



Zadání bakalářské práce

Název:	Využití metod super resolution pro zlepšení čitelnosti účetních dokladů
Student:	Petr Menci
Vedoucí:	Ing. Jakub Novák
Studijní program:	Informatika
Obor / specializace:	Umělá inteligence 2021
Katedra:	Katedra aplikované matematiky
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

Cílem práce je obrazové vylepšení účetních dokladů (např. faktura či účtenka) z důvodu budoucí možné automatizace extrakce informací. Obrázky dokladů jsou pořizovány mobilním telefonem, je tedy možné získat nejen jednu, ale i celou sérii fotografií. Hlavní důraz je kladen na metody super resolution, které využívají více snímků, pro vylepšení informace za účelem budoucího čtení např. pomocí OCR.

- 1) Proveďte rešerši možných metod a přístupů zabývajících se metodami pro zlepšení rozlišení obrazu s využitím více snímků.
- 2) Definujte metodiku měření a vytvořte či získejte obrazový dataset účetních dokladů.
- 3) Na základě nalezených přístupů navrhnete několik algoritmů zabývajících se zlepšením obrazové informace za využití zvýšení rozlišení snímku.
- 4) Implementujte algoritmy v programovacím jazyce Python.
- 5) Otestujte a zhodnoťte použitelnost algoritmu pro možné čtení informací z dokladů na obrazovém datasetu.
- 6) Diskutujte výsledky a možnosti rozšíření.

Bakalářská práce

VYUŽITÍ METOD SUPER
RESOLUTION PRO
ZLEPŠENÍ ČITELNOSTI
ÚČETNÍCH DOKLADŮ

Petr Měnl

Fakulta informačních technologií
Katedra aplikované matematiky
Vedoucí: Ing. Jakub Novák
16. května 2024

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2024 Petr Mencl. Všechna práva vyhrazena.

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

Odkaz na tuto práci: Mencl Petr. *Využití metod super resolution pro zlepšení čitelnosti účetních dokladů*.
Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

Obsah

Poděkování	vii
Prohlášení	viii
Abstrakt	ix
Seznam zkratek	x
1 Úvod	1
2 Rešerše	2
3 Teoretické pozadí	3
3.1 Mapování snímků	3
3.2 Zpracování snímků	7
3.3 Metriky snímků	7
3.4 Evaluace	9
4 Analýza problému	10
4.1 Metodika snímání	10
4.1.1 Charakteristika snímaného objektu	11
4.1.2 Snímací soustava	13
4.2 Návrh metod SR	13
4.2.1 Mapování snímků	13
4.2.2 Kombinování dat ze snímků	18
4.3 Volba knihoven pro implementaci	20
5 Praktická část	23
5.1 Dataset	23
5.2 Implementace metod	26
5.2.1 Načtení a zpracování	26
5.2.2 Metriky snímků	28
5.2.3 Mapování snímků	28
5.2.4 Kombinování dat ze snímků	32
5.2.5 SR pipeline	34
5.2.6 OCR	35
6 Evaluace výsledků	36
6.1 OCR čitelnost	36
6.2 Vizuelní dopady	40
7 Diskuze	44
8 Závěr	45

Obsah

iii

Obsah příloh

49

Seznam obrázků

3.1	Princip využití subpixel informace v jemnější mřížce [12]	4
3.2	Ilustrace různých projektivních transformací roviny [13]	4
3.3	Vizualizace nalezených KP a jejich okolí ve snímku [14]	5
3.4	Metody 1D a 2D interpolace [20]	6
4.1	Ilustrační snímek paragonu (vlevo) a faktury (vpravo)	11
4.2	Ilustrační snímky s vadami dokumentu, které se mohou vyskytovat na pořízených snímcích nehledě na zvolenou snímací soustavu (bez úpravy dokumentu)	12
4.3	Ilustrační snímky s výskytem vad způsobených vlastnosti dokumentu, pokud není správně zvolená snímací soustava	14
4.4	Výsledek interpolace při mapování ukázkové mřížky do mnohem jemnější mřížky pomocí metody nejbližšího souseda. V prvním a druhém obrázku probíhá mapování do stejně jemné mřížky a pro lepší přehlednost je v prvním obrázku vykreslena původní mřížka. V třetím obrázku se jedná a mnohonásobně jemnější cílovou mřížku	16
4.5	Simulace interpolace do 2x jemnější mřížky. Vrchní mřížka simuluje jak je hrana (zobrazena červenou přímkou) průměrována do pixelu. Vpravo od hrany se předpokládá oblast nejvyšší hodnoty pixelu, vlevo od hrany nejnižší. Mapování do spodní mřížky se předpokládá ideální	16
4.6	Grafy hodnot jednotlivých pixelů (indexovaných od 0) v závislosti na poloze hrany. Osa x odpovídá poloze hrany v procentech. Osa y udává hodnotu pixelu. Metody použité v grafech jsou v pořadí: lineární, metoda nejbližších sousedů, kubická	17
4.7	Za předpokladu, že rozdělení umístění hrany je uniformní, mřížky zobrazují průměrnou hodnotu pixelů. Metody použité jsou v pořadí: lineární, metoda nejbližších sousedů, kubická	17
4.8	Hodnoty z 4.7 pro vzájemné porovnání. Osa x odpovídá pořadí pixelu. Osa y odpovídá hodnotě pixelu	17
4.9	Degradace snímku (průsvit nebo vzniklé stíny v ohybech)	19
4.10	Popis principu vytváření „drops“ při mapování do jemnější mřížky [8]	20
4.11	Hodnoty vzdálenosti od středu odpovídajícího pixelu v jemnějších mřížkách. Hodnoty jsou pro vizualizaci normalizované	21
4.12	Váhy mapovaných pixelů hrubé mřížky do jemnějších mřížek. Váhy jsou pro vizualizaci normalizované	22
5.1	Snímací soustava pro získání datasetu	24
5.2	Ilustrativní podoba datasetu. Pro každý dokument je nasnímáno alespoň 10 snímků během pohybu	25
5.3	Rozmazání za využití maskování hodnot	27
5.4	Výsledky algoritmu odstranění stínů. Zleva doprava se zvyšuje velikost <i>kernelu</i> mazání. Ze shora dolů se zvyšuje intenzita rozmazání. Původní snímek lze nalézt ve vizualizaci 5.5	29
5.5	Výsledky algoritmu odstranění stínů. Zleva doprava se zvyšuje počet iterací algoritmu (kroky po 1 iteraci). První snímek je původní (počet iterací je 0)	30

5.6	Výsledky binarizace Otsuovou metodou. Vlevo je využit snímek bez odstranění stínů. Vpravo je využito posledního snímku z obrázku 5.5 – využitý algoritmus odstranění stínů s 3 iteracemi	30
5.7	Nalezené klíčové body ve snímku. Vlevo jsou všechny detekované body. Vpravo je detail snímku	31
5.8	Rozložení hodnoty počtu nalezených KP	31
5.9	Vizualizace párů klíčových bodů mezi dvěma snímky	33
5.10	Maticе s mírou spolehlivosti hodnot s rozdílným rozptylem ve výpočtu. Hodnoty matice jsou normalizovány pro vizualizaci	34
6.1	Rozložení hodnot zlepšení míry WER/CER podle rychlosti snímání. K snímání bylo využito videa	37
6.2	Rozložení hodnot zlepšení míry WER/CER podle rychlosti snímání. K snímání bylo využito sériového snímání	38
6.3	Rozložení hodnot zlepšení míry WER/CER. Snímky pořízené sériovým snímáním a z videa. Obojí snímání probíhalo při rychlosti posunu dokumentu 30 mm/s. Využita interpolace metodou nejbližších sousedů	38
6.4	Rozložení hodnot zlepšení míry WER/CER. Snímky pořízené sériovým snímáním a z videa. Obojí snímání probíhalo při rychlosti posunu dokumentu 30 mm/s. Využita lineární interpolace	39
6.5	Rozložení hodnot zlepšení míry CER. Snímky pořízené sériovým snímáním. Snímání probíhalo při rychlosti posunu dokumentu 30 mm/s. Využita lineární interpolace a metoda nejbližších sousedů	39
6.6	Rozložení hodnot zlepšení míry WER. Snímky pořízené sériovým snímáním. Snímání probíhalo při rychlosti posunu dokumentu 30 mm/s. Využita lineární interpolace a metoda nejbližších sousedů	39
6.7	První snímek ukazuje detail bázového snímku. Druhý snímek je detail výsledku SR algoritmu. Lze vidět redukcí šumu v druhém snímku	40
6.8	První snímek ukazuje detail bázového snímku. Druhý snímek je detail výsledku SR algoritmu. Lze vidět zvýraznění průsvitu textu z druhé strany dokumentu na druhém snímku	41
6.9	První snímek ukazuje detail bázového snímku. Druhý snímek je detail výsledku SR algoritmu. Lze vidět zvýšení rozlišení druhého snímku	41
6.10	Snímek bázového snímku a výsledek SR algoritmu s detailem do obou snímků. Lze vidět zvýšení rozlišení druhého snímku	42
6.11	První snímek ukazuje detail bázového snímku. Druhý snímek je detail výsledku SR algoritmu. Lze vidět efekt „fantomových písmen“	43

Seznam tabulek

5.1	Specifikace hlavní kamery modelu SM-A346B/DSN [32, 33]	26
6.1	Kvartily hodnot metriky zlepšení CER podle rychlosti posunu dokumentu. Konfigurace: interpolace metodou nej. soused, nevyužita segmentace textu	37
6.2	Kvartily hodnot metriky zlepšení CER podle konfigurace pro rychlost 30 mm/s a snímání sériovým snímáním	37

Seznam výpisů kódu

Chtěl bych poděkovat především Ing. Jakubu Novákovi, který mě podporoval při tvorbě této práce a byl mi ochotným vedoucím. Také bych chtěl poděkovat mé rodině, přátelům, přítelkyni a všem, kteří mi pomáhali se sběrem dat. Nesmím zapomenout ani na mého psa Tommyho, který mi byl často potřebným odračováním.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 16. května 2024

Abstrakt

Cílem práce je návrh algoritmu pro získání snímku účetního dokladu o vysokém rozlišení za využití informace z více snímků o nižším rozlišení pořízených kamerou mobilního telefonu. Algoritmus staví na principech MI-SR metod, kde dochází k mapování snímků na sebe a následné kombinaci překrývajících se dat. Aplikování algoritmu výrazně redukuje šum ve snímku a zvyšuje rozlišení. Při porovnání výsledků čitelnosti metodami OCR před a po aplikování algoritmu dochází při správné konfiguraci průměrně k jejímu zlepšení. Navržený algoritmus může být proto využit jako metoda předzpracování k detekci znaků ve snímku.

Klíčová slova předzpracování snímků, OCR, metody super resolution, MISR, účetní dokumenty, odstranění šumu, mapování snímků

Abstract

The purpose of this work is to propose an algorithm for obtaining a high resolution image of an accounting document using information from multiple lower resolution images taken by a mobile phone camera. The algorithm builds on the principles of MI-SR methods, where mapping of images onto each other and subsequent combination of overlapping data occurs. Applying the algorithm significantly reduces the noise in the image and increases the resolution. When comparing the readability results of OCR methods before and after the application of the algorithm, there is an average improvement when the algorithm is correctly configured. Therefore, the proposed algorithm can be used as a preprocessing method for character detection in an image.

Keywords image preprocessing, OCR, super resolution methods, MISR, accounting documents, noise removal, image mapping

Seznam zkratek

ASIFT	Affine-SIFT
CER	Character Error Rate (míra chybovosti znaků)
FLANN	Fast Library for Approximate Nearest Neighbors
ISO	International Organization for Standardization
KP	Keypoint (klíčový bod)
MAP	Maximum A Posteriori
MI-SR	Multi Image Super Resolution
ML	Maximum Likelihood (maximální věrohodnost)
NTSC	National Television System Committee
OCR	Optical Character Recognition (optické rozpoznávání znaků)
ORB	Oriented FAST and rotated BRIEF
POCS	Projections Onto Convex Sets
RANSAC	Random Sample Consensus
SER	Sentence Error Rate (míra chybovosti vět)
SI-SR	Single Image Super Resolution
SIFT	Scale-invariant feature transform
SR	Super Resolution
SURF	Speeded-Up Robust Features
WER	Word Error Rate (míra chybovosti slov)

Kapitola 1

Úvod

Aktuálně je snaha vše kolem sebe plně digitalizovat, ať už z důvodu ekologie, nebo zefektivnění postupů [1]. Některé oblasti ale nejsou ani zdaleka připravené na plnou digitalizaci a z různých důvodů jsou i společností odmítány. Příkladem toho jsou účetní doklady, kde i kvůli anonymitě ze strany nakupujícího je stále nutné vydat fyzické potvrzení a nelze ho přiřadit k známému subjektu v digitálním světě. [2]

Podnikatelé a firmy si kvůli účetnictví musí tyto doklady fyzicky uchovávat a následně z nich často ručně přepisovat data do účetních systémů. Ani jeden z těchto kroků nezapadá do digitální éry svou nepraktičností a neefektivitou. Nutnost uchování lze omezit nebo úplně vypustit tím, že je pořízena digitální reprezentaci dokumentu, z které jsou získatelná všechna potřebná data. K získání dat lze místo ručního přepisování využít nástroje na vytěžování dat z dokumentů pomocí metod strojového vidění. Takové nástroje ale mají vysoké požadavky na kvalitu reprezentace dokumentu, obzvláště pokud mají pracovat spolehlivě. Potřeba spolehlivosti je ale v oblasti administrativy vyšší než jinde. [3]

Dobrou kvalitu můžeme očekávat u naskenovaného snímku. Potřeba skeneru jde však proti omezení uchovávání dokumentu a vlastně i efektivitě práce. Obdobné se týká použití profesionálního fotoaparátu. Nejvhodnější možností se tedy nabízí pořídit snímek kamerou mobilního telefonu. Je to rychlé a může se tak stát v podstatě kdekoliv a kdykoliv. Problémem ale může být kvalita fotografie. Na tu může mít dopad už špatný stav dokumentu způsobený nekvalitním tiskem, papírem nebo nešetrným nakládáním s ním. Další dopady má i běžné prostředí, ve kterém se snímek pravděpodobně pořizuje, spojené s neideálním využitím kamery běžným uživatelem. Nakonec sama kamera mobilního telefonu má omezení na rozlišení a kvalitu snímku.

Většinu z těchto vlivů na kvalitu nelze předcházet, pokud má být zachována flexibilita a efektivita snímání. Naopak je vhodné se pokusit ve výsledném snímku takové vlivy potlačit a rozlišení vylepšit pomocí moderních metod a algoritmů počítačového vidění. Za tímto účelem lze využít informaci i z více snímků například pořízením krátkého videa nebo sériového snímání.

Cílem teoretické části práce je prozkoumat existující metody využití více snímků za účelem získu snímku o vyšším rozlišení. Dále definovat metodiku měření a navrhnout vlastní algoritmy pro zlepšení obrazové informace za využití zvýšení rozlišení snímku. V praktické části se práce věnuje získání obrazového datasetu účetních dokladů a implementaci navržených algoritmů v programovacím jazyce Python. Algoritmy se testují na získaném datasetu a zhodnocuje se jejich využitelnost. Nakonec jsou diskutovány výsledky a možnosti rozšíření.

Cílem práce není zhodnocovat kvalitu metod pro vytěžení dat. Tyto metody jsou v práci použity pouze jako metriky výsledků navrhovaných algoritmů na předzpracování dat.

Kapitola 2

Rešerše

Metody *super resolution* (dále *SR*) mají za cíl zlepšení obrazové informace snímku pomocí zvýšení rozlišení. K těmto metodám se často přistupuje, pokud je obtížné přímo získat snímek s vysokým rozlišením (dále jen *HR*), ale je dostupný alespoň jeden snímek s nižším rozlišením (dále jen *LR*). Omezením pro získání *HR* mohou být například optické limitace, s kterými se nejčastěji lze setkat v oblasti mikroskopie [4]. Dalším omezením může být i nedostupnost odpovídajících snímačů. Nedostupnost kvalitních snímačů může být způsobena například zastaralou technologií, kterou nelze modernizovat (vesmírné teleskopy), nebo i finančními důvody.

Základním rozdělením *SR* metod je, zda využívají pro vylepšení obrazové informace jeden snímek, nebo více. V případě použití jednoho snímku (*Single Image Super Resolution*, dále jen *SI-SR*) jsou informace o snímaném objektu obsažené výhradně v něm a další nejsou zpravidla k dispozici. V případě použití více snímků (*Multi Image Super Resolution*, dále jen *MI-SR*) je možné spojit informaci z těchto snímků za účelem získání jednoho o vyšším rozlišení. Více počáteční informace o dokumentu ale nemusí nutně znamenat lepší výsledky oproti předešlému přístupu. *MI-SR* algoritmy jsou totiž omezeny pouze na aktuální vstupní snímky, zatímco *SI-SR* může využívat teoreticky neomezeně snímků.

Myšlenka zvýšení rozlišení snímku pomocí analytických metod byla navržena a zpopularizována Tsaiem a Haungem v práci [5]. Ti využívali informace z frekvenčních domén více snímků s použitím Fourierovy transformace. Tento přístup ale spoléhal na snímky bez vážnějších degradací (např. šum nebo rozmazání). Na jejich práci proto navázali Kim s Bosem v [6]. Ti navrhli algoritmus, který dokáže pracovat i s do jisté míry degradovanými snímky. Frekvenční doména má ale obecně limity na práci se snímky s výraznými degradacemi, proto se další výzkumy věnovaly převážně informacím z prostorové domény.

Jedním z prvních přístupů k *SR* v prostorové doméně se zabývali Ur a Gross v [7], kteří metodu založili na principu frekvenční domény. Další nové přístupy využívaly například odhady *maximum a posteriori* (MAP), maximální věrohodnost (ML) nebo *projekci na konvexní množiny* (POCS). Výraznou popularitu si získala metoda Fruchtera a Hooka navržená v [8], která se využívá dodnes v astronomii [9].

S příchodem neuronových sítí a hlubokého učení začaly být intenzivně vyvíjeny i metody využívající pro *SR* pouze jeden snímek. Jak navrhl Freeman a spol. v [10], využívá se předučených sítí, které tzv. „halucinují“ chybějící data. Tyto sítě mohou dosahovat velmi dobrých výsledků, ale chybějící informace je stále doplněna na základě učení z jiných dokumentů, proto se jedná pouze o odhad a může být plně mylný. Jako moderní přelomový přístup k *SI-SR* uvedli Ledig a spol. v [11] metodu založenou na GAN (*Generative Adversarial Network* – generující soupeřící sítě), v které spolu soupeří generující a diskriminační síť. Generující síť se snaží vygenerovat takový snímek, aby diskriminující síť nepoznala, zda se jedná o reálný snímek, nebo vygenerovaný.

