y] = \begin{cases} 0, & \text{pokud } \exists k, k \in \{1, 2, \dots, 8\} : R_k[x, y] > F_{\text{Rot}}, \\ 255, & \text{jinak.} \end{cases} \quad (2.52)$$

Ukažme si závislost parametrů L_{Rot} a U_{Rot} na expozici τ . Pořídme několik sad vstupních snímků malé oblasti skla bez vad pro různá τ (od 50 ms do 800 ms s krokem 50 ms). Pro každou sadu vstupních snímků nechme metodu Rotace vypočítat parametry L_{Rot} , U_{Rot} a vykresleme je v závislosti na expozici τ (obr. 2.39). Aplikujeme-li vypočítané parametry na snímky s vadou matný pro různá τ , zjišťujeme, že pro expozice menší jak 200 ms je interval $(L_{\text{Rot}}, U_{\text{Rot}})$ příliš široký a na snímcích G_k se objevuje příliš mnoho pixelů o hodnotě 1. Pomocí vztahů (2.51) a (2.52) potom nelze jednoduše rozlišit pixely s vadami a bez vad. Pro expozice větší jak 400 ms je na druhou stranu interval $(L_{\text{Rot}}, U_{\text{Rot}})$ natolik úzký, že pixely v místě vady nemají ve svém rozšířeném okolí žádné pixely o hodnotě 1 a metoda v jejich identifikaci opět selhává.



Obrázek 2.39: Závislost parametrů L_{Rot} (LowerLimit) a U_{Rot} (UpperLimit) na expoziční kamery τ .

2.6.8 Určení parametrů metod

V této sekci jsou shrnuty parametry metod a dále je popsáno, jakým způsobem se určuje jejich hodnota. Je představen způsob automatického výpočtu parametrů, který je závislý pouze na kvalitě vstupního snímku. Každá z metod obsahuje v první části tento automatický výpočet, s jehož výstupem – vyhodnoceným parametrem metody – dále pracuje při analýze snímku nebo snímků krytů světlometů s vadami.

Shrnutí parametrů

Metody Maximum, Aliasing

Parametry D_{Max} , D_{Ali} určují minimální absolutní hodnotu rozdílu intenzit centrálního a sousedního pixelu, od které jsou všechny vyšší hodnoty významné.

Metoda Variance

Parametr D_{Var} určuje minimální hodnotu rozdílu pseudorozptylů dvou protilehlých množin centrálního pixelu, od které jsou všechny větší rozdíly významné.

Metoda Grey

Parametr D_{Gre} určuje minimální hodnotu rozdílu intenzity centrálního pixelu od intenzity sousedního pixelu, od které jsou všechny vyšší hodnoty významné.

Metoda Saturace

Parametr L_{Sat} určuje minimální hodnotu intenzity, od které jsou všechny vyšší hodnoty významné.

Metoda Rotace

Parametry L_{Rot} a U_{Rot} vymezují interval $(L_{\text{Rot}}, U_{\text{Rot}})$, v němž nacházející se gradienty $G(S_k[x, y])$ počítané pomocí Sobelova operátoru jsou významné.

Určení parametrů D_{Max} , D_{Ali} , D_{Var} , D_{Gre} a L_{Sat}

Pro zjednodušení si pod parametrem D představme kterýkoli z parametrů D_{Max} , D_{Ali} , D_{Var} , D_{Gre} nebo L_{Sat} . Hlavní část výpočtu parametru D je shodná s metodou, které parametr náleží. Odlišnost je ve vstupních snímcích a ve vyhodnocování výstupního snímku. Vstupní snímky musí obsahovat pouze samotné čiré sklo bez vad a jiných

neprůsvitných částí skla. Velikost snímků není podstatná. Přesto spíše volíme menší oblast, u které jsme si jisti, že skutečně neobsahuje žádné vady. Rovněž by tato oblast měla zahrnovat část šíbrů, což je zlom mezi dvěma plochami na skle.

S těmito snímky se dále pracuje stejně jako v samotné metodě. Metoda ale potřebuje znát velikost parametru. Tento parametr proto položíme na počátku roven nule, $D = 0$. Metoda poté vypracovává výstupní snímek, který obsahuje černé pixely indikující vadu. Sumu těchto pixelů označme P_0 , kde dolní index značí hodnotu parametru D . My ale předpokládáme, že na našich vstupních snímcích žádná vada není. Proto navyšujeme hodnotu parametru D o jedničku a proces opakujeme. Tím získáváme z nového černobílého snímku novou sumu černých pixelů P_1 , která je menší nebo rovna P_0 . Dalším iterováním metody vždy s vyšší hodnotou parametru D o jedničku snižujeme sumu černých pixelů P_D do té doby, dokud není rovna nule. To se shoduje s tvrzením, že metoda nenalezla žádné vady, což je v souladu s naším předpokladem. Hodnota parametru D , pro které platí $P_D = 0$, je potom jeho konečná hodnota.

Proto požadujeme pro vyhodnocení čiré sklo bez vad. Pokud by se ve vstupním snímku nacházela jakákoli vada o velikosti minimálně 1 pixel, potom by se hodnota parametru D navyšovala do té doby, kdy by i tato vada za vadu považována nebyla. Tím by se snížila intenzitní citlivost metody, čímž máme na mysli schopnost nacházet vady s ohledem na jejich čitelnost příslušnou metodou.

Určení parametrů L_{Rot} a U_{Rot}

Vyhodnocení parametrů L_{Rot} a U_{Rot} metody Rotace se od předchozího způsobu mírně odlišuje. Opět platí požadavek na vstupní snímky. Na rozdíl od samotné metody ale nepotřebujeme všech osm vstupních snímků, ale postačuje nám pouze jeden vstupní snímek S_1 .

Nejprve počítáme všechny gradienty ve vstupním snímku a z jejich znalosti definujeme parametr U_{Rot} jako polovinu největšího gradientu $U_{\text{Rot}} = \frac{1}{2} \max \{G(S_1)[x,y]\}$.

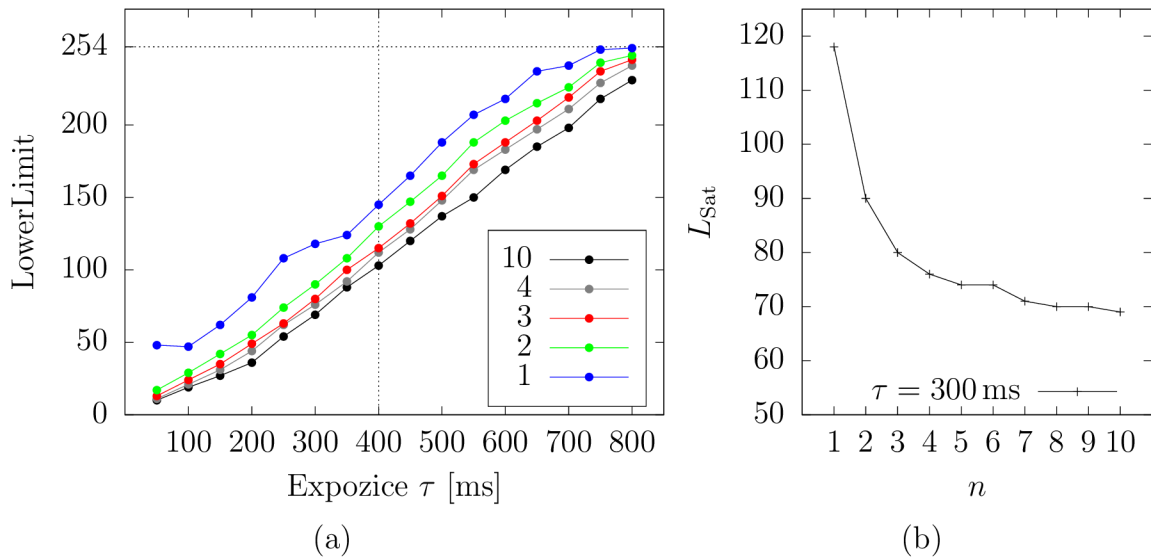
Poté provádíme vyhodnocení černobílého výstupního snímku podobně jako ve vlastní metodě Rotace s tím rozdílem, že vstupní snímek S_1 je opět jediný. Parametr U_{Rot} známe a parametr L_{Rot} položíme nejprve roven nule. Odtud je postup shodný s postupem pro určování parametru D .

Metoda nejprve vypracovává výstupní snímek, který obsahuje sumu $P_{L_{\text{Rot}}}$ černých pixelů pro $L_{\text{Rot}} = 0$. Metodu opakujeme a současně parametr L_{Rot} navyšujeme o jedničku do té doby, kdy je počet černých pixelů $P_{L_{\text{Rot}}}$ roven nule. Splněním této podmínky je hodnota parametru L_{Rot} i jeho konečná hodnota.

Určení parametru n metody Saturace

Abychom zjistili vhodnou hodnotu parametru n , pořídíme několik sad vstupních snímků malé oblasti skla bez vad pro různé expozice τ (od 50 ms do 800 ms s krokem 50 ms). Pro každou sadu vstupních snímků nechme program vypočítat parametr L_{Sat} pro různé hodnoty n (od 1 do 10). Nakonec vykresleme závislost L_{Sat} na expozici τ pro různé hodnoty parametru n [obr. 2.40(a)].

V případě $n = 1$ nejsou z kompozitního snímku vstupních snímků odstraněny pixely, které sice nejsou v saturaci, ale jejich intenzita je značně vysoká, což zbytečně zvyšuje hodnotu L_{Sat} . Naopak pro $n = 10$ je kompozitní snímek zcela bez rušivého rozmazaného přechodu mezi světlým a tmavým proužkem. Nicméně ztrácíme informaci o vadách,



Obrázek 2.40: (a) Závislost parametru L_{Sat} (LowerLimit) metody Saturace na expoziční době τ pro různé hodnoty parametru n . (b) Průběh L_{Sat} v závislosti na n pro expoziční dobu kamery v metodě Saturace $\tau = 300$ ms.

kteřé jsou detekovatelné pouze v blízkosti světlých proužků, neboli právě v oblasti tohoto přechodu.

Kompromisem obou extrémů je $n = 4$. Jde o nejnižší hodnotu, kdy je zachována detekce vad detekovatelných pouze v oblasti přechodu, a zároveň je parametr L_{Sat} dostatečně nízký, abychom zachovali detekci vad s nízkými hodnotami intenzity pixelů ve snímku. Na obr. 2.40(b) je navíc vykreslená závislost L_{Sat} na n pro expoziční dobu kamery v metodě Saturace $\tau = 300$ ms.

Kapitola 3

Výsledky a diskuze

Poslední kapitola tohoto textu se věnuje detekci dekorativních vad. Pro každou vadu zvlášť ukazuje, s jakou úspěšností je vada jednotlivými metodami lokalizována. Obsahuje výřezy výstupních i šedých snímků metod s vadami, případně také vstupní snímky pro lepší názornost. Konec kapitoly je věnován diskuzi.

3.1 Analýza dekorativních vad

Po nastavení senzoru a vyhotovení sérií snímků pro FCC přejdeme k pořízení vstupních snímků metod. Před LCD panel umístíme sklo tak, abychom ve středu snímku pořízeného kamerou viděli oblast skla s vadou. Následně na LCD panelu nechme postupně vykreslit všechny binární obrazce. Přitom pro každý z nich kamera pořizuje jeden snímek s příslušnou expozicí τ . Tento postup provádíme pro všechna skla s vadami.

Takto vzniklé vstupní snímky R opravme pomocí FCC. Výstupní snímky C metody FCC jsou nyní vstupními snímky metod pro detekci dekorativních vad a těmito metodami je analyzujeme. Výsledky analýzy si ukažme na výstupních snímcích metod postupně pro každou vadu.

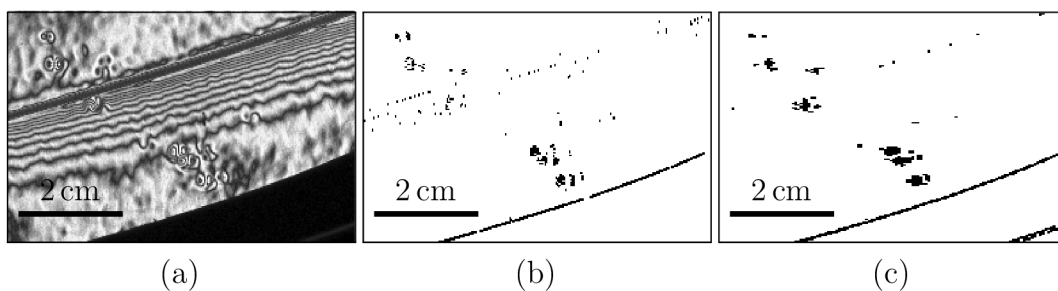
Vstupní snímky před korekcí metodou FCC a po korekci spolu s výstupními snímky metod jsou součástí přílohy C Ostatní na příloženém CD.