Teoretické pozadí

Obecně MI-SR přistupuje k tomu, že rozdílné posuny informace v mřížce v jednotlivých snímcích obsahují hlubší informaci o tom, jak by informace měla vypadat v *subpixel* doméně. *Subpixel* znamená úroveň nižší než je pixel. Vizualizaci tohoto principu lze nalézt na obrázku 3.1.

K tomuto přístupu v MI-SR jsou společné kroky, které je nutné blíže pochopit. K těmto krokům patří: mapování, zpracování a metriky snímků a evaluace výsledného snímku.

3.1 Mapování snímků

Pokud jsou snímky pořízené z identické pozice s nezměněnou scénou, pak se musí překrývat. Oblasti se překrývají, pokud je jejich rozdíl nulový nebo dostatečně malý, aby se dalo předpokládat, že je způsobený pouze rozdílným rozmístěním šumu, interpolací nebo rozdílným zprůměrováním světla v pixelové mřížce. Pokud jsou snímky pořízené z různých pozic nebo se scéna mění, pak se nemusí překrývat. Pro další zpracování je ale nutné, aby se alespoň v oblastech, kde se vyskytuje na snímku doklad, překrývaly. Za předpokladu, že na všech snímcích se doklad nachází, lze toho dosáhnout grafickou transformací snímků.

Snímek je výsledek projekce z tří-dimenzionálního prostoru P na dvou-dimenzionální rovinu Q za použití zobrazení $A \in \mathcal{L}(P, Q)$, kde \mathcal{L} označuje množinu lineárních zobrazení mezi prostory. Při projekci do nižších dimenzí se ztrácí informace a defekt $d(A)$ je nenulový (viz rovnice 3.3, kde h je hodnota zobrazení a \dim je dimenze prostoru).

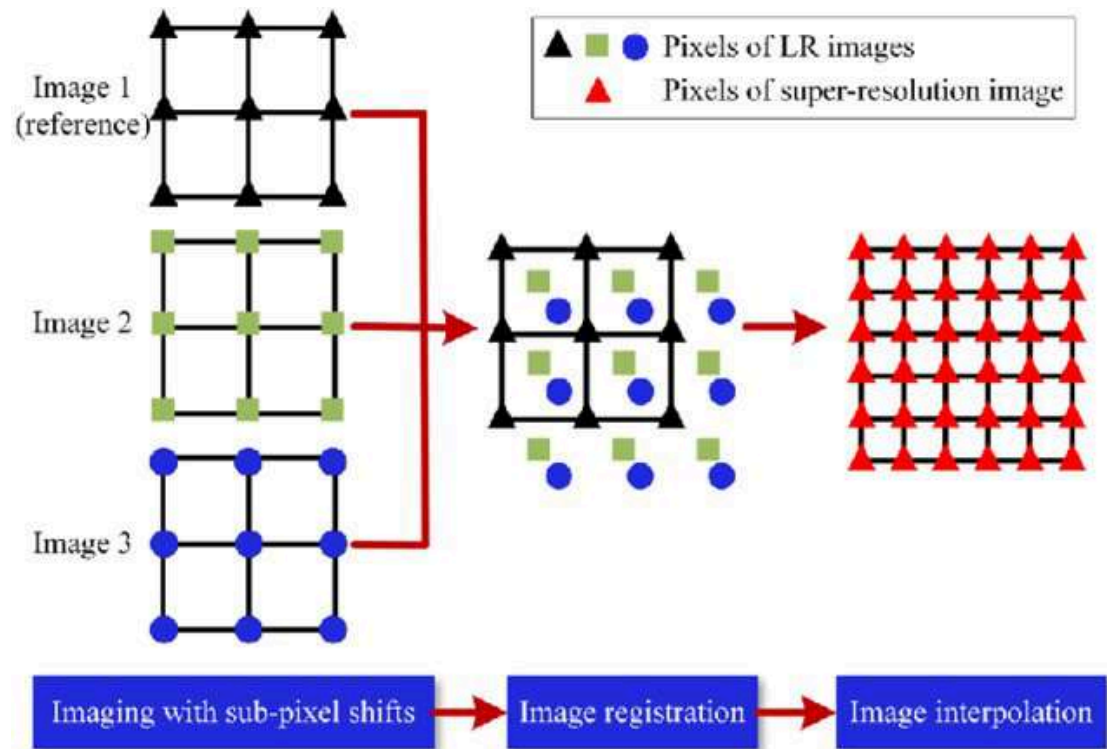
$$h(A) + d(A) = \dim(P) \quad (3.1)$$

$$2 + 1 = 3 \quad (3.2)$$

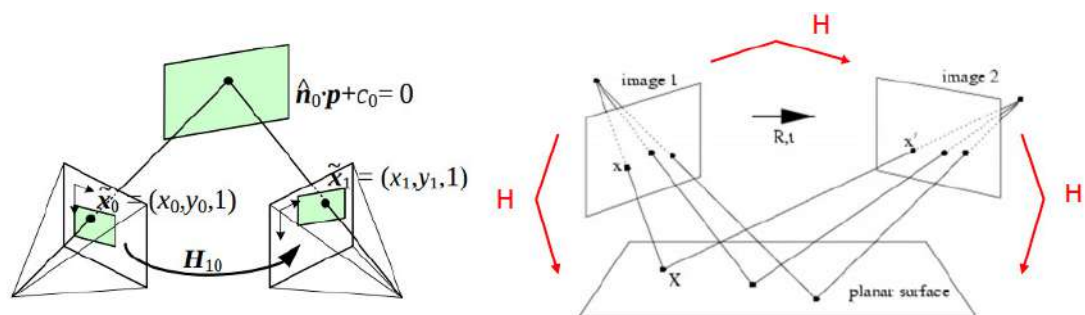
$$d(a) = 1 \quad (3.3)$$

Pokud by povrch snímaného objektu odpovídal rovině a zároveň by splýval s rovinou pozadí objektu, pak by se na jednotlivé snímky dalo nahlížet jako na výsledky postupné aplikace projektivních transformací (obrázek 3.2). Pro projektivní transformaci H nelze zajistit existenci inverzní transformace H^{-1} , kde platí $HH^{-1} = I$. Je nutné si tedy vystačit s pseudo-inverzí. Tedy aplikací (pseudo)inverzních transformací lze získat překrývající se snímky.

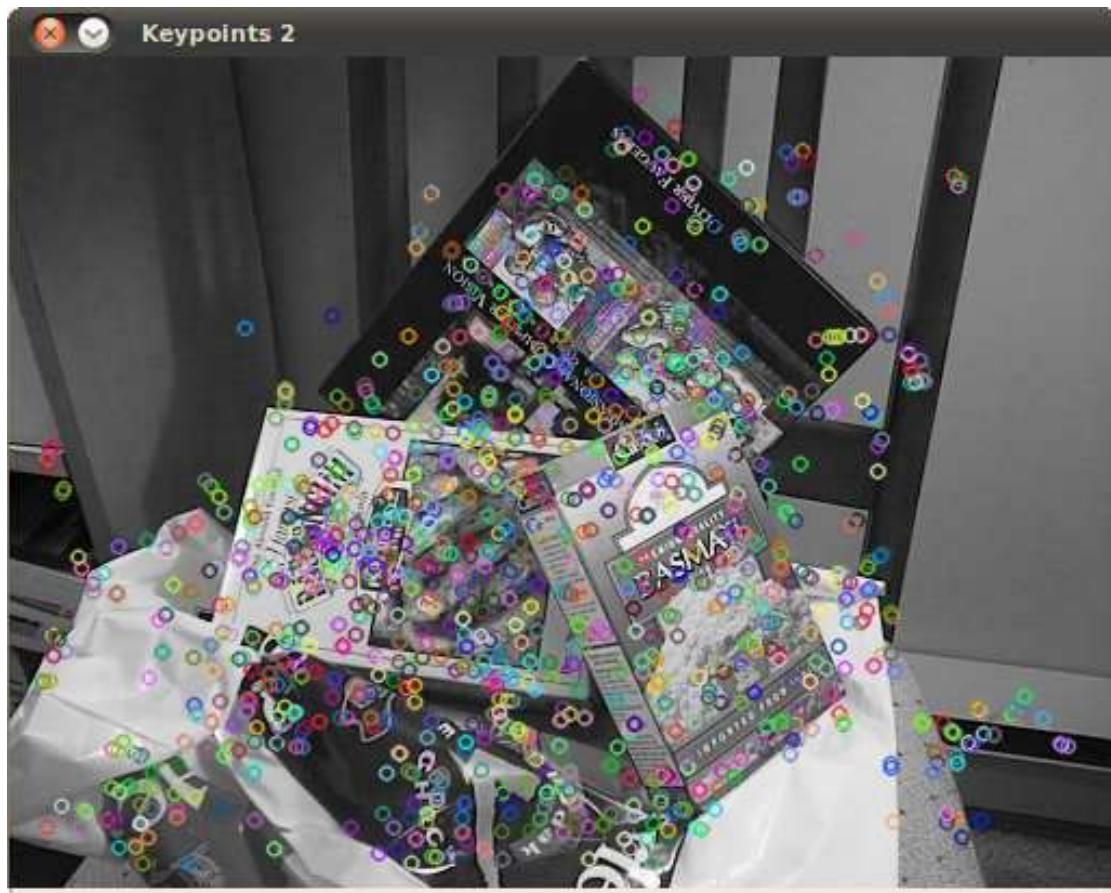
Pro účely mapování je potřeba množina párů bodů. Pár obsahuje bod z obou snímků, v kterých se nachází identické místo dokumentu. Algoritmy, které tyto páry tvoří, vyberou bod v jednom snímku a naleznou k němu odpovídající protějšek v druhém snímku. Samotný bod nelze spolehlivě porovnávat s jiným, proto se bere jeho blízké okolí, které má zajistit unikátní oblast, která bude rozeznatelná se spolehlivostí v obou snímcích – *deskriptor*. Okolí se určitou metodou porovná s okolím kandidáta na odpovídající bod v druhém snímku. Pokud jsou si okolí dostatečně



■ Obrázek 3.1 Princip využití subpixel informace v jemnější mřížce [12]



■ Obrázek 3.2 Ilustrace různých projektivních transformací roviny [13]



■ **Obrázek 3.3** Vizualizace nalezených KP a jejich okolí ve snímku [14]

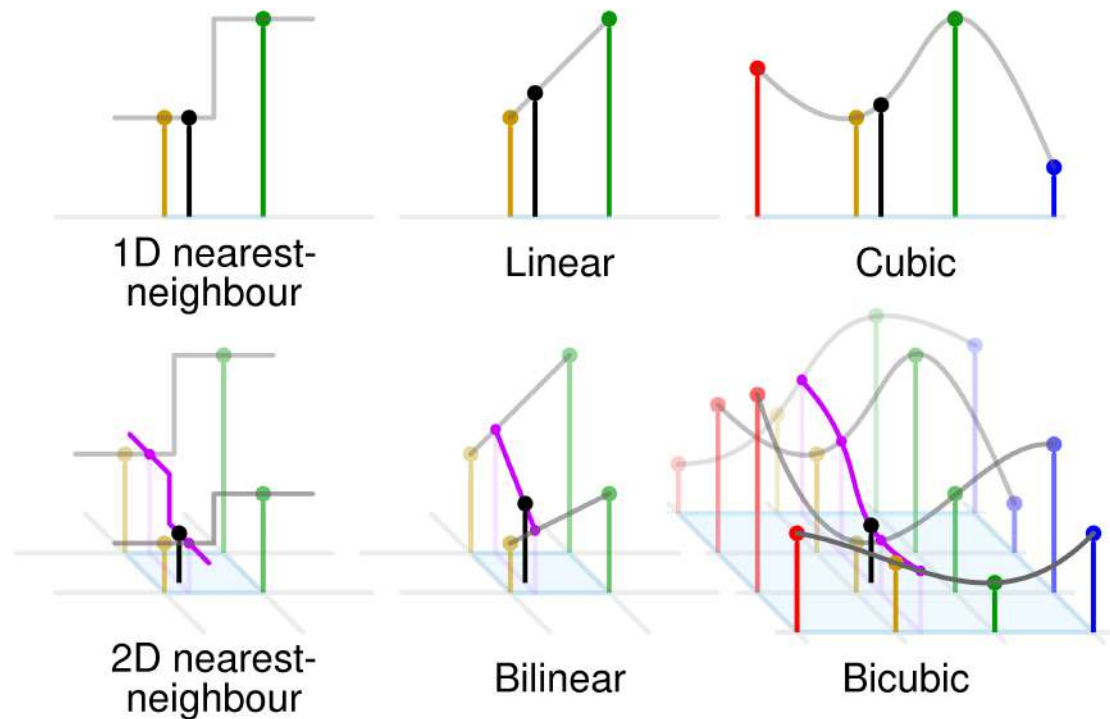
podobná a není podobnější kandidát, považují se za pár. Aby se body daly co nejjednoznačněji přiřadit k sobě, je dobré, aby jejich okolí byla co nejvíce unikátní ve snímcích. Takové unikátní body se nazývají *klíčovými body* (dále *KP*). KP se proto umísťuje v místech, kde je charakteristický rys snímku. Příklad jak vypadají nalezené KP s vyznačenými deskriptory je zobrazen v obrázku 3.3.

Mezi nejpoužívanější algoritmy pro hledání KP se řadí: *SURF*, *SIFT* a *ORB*. Nově vznikají i algoritmy pracující s neuronovými sítěmi, ale ty nedosahují tak výrazně lepších výsledků [15]. Každý z algoritmů přistupuje jinak k hledání kandidátních bodů i k popisu jejich okolí, proto některé mohou být vhodnější na specifický typ snímků.

SIFT je algoritmus navržený Lowem v [16] pro hledání a popis KP. Je měřítku a orientaci invariantní. To znamená, že je nezávislý na otočení a přiblížení. Bylo vyzorováno, že je invariantní i do jisté míry perspektivní transformaci [17]. Při hledání KP se nejdříve naleznou kandidátní body. Jejich okolí se následně interpoluje a pozice KP se určí přesněji [18]. Tento postup navíc podporuje cíl této práce, tím že vyhledává KP na pozicích mimo původní diskrétní mřížku, tedy pracuje s větší přesností než jaké je rozlišení snímku. Pro KP je sestaven deskriptor, který popisuje jeho okolí. Ten se následně využívá pro porovnání s deskriptory jiných KP z dalších snímků.

Jako rozšíření *SIFT* byl Yu a Morelem v [19] navržen algoritmus *ASIFT*, který je invariantní i afinním transformacím. Jejich práce také dokazuje (stejně jako [17]), že je možné projektivní transformaci lokálně aproximovat afinní transformací.

Po nalezení klíčových bodů následuje párování. To může probíhat hrubou silou, kdy se každý



■ **Obrázek 3.4** Metody 1D a 2D interpolace [20]

deskriptor porovnává s všemi deskriptory v druhém snímku a určí se míra podobnosti (*vzdálenost*). To umožňuje vytvořit seznam nejpodobnějších klíčových bodů a zhodnotit rozdíl podobnosti prvního a druhého nejpodobnějšího, což je tzv. *Lowe's ratio test* navržený v [18]. Tento přístup zakládá na tom, že pokud je velký rozdíl mezi prvním a druhým párem, pak je první pár spolehlivější. Pokud by nebyly příliš daleko od sebe, znamenalo by to, že mohlo dojít k záměně a deskriptor není dostatečně unikátní. Alternativou k párování hrubou silou je metoda *FLANN*, která najde nejpodobnější páry přibližně, ale výpočetně jednodušeji.

Při mapování dochází k aplikování projektivní transformace $P(x, y) : \mathbb{N}^2 \rightarrow \mathbb{R}^2$ na každý pixel zdrojového snímku. Je uvažována matice $A \in \mathbb{R}^{3,3}$ jako matice projektivní transformace a vektor $v \in \mathbb{N}^2$ souřadnic pixelu ve snímku, homogenní souřadnice odpovídající vektoru v jsou definovány jako \bar{v} , pak

$$P(v) := A\bar{v}$$

Zobrazením diskretních souřadnic zdrojové mřížky nejsou získány diskretní souřadnice. Hodnoty cílové pixelové mřížky musí být proto interpolovány. Je definována funkce $val(x, y)$, které hodnota odpovídá hodnotě pixelu na souřadnicích x, y . Interpolace znamená nalezení hodnoty $val(x_k, y_k)$ pokud jsou známy hodnoty $val(x_0, y_0), \dots, val(x_n, y_n)$ a

$$val(x_0, y_0) < val(x_k, y_k) < val(x_n, y_n)$$

. Nejčastěji se k interpolaci přistupuje jako k prokládání známých bodů křivkou (funkcí). Z křivky je pak možné získat hodnotu funkce val i v reálných souřadnicích. K interpolaci se proto přistupuje ze směru cílové mřížky. Princip interpolace ukazuje obrázek 3.4.

Nechť A^{-1} je (pseudo)inverzní matice k matici projektivní transformace A . Inverznímu zobrazení $P^{-1}(x, y)$ pak odpovídá

$$P^{-1}(v) := A^{-1}\bar{v}$$

. Takto lze pro souřadnice každého pixelu v cílové mřížce získat odpovídající reálné souřadnice v zdrojovém snímku. Na těchto reálných souřadnicích pak lze interpolovat hodnotu a zapsat ji do pixelu v cílové mřížce.

3.2 Zpracování snímků

Vzhledem k tomu, že důležité informace na dokladech bývají dvoubarevné – barva pozadí a barva textu – a často se jedná o kombinaci bílé a černé, snímky lze převést do stupňů šedi bez ztráty podstatné informace. K převodu lze využít obecně používanou metodu lineární kombinace barevných kanálů dle intenzity jejich vnímání bez gamma korekce. Ta je uvedena v rovnici 3.4. Zvolené koeficienty R – červená, G – zelená a B – modrá vychází z barevného schématu *NTSC*, což je mezinárodně uznávaný standard pro popis barevného rozsahu zobrazovacích zařízení. Díky tomu na zařízeních, které odpovídají *NTSC*, převod do stupňů šedi bere správně v úvahu jasové intenzity jednotlivých barev.

$$Y = 0.299R + 0.587G + 0.114B \quad (3.4)$$

Při segmentování dat je důležitá binarizační metoda, která zpravidla pracuje na obrázku o jednom kanálu (například stupně šedi). Ta určí způsob, kterým jsou rozděleny hodnoty do dvou tříd. Nejčastěji se hodnoty dělí na základě prahu (hodnoty větší a menší než práh). K nejpoužívanějším metodám patří Otsuovo prahování [21]. To má za cíl rozdělit hodnoty pixelů do dvou tříd tak, aby vážený součet rozptylů těchto tříd byl co nejmenší. Hledá se tedy prahová hodnota t , maximalizující hodnotu

$$\sigma_w^2 = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \quad (3.5)$$

kde

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \quad \omega_1(t) = \sum_{i=t}^{255} p(i) \quad (3.6)$$

ω_0 a ω_1 značí pravděpodobnosti tříd, σ_0^2 a σ_1^2 značí rozptyly tříd.

3.3 Metriky snímků

Metrika kvality jednoho snímku je těžko definovatelná. Snímek ale lze porovnat s ostatními, a tím určit vzájemné relativní míry. Jsou porovnávány tedy metriky vlastností snímku. Vlastnosti, které lze použít, jsou například kontrast snímku nebo intenzitu šumu.

Šum je náhodná veličina přičtená ke každému pixelu snímku. Výsledný snímek pak vypadá takto

$$I_\eta(x, y) = I(x, y) + N(x, y) \quad N(x, y) \approx \mathcal{N}(\mu, \sigma^2)$$

kde $I_\eta(x, y)$ je hodnota pixelu snímku, $I(x, y)$ je hodnotu pixelu před zašuměním a $N(x, y)$ je náhodná veličina šumu s normálním rozdělením s střední hodnotou μ a rozptylem σ^2 . Ze zákona velkých čísel průměr v nekonečnu konverguje k střední hodnotě. V případě souboru snímků (I_1, \dots, I_n) je $I_i(x, y)$ hodnota pixelu snímku I_i .

Při označení střední hodnoty pixelu jako $\hat{I}(x, y)$, je

$$\lim_{n \rightarrow \infty} \frac{\sum_{i=0}^n I_\eta(x, y)}{n} = \lim_{n \rightarrow \infty} \frac{\sum_{i=0}^n I(x, y) + N(x, y)}{n} = \mathbb{E}[I(x, y)] + \mathbb{E}[N(x, y)] = \hat{I}(x, y) + \mu$$

S rostoucím počtem snímků k průměrování tedy konverguje výsledný Gaussovský šum k hodnotě μ v celém snímku. Za naivního předpokladu, že snímky se lišily pouze o šum, je $\hat{I}(x, y) = I(x, y)$.

V případě odhadu míry šumu v snímku je použita metoda navržená Immerkærem v [22]. Tato metoda odhaduje lokální rozptyl šumu. Nevýhodou metody je vnímání velmi tenké linie jako šumu. Nepředpokládá se přirozený výskyt takové linie v kamerou pořízeném snímku. Případný výskyt nemá na využití míry dopad za předpokladu, že se vyskytuje ve všech snímcích. To by znamenalo navýšení míry o konstantu u všech snímků, které neovlivní relativní uspořádání. Podstata metody spočívá v rozdílu dvou kernelů aproximujících Laplaceův operátor. Tímto rozdílem (rovnice 3.7) se konvoluje snímek I o výšce H a šířce W . Výpočtem 3.8 je získán odhad lokální směrodatné odchylky. Jako metrika šumu je tedy definován operátor NOISE, který odpovídá tomuto odhadu.

$$L_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad L_2 = \frac{1}{2} \begin{bmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$N = 2(L_2 - L_1) = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad (3.7)$$

$$\text{NOISE} := \sqrt{\pi/2} \frac{\sum_I |I(x, y) * N|}{6(W-2)(H-2)} = \sigma_{noise} \quad (3.8)$$

Rozmazání lze aproximovat jako konvoluci ideálního snímku s maskou (*kernelem*). Toto rozmazání má charakteristiku danou hodnotami a velikostí masky. Pohybové rozmazání je typem rozmazání, které je intenzivnější ve směru pohybu. Pohybové rozmazání způsobené lineárním rovnoměrným pohybem lze aproximovat jako konvoluci snímku s horizontálním kernelem (definovaný v rovnici 3.9), který má velikost odpovídající vzdálenosti pohybu. Pokud lineární pohyb znamenal pohyb objektu o m pixelů, pak je velikost horizontálního kernelu m , tzn. $h \in \mathbb{R}^{1,m}$. Při rovnoměrném pohybu jsou hodnoty kernelu $1/m$, tedy $k_i = 1/m \quad \forall i \in \{1, 2, \dots, m\}$. Směr pohybu odpovídá rotaci kernelu pomocí matice lineární transformace o úhel θ . Rotovaný kernel se interpoluje do masky $k \in \mathbb{R}^{m,m}$. Proces rotace o úhel θ a interpolace je zde definovaný operátorem ROTATE $_{\theta}$.

$$h := [h_1 \quad h_2 \quad \dots \quad h_m] \quad (3.9)$$

$$k = \text{ROTATE}_{\theta} h \quad (3.10)$$

$$I(x, y) = I_0(x, y) * k \quad (3.11)$$

Pokud by pohyb (směr a rychlost) byl známý, bylo by možné odhadnout masku, kterou by bylo možné rozmazání aproximovat, a provést dekonvoluci maskou inverzní, pokud by existovala. Podobným technikám se věnovali Ben-Ezra a Nayar v práci [23].

Pro odhad míry rozmazání snímku lze využít Laplaceova operátoru. Tento přístup je vybrán na základě Pertuzovy práce [24], kde dochází k závěru, že metody založené na Laplacianu jsou nejvhodnější za běžných snímacích podmínek. Laplaceův operátor je divergence gradientu (rovnice 3.12). Lze ho využít jako míru změn intenzit v obraze (kontrastu) nebo k detekci hran. Metoda vychází z toho, že ostřejší snímek bude mít rapidnější změny intenzit v obraze než rozostřený/rozmazaný, proto je brán v úvahu jako míra ostrosti rozptyl hodnot Laplaceova operátoru aplikovaného na snímek I . Jako metrika ostrosti je tedy definován operátor BLUR, definovaný tímto rozptylem (výpočet 3.13).

$$\Delta f = \nabla \cdot (\nabla f) = \frac{\partial^2 f}{\partial x_1^2} + \dots + \frac{\partial^2 f}{\partial x_n^2} \quad (3.12)$$

$$\text{BLUR} := \text{var}(\Delta I) \quad (3.13)$$

3.4 Evaluace

K evaluaci výsledků aplikování SR algoritmů lze využít rozpoznání textu ve snímku (*optical character recognition*, dále jen *OCR*). Takto je získán text z bazového snímku před a po aplikování *SR* algoritmu. K určení zlepšení je potřeba referenčního textu každého dokladu. Míru správnosti rozpoznávaného textu (podobnosti k referenčnímu textu) udávají metriky *WER* (*word error rate*, míra chybovosti slov) a *CER* (*character error rate*, míra chybovosti znaků). Tyto míry pracují s *Levenshteinovou vzdáleností* – minimální editační vzdálenost. Jako editace se považují operace smazání, přidání a změna. Metrika je pak vypočítána jako minimální počet těchto editací, který vede k získání referenčního textu, vůči počtu slov nebo znaků (dle počítané metriky).

$$\text{WER} := \frac{\text{smazaných slov} + \text{přidaných slov} + \text{změněných slov}}{\text{slov referenčního textu}}$$

$$\text{CER} := \frac{\text{smazaných znaků} + \text{přidaných znaků} + \text{změněných znaků}}{\text{znaků referenčního textu}}$$

Takto získané míry se následně porovnávají. Jejich zlepšení/zhoršení je důsledkem aplikování *SR* algoritmu, proto lze takto zhodnotit jeho použitelnost.

Analýza problému

Úspěšnost algoritmů počítačového vidění (*computer vision*) se zpravidla odvíjí od kvality použitých dat. Požadovaná kvalita může být z různých důvodů těžko dosažitelná. Častým důvodem je neideální snímací soustava. Tento problém lze v nějaké míře kompenzovat využitím metod SR, ale ideální je mu předcházet a volit ideální snímací soustavu. Metody SR dovolují přidat do snímku informace, které v něm původně nejsou, a tím zlepšit rozlišení obrazu. Zdrojem těchto informací mohou být další snímky, data ze sensorů, nebo i předučené neuronové sítě.

Tato práce se soustředí na využití principů MI-SR a navržení vlastního algoritmu, který jako vstup považuje kolekci snímků účetního dokladu. Výstupem je zpracovaný snímek o vyšším rozlišení. Míra zlepšení je měřena pomocí OCR a porovnávání takto získaného textu s referenčním textem z dokladu. Jako metrika podobnosti textů je využito WER a CER algoritmů.

Jednotlivé kroky navrženého algoritmu jsou:

1. získání snímků (co nejvyšší kvality);
2. předzpracování snímků (převést do stupňů šedi, atd.);
3. filtrování snímků – vybrání n snímků nejvhodnějších pro další zpracování;
4. nalezení klíčových bodů a jejich ztotožnění napříč snímky;
5. nalezení nejvhodnějšího snímku, který bude tvořit bázi (referenční perspektivu);
6. mapování snímků na bázi za pomoci klíčových bodů;
7. kombinování snímků za váhování snímků a oblastí jednotlivých snímků;
8. zhodnocení výsledků pomocí OCR a referenčního textu.

4.1 Metodika snímání

Metodika snímání přímo ovlivňuje kvalitu získaných dat. Při snaze získat ze snímku maximum informace o snímaném objektu je prioritou nastavit snímací soustavu tak, aby přenesla nejvíce informace na snímek. Snímaný objekt udává, jaký vliv na něj má prostředí, a jak je potřeba přizpůsobit snímací soustavu.

Předem určené části snímací soustavy jsou snímací senzor a objektiv. Obojí je integrované v kameře mobilního telefonu, která je pro snímání použita. Oproti klasickým digitálním fotoaparátům má tento senzor výrazně menší rozměry. Menší senzor při zachování stejného počtu pixelů způsobuje, že na každý pixel dopadá méně světla. To může způsobit vyšší výskyt šumu a další



■ Obrázek 4.1 Ilustrační snímek paragonu (vlevo) a faktury (vpravo)

zhoršení kvality snímku. Při porovnání integrovaného objektivu s klasickým je důležitá hodnota světelnosti. Ta je zde srovnatelná a propustí tedy do snímače obdobné množství světla. Tedy i když se jedná o velmi rozdílné snímací aparáty, v době vzniku práce se v mnoha aspektech už vzájemně vyrovnají [25].

4.1.1 Charakteristika snímaného objektu

Účetní doklady se dají rozdělit na dva nejčastější typy dokumentů: faktura a paragon. Ilustrační snímky dokumentů lze vidět na obrázku 4.1.

Faktura se zpravidla tiskne na kancelářský papír formátu A4. Bývá tisknuta černě na obvykle bílý (v některých případech světlý odstín jiné barvy) papír a text je na ní řádně strukturován. Kvůli rozměrům bývá lidmi za účelem větší skladnosti překládána. Zanechané ohyby pak v místě ohybu degradují papír. Tato místa jsou pak náchylná na vznik stínů. Papír je díky svým vlastnostem nelesklý a neprůsvitný.

Paragon je zkrácený účetní doklad, tedy může obsahovat méně informací. Díky tomu se využívá menší formát papíru. Kvůli potřebě tisku velkého množství těchto dokladů se využívají termální tiskárny, které tisknou na role tzv. termo-papíru, ty jsou levnější na tisk i údržbu oproti jiným metodám. Výjimečně se objevuje i tisk na papír jiné barvy než bílé. Při tisknutí delšího dokumentu se projevuje právě ohyb z návínou v roli. Nevýhodami této metody tisku jsou obecně kvalita papíru a nechtěné termické reakce. Termo-papír má zlomek gramáže kancelářského papíru a tak se velmi jednoduše poškodí nebo deformuje a bývá mírně průsvitný. Tisk na něj probíhá lokálním zahřátím. To způsobí termickou reakci ve vrchní vrstvě termo-papíru, která vytvoří tmavý bod. Tato reakce ale jde nechtěně vyvolat i po tisku například zahřátím při tření. Samotná reakce také není trvalá a po určité době, která záleží na okolních podmínkách, začnou tmavé body blednout. Povrch papíru je navíc kvůli své vrchní vrstvě vždy v nějaké míře lesklý.

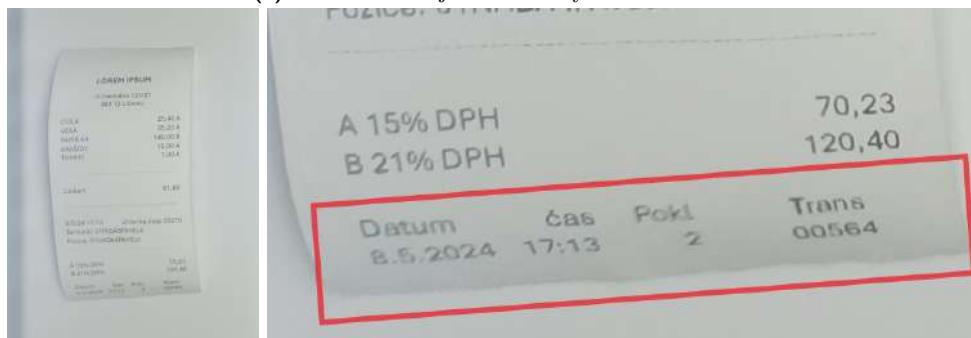
Obvyklé vady dokladů jsou zobrazené v obrázku 4.2.



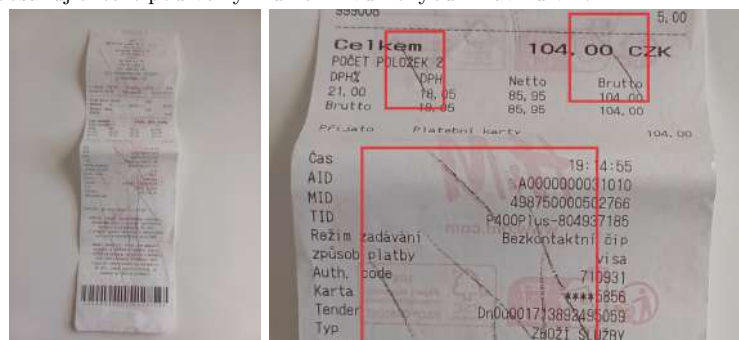
(a) Snímek obsahující vybledlé oblasti



(b) Snímek obsahující nedostatky tisku



(c) Snímek obsahující text pod velkým úhlem kvůli ohybu z návinnu v roli



(d) Snímek obsahující nechtěné termické reakce na dokladu

■ **Obrázek 4.2** Ilustrační snímky s vadami dokumentu, které se mohou vyskytovat na pořízených snímcích nehledě na zvolenou snímací soustavu (bez úpravy dokumentu)

4.1.2 Snímací soustava

Hlavní prvky snímací soustavy jsou snímací aparát a zdroj osvětlení. Snímací aparát je předem určený. Osvětlení je voleno dle prvků, které jsou v zorném poli snímacího aparátu. V zorném poli by se měl vyskytovat pouze snímáný objekt – účetní dokument. Pozadí je ideálně nereflexní, jednolitě a jednobarevné. Oba typy dokladů mohou mít velké množství záhybů, ve kterých mohou vznikat stíny. Záměrem je redukovat ve snímku stíny a tedy potlačit informaci o hloubce. K tomu je vhodné přední difuzní osvětlení s jasným obrazovým polem. To netvoří odlesky na lesklém povrchu, potlačuje stíny a zvyšuje kontrast dle absorpce světla (barvy). Příkladem takového osvětlení je *kopulové světlo*. [26, 27]

Pro ideální zaostření celého objektu je vhodné snímat senzorem, který je rovnoběžný s papírem. V případě velmi zdeformovaného papíru, který je v některých částech nadzvednutý, je potřeba, aby zůstal zaostřený nejbližší i nejdálší bod papíru k rovině snímače. To lze zajistit úpravou hloubky ostrosti. K získání větší hloubky ostrosti je nutné buď více uzavřít clonu objektivu nebo zaostřit objekt ve větší vzdálenosti od snímače [28]. Ve většině mobilních telefonů je ale clona fixní, proto pak nedostatečnou hloubku ostrosti lze řešit pouze snímáním z větší vzdálenosti od objektu.

Zkreslení objektivu je zde zanedbatelné, díky kalibraci kamery od výroby. Kamera dále obvykle disponuje optickou a softwarovou stabilizací. To snižuje efekt pohybového rozmazání, které může mít silný negativní dopad na navrhované algoritmy [29]. Pohyb mezi snímky je ale chtěný, proto není snímací aparát umísťovaný na stativ. Rychlost pohybu při snímání by měla být taková, aby nevytvářela znatelné rozmazání snímku. To je závislé na intenzitě osvětlení, objektivu a snímači, proto musí být řešeno individuálně. Směr pohybu a rotace jsou blíže nepodmíněné (jedinou podmínkou je, aby snímáný objekt – alespoň část s potřebnými daty – byla stále v zorném poli).

Při účelu, pro který je tato práce motivována, je nereálné zajištění ideální snímací soustavy. Dodatečně běžně přístupné osvětlení je pouze přisvětlovací dioda kamery („blesk“). Tu lze zařadit jako přímé směrové nasvícení, které není kvůli vzniku odlesků vhodné. Je nutné proto využít například denního světla nebo výrazně prosvětlené místnosti. Kvůli pořizování snímků z ruky, je rychlost snímače nepředvídatelná, proto lze očekávat, že část snímků bude rozmazaná. Možné další dopady neideální snímací soustavy jsou zobrazeny v obrázku 4.3.

4.2 Návrh metod SR

Návrh algoritmu staví na obvyklém přístupu k MI-SR za podmínek stanovených v sekcích 4.1.1 a 4.1.2. Díky homogenitě snímaného objektu je možné využít předpoklady, které v obecném MI-SR nelze.

Snímky musí projít předzpracováním, aby bylo možné kombinovat jejich data. Předzpracování proto zahrnuje mapování snímků tak, aby se informace v nich co nejvíce překrývaly. Podstatou SR je zvýšení rozlišení, proto mapování musí probíhat do jemnější mřížky, aby bylo možné využít *subpixel* informaci. Míra zjemnění cílové mřížky je v pozitivní korelaci s vylepšením rozlišení (zobrazeno v obrázku 4.4). Omezením ale je rychle rostoucí výpočetní a paměťová náročnost.

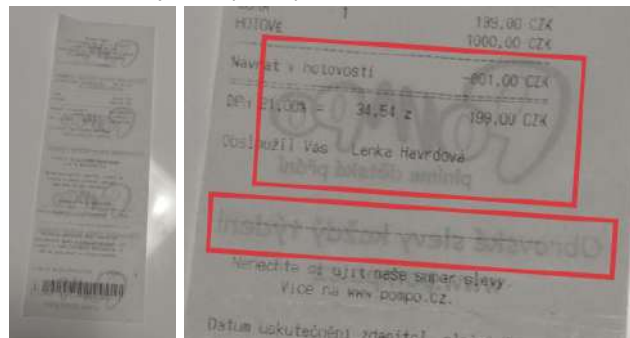
Při kombinaci informací z jednotlivých snímků je dobré jednotlivé informace váhovat, protože nemusí být stejně kvalitní nebo důvěryhodné. Váhování může probíhat na úrovni dat ve snímku a mezi snímky.