3.1.1 Voda na skle

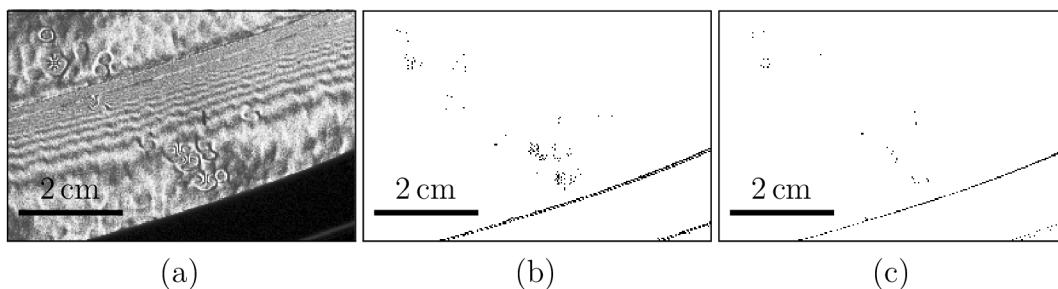
Vada voda na skle se na testovaném skle nachází v několika skupinách v těsné blízkosti šíbru. Ty jsou zřejmé na obou snímcích [obr. 3.1(a) a 3.2(a)], což jsou výřezy šedého snímku M metody Maximum a vstupního snímku metod Variance a Grey. Na výstupních snímcích M_{BW} a A_{BW} [obr. 3.1(b) 3.1(c)] je voda na skle výrazná, obsahují však patrnou linii šíbru [obr. 2.2(a)], přičemž výstupní snímek M_{BW} o něco více.

Srovnáme-li je s výstupními snímky V_{BW} [obr. 3.2(b)] a G_{BW} [obr. 3.2(c)], vada není tak jasná, ale přesto je se stejnou úspěšností zachycena. Jedná se o téměř nepatrný záchyt především u výstupního snímku G_{BW} . Zdůrazněme však absenci jakéhokoli náznaku šíbru u obou výstupních snímků.

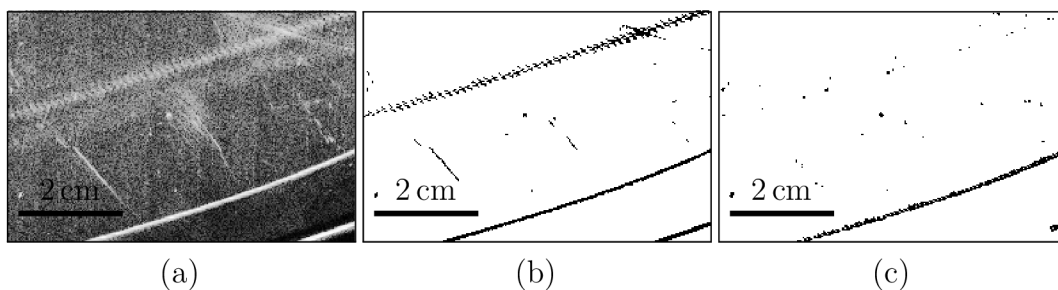
Na šedém snímku metody Saturace [obr. 3.3(a)] není vada voda na skle vůbec patrná. Proto i na jejím výstupním snímku S_{BW} [obr. 3.3(b)] nenacházíme žádné náznaky. Naopak šíbr je zde výrazně zaznačen. Stejně neúspěšně si vede i metoda Rotace s výstupním snímkem R_{BW} [obr. 3.3(c)], přičemž linie šíbru je také patrná.



Obrázek 3.1: Dekorativní vada voda na skle na výřezech (a) šedého snímku M s ekvalizovaným histogramem, (b) výstupního snímku M_{BW} metody Maximum a (c) výstupního snímku A_{BW} metody Aliasing.



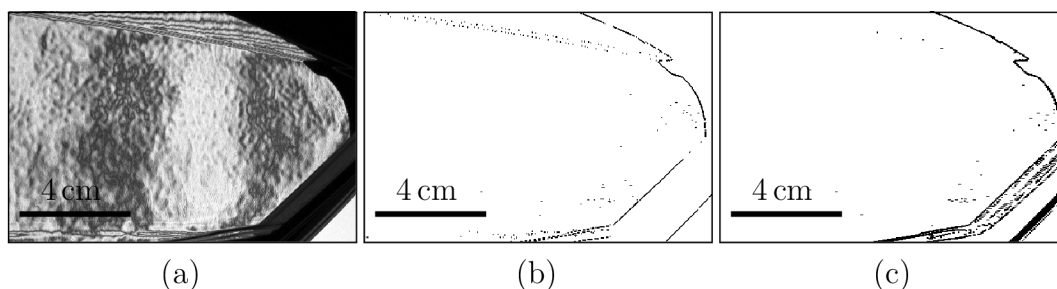
Obrázek 3.2: Dekorativní vada voda na skle na výřezech (a) vstupního snímku metod Variance a Grey s ekvalizovaným histogramem a výstupních snímcích (b) V_{BW} metody Variance a (c) G_{BW} metody Grey.



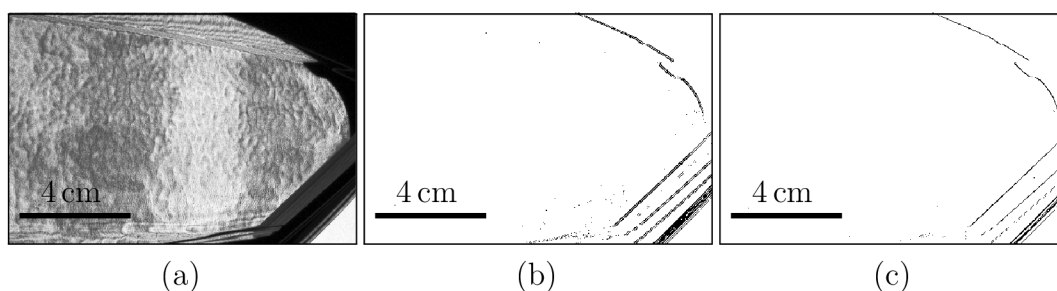
Obrázek 3.3: Dekorativní vada voda na skle na výřezech (a) šedého snímku S s ekvalizovaným histogramem, (b) výstupního snímku S_{BW} metody Saturace a (c) výstupního snímku R_{BW} metody Rotace.

3.1.2 Matný

Metody Maximum, Aliasing, Variance i Grey jsou v detekci vady matný zcela neúspěšné. Při pohledu na jejich výstupní snímky M_{BW} , A_{BW} , V_{BW} a G_{BW} [obr. 3.4(b), (c) a 3.5(b), (c)] nepozorujeme žádný náznak vady matný v oblasti, kde ji očekáváme. V případě metod Maximum a Aliasing je navíc patrný šíbr. Metoda Maximum detekuje šíbr výrazněji. Vada matný se neprojevuje ani na šedém snímku metody Maximum [obr. 3.4(a)], ani na vstupním snímku metod Variance a Grey [obr. 3.5(a)].



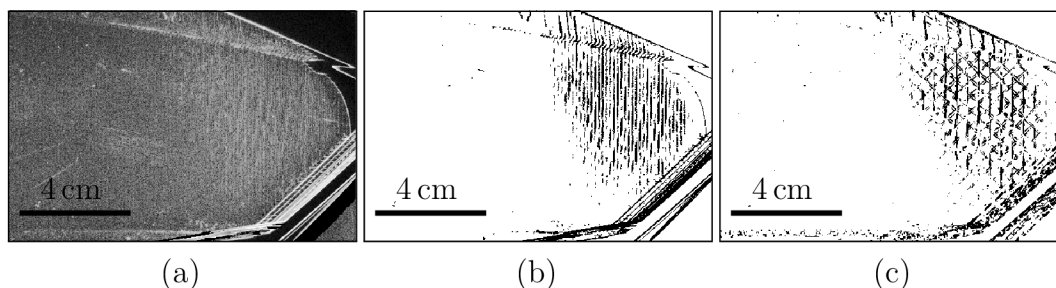
Obrázek 3.4: Dekorativní vada matné na výřezech (a) šedého snímku M s ekvalizovaným histogramem, (b) výstupního snímku M_{BW} metody Maximum a (c) výstupního snímku A_{BW} metody Aliasing.



Obrázek 3.5: Dekorativní vada matné na výřezech (a) vstupního snímku metod Variance a Grey s ekvalizovaným histogramem a výstupních snímcích (b) V_{BW} metody Variance a (c) G_{BW} metody Grey.

Metoda Rotace, která byla s ohledem na tuto vadu navržena, si pochopitelně vede v její detekci dobře. Ve výstupním snímku R_{BW} [obr. 3.6(c)] je jasně pozorovatelný vzor rotujících proužků binárního obrazce pokrývajících vadu matný. Díky tomuto vzoru velmi obtížně můžeme pozorovat vlastní strukturu vady matný. Pokud bychom tohoto chtěli dosáhnout, pravděpodobně by stačilo rozšířit soubor vstupních snímků o několik dalších rotovaných sad posunutých proužků. Aktuálně metoda pracuje s posunutím pouze o půl periody proužků a vznikají dvě sady. Více posunutím bychom získali jemnější strukturu vady matný a potlačili bychom vlastní strukturu rotujících proužků.

Názorné je to u metody Saturace, která pracuje se vstupními snímky vertikálních proužků posunutých celkem sedmkrát vůči počátečnímu stavu v rámci jedné periody. Náznak vertikálních proužků na obr. 3.6(b) pozorujeme, jasněji se ale projevuje struktura vady matný svými oblouky. Vizually je vada pěkně pozorovatelná na šedém snímku metody Saturace [obr. 3.6(a)].

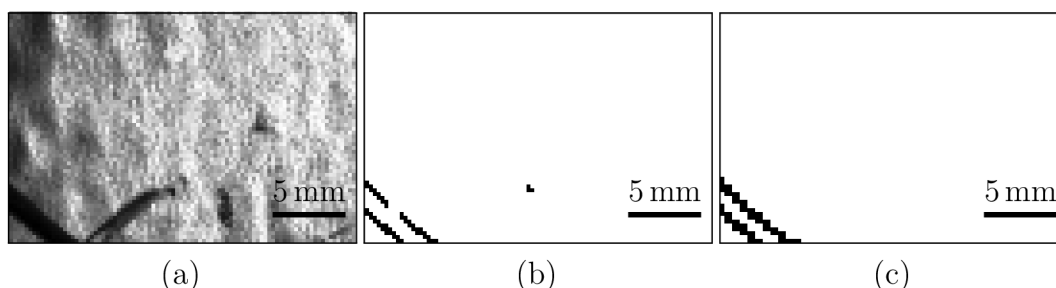


Obrázek 3.6: Dekorativní vada matné na výřezech (a) šedého snímku S s ekvalizovaným histogramem, (b) výstupního snímku S_{BW} metody Saturace a (c) výstupního snímku R_{BW} metody Rotace.

3.1.3 Zástřík

Drobná vada zástřík má na testovaném skle velikost přibližně 2 mm a tvar řeckého písmene α . Navíc je lokalizovaná černým fixem. Spatřit ji můžeme na šedém snímku M metody Maximum [obr. 3.7(a)], nepatrně na vstupním snímku metod Variance a Grey [obr. 3.8(a)] a zřetelně také na šedém snímku S metody Saturace [obr. 3.9(a)]. Metody Maximum a Aliasing si na výstupních snímcích M_{BW} a A_{BW} [obr. 3.7(b), 3.7(c)] s detekcí neporadí, protože se vada chová podobně jako šum způsobený fluktuacemi tloušťky vrstvy laku na skle v levé části snímku. Tento problém by mohl být řešen rozčleněním snímku na světlé a tmavé části, z nichž ty tmavé by byly ignorovány, protože v jejich místě je informace o vadě ztracena ve zmíněném šumu. Potom by metoda mohla být citlivější vzhledem ke světlým oblastem a vadu by mohla nalézt.

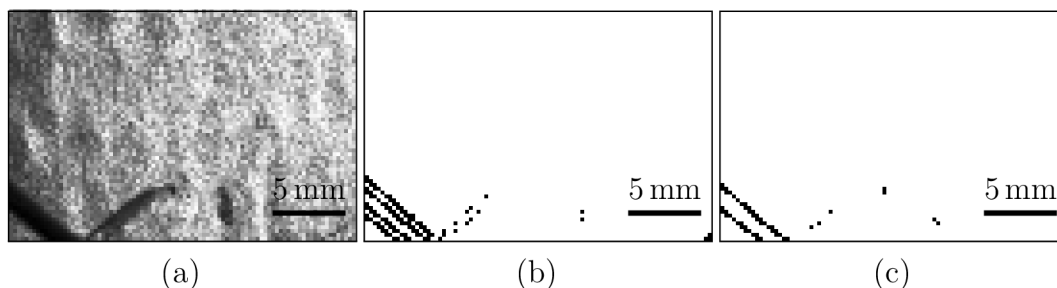
Poznamenejme také, že obě metody ne zcela identifikují čáry černého fixu. Ty části, které jsou více průsvitné, metoda ignoruje stejně jako vadu zástřík.



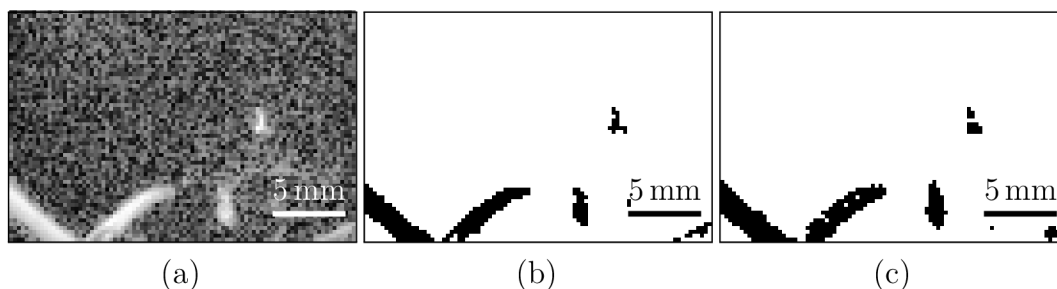
Obrázek 3.7: Dekorativní vada zástřík na výřezech (a) šedého snímku M s ekvalizovaným histogramem, (b) výstupního snímku M_{BW} metody Maximum a (c) výstupního snímku A_{BW} metody Aliasing.