4.2.1 Mapování snímků

Pro využití mapování, jak je popsáno v sekci 3.1, jsou snímky neideální. Povrch snímaného objektu neodpovídá rovině a nesplývá s pozadím, jak je popsáno v sekci 4.1.1. Lze k tomu přistupovat dvěma způsoby. Povrch naivně aproximovat jako rovinu, nebo rozdělit objekt na více



(a) Snímek obsahující stíny v ohybech



(b) Snímek obsahující průsvit druhé strany dokumentu



(c) Snímek obsahující odlesk přisvětlovací diody na pozadí



(d) Snímek obsahující výrazné odlesky na povrchu termo-papíru

■ **Obrázek 4.3** Ilustrační snímky s výskytem vad způsobených vlastnosti dokumentu, pokud není správně zvolená snímací soustava

části, kde každá z nich bude odpovídat rovině. Pokud není možné objekt rozdělit na části, které obsahují pouze roviny, lze zkombinovat oba přístupy a části, které po dělení neodpovídají rovinám, aproximovat jako roviny. Z informací, které jsou ze snímků ale k dispozici, nelze spolehlivě segmentovat objekt na roviny, proto je povrch dále vždy aproximován jako jedna rovina.

Matice projektivní transformace má 8 stupňů volnosti (volných parametrů). Pro jednoznačné dopočítání koeficientů jsou proto potřeba 4 páry bodů. Každý pár obsahuje bod v zdrojovém snímku a jemu odpovídající bod v cílovém snímku. Kvůli aproximaci roviny ale neexistuje transformace, která by všechny body správně zobrazila. Cílem tedy je najít takovou transformaci, která způsobí v součtu přes všechny body nejmenší odchylku mezi zobrazenými zdrojovými body a cílovými body. K tomu je potřeba správně vyfiltrovat mylně párované body a následně využít všech bodů pro minimalizaci vzniklé chyby po aplikování transformační matice.

4.2.1.1 Hledání a párování klíčových bodů

Jako algoritmus pro hledání klíčových bodů je využit SIFT. Jeho alternativa ASIFT není kvůli vysoké výpočetní náročnosti v práci vhodná, i když by její využití mělo velký vliv na výsledky díky očekávaným změnám perspektivy. K párování je využito párování hrubou silou kvůli potřebě zisku maximální možné informace, aby bylo mapování co nejspolehlivější. Kvůli nepřiliš unikátním oblastem na dokumentu je využito i filtrování správných párů pomocí *Lowe's ratio test*.

Počty nalezených KP se pohybují často ve vyšších stovkách až tisících. Za účelem zvýšení věrohodnosti mapování jsou body filtrovány. Při výpočtu trans. matice se využije algoritmus *RANSAC* pro výběr KP, z kterých se matice zobrazení A vypočítá. *RANSAC* opakovaně vybírá náhodné 4 páry a vypočte z nich matici A . Maticí zprojektuje všechny KP zdrojového snímku a vypočítá vzdálenost od odpovídajících KP cílového snímku. Pokud je vzdálenost menší než daný práh $t \in \mathbb{R}_{>=0}$, pak se projekce považuje za korektní a spočítá se počet korektně projektovaných bodů – počet *inlierů* (neodlehklých bodů). Následně se k výpočtu využijí ty 4 body, které zprojektovaly nejvíce inlierů. Inlier lze považovat za skoro jistě správně napárovaný bod. Díky aproximaci roviny dokladu by tento přístup vedl k neurčitému výsledku. Vyberou se proto pouze inlieri a z nich se metodou *least square* vypočítá znovu projektivní matice A , která minimalizuje odchylku mezi projektovanými a cílovými body.

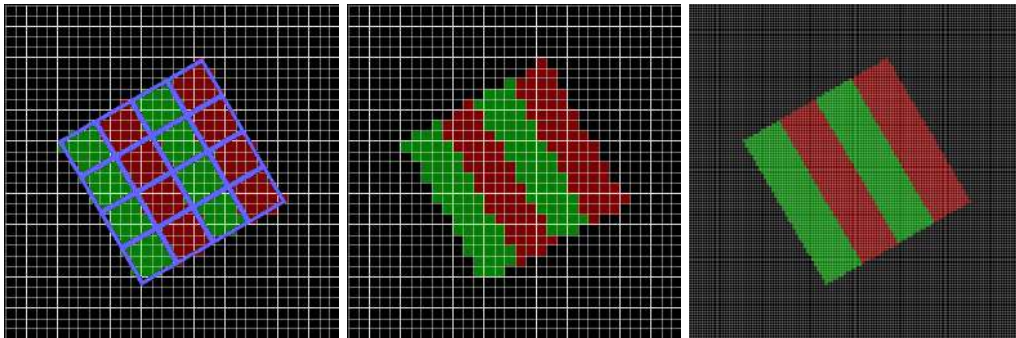
4.2.1.2 Interpolace

Interpolace různými křivkami přináší různé výsledky. V případě, kdy se snímek mapuje do výrazně jemnější mřížky, nejlépe informaci zachová metoda nejbližšího souseda, která pro jemnost mřížky jdoucí limitně do nekonečna dává nejpřesnější reprezentaci, jak ukazuje obrázek 4.4. Pokud z výpočetních nebo jiných důvodů nelze využít tak jemné mřížky, je ideální vybrat interpolační metodu, která nejvíce zúží hranu.

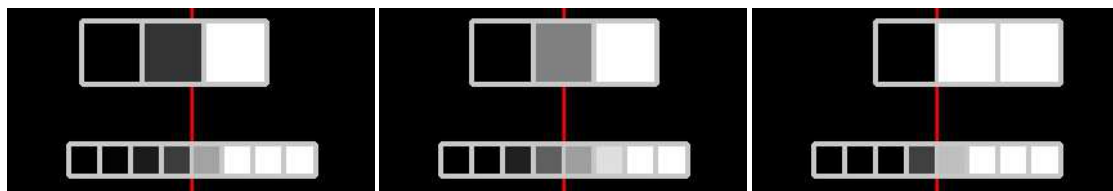
Jednotlivé snímky budou hranu (její přechod) průměrovat v určitém pixelu. V každém snímku ale díky rozdílným polohám hrany jinak. V obrázku 4.5 vidíme jak se nejpoužívanější interpolace – lineární, kubická a metoda nejbližšího souseda – chovají při zobrazování hrany. Z grafů v 4.6 lze vyčíst, že výběr interpolační metody ovlivňuje na jak široké okolí má přenos hrany vliv. Také má dopad na to jak je zachována hrana, jak ukazují obrázky v 4.7. Z porovnání ve snímku 4.8 lze vypořadovat, že interpolace pomocí metody nejbližších sousedů zachovává hranu nejužší a tedy nejostřejší. Takový přístup ale nevyužívá plnou informaci snímku, protože zanedbává vliv vzdálenosti od pixelu (za podmínky, že je stále nejbližší).

4.2.1.3 Hledání báze

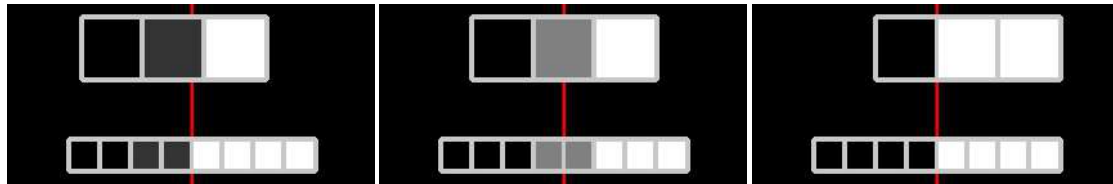
Pro mapování je potřeba vybrat jeden ze snímků jako bázi, tedy snímek do jehož perspektivy se ostatní namapují. Musí se vzít v úvahu že mapováním na špatnou bázi lze ztratit spoustu



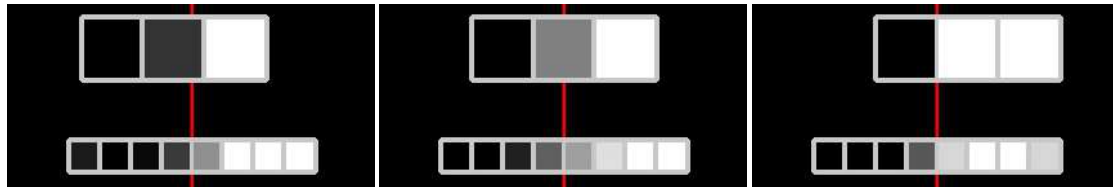
■ **Obrázek 4.4** Výsledek interpolace při mapování ukázkové mřížky do mnohem jemnější mřížky pomocí metody nejbližšího souseda. V prvním a druhém obrázku probíhá mapování do stejné jemné mřížky a pro lepší přehlednost je v prvním obrázku vykreslena původní mřížka. V třetím obrázku se jedná o mnohonásobně jemnější cílovou mřížku



(a) Použita lineární interpolace. Obrázky zobrazují hranu procházející pixelem v 80 % jeho šířky, 50 % a 0 % v pořadí zleva

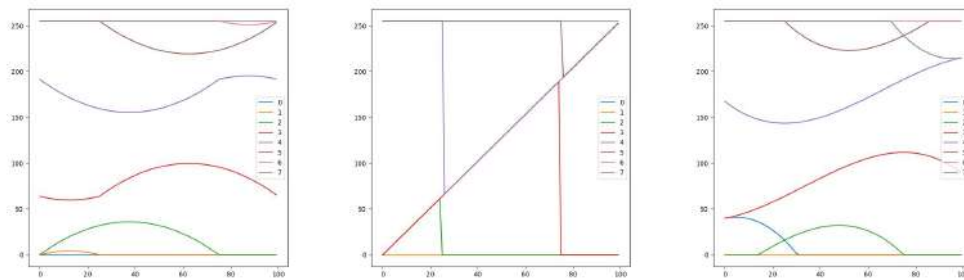


(b) Použita interpolace metodou nejbližšího souseda. Obrázky zobrazují hranu procházející pixelem v 80 % jeho šířky, 50 % a 0 % v pořadí zleva



(c) Použita kubická interpolace. Obrázky zobrazují hranu procházející pixelem v 80 % jeho šířky, 50 % a 0 % v pořadí zleva

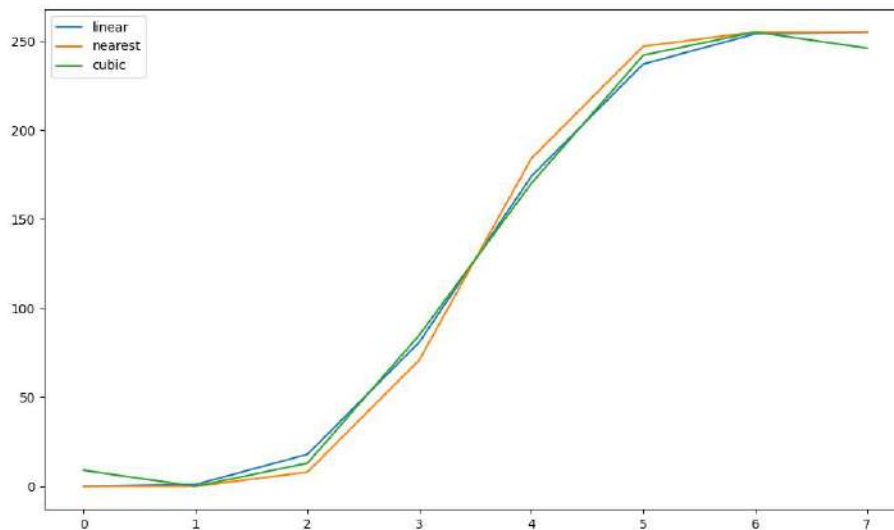
■ **Obrázek 4.5** Simulace interpolace do 2x jemnější mřížky. Vrchní mřížka simuluje jak je hrana (zobrazena červenou přímkou) průměrována do pixelu. Vpravo od hrany se předpokládá oblast nejvyšší hodnoty pixelu, vlevo od hrany nejnižší. Mapování do spodní mřížky se předpokládá ideální



■ **Obrázek 4.6** Grafy hodnot jednotlivých pixelů (indexovaných od 0) v závislosti na poloze hrany. Osa x odpovídá poloze hrany v procentech. Osa y udává hodnotu pixelu. Metody použité v grafech jsou v pořadí: lineární, metoda nejblížeších sousedů, kubická



■ **Obrázek 4.7** Za předpokladu, že rozdělení umístění hrany je uniformní, mřížky zobrazují průměrnou hodnotu pixelů. Metody použité jsou v pořadí: lineární, metoda nejblížeších sousedů, kubická



■ **Obrázek 4.8** Hodnoty z 4.7 pro vzájemné porovnání. Osa x odpovídá pořadí pixelu. Osa y odpovídá hodnotě pixelu

informace. Nejpřímočařejší přístup tedy je vyzkoušet všechny možné báze. Pro každou bázi se ostatní snímky namapují a zhodnotí se o kolik informace se přišlo.

K hodnocení ztráty informace lze vypočítat rozdíl namapovaného snímku a báze. Přílišná změna perspektivy, která by vedla k ztrátě detailu, by se projevila větším rozdílem. Ten můžeme sčítat přes celý snímek a to akumulovat přes všechny mapované snímky. Následně se jako báze vybere snímek, který vytvoří při mapování dohromady nejmenší chybu, protože se dá předpokládat, že provedl nejmenší změny perspektivy.

4.2.2 Kombinování dat ze snímků

Jak bylo uvedeno dříve, každý snímek prošel jinou změnou perspektivy. Některé obrázky nebyly spolehlivě namapované, to lze odvodit z počtu bodů, které byly použity pro výpočet homografie (inlierů), oproti počtu všech detekovaných bodů. Dále lze zhodnotit kvalitu jednotlivých snímků různými metrikami a ty využít k váhování.

Kromě váhování celých snímků je možnost také naváňovat data ve snímcích. Předpoklad je, že některá data v snímku nejsou důležitá a tedy nemusí příliš zasahovat do výsledného obrazu.

4.2.2.1 Porovnání kvalitativní metrikou snímků

Kvůli kombinaci snímků a evaluaci výsledků je nutné určit metriky kvality jednotlivých snímků. Tomu se věnovala sekce 3.3. Jako kvalita snímku je považována metrika intenzity šumu snímku NOISE a metrika zaostření snímku BLUR.

Kvůli omezenému počtu snímků je nemožné postupem popsaným v sekci 3.3 odstranit šum z celého snímku, proto je alespoň snaha o jeho největší možnou redukci. K tomu se přistupuje jako k minimalizační úloze. Je hledáno takové

$$w \in \mathbb{R}_{\geq 0}^n \wedge \sum_{i=0}^n w_i = 1$$

které minimalizuje metriku šumu na finálním snímku – výsledku váhované kombinace. Minimalizující váha w_{noise}^* je získána výpočtem 4.1. [30, 31]

$$w_{noise}^* = \arg \min_w \text{NOISE} \sum_{i=0}^n I_n(x, y) w_i \quad (4.1)$$

V případě znalosti pohybu kamery by bylo možné redukovat pohybové rozmazání. Pohyb ale není znám, proto bude v algoritmu vliv snímku obsahujícího rozmazání pouze penalizován na základě metriky rozmazání. Vyšší hodnota operátoru BLUR znamená ostřejší snímek. Kvůli vyrovnaní extrémů je tato hodnota odmocněna a následně brána v kontextu ostatních snímků jako váha před normalizováním. Výpočtem 4.2 je získána váha před normalizací. Normalizace, která zajistí součet vah roven 1 se nachází ve výpočtu 4.3.

$$\rho_i = \sqrt{\text{BLUR } I_i(x, y)} \quad \text{pro } i \in \{1, 2, \dots, n\} \quad (4.2)$$

$$w_i = \frac{\rho_i}{\sum_{j=0}^n \rho_j} \quad (4.3)$$

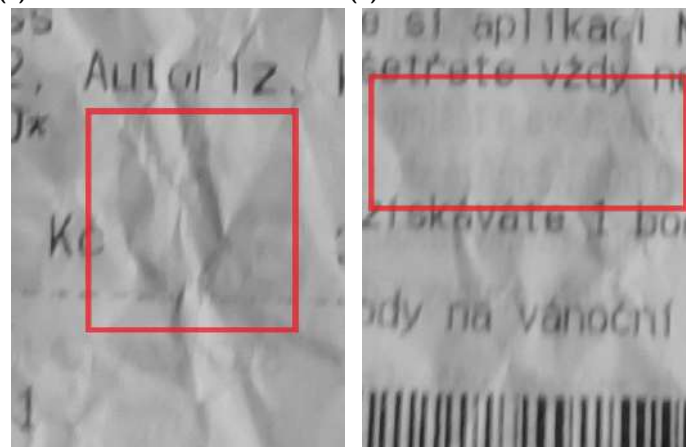
4.2.2.2 Segmentace dat ve snímku

Pokud dojde k zvýšení rozlišení celého snímku, dojde k doplnění informace i k nechtěným prvkům ve snímku. Různé artefakty nebo průsvit kontur z druhé strany papíru pak mohou zlepšením jejich rozlišení naopak zhoršovat kvalitu výsledného čtení. Je proto možnost segmentovat text od bílého



(a) Ilustrační snímek

(b) Detail snímku



(c) Vizualizace ohybů

(d) Vizualizace průsvitu

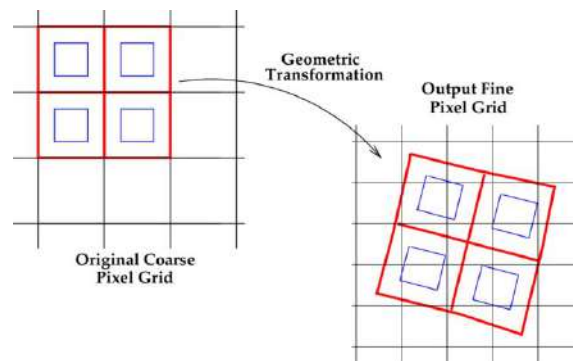
■ **Obrázek 4.9** Degradace snímku (průsvit nebo vzniklé stíny v ohybech)

pozadí a zlepšení obrazové informace provádět pouze v této oblasti. Cílené zanechání původní kvality v určitých oblastech snímku má potenciál zlepšit výsledky OCR čtení.