Přestože si vady zástřík, pokud známe její souřadnice, můžeme na snímku všimnout [obr. 3.8(a)], není ani pro jednu z metod Variance ani Grey viditelná. Její projev ve vstupním snímku je příliš slabý a srovnatelný s projevem šumu. Proto ji na výstupních snímcích V_{BW} a G_{BW} [obr. 3.8(b), 3.8(c)] nepozorujeme. Černý fix metody identifikují o něco lépe nežli metody Maximum a Aliasing.

Úspěšně se jí daří detekovat pouze oběma metodám Saturace a Rotace, jak vidíme na výstupních snímcích S_{BW} a R_{BW} [obr. 3.9(b), 3.9(c)]. Rovněž na nich spolehlivě vidíme téměř celou část ohraničení černým fixem ve spodních částech snímků. Na výstupním snímku metody Saturace je znatelný i tvar vady zástřík „ α “.



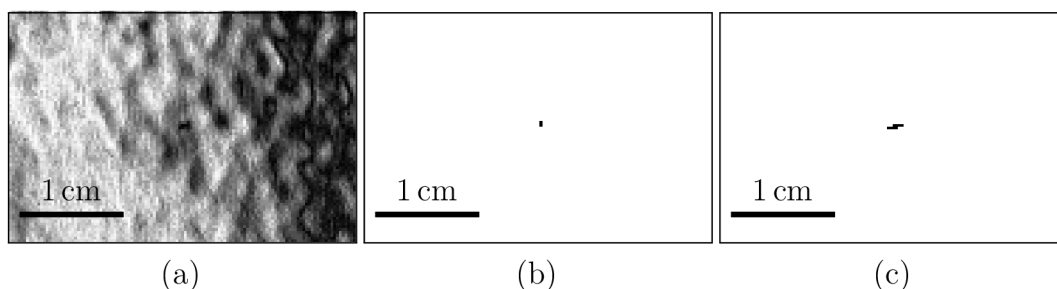
Obrázek 3.8: Dekorativní vada zástřík na výřezech (a) vstupního snímku metod Variance a Grey s ekvalizovaným histogramem a výstupních snímcích (b) V_{BW} metody Variance a (c) G_{BW} metody Grey.



Obrázek 3.9: Dekorativní vada zástřík na výřezech (a) šedého snímku S s ekvalizovaným histogramem, (b) výstupního snímku S_{BW} metody Saturace a (c) výstupního snímku R_{BW} metody Rotace.

3.1.4 Cizí tělísko

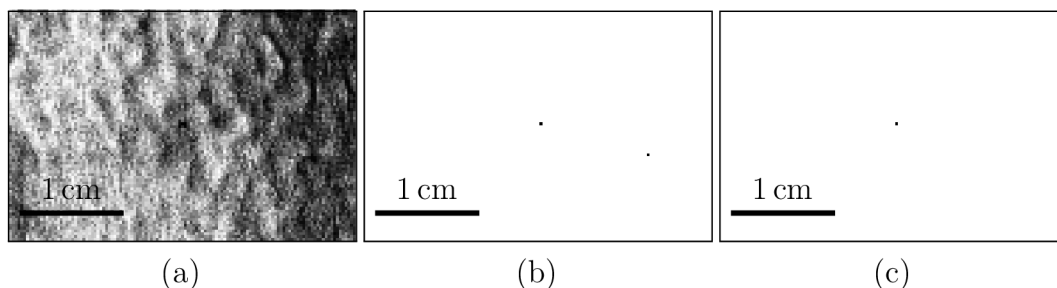
Na šedém snímku M metody Maximum [obr. 3.10(a)] je drobná vada cizí tělísko o velikosti přibližně 1 mm viditelná i přes vliv šumu fluktuací tloušťky laku na skle. Nicméně i přes její detekci na snímku M_{BW} [obr. 3.10(b)] není tato metoda robustní. Lépe je na tom metoda Aliasing, jejíž výstupní snímek A_{BW} je na obr. 3.10(c).



Obrázek 3.10: Dekorativní vada cizí tělísko na výřezech (a) šedého snímku M s ekvalizovaným histogramem, (b) výstupního snímku M_{BW} metody Maximum a (c) výstupního snímku A_{BW} metody Aliasing.

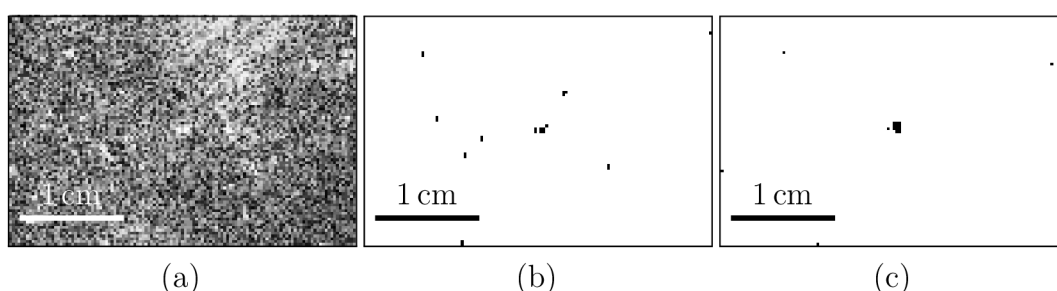
Podobně jako metoda Maximum jsou na tom i metody Variance a Grey. Na jejich vstupním snímku [obr. 3.11(a)] je sice vada zřejmá, avšak vzhledem k její velikosti je na výstupních snímcích V_{BW} , G_{BW} [obr. 3.11(b), (c)] identifikována pouze jedním pixelem.

Metody s poměrně dlouhou expoziční dobou τ , jakými jsou Saturace a Rotace, si v detekci vedou nejlépe. Vada je oběma metodami lokalizována několika pixely. U obou



Obrázek 3.11: Dekorativní vada cizí tělísko na výřezech (a) vstupního snímku metod Variance a Grey s ekvalizovaným histogramem a výstupních snímcích (b) V_{BW} metody Variance a (c) G_{BW} metody Grey.

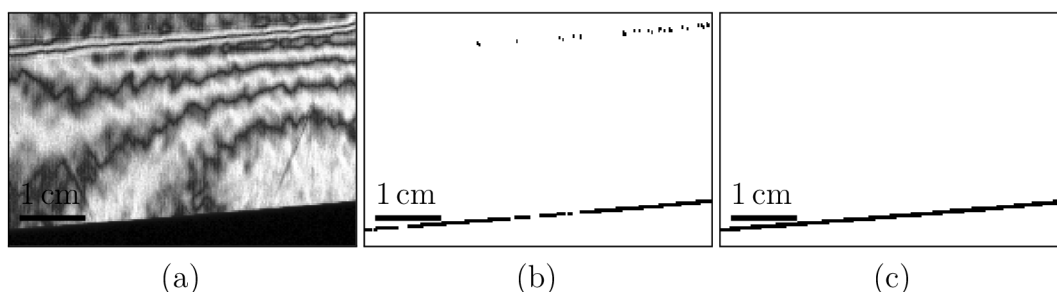
výstupních snímků S_{BW} , R_{BW} [obr. 3.12(b), (c)] si povšimněme detekce dalších vad patrných na šedém snímku S [obr. 3.12(a)], přestože to jindy nezmiňujeme.



Obrázek 3.12: Dekorativní vada cizí tělísko na výřezech (a) šedého snímku S s ekvalizovaným histogramem, (b) výstupního snímku S_{BW} metody Saturace a (c) výstupního snímku R_{BW} metody Rotace.

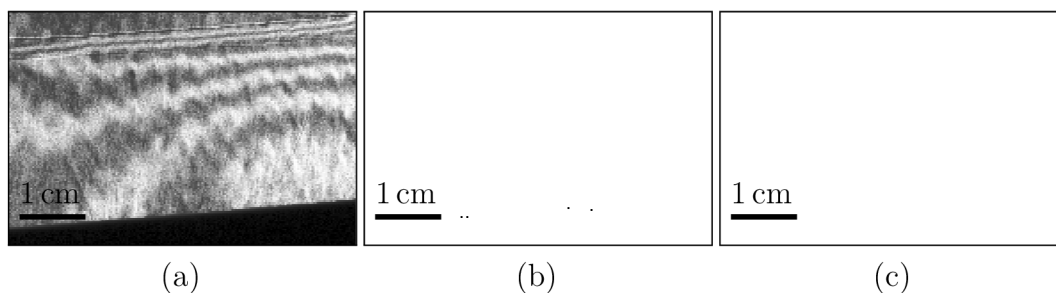
3.1.5 Šlíra v čiré

Vada šlíra v čiré je podélného charakteru a je patrná na šedém snímku M na obr. 3.13(a). Jedná se o podélný útvar napravo dole. Nicméně se vada projevuje podobně jako její okolí a pro metodu Maximum je neviditelná. Ta vidí pouze linii šíbru [výstupní snímek M_{BW} na obr. 3.13(b)]. Metoda Aliasing šíbr v tomto případě nedetekuje, avšak ani samotnou vadu [výstupní snímek A_{BW} na obr. 3.13(c)]



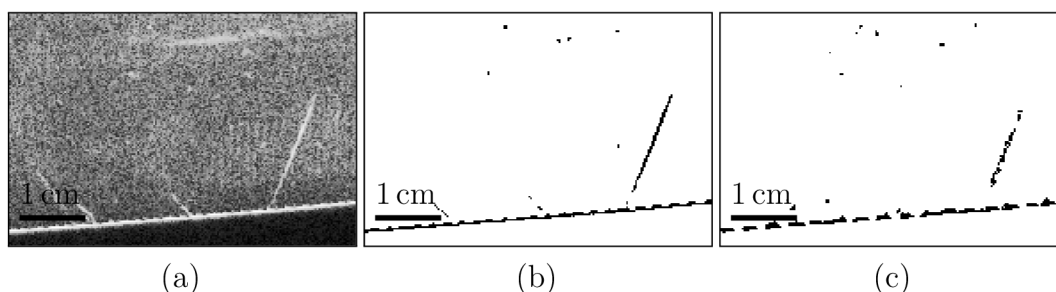
Obrázek 3.13: Dekorativní vada šlíra v čiré na výřezech (a) šedého snímku M s ekvalizovaným histogramem, (b) výstupního snímku M_{BW} metody Maximum a (c) výstupního snímku A_{BW} metody Aliasing.

Pro metody Variance a Grey je vada rovněž neviditelná, jak napovídá vstupní snímek obou metod [obr. 3.14(a)]. Proto na výstupních snímcích V_{BW} , G_{BW} [obr. 3.14(b), (c)] nepozorujeme v oblasti skla ani jeden černý pixel.



Obrázek 3.14: Dekorativní vada šlíra v čiré na výřezech (a) vstupního snímku metod Variance a Grey s ekvalizovaným histogramem a výstupních snímcích (b) V_{BW} metody Variance a (c) G_{BW} metody Grey.

Zcela opačná situace se odehrává v případě metod Saturace a Rotace. Na obou výstupních snímcích S_{BW} , R_{BW} [obr. 3.15(b), (c)] je vada šlíra v čiré jasně viditelná. Všimněme si velmi slabého projevu šíbru na obou snímcích. To je dáno jeho natočením vůči směru vertikálních proužků binárního obrazce. S menším úhlem svíraným mezi nimi dochází k výraznější deformaci proužků na snímku kamery, s čímž mají metody problém se vypořádat. V podstatě by bylo nutné v tomto případě zvolit jiný binární obrazec úměrně s deformací proužků na šíbru. Zřetelně to lze spatřit při srovnání šedých snímků S metody Saturace [obr. 3.3(a) a 3.15(a)], kde jsou odlišné úhly mezi šíbrem a vertikálními proužky binárního obrazce, a tím je i odlišná odezva na šíbr ve výstupních snímcích [obr. 3.3(b) a 3.15(b)].



Obrázek 3.15: Dekorativní vada šlíra v čiré na výřezech (a) šedého snímku S s ekvalizovaným histogramem, (b) výstupního snímku S_{BW} metody Saturace a (c) výstupního snímku R_{BW} metody Rotace.

3.1.6 Shrnutí a kompozitní výstupní snímky

Výsledky detekce dekorativních vad na sklech krytů světlometů na základě uvedených výstupních snímků shrnuje tabulka 3.1. Symbol „✓“ značí úspěšnou detekci vady a symbol „✗“ nulovou detekci. Pokud je u symbolu otazník, znamená to nejistotu v tvrzení daném symbolem. Například u metody Maximum ve sloupci šíbr je buňka tabulky uvedeno „✓?“, což znamená, že metoda šíbr detekuje, ale ne vždy a jasně. Nemůžeme naopak tvrdit, že její metoda vůbec, nebo téměř vůbec nedetekuje jako naopak metoda Aliasing.

Tabulka 3.1: Srovnání metod a jejich schopností detekce různých vad a vlivů.