Segmentace textu nemusí být dokonalá. Požadavkem je segmentovat co nejvíce textu a nesegmentovat plochy pozadí na dokumentu, na kterých mohou vyniknout ohybové degradace, v nichž vzniklé stíny nebo průsvit skrze papír. Degradace, které nejsou žádoucí vylepšovat jsou uvedené v obrázku 4.9)

Segmentaci ztěžují rozdílné světelné podmínky v oblastech snímku. K vyrovnání intenzit světelných oblastí lze využít algoritmus na odstranění jasového gradientu. Ten zanechá pouze relativně malé tmavé oblasti (písmena) a vyrovná světelné podmínky ve zbytku snímku. Algoritmus funguje na principu odečtení jasového gradientu ze snímku. Jasový gradient je odhadnut rozmazáním snímku, tak aby nebyly rozpoznatelné obrazové struktury, ale pouze změny intenzity jasu. Rozmazání probíhá opakovaným aplikováním normalizovaného krabicového filtru na snímek. Ideální rozmazání je takové, aby nebyl rozeznatelný text, ale oblasti se stíny byly rozeznatelné. Pokud bude rozmazání moc intenzivní, vyrovnají se světelné podmínky pouze v rámci celého snímku, ne menších oblastí. Pokud málo intenzivní, algoritmus se bude chovat i k textu jako stínu a vyrovnává jeho hodnotu k okolí. Postup odstranění jasového gradientu je možné pro dosažení spolehlivějších výsledků iterativně na snímku opakovat.

Po odstranění stínů a některých degradací snímku projevujících se ztmavením oblastí lze



■ **Obrázek 4.10** Popis principu vytváření „drops“ při mapování do jemnější mřížky [8]

přistoupit k hledání masky segmentovaného textu. Otsuova metoda dosahuje nejlepších výsledků pro bimodální rozdělení. Vzhledem k tomu, že dokument je dvoubarevný, a pokud byly stíny ze snímku správně odstraněny odstraněním jasového gradientu, rozdělení by se mělo podobat bimodálnímu, proto je zvolena k binarizaci tato metoda.

4.2.2.3 Důvěryhodnost interpolované hodnoty

Výpočet váhy dle pozice vzhledem k interpolaci je inspirovaný přístupem Fruchtera a Hooka v [8]. Ti mapované pixely zmenšili (v práci nazýváno jako „drops“ – kapky) na půlku původní velikosti pixelu. Princip je zobrazen v obrázku 4.10. Podobný přístup je zde navržen pro získání váhy jednotlivých pixelů snímků. Předpoklad je, že každý pixel prochází aproximací polohy/hodnoty. Nepřesnosti tímto způsobem vznikající by měly být více výraznější k okrajům původních pixelů, a to nejen kvůli aproximaci, ale i interpolaci a důsledcích rozmazání. Proto se hodnotám v těchto krajích dává menší důraz ve výsledném snímku. Hodnota finálního váhy pixelu v jednotlivých snímcích se následně normalizuje, aby se suma vah rovnala 1 a snímek byl složen z pixelů za využití nejspolehlivější informace. Tento přístup zaručí, že každý pixel výsledného snímku má stejnou intenzitu.

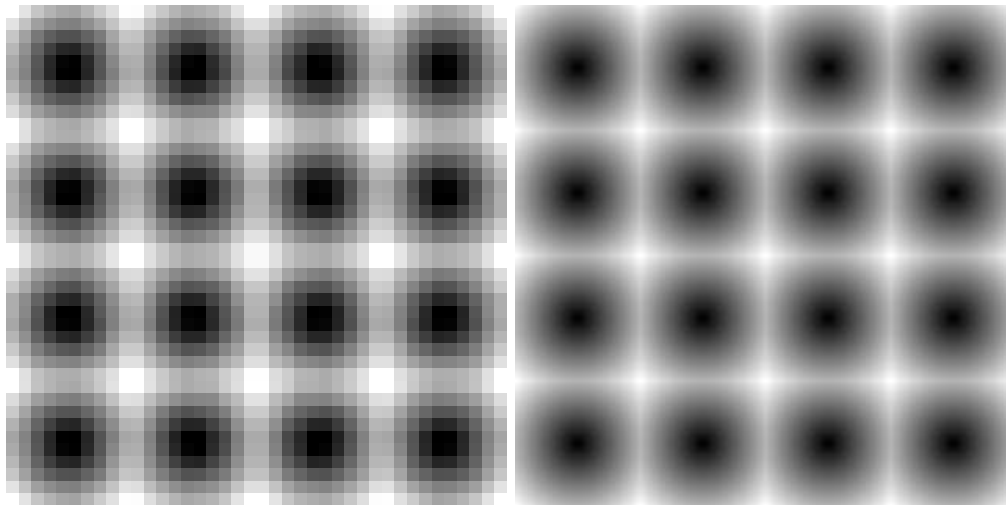
Prvně se tedy získají hodnoty vzdálenosti od středu pixelu v jemnější mřížce. Jako šířka původního pixelu je brána hodnota 1. Hodnoty vzdálenosti pak nabývají hodnot z rozsahu $\left(0, \sqrt{\frac{1}{2}}\right)$. Princip je zobrazen v obrázku 4.11. Tato vzdálenostní maska je pak mapována stejně jako jí odpovídající snímek. Metoda interpolace je využita lineární, ta zajistí zachování linearitu vzdálenosti od středu, tedy aproximuje reálnou vzdálenost po transformaci (mapování). Váha pixelu odpovídá hodnotě Gaussovy funkce ve vzdálenosti od středu. Rozdělení má střední hodnotu 0 a rozptyl σ^2 je parametr, který lze optimalizovat. Tento princip je zobrazen v obrázku 4.12.

4.3 Volba knihoven pro implementaci

K implementaci jednotlivých kroků lze využít některých již implementovaných algoritmů ve volně dostupných knihovnách. Tyto knihovny jsou pro implementaci algoritmů počítačového vidění často esenciální, jelikož jsou velmi efektivní a optimalizované. V praktické části práce budou proto využity algoritmy z následujících knihoven:

OpenCV Implementuje základní a některé pokročilejší nástroje pro počítačové vidění a zpracování obrazu. Je napsaná v C++ a optimalizovaná, aby výpočty mohly probíhat v reálném čase. Knihovna je multiplatformní a má rozhraní pro Python, C++, Javu a MATLAB.

NumPy Určená především pro práci s maticemi a vektory a matematickými operacemi nad



■ **Obrázek 4.11** Hodnoty vzdálenosti od středu odpovídajícího pixelu v jemnějších mřížkách. Hodnoty jsou pro vizualizaci normalizované

nimi. Dokáže efektivně pracovat s vícedimenzionálními datovými strukturami (tenzory). Obsahuje také implementované funkce pro statistiku, lineární algebru a Fourierovu transformaci.

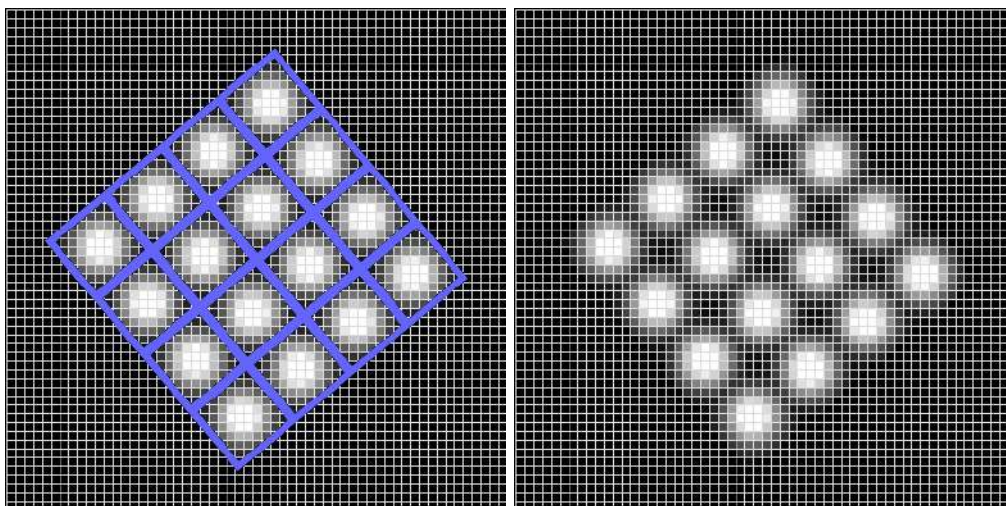
SciPy Postavená na knihovně NumPy, ke které přidává pokročilejší analýzu a práci s daty. Věnuje se oblastem interpolace, optimalizace, integraci, lineárním rovnicím a dalším.

Scikit-learn (sklearn) Knihovna pro strojové učení. Implementuje funkce pro zpracování a analýzu dat i samotné učení. Nabízí konzistentní rozhraní pro modely strojového učení jako jsou regrese, klasifikace, shlukování a další.

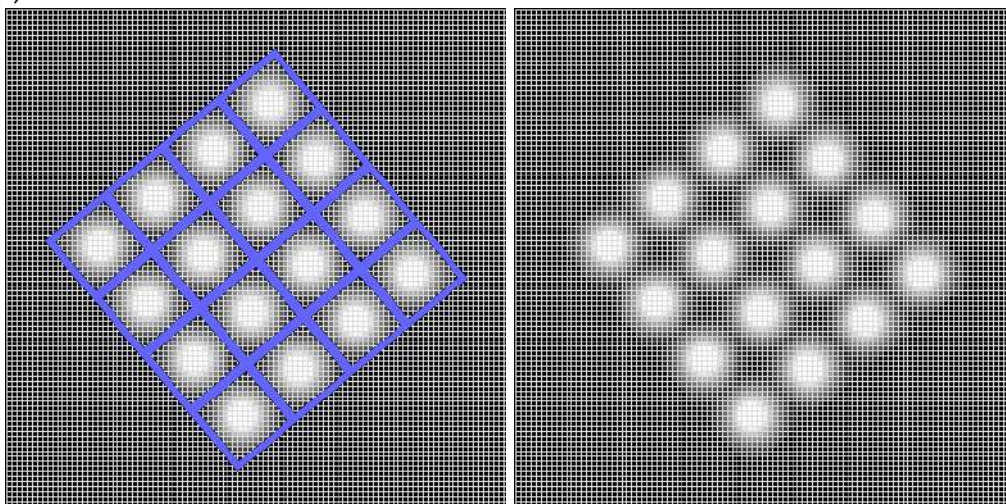
Jiwer Nabízí nástroje pro výpočet chybovosti v zpracování textu. Vypočítává míru podobnosti dvou řetězců pomocí *Levenshteinovy vzdálenosti* (minimální editační vzdálenost). Implementované algoritmy jsou například CER, WER nebo SER (*Sentence Error Rate*).

PyTesseract Jedná se o rozhraní pro práci s *Tesseract OCR*. Tesseract OCR je nástroj pro detekci a extrakci textu z obrázků.

Improutils Implementuje obecné podpůrné nástroje pro práci se snímky a jejich zpracování. Je zaměřena na využití v oblasti strojového vidění. Vznikla za účelem využití při výuce v oblasti strojového vidění a zpracování na FIT ČVUT v Praze.



(a) Hrubá mřížka



(b) Jemnější mřížka

■ **Obrázek 4.12** Váhy mapovaných pixelů hrubé mřížky do jemnějších mřížek. Váhy jsou pro vizualizaci normalizované

Kapitola 5

Praktická část

Cílem praktické části je získání obrazového datasetu a implementaci navržených algoritmů. Algoritmy byly realizovány v programovacím jazyce Python za využití *Jupyter Notebook*.

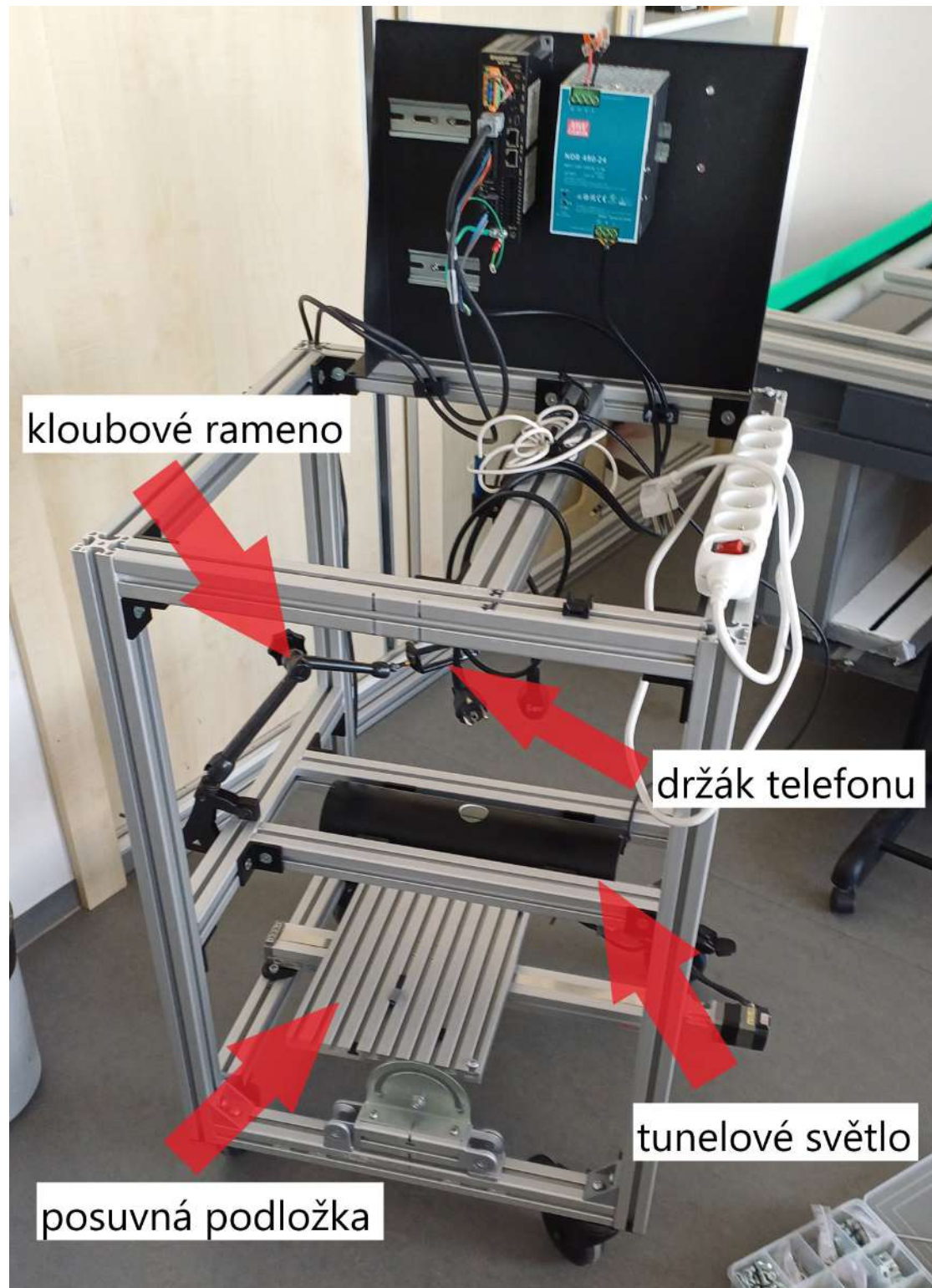
Pro každý doklad je nasnímáno alespoň 8 snímků a video skládající se alespoň z 8 snímků. Sám navržený algoritmus SR není omezen na velikost vstupu (počet snímků, z kterých se skládá výsledný snímek), ale podléhá vysokým výpočetním nárokům. Algoritmus tedy dostává na vstup již pouze „rozumný“ počet snímků (zvolen počet 10 snímků) nebo jsou v určité fázi algoritmu vybrány jen některé snímky. Druhý postup může vést k lepším výsledkům, protože výběr je založený na metrice kvality mapování.

5.1 Dataset

Dataset byl získán ve spolupráci s Laboratoří zpracování obrazu (ImproLab) FIT ČVUT v Praze. Pro získání byla využita ideální snímávací soustava popsaná v 4.1.2). Skládá se ze snímků 18 dokladů, ke kterým byly ručně sepsány odpovídající referenční texty.

Snímávací soustava se skládala z bílého matného pozadí, snímaného objektu, osvětlení typu *dome* – zde tunelové světlo – a mobilního telefonu s kamerou (dále bude referováno pouze jako *kamerě*). Kamera byla umístěna v takové vzdálenosti, aby byl celý objekt v zorném poli kamery a navíc byl nejbližší i nejvzdálenější bod dokumentu od senzoru zaostřený. Vzdálenost senzoru od podložky byla 30 cm a v průběhu celého snímání byla konstantní. K pohybu dokladu na snímku byl využit automatický posuvník, který umožnil zajištění konstantní rychlosti. Snímávací soustava je zobrazena v obrázku 5.1. Snímky byly získávány při rychlostech posuvníku 30 mm/s, 50 mm/s a 60 mm/s. Pro všechny rychlosti byly pořízeny snímky sériovým snímáním i krátkým videem (*živá fotografie*). Pro každý dokument je nasnímáno 10 snímků během pohybu pro oba typy snímání. Ilustrativní podoba datasetu je zobrazena v obrázku 5.2.

Pro snímání byla použita kamera mobilního telefonu značky Samsung, model A34 (přesné označení: SM-A346B/DSN). Na tomto modelu jsou umístěné tři samostatné senzory s objektivy: hlavní, širokoúhlý a makro. K snímání je využit hlavní snímač a objektiv. Jejich specifikace jsou uvedeny v tabulce 5.1. Parametry kamery při snímání byly nastaveny kamerou automaticky. Hodnota ISO byla ve všech snímcích 50, díky dostatečnému nasvětlení. Délka expozice se pohybovala proměnlivě mezi hodnotami 1/500 s až 1/600 s. Kamera v běžném nastavení fotí snímky o rozlišení 12 MPx (3000 x 4000 pixelů), díky kombinaci pixelů 48 MPx snímače do výsledného snímku. V případě pořizování *živé fotografie* je rozlišení snímků 1080 x 1440 pixelů.



■ Obrázek 5.1 Snímací soustava pro získání datasetu



■ **Obrázek 5.2** Ilustrativní podoba datasetu. Pro každý dokument je nasnímáno alespoň 10 snímků během pohybu

■ **Tabulka 5.1** Specifikace hlavní kamery modelu SM-A346B/DSN [32, 33]

clonové číslo	F1.8
rozlišení	48 MPx
velikost senzoru	1/2"
ohnisková vzdálenost objektivu	26 mm
horizontální úhel snímání	69°
vertikální úhel snímání	54°
stabilizace obrazu	optická
zoom (přiblížení)	digitální
přisvětlovací dioda	ano

5.2 Implementace metod

V rámci práce vznikla knihovna implementovaných algoritmů. Ta se nachází v souboru *lib.ipynb*, který lze nalézt v příloze této práce, a je exportovaná i jako modul jazyka Python *lib.py*. Knihovna implementuje účelové algoritmy, ale i komplexní algoritmy – *pipeline*, které zpracují více dat nebo provádí více kroků celého SR algoritmu (tzv. *end-to-end*).

Následující podsekcce se věnují rozboru a vysvětlení implementace a výsledků některých algoritmů rozdělených podle kategorií, do kterých je lze vůči funkci v SR algoritmu zařadit. Jsou vysvětlené pouze implementačně zajímavé nebo klíčové funkce, případně jejich části. Pro kompletní implementaci i s dokumentací je k práci přiložena kompletní knihovna.

5.2.1 Načtení a zpracování

Pro načtení dat jsou využity funkce z *OpenCv* (načteno jako *cv*). Pro načtení snímků je použita funkce *cv.imread* v cyklu na jednotlivé soubory (snímky) v adresáři zadaném cestou. Pro načtení dat z video-souboru, je využita funkce *cv.VideoCapture*, která nahraje soubor a následně se na navráceném objektu v cyklu volá metoda *read*, které návratová hodnota jsou jednotlivé snímky. Takto je video rozloženo na jednotlivé snímky. Případně je využito i možnosti video-souboru rozložit do adresáře na jednotlivé snímky funkcí *video_to_burst* a následně pracovat s tímto adresářem. K zmenšení paměťových nároků je načítání kolekcí snímků řešeno přes generátor – načtení dat do paměti se děje až v okamžiku kdy jsou data využívána.

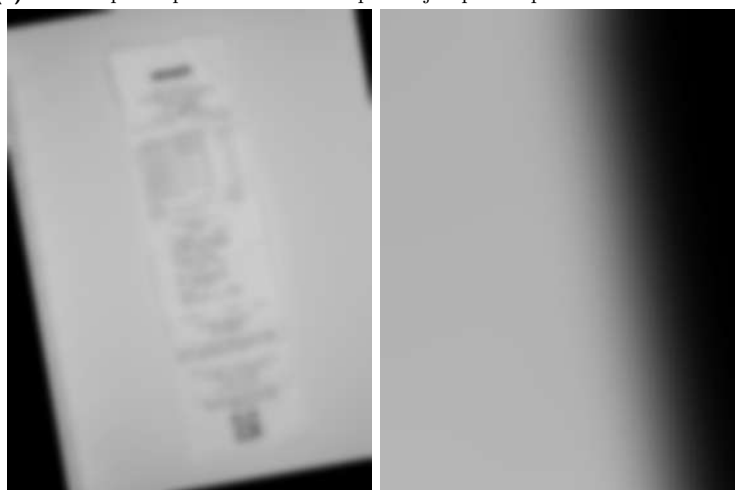
Následuje zpracování snímků. Prvním krokem ve zpracování je převod snímků do stupňů šedi. Pro tento převod je využita funkce *to_gray* z knihovny *Improutils*. Z implementace této funkce lze zjistit, že využívá k převodu *cv.cvtColor(img_bgr, cv.COLOR_BGR2GRAY)*, kde *img_bgr* je snímek k převodu. Tato funkce využívá pro převod výpočet 3.4 definovaný v sekci 3.2. Pro převod celého seznamu snímků je implementována metoda *get_bw_imgs*.

Dále je implementována funkce pro rozmazání snímku *blur*. Jedná se o nadstavbu metody *cv.blur*, která umožňuje pouze jednu iteraci rozmazání (konvoluci s kernelem pro rozmazání) a nelze v ní zvolit masku, aby bylo možné provést rozmazání pouze v některých oblastech snímku. Navržená funkce v cyklu aplikuje rozmazání a maskuje hodnoty, které nemají mít vliv na tento proces. Maskování je založené na navrženém řešení v [34]. Toto řešení využívá rozmazání masky společně se snímkem a následně vydělení hodnot snímku hodnotami této masky v místech, kde byla původní maska nenulová. Význam této funkcionality je v případech, kdy snímek po mapování má rámečky s nulovou hodnotou, které nenesou informaci a nemají ovlivňovat kraj mapovaného snímku. Podstata myšlenky tedy je, že do těchto krajů bude při rozmazání vnášena informace z černých rámečků stejnou intenzitou ve snímku i v masce. Práci funkce lze vidět na obrázku 5.3.

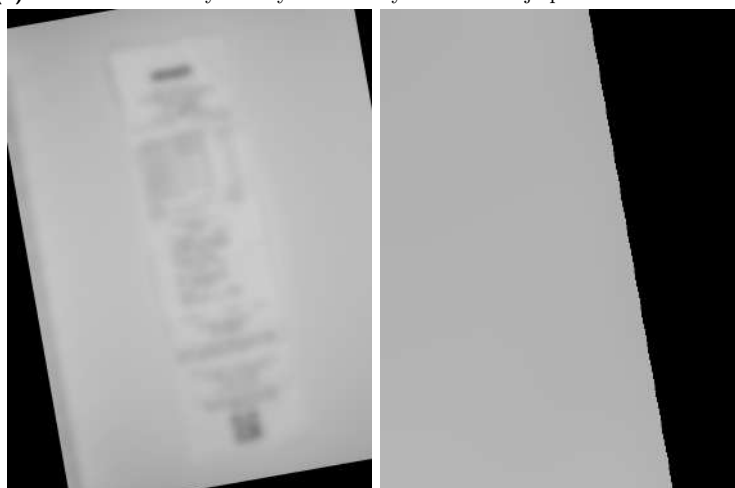
Dalším implementovaným krokem zpracování snímku, který je využit pouze pro účely segmentace textu, je odstranění stínů pomocí jasového gradientu funkcí *remove_shadows*. V cyklu se provádí samotný výpočet jasového gradientu a jeho odstranění. K získání jas. gradientu dochází



(a) Snímek po mapování a maska odpovídající poloze původního snímku



(b) Snímek rozmazaný bez využití masky a detail kraje původního snímku



(c) Snímek rozmazaný s využitím masky a detail kraje původního snímku

■ **Obrázek 5.3** Rozmazání za využití maskování hodnot

k rozmazání snímku pomocí funkce `blur` s zadanou velikostí kernelu a maskou. Jako maska je zde využita maska podle polohy snímku po jeho mapování. Ta je získána transformací jedničkové masky stejnou trans. maticí jako mapovaný snímek s využitím stejné interpolační metody a prahováním nenulových hodnot. Velikost kernelu je zde získána průběžným testováním při vývoji jako pro většinu snímků dostačující hodnota, která se dále škáluje s globálním koeficientem určujícím, o kolik je výsledná mřížka po aplikaci SR jemnější. Takto získaný jasový gradient je následně převrácen (získán negativ) a přičten k původnímu snímku. Získaný snímek má hodnoty z pohledu jasového gradientu vyrovnanější, ale nemá hodnoty v rozsahu 0 až 255, proto je následně ještě potřeba snímek do rozsahu normalizovat. Pokud je zadaná maska, pak se ještě před tím nastaví hodnoty mimo ní na průměrnou hodnotu ve snímku, aby neovlivňovaly proces normalizace. Opakováním tohoto postupu se oblasti v snímku přesněji vyrovnávají. Nakonec je dobré odstranit tmavé rámečky mimo masku, aby nemanipulovaly příliš metody prahování. Tato místa se proto nastaví na nejčastější hodnotu (*medián*) v snímku bez hodnot nacházejících se mimo masku. Díky odstranění jas. gradientu se předpokládá vyrovnání pozadí a okolí textu do podobné barvy, a ta by měla ve snímku převládat. Výsledky odstranění stínů podle intenzity rozmazání a velikosti kernelu k rozmazání jsou zobrazené v obrázku 5.4. Lze vypořadovat, že nedostatečné rozmazání dělá snímek nečitelným a přílišné rozmazání nemá požadovaný efekt odstranění stínů. Volba většího kernelu má zase negativní vliv na odstranění menších stínových oblastí. Spolehlivější tedy je snímek rozmazat více s menším kernelem a celý algoritmus opakovat vícekrát. Výsledky tohoto přístupu jsou v obrázku 5.5.

Poslední funkcí pro zpracování je binarizační funkce `otsu`. Ta má za účel provést prahování na snímku, které oddělí text od zbytku snímku. Využívá se Otsuova metoda. Tato funkce je pouze obalem pro funkci

```
cv.threshold(img,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
```

kteřá provádí nad snímek `img` Otsuovo prahování do hodnot 0 a 255. Přiblížení k *bimodálnímu rozdělení* potvrzuje zlepšení segmentace textu, která je pravidelná a správná (zobrazeno v obrázku 5.6).

5.2.2 Metriky snímků

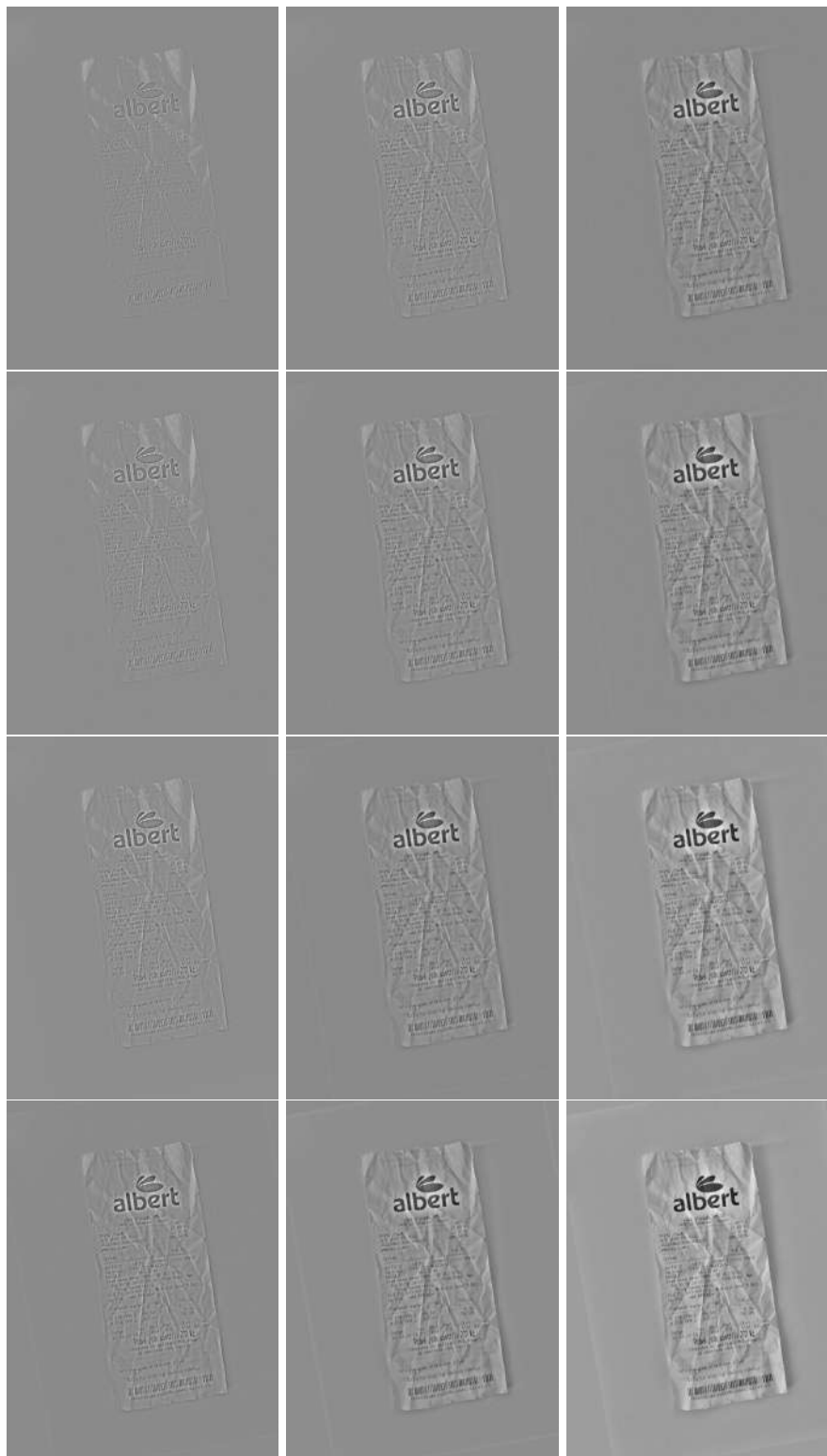
V sekci 3.3 byly teoreticky popsány metriky kvality snímku v kontextu ostatních snímků. Metoda pro odhad míry šumu zakládá na konvoluci snímku s definovaným kernelem. K tomu je využita funkce `convolve2d` z knihovny *Scipy*. Metoda pro odhad míry rozmazání je založená na aplikaci Laplaceova operátoru. K tomu je využita funkce `cv.Laplacian`, která aplikaci provede. Následuje využití metody `var` z knihovny *Numpy* pro získání rozptylu hodnot.

Poslední metrika, která je implementována pro účely odhadu podobnosti snímků s referenčním snímek a využívána při hledání báze, je funkce `get_diff`. Ta vypočte sumu absolutního rozdílu dvou snímků v oblasti dané maskou pro každý snímek vůči referenčnímu. Tyto sumy jsou následně sečteny a použity jako odhad podobnosti kolekce snímků s referenčním snímek.

5.2.3 Mapování snímků

Funkce `warp_imgs_to_base` zaobaluje celý proces mapování snímků do známé báze. Provede se detekce a párování KP, filtrování párů, hledání transformační matice a nakonec samotná transformace.

Pro mapování snímků je potřeba transformační matice. Její výpočet vyžaduje znalost zdrojových a cílových bodů. Pro získání těchto bodů je využito hledání KP pomocí algoritmu SIFT. Pro tento účel je využito třídy `cv.SIFT`. Dochází tedy v cyklu pro každý snímek k detekování KP, výpočtu deskriptorů a párování těchto deskriptorů. Vizualizace nalezených KP na příkladu lze vidět v obrázku 5.7. Rozložení hodnoty počtu nalezených KP ve snímcích z datasetu lze nalézt v grafu 5.8. K provedení filtrování KP je tedy bodů dostatek.



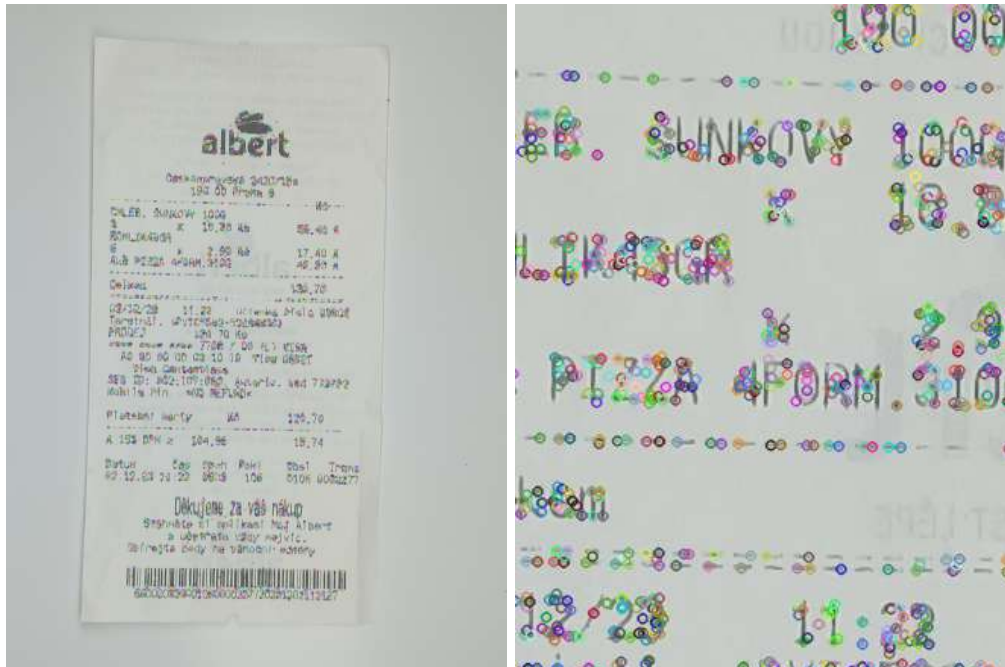
■ **Obrázek 5.4** Výsledky algoritmu odstranění stínů. Zleva doprava se zvyšuje velikost *kernelu* mazání. Ze shora dolů se zvyšuje intenzita rozmazání. Původní snímek lze nalézt ve vizualizaci 5.5



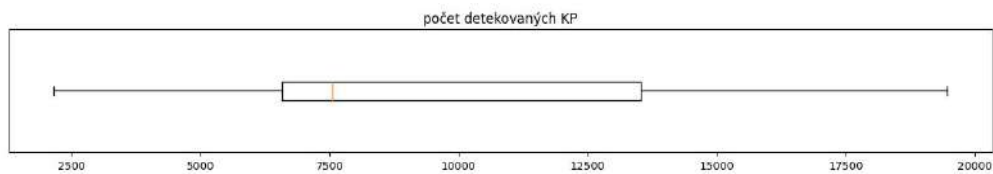
■ **Obrázek 5.5** Výsledky algoritmu odstranění stínů. Zleva doprava se zvyšuje počet iterací algoritmu (kroky po 1 iteraci). První snímek je původní (počet iterací je 0)



■ **Obrázek 5.6** Výsledky binarizace Otsuovou metodou. Vlevo je využit snímek bez odstranění stínů. Vpravo je využito posledního snímku z obrázku 5.5 – využitý algoritmus odstranění stínů s 3 iteracemi



■ **Obrázek 5.7** Nalezené klíčové body ve snímku. Vlevo jsou všechny detekované body. Vpravo je detail snímku



■ **Obrázek 5.8** Rozložení hodnoty počtu nalezených KP

Pokud mají být filtrovány páry podle spolehlivosti (jistoty párování), pak je potřeba využít metody `knnMatch(des1,des2,k=2)` z modulu `cv.BFMatcher`, která při párování nalezne první i druhý nejpodobnější deskriptor. Argumenty `des1` a `des2` jsou deskriptory bazového a mapovaného snímku. Pokud by kvůli nedostatku detekovatelných bodů nebylo možné získat požadované minimum bodů, pak páry nejsou filtrovány a je volána metoda `match(des1,des2)` ze stejného modulu. Vzhledem k homogenitě snímku, a tedy malé unikátnosti KP, je potřeba filtrovat páry. Tomu se věnuje funkce `filter_matches`. Filtrování probíhá metodou *Lowe's ratio test*, jeho výsledek lze vidět v obrázku 5.9.

Následně jsou cílové body přeškálovány koeficientem zjemnění, aby mapování probíhalo do jemnější mřížky. Pak již lze vypočítat transformační matici. Hledání transformační matice probíhá ve funkci `find_H`. Využívá se postup vyloučení chybných párů a následný výpočet matice tak, aby transformace způsobila nejmenší odchylku mezi transformovaným zdrojovým bodem a cílovým bodem v sumě kvadrátů. Vyloučení chybných párů a výpočet trans. matice zajišťuje funkce `get_transform_matrix_reject_outliers`. Ta nejprve využije algoritmu RANSAC při výpočtu trans. matice ve funkci `cv.findHomography`, která navrácí i masku bodů. Maskou je možné určit body, které byly algoritmem označeny jako *inliers* – neodlehle. Z těchto neodlehlejších bodů je následně vypočítána trans. matice, která je získána algoritmem *least squares* (metoda nejmenších čtverců). Motivace tohoto postupu je popsána v sekci 4.2.1. Počet neodlehlejších bodů vůči všem bodům je dále využit jako míra *spolehlivosti mapování*. V případě že není získána požadovaná matice, například kvůli nedostatku bodů pro výpočet, je pozvolně zvolňován práh, který určuje, které body jsou odlehle. S tím je ale zároveň snižována míra spolehlivosti mapování.