-	voda na skle	matný	zástřík	cizí tělísko	šlára v čiré	černý fix	šíbr	fluktuační laku
Maximum	✓	✗	✗	✓	✗	✓	✓?	✓?
Aliasing	✓	✗	✗	✓	✗	✓	✗?	✓?
Variance	✓	✗	✗	✓	✗	✓	✗	✗
Grey	✓	✗	✗	✓	✗?	✓	✓	✗
Saturace	✗	✓	✓	✓	✓	✓	✓?	✗?
Rotace	✗	✓	✓	✓	✓	✓	✓	✗

Výstupem senzoru kromě výstupních snímků metod je i kompozitní výstupní snímek (příloha B Kompozitní výstupní snímky). Ten má celkem sedm úrovní intenzity pixelů. Maximální intenzita pixelu (hodnota 255) reprezentuje nenalezení žádné vady v místě tohoto pixelu. Nalezlo-li vadu v místě stejného pixelu více metod, potom je hodnota intenzity 255 pixelu ponížena o hodnotu intenzity 40 právě tolikrát, kolik metod ji v tomto místě detekovalo. Hodnota pixelu má potom intenzitu $(255 - 40m)$, kde m je počet metod detekujících vadu v místě stejného pixelu. Kompozitní výstupní snímky jsou v příloze C Ostatní na příloženém CD.

3.2 Diskuze

V detekci dekorativních vad na průhledných částech krytů světlometů si metody Saturace a Rotace vedou ze všech šesti metod nejlépe. Jsou schopné lokalizovat čtyři z pěti vad. Na druhou stranu jako vadu detekují i zlom mezi plochami na skle, tzv. šibr.

Pokud bychom chtěli rozlišovat fluktuace laku (výrazné změny tloušťky tenké vrstvy laku na skle, které mohou být považovány za vadu), potom k tomu nejsou tyto metody vhodné. Obdobu těchto metod založených na delší expozici kamery uvádí jako součást senzoru pro detekci vad Satorres^[6]. Tento senzor s robotickou paží pracuje také s rovnoměrným osvětlením za krytem světlometu^[6] podobně jako metoda Grey a Variance. Avšak metody Grey a Variance snímají binární obrazec zobrazený na LCD panelu, a proto se některé vady (voda na skle) projevují na vstupním snímku specifickým způsobem (obr. 2.28). S rovnoměrným osvětlením na LCD panelu pracuje také Martínez^[3]. Jiný přístup volí Šmíd^[48], kdy kromě jiných zdrojů světla je také použita žárovka, jejíž světlo je fokusováno a prochází sklem do obrazového ohniska čočky. Tím dochází k osvětlení skla. Plocha skla s vadou je zobrazena druhou čočkou na matnici, kterou snímá kamera. Vstupní snímek pořízený kamerou je podobný snímkům pořízených s rovnoměrných osvětlením a studované vady jsou dobře viditelné. Metoda Grey přes svoji jednoduchost detekuje vadu na skle a také cizí tělísko. Pokud bychom zvolili slabší podmínku pro detekci vady [vztah (2.40)], byla by metoda schopná nalézt i výrazné fluktuace tloušťky laku.

Metoda Variance jako jediná jistě nedetekuje šibr, což je důležité. Přestože lze sklo senzorem analyzovat z různých úhlů např. s využitím robotické paže^[6], nachází-li se vada v místě šíbru, je pro ostatní metody velmi obtížné tuto vadu odlišit od detekovaného šíbru. Pokud se na výstupním snímku takové metody vada společně se šibrem objevuje, může ji právě metoda Variance potvrdit. Do jisté míry šibr nedetekuje ani metoda Aliasing. Obecně je detekce v oblasti šíbru obtížná, protože jde o zlom mezi plochami a směr dráhy paprsků ze světelného zdroje se zde prudce mění.

Výstupní snímky metody Aliasing a Maximum jsou si velmi podobné. Počet pixelů detekované vady je v případě metody Maximum nižší, navíc metoda výrazně detekuje linii šíbru. Ve srovnání s metodou Aliasing si vede hůře a v rámci senzoru není užitečná.

Podobně můžeme hovořit o metodě Rotace. Nicméně, připomeňme si závislost některých vad na směru proužků binárního obrazce (obr. 2.37). Pokud by metoda Rotace byla ze senzoru vyjmuta, bylo by nutné metodu Saturace rozšířit právě o rotaci binárního obrazce.

Výstupem senzoru kromě výstupních snímků metod je i kompozitní výstupní snímek (příloha B Kompozitní výstupní snímky). Na jeho základě je teoreticky možné odhadnout typ vady podle toho, kolik metod ji detekuje (tab. 3.1).

Pokud by měl autor k dispozici jiné kryty světlometů s jinak umístěnými dekorativními vadami, případně skla s jiným tvarem a jinou kvalitou laku na povrchu, potom lze předpokládat, že by během této práce vznikly do nějaké míry odlišné metody s jinou úspěšností detekce vad. Toto je třeba brát v potaz a uvědomit si, že senzor je nutné adaptovat na konkrétní podmínky. Tvar krytu světlometu je komplikovaný a důležitost detekce jednoho typu vady nemusí být všude na povrchu skla stejná. Potom je možné upravit parametry metod v závislosti na analyzovaném místě na skle. K jiným závěrům by se pravděpodobně došlo, pokud by autor věnoval více pozornosti kvalitě skel. Na nich je totiž zanecháno mnoho škrábanců, mastnoty a prachu vlivem manipulace s nimi. Následkem je nižší citlivost metod a možná i špatný směr jejich vývoje. Autor se domnívá, že potenciální užitečnost senzoru lze vyhodnotit na větších

souborech vzorků srovnáním s nynějšími manuálními postupy nebo při analýze skel kvalitativní analýzou (pravdivě/falešně pozitivní, falešně negativní^[4]).

Další postup by se mohl věnovat optimalizaci metod. Některé metody lze ze senzoru vynechat a jiné zdokonalit. Dále je otázkou závislost dekorativních vad na úhlu pozorování vůči normále plochy skla. Různé typy vad se mohou lišit a tímto způsobem by se možná daly rozeznat.

Zajímavé by bylo srovnání senzoru představeném v této práci s jiným senzorem na stejné sadě vzorků, např. s těmi, které používají filtry obrazové analýzy ve větší míře^[3]^[4]. Také by namísto vysoce kontrastního obrazce bylo zajímavé snímat obrazec s pozvolnou změnou intenzity (např. sinusový průběh^[4]) s difuzérem umístěným před LCD panelem. Zpracováním vstupního snímku by se generovala fázová mapa, kterou by bylo nutné rozbalit a analyzovat^[4]^[49]. Mezi mřížkami pixelů LCD panelu a CMOS snímače by nedocházelo k jevu moaré a sensor by nebyl tak citlivý na vzdálenost mezi kamerou a předmětem. Otázkou je, jaké rozměrové a intenzitní citlivosti by tyto přístupy dosáhly ve srovnání mezi sebou.

Závěr

Úkolem práce bylo sestavit senzor pro optickou detekci dekorativních vad na transparentních částech krytů světlometů – sklo. Byl navržen program `pythonMattice` v jazyce `python`, který komunikoval s kamerou, vykresloval binární obrazce na LCD panelu, pořizoval a ukládal snímky. Uměl také nasnímat a zpracovat snímky pro korekci obrazu metodou FFC.

Senzor si s využitím tohoto programu automaticky vykresloval binární obrazce a pořizoval jeho snímky přes sklo krytu světlometu. Tyto snímky následně korigoval metodou FFC a z nich poté analyzoval lokální deformace binárního obrazce celkem šesti navrženými metodami v jazyce `python`. Metody si v detekci každé z pěti vad vedly odlišně. Celkově však metody lokalizovaly všechny vady. Senzor po analýze vytvořil výstupní snímky metod a také kompozitní výstupní snímek, který je jejich kombinací a nese komplexní informaci o analyzovaném krytu světlometu.

V práci bylo podrobně rozepsáno nastavení senzoru s ohledem na optimalizaci zobrazení binárního obrazce kamerou a na kontrast pořízeného snímku. Za tímto účelem bylo vytvořeno několik menších programů v jazyce `python`, které usnadňovaly proces nastavení senzoru.

Bylo by zajímavé se dále zabývat skutečnými limity rozměrové a intenzitní citlivosti senzoru proto, že se tato práce pečlivě věnovala maximalizaci kontrastu binárního obrazce a přesnému zobrazení mřížky pixelů LCD panelu na CMOS snímač kamery. Povrchy skel byly totiž manipulací znečištěny a metody byly vyvíjeny na těchto površích. Pokud by byla k dispozici čistá skla, tzn. vady na nich by vznikly pouze během procesu jejich výroby, potom by se ukázalo skutečné omezení navržených metod.

Seznam použitých zdrojů

- [1] MAHMUD, Tanjim, Juel SIKDER, Rana Jyoti CHAKMA a Jannat FARDOUSH. Fabric Defect Detection System. VASANT, Pandian, Ivan ZELINKA a Gerhard-Wilhelm WEBER, ed. *Intelligent Computing and Optimization* [online]. Cham: Springer International Publishing, 2021, 2021-02-08, s. 788-800 [cit. 2022-01-10]. Advances in Intelligent Systems and Computing. ISBN 978-3-030-68153-1. Dostupné z: doi:10.1007/978-3-030-68154-8_68
- [2] LIU, Qiang, Chuan WANG, Yusheng LI, Mingwang GAO a Jingao LI. A fabric defect detection method based on deep learning. *IEEE Access.* , 1-1. ISSN 2169-3536. Dostupné z: doi:10.1109/ACCESS.2021.3140118
- [3] MARTÍNEZ, S. Satorres, J. Gómez ORTEGA, J. Gámez GARCÍA, A. Sánchez GARCÍA a E. Estévez ESTÉVEZ. An industrial vision system for surface quality inspection of transparent parts. *The International Journal of Advanced Manufacturing Technology* [online]. 2013, **68**(5-8), 1123-1136 [cit. 2022-01-10]. ISSN 0268-3768. Dostupné z: doi:10.1007/s00170-013-4904-2
- [4] TANDIYA, Adarsh, Sohail AKTHAR, Medhat MOUSSA a Cole TARRAY. Automotive Semi-specular Surface Defect Detection System. In: *2018 15th Conference on Computer and Robot Vision (CRV)*. IEEE, 2018, 2018, s. 285-291. ISBN 978-1-5386-6481-0. Dostupné z: doi:10.1109/CRV.2018.00047
- [5] MING, Wuyi, Fan SHEN, Xiaoke LI, Zhen ZHANG, Jinguang DU, Zhijun CHEN a Yang CAO. A comprehensive review of defect detection in 3C glass components. *Measurement* [online]. 2020, **158** [cit. 2022-01-10]. ISSN 02632241. Dostupné z: doi:10.1016/j.measurement.2020.107722
- [6] SATORRES, Silvia, Juan GOMEZ, Javier GAMEZ a Alejandro SANCHEZ. A Machine Vision for Automated Headlamp Lens Inspection. FRANCISCO, ed. *Vision Sensors and Edge Detection* [online]. Sciyo, 2010, 2010-08-12 [cit. 2022-01-11]. ISBN 978-953-307-098-8. Dostupné z: doi:10.5772/10134
- [7] JIN, Rui, Qiang NIU a Paolo SPAGNOLO. Automatic Fabric Defect Detection Based on an Improved YOLOv5. *Mathematical Problems in Engineering* [online]. 2021, 2021, 1-13 [cit. 2022-01-10]. ISSN 1563-5147. Dostupné z: doi:10.1155/2021/7321394
- [8] WÜRSCHINGER, Hubert, Matthias MÜHLBAUER, Michael WINTER, Michael ENGELBRECHT a Nico HANENKAMP. Implementation and potentials of a machine vision system in a series production using deep learning and low-cost hardware. *Procedia CIRP* [online]. 2020, **90**, 611-616 [cit. 2022-01-11]. ISSN 22128271. Dostupné z: doi:10.1016/j.procir.2020.01.121