Při mapování je vypočítána míra *spolehlivosti interpolace*. K tomuto výpočtu je využita funkce `get_inter_distance`, která vytvoří matici vzdáleností k nejbližšímu středu pixelu hrubé mřížky. Toho je dosaženo vytvořením matice, která obsahuje jako hodnoty pozice v rámci původního pixelu (hodnoty od 0 do 1) v obou osách. Následně se od hodnot v obou osách odečte 0,5, aby tvořili v rámci původního pixelu posloupnost od -0,5 do 0,5. Tak odpovídá absolutní hodnota vzdálenosti od středu v dané ose. K získání vzdálenosti je využito *Euklidovy vzdálenosti*. K využití této informace jako míry spolehlivosti, je na hodnoty aplikována *Gaussova funkce* se střední hodnotou 0. Hodnota rozptylu této funkce je parametr, jehož změna určuje rozložení míry vzhledem k vzdálenosti. Výsledek s různými hodnotami rozptylu lze vidět na obrázku 5.10.

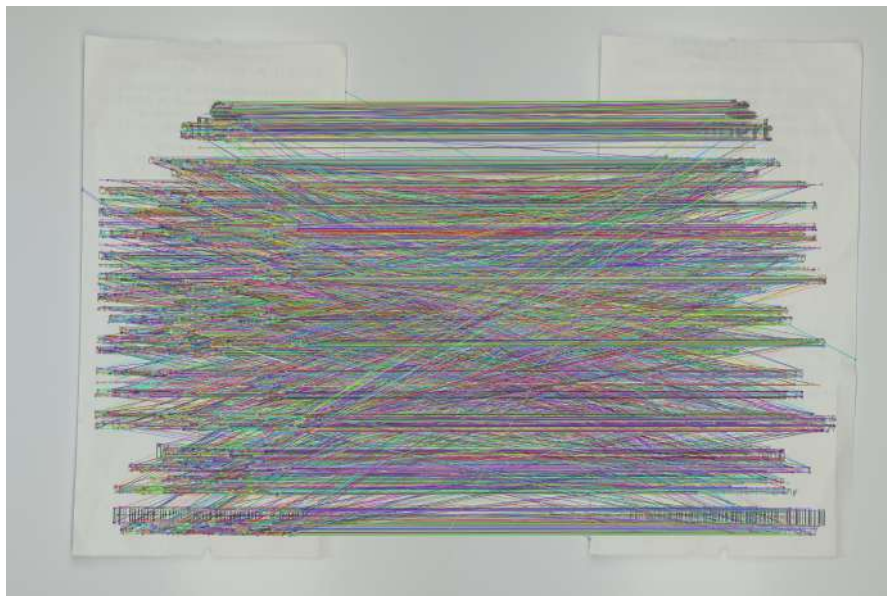
Pro výběr báze je využita funkce `detect_base_index`. Funkce vyzkouší každý snímek zvolit jako bázi, provede na něj mapování ostatních a vypočítá metriku `get_diff`, která slouží k vypočítání absolutní akumulované vzdálenosti mapovaných snímků od zvolené báze. Jako ideální báze je vybrána taková, která měla tuto metriku nejnižší – měla nejmenší akumulovanou vzdálenost – a lze tak naivně předpokládat, že způsobí nejmenší chybu.

5.2.4 Kombinování dat ze snímků

Snímky jsou váhovány na úrovni dat v nich i v rámci celých snímků v kolekci. Tyto váhy jsou spolu využity tvorbou matice vah pro každý snímek. Každá matice vznikne zkombinováním váhy dat ve snímku a váhy snímku v rámci kolekce, a to pro každou míru, která udává váhu nějaké informace. Procesu získání vah snímků v rámci kolekce se věnuje funkce `get_w`. Ta kombinuje váhy podle jednotlivých metrik. Kombinace je také vážená koeficienty, které kombinují váhy šumu, rozmazání a *spolehlivosti mapování*.

Funkce `get_w_minimize_noise` získává váhy minimalizací metriky výsledného snímku získaného kombinací snímků. K minimalizaci je využita metrika `estimate_noise` – dříve popsaná metrika NOISE – ve funkci `estimate_noise_fnc`, která je minimalizována. Funkce `estimate_noise_fnc` provádí kombinaci snímků vahami z parametru, na které metriku vypočítává. K minimalizaci je využita funkce `minimize` z knihovny *Scipy*. Té jsou předány argumenty nastavující aby parametry minimalizované funkce (váhy) byly kladné a jejich suma byla rovna 1. Metodu, kterou minimalizace probíhá, si volí funkce sama.

K získání vah podle rozmazání slouží funkce `get_w_blur`, která využívá metriku rozmazání

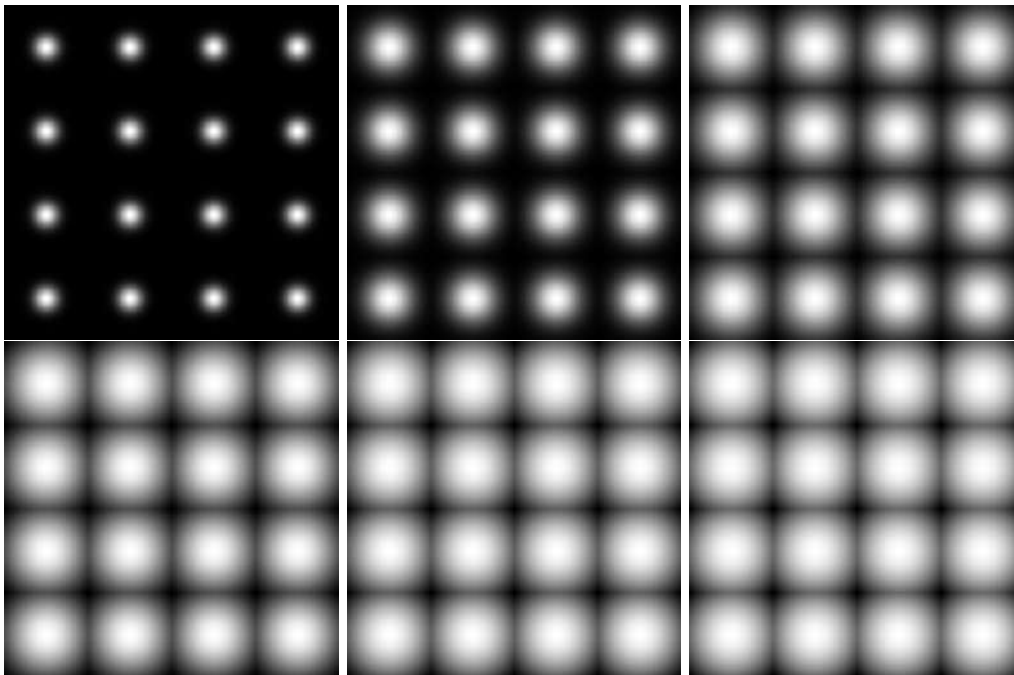


(a) Všechny páry. Celkem 7140 párů s průměrnou vzdáleností deskriptorů 133,45



(b) Filtrované páry. Celkem 3103 párů s průměrnou vzdáleností deskriptorů 90,34

■ **Obrázek 5.9** Vizualizace párů klíčových bodů mezi dvěma snímky



■ **Obrázek 5.10** Matice s mírou spolehlivosti hodnot s rozdílným rozptylem ve výpočtu. Hodnoty matice jsou normalizovány pro vizualizaci

BLUR. Hodnota metriky je odmocněna, aby se vyrovnaly rozdíly ve vyšších hodnotách. Následně jsou hodnoty metriky napříč snímky normalizovány tak, aby jejich suma byla rovna 1.

Dalším způsobem váhování informace je váhování v rámci snímku. To je již dříve vypočítaná matice *spolehlivosti interpolate*. Další způsob je segmentace textu váhováním. Tedy váhy jsou nenulové pouze tam, kde se vyskytuje na snímku text. Segmentace vede ke vzniku masky o hodnotách v rozsahu $\langle 0, 1 \rangle$, které označují přítomnost textu na této pozici ve snímku. Masku nemá binární hodnoty, jelikož obsahuje i blízké okolí textu, kterému dává menší váhu. Ze snímku se nejprve odstraní stíny metodou popsanou v 5.2.1 (funkce `remove_shadows`). Snímek je pak prahován Otsuovou metodou a následně získán negativ, což vede k tomu, že text je v masce označen hodnotou 255. Na masce je provedena morfologická operace otevření a následně dilatace za využití funkce `cv.morphologyEx` s kernely vytvořenými funkcí `cv.getStructuringElement`. Jako poslední operace se provede rozmazání, které zajistí nenulovou masku i v blízkém okolí textu, ale s postupně klesající hodnotou. Masku je pak normalizována do rozsahu 0 až 1 za využití funkce `minmax_scale` z knihovny *Sklearn*. Pokud je využita segmentace textu, pak je vylepšeno rozlišení pouze v těchto oblastech a zbytek snímku má nulové hodnoty. Tyto nulové hodnoty jsou proto nahrazeny hodnotami z původního bazového snímku, o nižším rozlišení.

5.2.5 SR pipeline

Předešlé metody jsou spojeny do jednoho komplexního algoritmu – *pipeline* – funkce `sr_pipeline`. Ta na vstupu předpokládá kolekci snímků, v kterých rozezná funkcí `warp_imgs_detect_base` bázi a namapuje na ní ostatní snímky z kolekce. Následně se vyfiltruje pouze limitem určený požadovaný počet snímků vybraných podle míry *spolehlivosti mapování*. Následuje získání vah podle metrik. Kolekce masek vah je znormalizována v ose napříč maskami, tedy pro každý pixel snímku platí $\sum_{i=1}^m M_i(x, y) = 1$, kde m je velikost kolekce a M_i je maska vah snímku na pozici i v kolekci M . Následuje samotná vážená kombinace snímků, v které se parametrem rozhodne zda se aplikuje na celý snímek nebo jen segmentovaný text. Zkombinovaný snímek pak ještě může být

volbou parametru normalizován za použití funkce `normalize` z *Improutils*.

Vzhledem k velkému množství parametrů, které je možné nastavit a ovlivňují funkce v průběhu celého algoritmu, je možné do *pipeline* předat konfigurační slovník. Tam lze definovat hodnoty všech parametrů, které lze v algoritmech nastavit.

5.2.6 OCR

Pro evaluaci je využito OCR a metriky správnosti rozeznaného textu. K rozeznání textu ze snímku je využito metody `image_to_string` z knihovny *Pytesseract*, která využívá českou jazykovou sadu. Není využito vylepšení nebo hlubší zpracování snímku modelem, aby bylo možné přesněji odhadnout jaký vliv na čitelnost má navržený algoritmus.

Pro metriku podobnosti textu je využito knihovny *Jiwer* a z ní metod `wer` a `cer`, které vypočítají podobnost na úrovni slov a znaků. Pro co nejspolehlivější porovnání musí být text nejdříve předzpracován. To zahrnuje odstranění všech netisknutelných znaků (mezery, nové řádky, apod.) a jejich nahrazení mezerou. Následně je více mezer u sebe spojeno do jedné. Jsou odstraněny mezery ze začátku a konce textu a všechny znaky jsou převedeny na *minuskule* (malá písmena). Pro zpracování k výpočtu metriky *WER* je následně text rozdělen na seznam slov, k výpočtu metriky *CER* na seznam znaků. Metody `wer` a `cer` pak vypočítají odpovídající metriky.

Evaluace výsledků

V algoritmu je možné zvolit množství prahů a koeficientů a zvolit různé varianty využití dílčích algoritmů. Tato kapitola je proto věnována evaluaci výsledků získaných pomocí různých konfigurací celého procesu – SR algoritmu. Různé konfigurace mají různý dopad na metriky zlepšení čitelnosti – zlepšení WER a CER. Dále jsou zhodnoceny vizuální dopady, tedy dopady vysvětlitelné pouhým pohledem na snímek.

6.1 OCR čitelnost

K evaluaci je využito míry zlepšení metrik WER a CER (*enhancement*). To je vypočítáno jako rozdíl míry bazového snímku a míry SR vylepšeného snímku. Kladná hodnota proto odpovídá zlepšení míry. Výsledky jsou zobrazeny průběžně v grafech i na konci kapitoly v tabulkách 6.2 a 6.1.

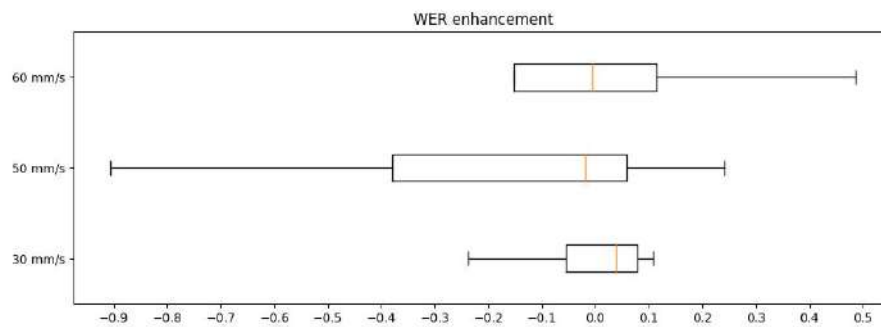
Nejprve je třeba určit vliv rychlosti snímání na výsledky. Porovnání mezi rychlostmi je zobrazené v grafech 6.2 a 6.1. Zde lze pozorovat podobné výsledky mezi rychlostmi. Při vyšších rychlostech grafy naznačují možnosti dosáhnout vyššího zlepšení. To může být způsobeno tím, že snímky o vyšší kvalitě (snímané při nižších rychlostech) mají menší potenciál k dalšímu vylepšení. Změny ale nejsou dostatečně výrazné, proto se předpokládá, že změny rychlosti neměly na algoritmus významný vliv a je dále využívána pouze část datasetu vztahující se k rychlosti 30 mm/s.

První konfigurace využívá všech dříve popsaných metod kromě segmentace textu – je zvyšováno rozlišení celého snímku. Jako interpolační metoda je zvolena metoda nejbližších sousedů. Rozložení zlepšení na snímcích pořízených při rychlosti 30 mm/s je zobrazené v grafu 6.3. Lze vidět, že většina hodnot je kladná, tedy převážně dochází ke zlepšení čitelnosti pro OCR. Lepších výsledků navíc dosahují snímky pořízené sériovým snímáním.

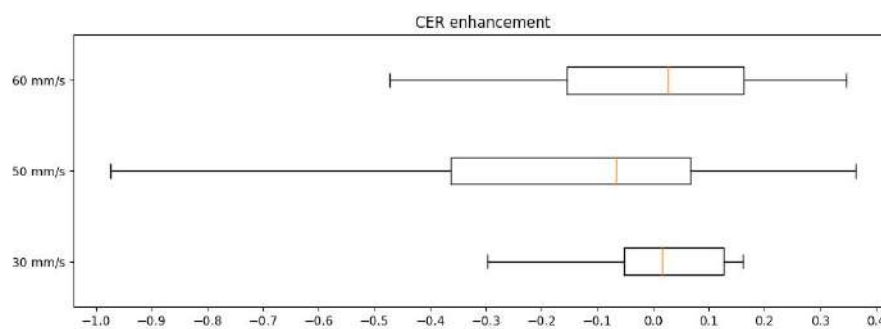
Při využití lineární interpolace dochází k možnosti vyššího zlepšení. To je pravděpodobně způsobeno lepším využitím *subpixel* informace. Obecně jsou ale výsledky srovnatelné. Zobrazeno v grafu 6.4.

V předchozích porovnáních lze vysledovat, že výsledky snímků z video-souboru jsou obecně horší. Při dalším srovnávání konfigurací se proto pracuje pouze s částí datasetu pořízenou sériovým snímáním. Lze předpokládat, že změny konfigurace budou mít shodný vliv (zlepšení nebo zhoršení) i na část datasetu pořízenou videem.

Pokud je v algoritmu použit krok s segmentací textu a vylepšováním pouze této oblasti, pak dochází k lehkému zhoršení WER, jak pro interpolaci metodou nejbližších sousedů, tak i pro lineární interpolaci, jak lze vidět v grafu 6.6. U vylepšení CER dochází k zlepšení u lineární interpolace, zatímco u metody nejbližších sousedů dochází k zhoršení, jak je vidět z grafu 6.5.