- [9] SOBEL, Irwin. An Isotropic 3x3 Image Gradient Operator. *Presentation at Stanford A.I. Project 1968*. February 2014.
- [10] AMIDROR, Isaac. *The Theory of the Moiré Phenomenon: Volume I: Periodic Layers*. London: Springer London, 2009, 529 s. ISBN 9781848821811. 2009921442.
- [11] REID, G.T. Moiré fringes in metrology. *Optics and Lasers in Engineering*. 1984, **5**(2), 63-93. ISSN 01438166. Dostupné z: doi:10.1016/0143-8166(84)90012-5
- [12] KING, M. C. a D. H. BERRY. Photolithographic Mask Alignment Using Moiré Techniques. *Applied Optics*. 1972, **11**(11). ISSN 0003-6935. Dostupné z: doi:10.1364/AO.11.002455
- [13] PALLEK, D., K.A. BUTEFISCH, J. QUEST a W. STRUDTHOFF. Model deformation measurement in ETW using the Moire technique. *20th International Congress on Instrumentation in Aerospace Simulation Facilities, 2003. ICIASF '03*. IEEE, 2003, , 110-114. ISBN 0-7803-8149-1. Dostupné z: doi:10.1109/ICIASF.2003.1274859
- [14] RI, S., M. FUJIGAKI a Y. MORIMOTO. Sampling Moiré Method for Accurate Small Deformation Distribution Measurement. *Experimental Mechanics*. 2010, **50**(4), 501-508. ISSN 0014-4851. Dostupné z: doi:10.1007/s11340-009-9239-4
- [15] LEE, Sang Won, Ryungkee CHANG a Sucbei MOON. Optimized Optomechanical Anti-Aliasing Filter for Digital Camera Photography. *Journal of the Optical Society of Korea* [online]. 2015, **19**(5), 456-466 [cit. 2022-01-16]. ISSN 1226-4776. Dostupné z: doi:10.3807/JOSK.2015.19.5.456
- [16] SCHÖBERL, Michael, Francisco H. IMAI, Feng XIAO, Jürgen ERNST, Wolfgang SCHNURRER, Siegfried FÖSSEL a André KAUP. *Moving refractive optical low pass filter for digital cameras*. 2011-1-23, , 78760Z-. Dostupné z: doi:10.1117/12.872402
- [17] SHANNON, C.E. Communication in the Presence of Noise. *Proceedings of the IRE* [online]. 1949, **37**(1), 10-21 [cit. 2022-01-16]. ISSN 0096-8390. Dostupné z: doi:10.1109/JRPROC.1949.232969
- [18] BAJER, Jiří. *Optika 1*. Olomouc: Univerzita Palackého v Olomouci, 2018. ISBN 978-80-244-4532-8.
- [19] MALÝ, Petr. *Optika*. Vyd. 2., přepřac. Praha: Karolinum, 2013. ISBN 978-80-246-2246-0.
- [20] KIM, Hyang-Yul, Yong-Je JEON, Jeoung-Yeon HWANG, Jae-Hyung KIM a Dae-Shik SEO. Residual DC characteristics in the IPS-LCD by capacitance-voltage hysteresis method on a polymer layer. *Current Applied Physics*. 2002, **2**(3), 237-240. ISSN 15671739. Dostupné z: doi:10.1016/S1567-1739(02)00094-9
- [21] BHOWMIK, Achintya K., Zili LI a Philip J. BOS. *Mobile Displays: Technology and Applications*. John Wiley, 2008, 650 s. ISBN 9780470994634.
- [22] OH-E, MASAHITO, MAKOTO YONEYA, MASUYUKI OHTA a KATSUMI KONDO. Dependence of viewing angle characteristics on pretilt angle in the in-plane switching mode. *Liquid Crystals*. 2010, **22**(4), 391-400. ISSN 0267-8292. Dostupné z: doi:10.1080/026782997209108

- [23] HONG, Hyunki, Hyunho SHIN a InJae CHUNG. In-Plane Switching Technology for Liquid Crystal Display Television. *Journal of Display Technology*. January 2008, **3**(4), 361 - 370. ISSN 1551-319X. Dostupné z: doi:10.1109/JDT.2007.901562
- [24] ČERNOCH, Antonín a Jan SOUBUSTA. *Detekce světla*. Olomouc: Univerzita Palackého v Olomouci, 2020. ISBN 978-80-244-5706-2.
- [25] EV76C570 2 Mpixels B&W and Color CMOS Image sensor: Datasheet. *1stVision* [online]. 29/03/2013 [cit. 2022-01-05]. Dostupné z: https://www.1stvision.com/cameras/sensor_specs/DSC_EV76C570.pdf
- [26] GOW, R.D., D. RENSHAW, K. FINDLATER, L. GRANT, S.J. MCLEOD, J. HART a R.L. NICOL. A Comprehensive Tool for Modeling CMOS Image-Sensor-Noise Performance. *IEEE Transactions on Electron Devices*. 2007, **54**(6), 1321-1329. ISSN 0018-9383. Dostupné z: doi:10.1109/TED.2007.896718
- [27] LOUKIANOVA, N.V., H.O. FOLKERTS, J.P.V. MAAS, D.W.E. VERBUGT, A.J. MIEROP, W. HOEKSTRA, E. ROKS a A.J.P. THEUWISSEN. Leakage current modeling of test structures for characterization of dark current in CMOS image sensors. *IEEE Transactions on Electron Devices*. 2003, **50**(1), 77-83. ISSN 0018-9383. Dostupné z: doi:10.1109/TED.2002.807249
- [28] Non-linear dark current fixed pattern noise compensation for variable frame rate moving picture cameras. In: SCHÖBERL, Michael, Cihan SENEL, Siegfried FÖSSEL, Hans BLOSS a André KAUP. *2009 17th European Signal Processing Conference* [online]. 2009, s. 268-272 [cit. 2022-01-09].
- [29] VAN NIEUWENHOVE, Vincent, Jan DE BEENHOUWER, Francesco DE CARLO, Lucia MANCINI, Federica MARONE a Jan SIJBERS. Dynamic intensity normalization using eigen flat fields in X-ray imaging. *Optics Express*. 2015, **23**(21). ISSN 1094-4087. Dostupné z: doi:10.1364/OE.23.027975
- [30] WILLIAMS, Timothy C. a Christopher R. SHADDIX. Simultaneous correction of flat field and nonlinearity response of intensified charge-coupled devices. *Review of Scientific Instruments*. 2007, **78**(12). ISSN 0034-6748. Dostupné z: doi:10.1063/1.2821616
- [31] RAY, Sidney F. *Applied Photographic Optics* [online]. Routledge, 2002 [cit. 2022-01-11]. ISBN 9780080499253. Dostupné z: doi:10.4324/9780080499253
- [32] Evaluation of measurement data - Guide to the expression of uncertainty in measurement, JCGM 100:2008, GUM 1995 with minor corrections.
- [33] International vocabulary of Metrology - Basic and general concepts and associated terms (VIM), JCGM 200:2008.
- [34] Pokyn pro vyjadřování nejistoty měření (GUM). Sborníky technické harmonizace, ÚNMZ, Praha, 2012. www.unmz.cz/les/SbornikyTH/GUM-celek-DEF.pdf
- [35] LUDVÍK, Vladimír. *Nejistoty měření, přesnost měření, správnost měření a otázky spojené se vzájemnou porovnatelností výsledků měření a s prohlášením o shodě s technickými specifikacemi.: Sborníky technické dokumentace*. Praha: Q-art, Praha., 2005, 68 s.

- [36] SCHOVÁNEK, Petr a Vítězslav HAVRÁNEK. *Chyby a nejistoty měření: doplňující text k laboratornímu cvičení*.
- [37] ŽLEBEK, M. *Osvětlení automobilu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013. 79 s. Vedoucí bakalářské práce doc. Ing. Josef Štětina, Ph.D.
- [38] AOC I2460PXQU 24 inch (61 cm) LCD IPS MONITOR: Datasheet. *Icccat* [online]. [cit. 2022-01-05]. Dostupné z: https://objects.icccat.biz/objects/mmo_20085751_1604932042_8505_26084.pdf
- [39] Economical Backlight FR-BL89x155: datasheet. *Metaphase Technologies* [online]. September 2018 [cit. 2022-01-09]. Dostupné z: https://www.metaphase-tech.com/wp-content/uploads/2018/03/FR-BL_PDS-09-21-2018-1-1.pdf
- [40] AcA1600-60gm: datasheet. *QATTA* [online]. 2018/2/9 [cit. 2022-01-09]. Dostupné z: <http://www.qataa.com/upLoad/file/20180209/15181616387428755.pdf>
- [41] Basler pylon Camera Software Suite: Overview of the pylon Viewer. *Basler* [online]. [cit. 2022-01-16]. Dostupné z: <https://docs.baslerweb.com/overview-of-the-ylon-viewer>
- [42] Basler pylon Camera Software Suite: pypylon. *Github* [online]. [cit. 2022-01-16]. Dostupné z: <https://github.com/basler/pypylon>
- [43] Xenoplan 1.4/23: datasheet. *Schneider Kreuznach* [online]. 10/2018 [cit. 2022-01-09]. Dostupné z: https://schneiderkreuznach.com/application/files/1315/4115/0849/1001917_Xenoplan_1-4_23.pdf
- [44] Sill Optics Compact S5LPJ1199 - Correctal® T60/0.2D telecentric lens with c-mount: Download manual. *Manualzz: the universal manuals library* [online]. [cit. 2022-01-05]. Dostupné z: <https://manualzz.com/download/10202175>
- [45] ITTIYAVIRAH, Sibi, S. JONES a P. SIDDARTH. Analysis of different activation functions using Backpropagation Neural Networks. In: *Journal of Theoretical and Applied Information Technology*. 2013/01/01. 2013, s. 1344-1348. ISSN 1992-8645.
- [46] COLIN CAMERON, A. a Frank A.G. WINDMEIJER. An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics* [online]. 1997, **77**(2), 329-342 [cit. 2022-01-16]. ISSN 03044076. Dostupné z: doi:10.1016/S0304-4076(96)01818-0
- [47] CHICCO, Davide, Matthijs J. WARRENS a Giuseppe JURMAN. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science* [online]. 2021, **7** [cit. 2022-01-16]. ISSN 2376-5992. Dostupné z: doi:10.7717/peerj-cs.623
- [48] ŠMÍD, P., HAVRÁNEK, V., PAVLÍČEK, P., IVANOV, G. *Detekce dekorativních vad krytů předních světlometů*. Zpráva SLO UP a FZÚ AV ČR, Olomouc, 2018.
- [49] STANKE, Ladislav. *Elektronická interferometrie na bázi koherenční zrnitosti a její aplikace*. 2013, 105 s. Diplomové práce. Univerzita Palackého, Katedra optiky. Vedoucí práce Petr Šmíd.

Seznam použitých symbolů a zkratek

A_{BW}	výstupní snímek metody Aliasing
a, a'	předmětová, obrazová vzdálenost
B	elementární matice binárního obrazce
c	clonové číslo
C	korigovaný snímek
$\Delta y'_0$	tolerovaná velikost obrazu
Δ_g	hloubka ostrosti
$D_{\text{Max}}, D_{\text{Ali}}$	parametr metod Maximum a Aliasing
$D_{\text{Var}}, D_{\text{Gre}}$	parametr metod Variance a Grey
$d_{\text{papír}}$	tloušťka optického papíru
Δ_{pm}	rozteč čelistí posuvného měřítka
Δ'_g	mezní hloubka ostrosti
D, D'	průměr vstupní, výstupní pupily
d_c	délka celistvých period
D^{raw}	temný snímek
\vec{E}	elektrická intenzita
f_v	vzorkovací frekvence
f_{max}	nejvyšší frekvence
f, f'	předmětová, obrazová ohnisková vzdálenost
F_{Rot}	parametr metody Rotace
F^{raw}	snímek rovnoměrného osvětlení
\bar{f}	normalizovaný snímek za rovnoměrného osvětlení se středními hodnotami
G	hyperfokální vzdálenost
Γ	konstantní příčné zvětšení teleobjektivu
g	předmětová vzdálenost vztažená ke vstupní pupile
g_1, g_2	krajní meze hloubky ostrosti vztažené ke vstupní pupile
G_{BW}	výstupní snímek metody Grey
HH'	vzdálenost mezi předmětovým a obrazovým hlavním bodem
H_x, H_y	Sobelovy masky
I	intenzita
I_n	normovaná intenzita
$I_{\text{min}}, I_{\text{max}}$	minimální, maximální intenzita
ID	interval délek
K	kontrast
k_p	vzdálenost hrany přední objímky objektivu k přední optické ploše
k_z	vzdálenost hrany zadní objímky objektivu k zadní optické ploše
k_{pz}	vzdálenost mezi hranami přední a zadní objímky objektivu

l	vzdálenost předmětu od hrany objímky
l'	vzdálenost od hrany objímky k předmětové hlavní rovině
L_{Sat}	parametr metody Saturace
$L_{\text{Rot}}, U_{\text{Rot}}$	parametry metody Rotace
m	příčné zvětšení
M_{BW}	výstupní snímek metody Maximum
μ	střední hodnota intenzity
M	šedý snímek metody Maximum
$n_{\mathbb{R}}$	neceločíselný násobek
NA	numerická apertura
OD	optická hustota
PP, PP_c	počet period, počet celistvých period
R^2	koeficient determinace
R_{BW}	výstupní snímek metody Rotace
R	snímek před korekcí
\tilde{s}^2	pseudorozptyl
s^2	rozptyl
S_{BW}	výstupní snímek metody Saturace
S	šedý snímek metody Saturace
$s_F, s'_{F'}$	poloha předmětového, obrazového ohniskového bodu
$\sum d$	celková optická délka
T	prostorová perioda
T_{px}	příčná velikost obrazu pixelu LCD panelu na CMOS snímači
τ	expoziční doba kamery
v	výrok
VH	vzdálenost mezi zadní optickou plochou a předmětovým hlavním bodem
V_{BW}	výstupní snímek metody Variance
$y_{\text{LCD}}, y_{\text{CMOS}}$	příčná velikost pixelu LCD panelu, CMOS snímače
y_d	příčná velikost pixelu detektoru
y, y'	příčná poloha předmětu, obrazu

Příloha A

Metody

A.1 Metoda Maximum

```
1  #!/usr/bin/python3
2  import tkinter as tk
3  import numpy as np
4  import sys
5  import imageio
6  import scipy.misc
7  from PIL import ImageTk, Image
8  import PIL
9  import os
10 import datetime
11 import math
12 =====
13 # Predprogram - vypocita parametr DMax
14 =====
15 array_1 = np.array(PIL.Image.open("./data/00/00_ver_a.png").convert("L"))
16 array_2 = np.array(PIL.Image.open("./data/00/00_ver_b.png").convert("L"))
17 Width, Height = len(array_1[0]), len(array_1)
18 arrayMaximum = np.maximum(array_1, array_2)
19 DMax = 0
20
21 while (1):
22     arrayMaximum_BW = np.array([np.array([255 for i in range(Width) ],
23                                     dtype=np.uint8) for j in range(Height) ], dtype=np.uint8)
24
25     for i in range(0+1, Height-1):
26         for j in range(0+1, Width-1):
27             PixelCenter = int(arrayMaximum[i][j])
28             PixelUp = int(arrayMaximum[i-1][j])
29             PixelDown = int(arrayMaximum[i+1][j])
30             PixelLeft = int(arrayMaximum[i][j-1])
31             PixelRight = int(arrayMaximum[i][j+1])
32
33             DifferenceUp = abs(PixelCenter - PixelUp)
34             DifferenceDown = abs(PixelCenter - PixelDown)
35             DifferenceLeft = abs(PixelCenter - PixelLeft)
36             DifferenceRight = abs(PixelCenter - PixelRight)
37
38             if DifferenceUp > DMax or DifferenceDown > DMax or
39                 DifferenceLeft > DMax or DifferenceRight > DMax:
40                 arrayMaximum_BW[i][j] = 0
```

```

40
41     for i in range(0, Height):
42         for j in range(0, Width):
43             if arrayMaximum_BW[i][j] == 0:
44                 SumBW = SumBW + 1
45     if SumBW > 0 :
46         DMax = DMax + 1
47     else:
48         print(" Parametr DMax:" , DMax)
49         break
50     print("SumBW:" , SumBW, "\t DMax:" , DMax)
51     =====
52     # Hlavni cast programu – prace s obrazky s vadami
53     =====
54     Files = ("03" , "05" , "10" , "12" , "13")
55
56     for f in range(len(Files)):
57         array_1 = np.array(PIL.Image.open("./data/{0}/{0}_ver_a.png".format(
58             Files[f])).convert("L"))
59         array_2 = np.array(PIL.Image.open("./data/{0}/{0}_ver_b.png".format(
60             Files[f])).convert("L"))
61         Width, Height = len(array_1[0]), len(array_1)
62         arrayMaximum = np.array([np.array([0 for i in range(Width) ], dtype=
63             np.uint8) for j in range(Height) ], dtype=np.uint8)
64         arrayMaximum_BW = np.array([np.array([255 for i in range(Width) ],
65             dtype=np.uint8) for j in range(Height) ], dtype=np.uint8)
66         arrayMaximum = np.maximum(array_1 , array_2)
67
68         for i in range(0+1, Height-1):
69             for j in range(0+1, Width-1):
70                 PixelCenter = int(arrayMaximum[i][j])
71                 PixelUp = int(arrayMaximum[i-1][j])
72                 PixelDown = int(arrayMaximum[i+1][j])
73                 PixelLeft = int(arrayMaximum[i][j-1])
74                 PixelRight = int(arrayMaximum[i][j+1])
75
76                 DifferenceUp = abs(PixelCenter - PixelUp)
77                 DifferenceDown = abs(PixelCenter - PixelDown)
78                 DifferenceLeft = abs(PixelCenter - PixelLeft)
79                 DifferenceRight = abs(PixelCenter - PixelRight)
80
81                 if DifferenceUp > DMax or DifferenceDown > DMax or
82                 DifferenceLeft > DMax or DifferenceRight > DMax:
83                     arrayMaximum_BW[i][j] = 0
84         imageMaximum = Image.fromarray(arrayMaximum)
85         imageMaximum.save('./metoda_Maximum/Maximum_{0}.png'.format(Files[f]))
86     imageMaximum_BW = Image.fromarray(arrayMaximum_BW)
87     imageMaximum_BW.save('./metoda_Maximum/Maximum_{0}.BW.png'.format(
88         Files[f]))

```

A.2 Metoda Aliasing

```

1 #!/usr/bin/python3
2 import tkinter as tk
3 import numpy as np
4 import sys
5 import imageio
6 import scipy.misc

```

```

7 from PIL import ImageTk, Image
8 import PIL
9 import os
10 import datetime
11 import math
12 #=====
13 # Predprogram – vypocita parametr DAli
14 #=====
15 array_1 = np.array(PIL.Image.open("./data/00/00_ver_a.png").convert("L"))
16 array_2 = np.array(PIL.Image.open("./data/00/00_ver_b.png").convert("L"))
17 Width, Height = len(array_1[0]), len(array_1)
18 arrayAliasing = np.array([np.array([0 for i in range(Width) ], dtype=np.
    uint8) for j in range(Height) ], dtype=np.uint8)
19
20 for i in range(0, Height):
21     for j in range(0, int(Width/2)):
22         Pixel_1 = int(array_1[i][2*j])
23         NextPixel_1 = int(array_1[i][2*j+1])
24         Pixel_2 = int(array_2[i][2*j])
25         NextPixel_2 = int(array_2[i][2*j+1])
26
27         Value_1 = round((Pixel_1 + NextPixel_1)/2)
28         arrayAliasing[i][2*j] = Value_1
29         Value_2 = round((Pixel_2 + NextPixel_2)/2)
30         arrayAliasing[i][2*j+1] = Value_2
31 DAli = 0
32
33 while (1):
34     arrayAliasing_BW = np.array([np.array([255 for i in range(Width) ],
    dtype=np.uint8) for j in range(Height) ], dtype=np.uint8)
35
36     for i in range(0+1, Height-1):
37         for j in range(0+1, int(Width/2)-1):
38             ValueCenter_1 = int(arrayAliasing[i][2*(j)])
39             ValueLeft_1 = int(arrayAliasing[i][2*(j-1)])
40             ValueRight_1 = int(arrayAliasing[i][2*(j+1)])
41             ValueUp_1 = int(arrayAliasing[i-1][2*(j)])
42             ValueDown_1 = int(arrayAliasing[i+1][2*(j)])
43
44             DifLeft_1 = abs(ValueCenter_1-ValueLeft_1)
45             DifRight_1 = abs(ValueCenter_1-ValueRight_1)
46             DifUp_1 = abs(ValueCenter_1-ValueUp_1)
47             DifDown_1 = abs(ValueCenter_1-ValueDown_1)
48
49             if DifLeft_1 > DAli or DifRight_1 > DAli or DifUp_1 > DAli
    or DifDown_1 > DAli :
50                 ValueCenter_2 = int(arrayAliasing[i][2*(j)+1])
51                 ValueLeft_2 = int(arrayAliasing[i][2*(j-1)+1])
52                 ValueRight_2 = int(arrayAliasing[i][2*(j+1)+1])
53                 ValueUp_2 = int(arrayAliasing[i-1][2*(j)+1])
54                 ValueDown_2 = int(arrayAliasing[i+1][2*(j)+1])
55
56                 DiffLeft_2 = abs(ValueCenter_2-ValueLeft_2)
57                 DiffRight_2 = abs(ValueCenter_2-ValueRight_2)
58                 DiffUp_2 = abs(ValueCenter_2-ValueUp_2)
59                 DiffDown_2 = abs(ValueCenter_2-ValueDown_2)
60
61                 if DiffLeft_2 > DAli or DiffRight_2 > DAli or DiffUp_2 >
    DAli or DiffDown_2 > DAli :

```

```

62         arrayAliasing_BW[i][2*j] = 0
63         arrayAliasing_BW[i][2*j+1] = 0
64     SumBW = 0
65
66     for i in range(0, Height):
67         for j in range(0, int(Width/2)):
68             if arrayAliasing_BW[i][2*j] == 0:
69                 SumBW = SumBW + 1
70     if SumBW > 0 :
71         DAli = DAli + 1
72     else:
73         print(" Parametr_DAli:" , DAli)
74         break
75     print("SumBW:" , SumBW, "\t_DAli:" , DAli)
76     =====
77     # Hlavni cast programu – prace s obrazky s vadami
78     =====
79     Files = ("03" , "05" , "10" , "12" , "13")
80
81     for f in range(len(Files)):
82         array_1 = np.array(PIL.Image.open("./data/{0}/{0}_ver_a.png".format(
83             Files[f])).convert("L"))
84         array_2 = np.array(PIL.Image.open("./data/{0}/{0}_ver_b.png".format(
85             Files[f])).convert("L"))
86         Width, Height = len(array_1[0]) , len(array_1)
87         arrayAliasing = np.array([np.array([0 for i in range(Width) ] , dtype=
88             np.uint8) for j in range(Height) ] , dtype=np.uint8)
89         arrayAliasing_BW = np.array([np.array([255 for i in range(Width) ] ,
90             dtype=np.uint8) for j in range(Height) ] , dtype=np.uint8)
91
92         for i in range(0, Height):
93             for j in range(0, int(Width/2)):
94                 Pixel_1 = int(array_1[i][2*j])
95                 NextPixel_1 = int(array_1[i][2*j+1])
96                 Pixel_2 = int(array_2[i][2*j])
97                 NextPixel_2 = int(array_2[i][2*j+1])
98
99                 Value_1 = round((Pixel_1 + NextPixel_1)/2)
100                arrayAliasing[i][2*j] = Value_1
101                Value_2 = round((Pixel_2 + NextPixel_2)/2)
102                arrayAliasing[i][2*j+1] = Value_2
103
104         for i in range(0+1, Height-1):
105             for j in range(0+1, int(Width/2)-1):
106                 ValueCenter_1 = int(arrayAliasing[i][2*(j)])
107                 ValueLeft_1 = int(arrayAliasing[i][2*(j-1)])
108                 ValueRight_1 = int(arrayAliasing[i][2*(j+1)])
109                 ValueUp_1 = int(arrayAliasing[i-1][2*(j)])
110                 ValueDown_1 = int(arrayAliasing[i+1][2*(j)])
111
112                 DiffLeft_1 = abs(ValueCenter_1-ValueLeft_1)
113                 DiffRight_1 = abs(ValueCenter_1-ValueRight_1)
114                 DiffUp_1 = abs(ValueCenter_1-ValueUp_1)
115                 DiffDown_1 = abs(ValueCenter_1-ValueDown_1)
116
117                 if DiffLeft_1 > DAli or DiffRight_1 > DAli or DiffUp_1 > DAli
118                     or DiffDown_1 > DAli :
119                     ValueCenter_2 = int(arrayAliasing[i][2*(j)+1])
120                     ValueLeft_2 = int(arrayAliasing[i][2*(j-1)+1])

```

```

116     ValueRight_2 = int(arrayAliasing[i][2*(j+1)+1])
117     ValueUp_2 = int(arrayAliasing[i-1][2*(j)+1])
118     ValueDown_2 = int(arrayAliasing[i+1][2*(j)+1])
119
120     DiffLeft_2 = abs(ValueCenter_2-ValueLeft_2)
121     DiffRight_2 = abs(ValueCenter_2-ValueRight_2)
122     DiffUp_2 = abs(ValueCenter_2-ValueUp_2)
123     DiffDown_2 = abs(ValueCenter_2-ValueDown_2)
124
125     if DiffLeft_2 > DALi or DiffRight_2 > DALi or DiffUp_2 >
        DALi or DiffDown_2 > DALi :
126         arrayAliasing_BW[i][2*j] = 0
127         arrayAliasing_BW[i][2*j+1] = 0
128     imageAliasing_BW = Image.fromarray(arrayAliasing_BW)
129     imageAliasing_BW.save('./metoda_Aliasing/Aliasing-{}_BW.png'.format(
        Files[f]))

```

A.3 Metoda Variance

```

1  #!/usr/bin/python3
2  import tkinter as tk
3  import numpy as np
4  import sys
5  import imageio
6  import scipy.misc
7  from PIL import ImageTk, Image
8  import PIL
9  import os
10 import datetime
11 import math
12
13 def asVar(array):
14     return np.sum((array-np.mean(array))**2)
15 =====
16 # Predprogram - vypocita parametr DVar
17 =====
18 n = 2 # 2-6
19 array = np.array(PIL.Image.open("./data/00/00_grey.png").convert("L"))
20 Width, Height = len(array[0]), len(array)
21 arrayVarLeftRight = np.array([np.array([0 for i in range(Width) ]), dtype=
        np.uint8) for j in range(Height) ], dtype=np.uint8)
22 arrayVarUpDown = np.array([np.array([0 for i in range(Width) ]), dtype=np.
        uint8) for j in range(Height) ], dtype=np.uint8)
23
24 for i in range(0+n, Height-n):
25     for j in range(0+n, Width-n):
26         VarLeft = asVar(array[i:i+1,j-n:j][0])
27         VarRight = asVar(array[i:i+1,j+1:j+1+n][0])
28         VarUp = asVar(np.transpose(array[i-n:i, j:j+1])[0])
29         VarDown = asVar(np.transpose(array[i+1:i+1+n, j:j+1])[0])
30
31         arrayVarLeftRight[i][j] = abs(VarLeft-VarRight)
32         arrayVarUpDown[i][j] = abs(VarUp-VarDown)
33 DVar = 0
34
35 while (1):
36     arrayVariance_BW = np.array([np.array([255 for i in range(Width) ],
        dtype=np.uint8) for j in range(Height) ], dtype=np.uint8)
37

```

```

38     for i in range(0+n, Height-n):
39         for j in range(0+n, Width-n):
40             if arrayVarLeftRight[i][j] > DVar and arrayVarUpDown[i][j] >
                DVar:
41                 arrayVariance_BW[i][j] = 0
42 SumBW = 0
43
44     for i in range(0, Height):
45         for j in range(0, Width):
46             if arrayVariance_BW[i][j] == 0:
47                 SumBW = SumBW + 1
48     if SumBW > 1 :
49         DVar = DVar + 1
50     else:
51         print("Parametr_DVar:" , DVar)
52         break
53     print("SumBW:" , SumBW, "\t_DVar:" , DVar)
54     =====
55     # Hlavni cast programu - prace s obrazky s vadami
56     =====
57     Files = ("03" , "05" , "10" , "12" , "13")
58
59     for f in range(len(Files)):
60         array = np.array(PIL.Image.open("./data/{0}/{0}_grey.png".format(
                Files[f])).convert("L"))
61
62         if l == 0:
63             X = 400
64             dX = 300
65             Y = 450
66             dY = 600
67             array = array[X:X+dX,Y:Y+dY]
68         Width, Height = len(array[0]), len(array)
69         arrayVariance_BW = np.array([np.array([255 for i in range(Width) ] ,
                dtype=np.uint8) for j in range(Height) ], dtype=np.uint8)
70
71         for i in range(0+n, Height-n):
72             for j in range(0+n, Width-n):
73                 VarLeft = asVar(array[i:i+1,j-n:j][0])
74                 VarRight = asVar(array[i:i+1,j+1:j+1+n][0])
75                 VarUp = asVar(np.transpose(array[i-n:i , j:j+1])[0])
76                 VarDown = asVar(np.transpose(array[i+1:i+1+n , j:j+1])[0])
77                 if abs(VarLeft-VarRight) > DVar and abs(VarUp-VarDown) > DVar
                    :
78                     arrayVariance_BW[i][j] = 0
79         imageVariance_BW = Image.fromarray(arrayVariance_BW)
80         imageVariance_BW.save('./metoda_Variance/Variance_{0}_BW.png'.format(
                Files[f]))

```

A.4 Metoda Grey

```

1 #!/usr/bin/python3
2 import tkinter as tk
3 import numpy as np
4 import sys
5 import imageio
6 import scipy.misc
7 from PIL import ImageTk, Image
8 import PIL

```



```

9  import os
10 import datetime
11 import math
12 =====
13 # Predprogram - vypocita parametr DGre
14 =====
15 array = np.array(PIL.Image.open("./data/00/00_grey.png").convert("L"))
16 Width, Height = len(array[0]), len(array)
17 DGre = 0
18
19 while (1):
20     arrayGrey_BW = np.array([np.array([255 for i in range(Width) ], dtype
21                               =np.uint8) for j in range(Height) ], dtype=np.uint8)
22
23     for i in range(0+1,Height-1):
24         for j in range(0+1,Width-1):
25             PixelCenter = int(array[i][j])
26             PixelUp = int(array[i-1][j])
27             PixelDown = int(array[i+1][j])
28             PixelLeft = int(array[i][j-1])
29             PixelRight = int(array[i][j+1])
30
31             DiffUp = PixelUp - PixelCenter
32             DiffDown = PixelDown - PixelCenter
33             DiffLeft = PixelLeft - PixelCenter
34             DiffRight = PixelRight - PixelCenter
35
36             if (DiffLeft > DGre or DiffRight > DGre) and (DiffUp > DGre
37                 or DiffDown > DGre):
38                 arrayGrey_BW[i][j] = 0
39
40     SumBW = 0
41
42     for i in range(0,Height):
43         for j in range(0,Width):
44             if arrayGrey_BW[i][j] == 0:
45                 SumBW = SumBW + 1
46
47     if SumBW > 0 :
48         DGre = DGre + 1
49     else:
50         print(" Parametr_DGre:" , DGre)
51         break
52     print("SumBW:" , SumBW, "\t_DGre:" , DGre)
53 =====
54 # Hlavni cast programu - prace s obrazky s vadami
55 =====
56 Files = ("03" , "05" , "10" , "12" , "13")
57
58 for f in range(len(Files)):
59     array = np.array(PIL.Image.open("./data/{0}/{0}_grey.png".format(
60         Files[f])).convert("L"))
61     Width, Height = len(array[0]), len(array)
62     arrayGrey_BW = np.array([np.array([255 for i in range(Width) ], dtype
63                               =np.uint8) for j in range(Height) ], dtype=np.uint8)
64
65     for i in range(0+1,Height-1):
66         for j in range(0+1,Width-1):
67             PixelCenter = int(array[i][j])
68             PixelUp = int(array[i-1][j])
69             PixelDown = int(array[i+1][j])

```

```

64         PixelLeft = int(array[i][j-1])
65         PixelRight = int(array[i][j+1])
66
67         DiffUp = PixelUp - PixelCenter
68         DiffDown = PixelDown - PixelCenter
69         DiffLeft = PixelLeft - PixelCenter
70         DiffRight = PixelRight - PixelCenter
71
72         if (DiffLeft > DGre or DiffRight > DGre) and (DiffUp > DGre
73             or DiffDown > DGre):
74             arrayGrey_BW[i][j] = 0
75         imageGrey_BW = Image.fromarray(arrayGrey_BW)
76         imageGrey_BW.save(' ./metoda_Grey/Grey-{}_BW.png'.format(Files[f]))

```

A.5 Metoda Saturace

```

1  #!/usr/bin/python3
2  import tkinter as tk
3  import numpy as np
4  import sys
5  import imageio
6  import scipy.misc
7  from PIL import ImageTk, Image
8  import PIL
9  import os
10 import datetime
11 import math
12 =====
13 # Predprogram - vypocita parametr LSat
14 =====
15 n = 4
16 array_1 = np.array(PIL.Image.open("./data/00/00_sat_a.png").convert("L"))
17 array_2 = np.array(PIL.Image.open("./data/00/00_sat_b.png").convert("L"))
18 array_3 = np.array(PIL.Image.open("./data/00/00_sat_c.png").convert("L"))
19 array_4 = np.array(PIL.Image.open("./data/00/00_sat_d.png").convert("L"))
20 array_5 = np.array(PIL.Image.open("./data/00/00_sat_e.png").convert("L"))
21 array_6 = np.array(PIL.Image.open("./data/00/00_sat_f.png").convert("L"))
22 array_7 = np.array(PIL.Image.open("./data/00/00_sat_g.png").convert("L"))
23 array_8 = np.array(PIL.Image.open("./data/00/00_sat_h.png").convert("L"))
24 Width, Height = len(array_1[0]), len(array_1)
25 array = (array_1, array_2, array_3, array_4, array_5, array_6, array_7,
26         array_8)
27 arrayT = np.array([np.array([0 for i in range(Width) ]), dtype=np.uint8)
28                 for j in range(Height) ], dtype=np.uint8)
29 arraySaturace = np.array([np.array([0 for i in range(Width) ]), dtype=np.
30                 uint8) for j in range(Height) ], dtype=np.uint8)
31
32 for t in range(len(array)):
33     arrayT = np.copy(array[t])
34
35     for i in range(0, Height):
36         for j in range(0, n):
37             if array[t][i][j] == 255:
38                 arrayT[i:i+1, j:j+1+n] = arraySaturace[i:i+1, j:j+1+n]
39
40         for j in range(0+n, Width-n):
41             if array[t][i][j] == 255:
42                 arrayT[i:i+1, j-n:j+1+n] = arraySaturace[i:i+1, j-n:j+1+n]

```

```

41     for j in range(Width-n, Width):
42         if array[t][i][j] == 255:
43             arrayT[i:i+1, j-n:j+1] = arraySaturace[i:i+1, j-n:j+1]
44     array = list(array)
45     array[t] = np.copy(arrayT)
46     array = tuple(array)
47 arraySaturace = np.maximum(np.maximum(np.maximum(array[0], array[1]), np.
    maximum(array[2], array[3])), np.maximum(np.maximum(array[4], array
    [5]), np.maximum(array[6], array[7])))
48 arraySaturace[0:Height, 0:1+n] = np.zeros([Height, Width])[0:Height, 0:1+n]
49 arraySaturace[0:Height, Width-n:Width+1] = np.zeros([Height, Width])[0:
    Height, Width-n:Width+1]
50 LSat = 0
51
52 while (1):
53     arraySaturace_BW = np.array([np.array([255 for i in range(Width) ],
    dtype=np.uint8) for j in range(Height) ], dtype=np.uint8)
54
55     for i in range(0, Height):
56         for j in range(0, Width):
57             if arraySaturace[i][j] > LSat:
58                 arraySaturace_BW[i][j] = 0
59     SumBW = 0
60
61     for i in range(0, Height):
62         for j in range(0, Width):
63             if arraySaturace_BW[i][j] == 0:
64                 SumBW = SumBW + 1
65     if SumBW > 1 :
66         LSat = LSat + 1
67     else:
68         print(" Parametr LSat:" , LSat)
69         break
70     print("SumBW:" , SumBW, "\tLSat:" , LSat)
71 

---



---


72 # Hlavni cast programu - prace s obrazky s vadami
73 

---



---


74 Files = ("03" , "05" , "10" , "12" , "13")
75
76 for f in range(len(Files)):
77     array_1 = np.array(PIL.Image.open("./data/{0}/{0}_sat_a.png".format(
    Files[f])).convert("L"))
78     array_2 = np.array(PIL.Image.open("./data/{0}/{0}_sat_b.png".format(
    Files[f])).convert("L"))
79     array_3 = np.array(PIL.Image.open("./data/{0}/{0}_sat_c.png".format(
    Files[f])).convert("L"))
80     array_4 = np.array(PIL.Image.open("./data/{0}/{0}_sat_d.png".format(
    Files[f])).convert("L"))
81     array_5 = np.array(PIL.Image.open("./data/{0}/{0}_sat_e.png".format(
    Files[f])).convert("L"))
82     array_6 = np.array(PIL.Image.open("./data/{0}/{0}_sat_f.png".format(
    Files[f])).convert("L"))
83     array_7 = np.array(PIL.Image.open("./data/{0}/{0}_sat_g.png".format(
    Files[f])).convert("L"))
84     array_8 = np.array(PIL.Image.open("./data/{0}/{0}_sat_h.png".format(
    Files[f])).convert("L"))
85
86     Width, Height = len(array_1[0]), len(array_1)

```

```

87     array = (array_1, array_2, array_3, array_4, array_5, array_6,
88             array_7, array_8)
89     arrayT = np.array([np.array([0 for i in range(Width) ], dtype=np.
90                          uint8) for j in range(Height) ], dtype=np.uint8)
91     arraySaturace = np.array([np.array([0 for i in range(Width) ], dtype=
92                               np.uint8) for j in range(Height) ], dtype=np.uint8)
93     arraySaturace_BW = np.array([np.array([255 for i in range(Width) ],
94                                         dtype=np.uint8) for j in range(Height) ], dtype=np.uint8)
95
96     for t in range(len(array)):
97         arrayT = np.copy(array[t])
98         for i in range(0, Height):
99             for j in range(0+n, Width-n):
100                 if array[t][i][j] == 255:
101                     arrayT[i:i+1, j-n:j+1+n] = np.zeros([1, 2*n+1])
102
103         array = list(array)
104         array[t] = np.copy(arrayT)
105         array = tuple(array)
106     arraySaturace = np.maximum(np.maximum(np.maximum(array[0], array[1]),
107                                         np.maximum(array[2], array[3])), np.maximum(np.maximum(array[4],
108                                         array[5]), np.maximum(array[6], array[7])))
109
110     for i in range(0, Height):
111         for j in range(0, Width):
112             if arraySaturace[i][j] > LSat:
113                 arraySaturace_BW[i][j] = 0
114     imageSaturace = Image.fromarray(arraySaturace)
115     imageSaturace.save('./metoda_Saturace/Saturace_{0}.png'.format(Files[
116         f]))
117     imageSaturace_BW = Image.fromarray(arraySaturace_BW)
118     imageSaturace_BW.save('./metoda_Saturace/Saturace_{0}_BW.png'.format(
119         Files[f]))

```

A.6 Metoda Rotace

```

1  #!/usr/bin/python3
2  import tkinter as tk
3  import numpy as np
4  import sys
5  import imageio
6  import scipy.misc
7  from PIL import ImageTk, Image
8  import PIL
9  import os
10 import datetime
11 import math
12 #=====
13 # Predprogram - vypocita parametr URot a LRot
14 #=====
15 array = np.array(PIL.Image.open("./data/00/00_rot_a_90.png").convert("L")
16 )
17 Width, Height = len(array[0]), len(array)
18 arrayURot_parameter = np.array([np.array([0 for i in range(Width) ],
19                                         dtype=np.uint64) for j in range(Height) ], dtype=np.uint64)
20
21 for i in range(0+1, Height-1):
22     for j in range(0+1, Width-1):
23         Central = int(array[i][j])
24         Northwest = int(array[i-1][j-1])

```

```

23     North = int(array[i-1][j])
24     Northeast = int(array[i-1][j+1])
25     East = int(array[i][j+1])
26     Southeast = int(array[i+1][j+1])
27     South = int(array[i+1][j])
28     Southwest = int(array[i+1][j-1])
29     West = int(array[i][j-1])
30     GradientX = - Northwest - 2*West - Southwest + Northeast + 2*East
        + Southeast
31     GradientY = - Northwest - 2*North - Northeast + Southwest + 2*
        South + Southeast
32     arrayURot_parameter[i][j] = math.sqrt(GradientX**2 + GradientY
        **2)
33 URot = np.round(np.max(arrayURot_parameter)/2)
34 print("Parametr_URot:", URot)
35 LRot = 0
36 FRot = 5
37
38 while (1):
39     arrayStripRotation = np.array([np.array([255 for i in range(Width) ],
        dtype=np.uint8) for j in range(Height) ], dtype=np.uint8)
40     arrayGradient = np.array([np.array([0 for i in range(Width) ], dtype=
        np.uint8) for j in range(Height) ], dtype=np.uint8)
41
42     for i in range(0+1,Height-1):
43         for j in range(0+1,Width-1):
44             Central = int(array[i][j])
45             Northwest = int(array[i-1][j-1])
46             North = int(array[i-1][j])
47             Northeast = int(array[i-1][j+1])
48             East = int(array[i][j+1])
49             Southeast = int(array[i+1][j+1])
50             South = int(array[i+1][j])
51             Southwest = int(array[i+1][j-1])
52             West = int(array[i][j-1])
53             GradientX = - Northwest - 2*West - Southwest + Northeast + 2*
                East + Southeast
54             GradientY = - Northwest - 2*North - Northeast + Southwest +
                2*South + Southeast
55             Gradient = math.sqrt(GradientX**2 + GradientY**2)
56             if Gradient > LRot and Gradient < URot:
57                 arrayGradient[i][j] = 1
58
59     for i in range(0+1,Height-1):
60         for j in range(0+1,Width-1):
61             Central = int(arrayGradient[i][j])
62
63             if Central == 1:
64                 Northwest = int(arrayGradient[i-1][j-1])
65                 North = int(arrayGradient[i-1][j])
66                 Northeast = int(arrayGradient[i-1][j+1])
67                 East = int(arrayGradient[i][j+1])
68                 Southeast = int(arrayGradient[i+1][j+1])
69                 South = int(arrayGradient[i+1][j])
70                 Southwest = int(arrayGradient[i+1][j-1])
71                 West = int(arrayGradient[i][j-1])
72
73                 if Northwest + North + Northeast + East + Southeast +
                    South + Southwest + West >= FRot*1:

```

```

74         arrayStripRotation[i][j] = 0
75     SumBW = 0
76
77     for i in range(0, Height):
78         for j in range(0, Width):
79             if arrayStripRotation[i][j] == 0:
80                 SumBW = SumBW + 1
81     if SumBW > 1 :
82         LRot = LRot + 1
83     else:
84         print("Parametr_LRot:" , LRot)
85         break
86     print("SumBW:" , SumBW, "\t_LRot:" , LRot)
87     print("Parametry_LRot/URot:_", LRot, "/" , URot, sep="")
88     #=====
89     # Hlavni cast programu - prace s obrazky s vadami
90     #=====
91     Files = ("03" , "05" , "10" , "12" , "13")
92
93     for f in range(len(Files)):
94         array_1 = np.array(PIL.Image.open("./data/{0}/{0}_rot_a_0.png".format
95             (Files[f])).convert("L"))
96         array_2 = np.array(PIL.Image.open("./data/{0}/{0}_rot_a_45.png".
97             format(Files[f])).convert("L"))
98         array_3 = np.array(PIL.Image.open("./data/{0}/{0}_rot_a_90.png".
99             format(Files[f])).convert("L"))
100        array_4 = np.array(PIL.Image.open("./data/{0}/{0}_rot_a_135.png".
101            format(Files[f])).convert("L"))
102        array_5 = np.array(PIL.Image.open("./data/{0}/{0}_rot_b_0.png".format
103            (Files[f])).convert("L"))
104        array_6 = np.array(PIL.Image.open("./data/{0}/{0}_rot_b_45.png".
105            format(Files[f])).convert("L"))
106        array_7 = np.array(PIL.Image.open("./data/{0}/{0}_rot_b_90.png".
107            format(Files[f])).convert("L"))
108        array_8 = np.array(PIL.Image.open("./data/{0}/{0}_rot_b_135.png".
109            format(Files[f])).convert("L"))
110        Width, Height = len(array_1[0]) , len(array_1)
111        arrayStripRotation = np.array([np.array([255 for i in range(Width) ] ,
112            dtype=np.uint8) for j in range(Height) ] , dtype=np.uint8)
113        array = (array_1 , array_2 , array_3 , array_4 , array_5 , array_6 ,
114            array_7 , array_8)
115
116        for t in range(len(array)):
117            arrayGradient = np.array([np.array([0 for i in range(Width) ] ,
118                dtype=np.uint8) for j in range(Height) ] , dtype=np.uint8)
119
120            for i in range(0+1, Height-1):
121                for j in range(0+1, Width-1):
122                    Central = int(array[t][i][j])
123                    Northwest = int(array[t][i-1][j-1])
124                    North = int(array[t][i-1][j])
125                    Northeast = int(array[t][i-1][j+1])
126                    East = int(array[t][i][j+1])
127                    Southeast = int(array[t][i+1][j+1])
128                    South = int(array[t][i+1][j])
129                    Southwest = int(array[t][i+1][j-1])
130                    West = int(array[t][i][j-1])
131                    GradientX = - Northwest - 2*West - Southwest + Northeast
132                        + 2*East + Southeast

```

```

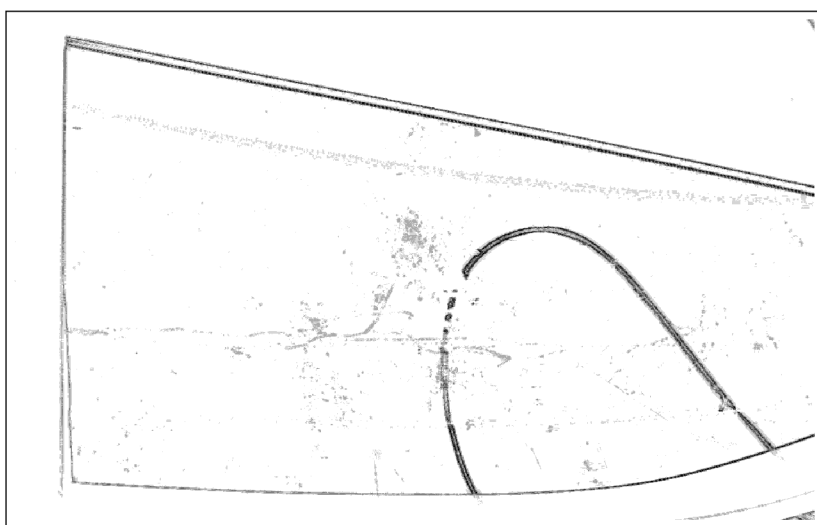
121         GradientY = - Northwest - 2*North - Northeast + Southwest
           + 2*South + Southeast
122         Gradient = math.sqrt(GradientX**2 + GradientY**2)
123
124         if Gradient > LRot and Gradient < URot:
125             arrayGradient[i][j] = 1
126             for i in range(0+1,Height-1):
127
128     for j in range(0+1,Width-1):
129         Central = int(arrayGradient[i][j])
130
131         if Central == 1:
132             Northwest = int(arrayGradient[i-1][j-1])
133             North = int(arrayGradient[i-1][j])
134             Northeast = int(arrayGradient[i-1][j+1])
135             East = int(arrayGradient[i][j+1])
136             Southeast = int(arrayGradient[i+1][j+1])
137             South = int(arrayGradient[i+1][j])
138             Southwest = int(arrayGradient[i+1][j-1])
139             West = int(arrayGradient[i][j-1])
140
141             if Northwest + North + Northeast + East + Southeast +
               South + Southwest + West >= FRot*1:
142                 arrayStripRotation[i][j] = 0
143     imageStripRotation = Image.fromarray(arrayStripRotation)
144     imageStripRotation.save('./metoda_Rotace/Rotace_{0}_BW.png'.format(
           Files[f]))

```

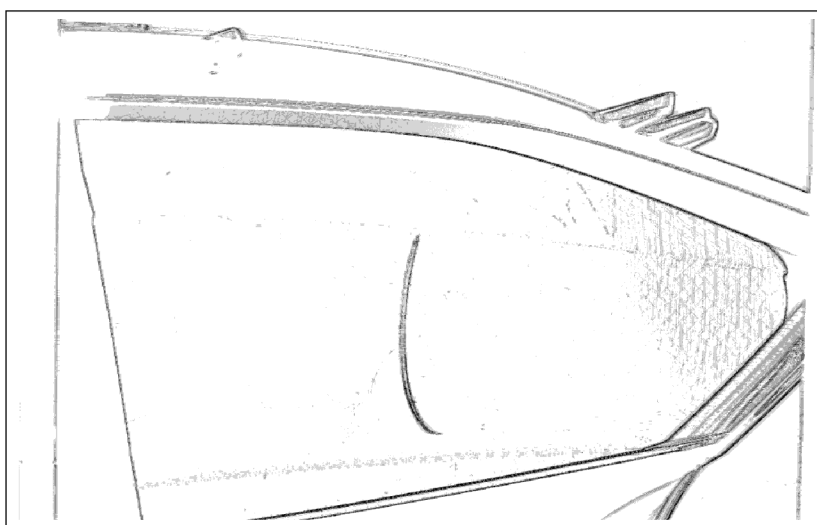
Příloha B

Kompozitní výstupní snímky

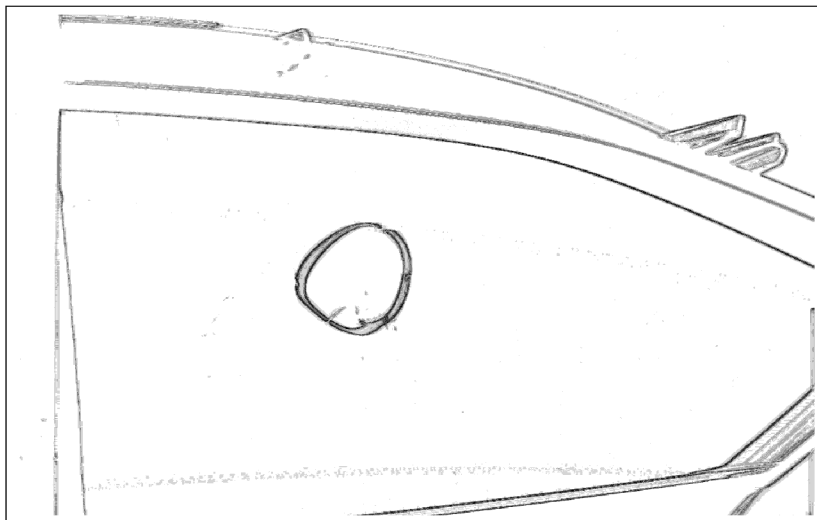
B.1 Sklo s dekorativní vadou voda na skle



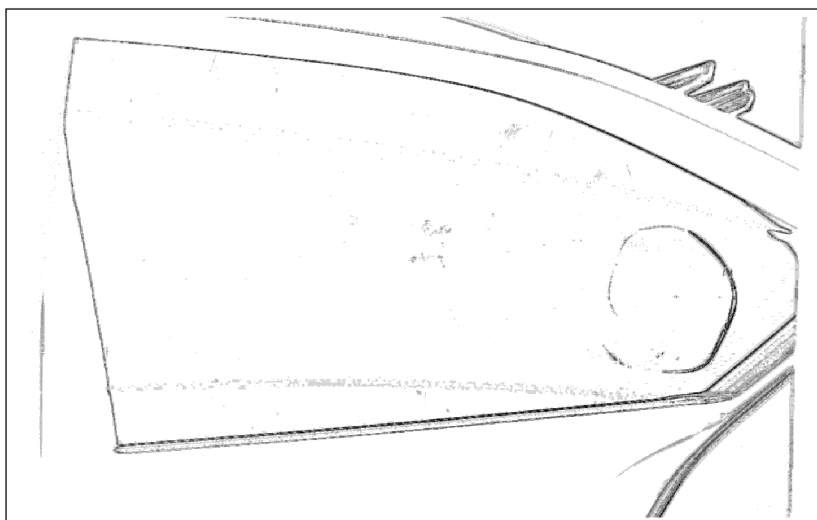
B.2 Sklo s dekorativní vadou matný



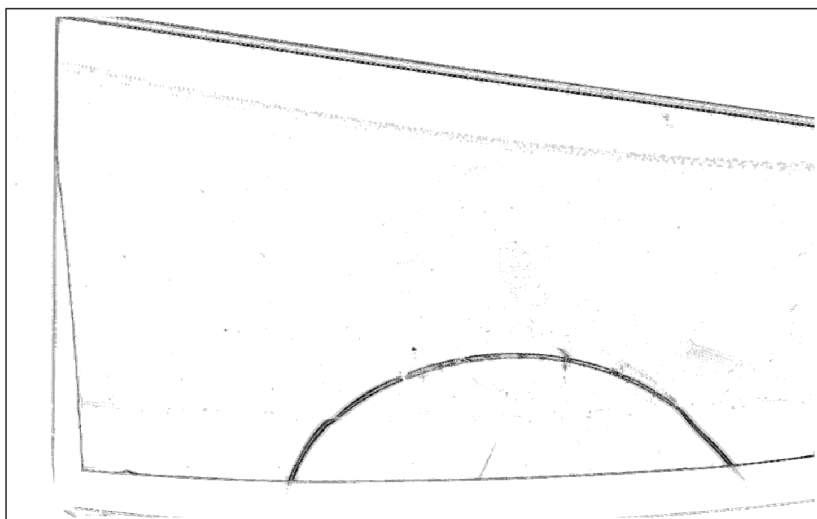
B.3 Sklo s dekorativní vadou zástřík



B.4 Sklo s dekorativní vadou cizí tělísko



B.5 Sklo s dekorativní vadou šlíra v čiré



Příloha C

Ostatní

Příloha C Ostatní odkazuje na přiložené CD, které obsahuje:

Programy `pythonMatice`, `pythonOstrost`, `pythonMoare` a `pythonKontrast`

Vstupní snímky metod před korekcí FFC

Vstupní snímky metod po korekci FFC

Výstupní a šedé snímky metod

Kompozitní výstupní snímky