(a) zlepšení WER



(b) zlepšení CER

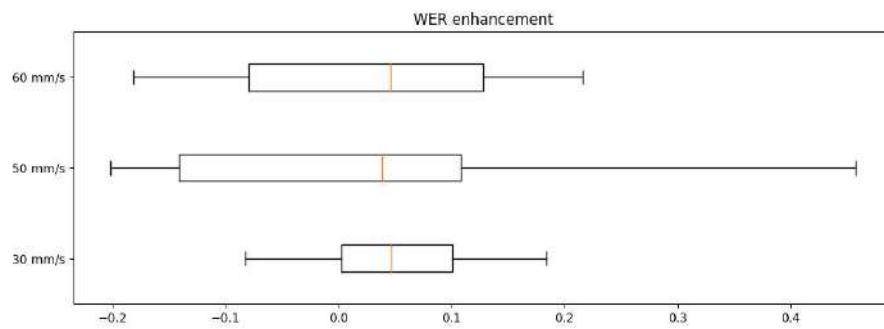
■ **Obrázek 6.1** Rozložení hodnot zlepšení míry WER/CER podle rychlosti snímání. K snímání bylo využito videa

■ **Tabulka 6.1** Kvartily hodnot metriky zlepšení CER podle rychlosti posunu dokumentu. Konfigurace: interpolace metodou nej. soused, nevyužita segmentace textu

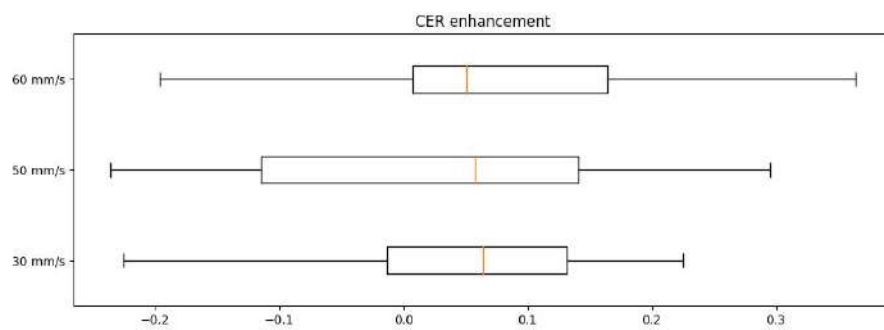
typ snímání	rychlost	Q1	Q2	Q3
série	30 mm/s	0.0024832	0.046639	0.10105
video	30 mm/s	-0.054925	0.038558	0.0777
série	50 mm/s	-0.14114	0.038558	0.10822
video	50 mm/s	-0.37911	-0.018245	0.059071
série	60 mm/s	-0.079594	0.046447	0.1279
video	60 mm/s	-0.15213	-0.0049957	0.11393

■ **Tabulka 6.2** Kvartily hodnot metriky zlepšení CER podle konfigurace pro rychlost 30 mm/s a snímání sériovým snímáním

interpolace	seg. text	Q1	Q2	Q3
metodou nej. sousedů	ne	0.0024832	0.046639	0.10105
lineární	ne	0.000363	0.029004	0.15329
metodou nej. sousedů	ano	-0.04984	0.029872	0.10971
lineární	ano	-0.0077342	0.042477	0.12553

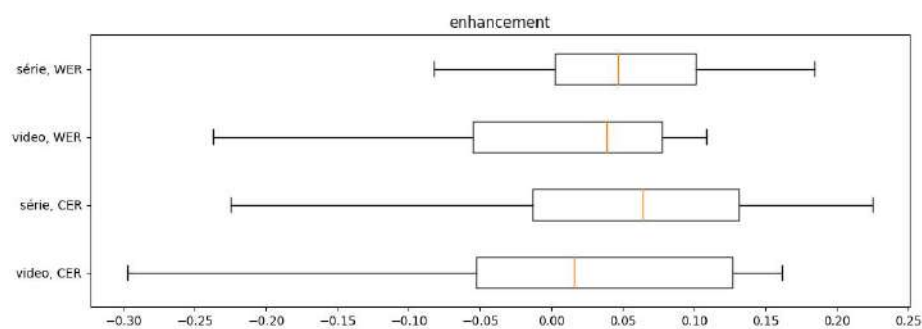


(a) zlepšení WER

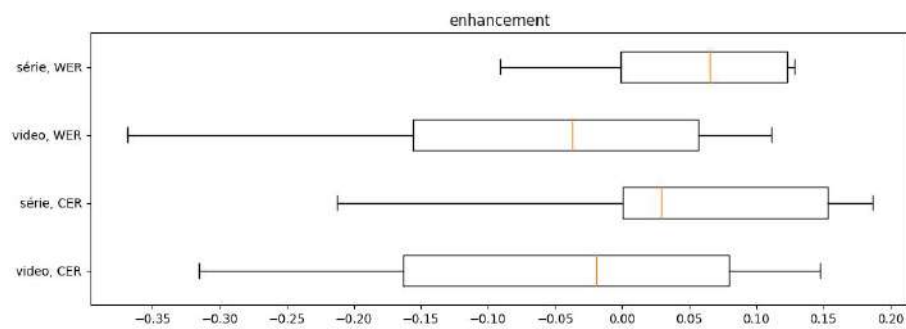


(b) zlepšení CER

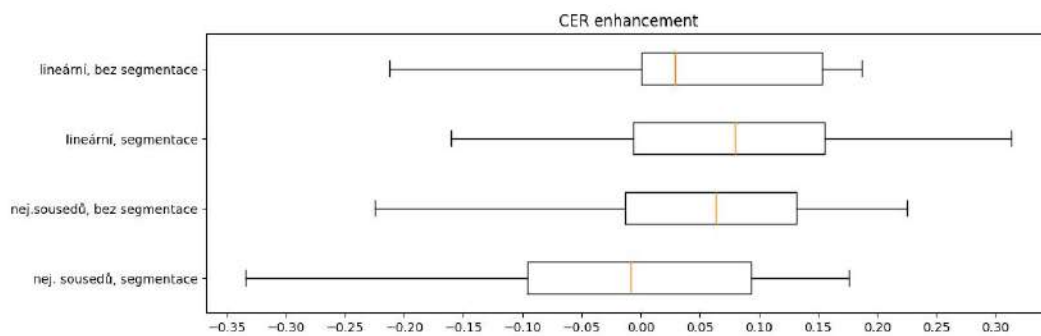
■ **Obrázek 6.2** Rozložení hodnot zlepšení míry WER/CER podle rychlosti snímání. K snímání bylo využito sériového snímání



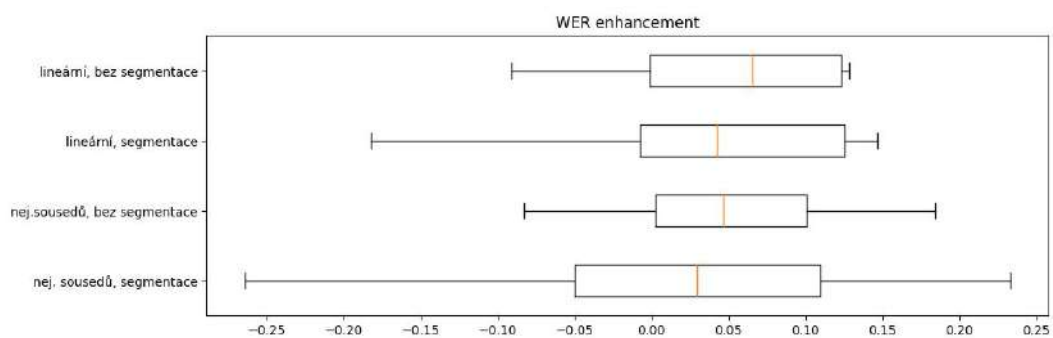
■ **Obrázek 6.3** Rozložení hodnot zlepšení míry WER/CER. Snímky pořízené sériovým snímáním a z videa. Obojí snímání probíhalo při rychlosti posunu dokumentu 30 mm/s. Využita interpolace metodou nejbližších sousedů



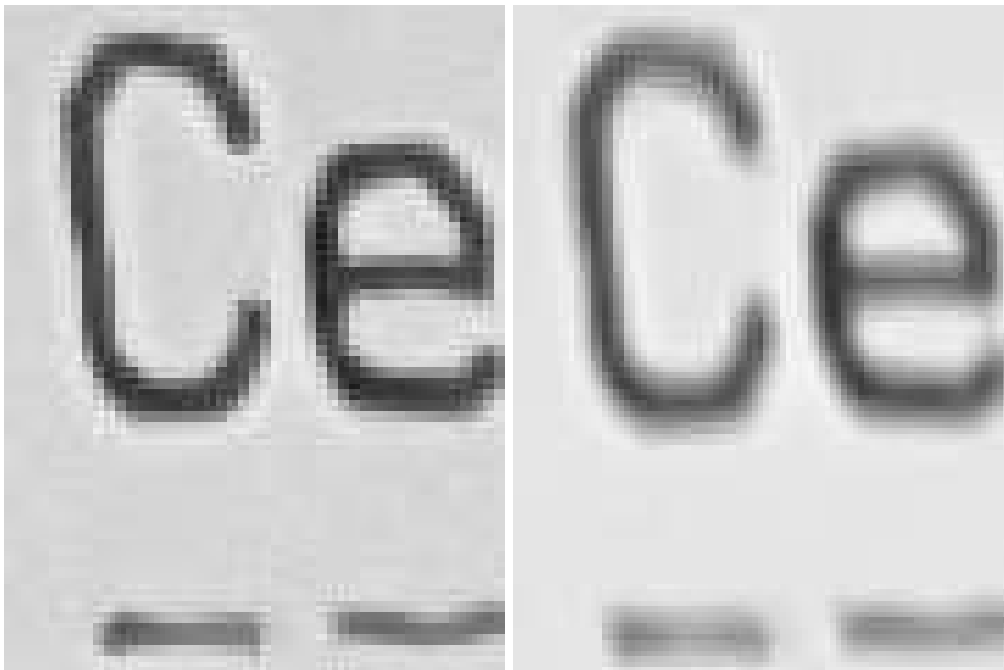
■ **Obrázek 6.4** Rozložení hodnot zlepšení míry WER/CER. Snímky pořízené sériovým snímáním a z videa. Obojí snímání probíhalo při rychlosti posunu dokumentu 30 mm/s. Využita lineární interpolace



■ **Obrázek 6.5** Rozložení hodnot zlepšení míry CER. Snímky pořízené sériovým snímáním. Snímání probíhalo při rychlosti posunu dokumentu 30 mm/s. Využita lineární interpolace a metoda nejbližších sousedů



■ **Obrázek 6.6** Rozložení hodnot zlepšení míry WER. Snímky pořízené sériovým snímáním. Snímání probíhalo při rychlosti posunu dokumentu 30 mm/s. Využita lineární interpolace a metoda nejbližších sousedů

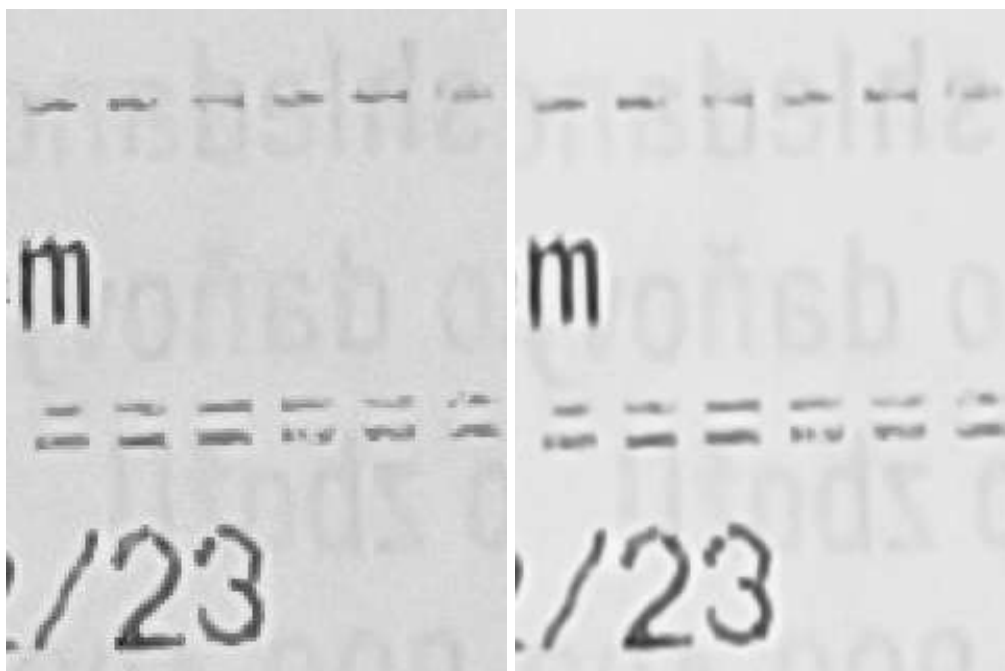


■ **Obrázek 6.7** První snímek ukazuje detail základního snímku. Druhý snímek je detail výsledku SR algoritmu. Lze vidět redukci šumu v druhém snímku

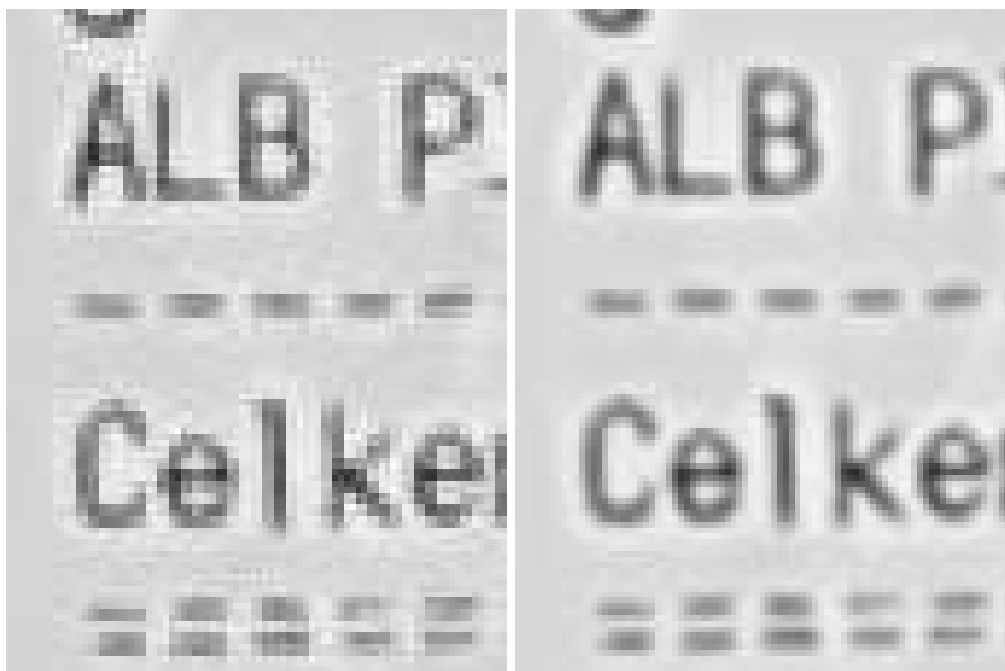
6.2 Vizuální dopady

V obrázku 6.7 lze pozorovat výraznou redukci šumu ve výsledném snímku. Lze také vidět, že šum se na původním snímku vyskytuje i v ideálním osvětlení, tedy jeho odstranění je vždy žádoucí. Fakt že využití SR algoritmu na celý snímek vede k zvýraznění průsvitů lze pozorovat v obrázku 6.8. Zlepšení rozlišení lze snadněji rozpoznat na snímcích pocházejících z videa (obrázek 6.9), ale je znatelný i na snímcích ze sériového snímání (obrázek 6.10).

Negativní následek SR algoritmu lze vidět na obrázku 6.11. Jedná se o efekt podobný rozmazání, který vytváří „*fantomová písmena*“, která jsou lehce posunutá oproti původním. Tento efekt je způsoben špatným mapováním této určité oblasti pravděpodobně kvůli příliš zdeformovanému dokumentu.



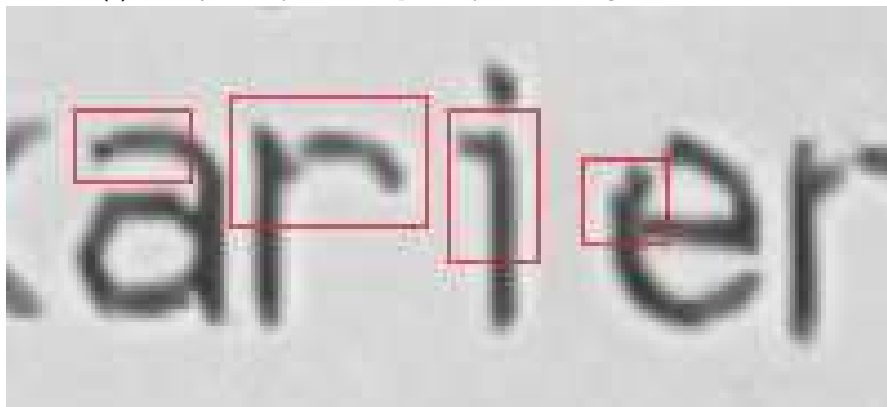
■ **Obrázek 6.8** První snímek ukazuje detail základního snímku. Druhý snímek je detail výsledku SR algoritmu. Lze vidět zvýraznění průsvitu textu z druhé strany dokumentu na druhém snímku



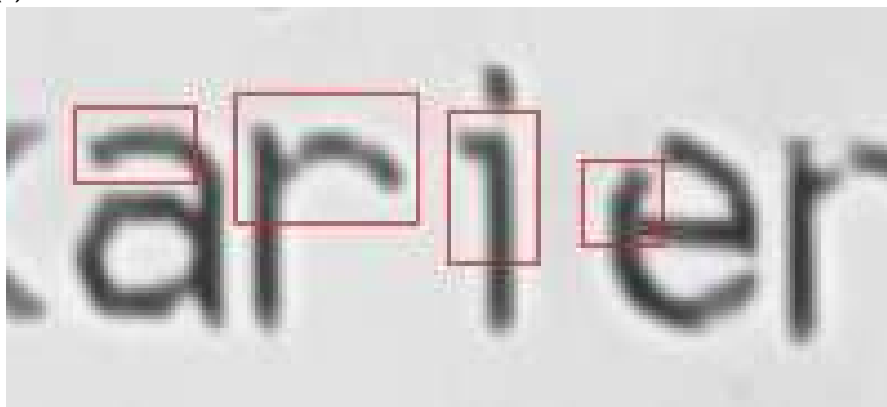
■ **Obrázek 6.9** První snímek ukazuje detail základního snímku. Druhý snímek je detail výsledku SR algoritmu. Lze vidět zvýšení rozlišení druhého snímku



(a) Vlevo je bazový snímek. Vpravo výsledek SR algoritmu



(b) Detail bazového snímku



(c) Detail výsledku SR algoritmu

■ **Obrázek 6.10** Snímek bazového snímku a výsledek SR algoritmu s detailem do obou snímků. Lze vidět zvýšení rozlišení druhého snímku



■ **Obrázek 6.11** První snímek ukazuje detail základního snímku. Druhý snímek je detail výsledku SR algoritmu. Lze vidět efekt „fantomových písmen“

Kapitola 7

Diskuze

Algoritmus nabízí spoustu modifikací, které by mohly vést k lepším výsledkům. Problém s nepřesným mapováním kvůli deformacím dokumentu by bylo možné zmírnit rozdělením snímku na více částí a mapováním těchto částí nezávisle na sobě. Rozpoznání těchto oblastí pomocí hran je nespolehlivé. Vyřazování odlehlých bodů metodou RANSAC by mohlo být použito iterativně k tvoření shluků bodů, které odpovídají přesnější rovině. Postup by byl nalezt metodou RANSAC transformační matici, která určí masku bodů, které považovala za neodlehle (nejvíce nalezitelných ve snímku). Tento shluk bodů by byl uložen do kolekce a hledání trans. matice metodou RANSAC by se opakovalo na bodech, které byly označeny jako odlehlé, a nyní získané neodlehle body by tvořili nový shluk. Tento postup by se opakoval do vyčerpání bodů. Nakonec by bylo třeba snímek rozdělit podle jednotlivých shluků bodů v kolekci. K tomu by mohl sloužit algoritmus založený na zaplavování okolí bodů. Taktó by vznikly samostatné oblasti, které by se měly více podobat rovině než celý snímek, a tedy by jejich mapování bylo přesnější. Tento přístup by byl limitovaný pouze počtem a kvalitou nalezených KP ve snímku.

V oblasti hledání klíčových bodů by jednou z takových modifikací mohlo být využití *ASIFT* [19] místo *SIFT* pro detekci a popis KP. Párování dokáže pak lépe pracovat s afinně transformovanými KP za cenu výrazně náročnějších výpočetních nároků. Tento přístup by ale mohl zajistit větší počet KP, což by pomohlo návrhu z předešlého odstavce.

Vysoká výpočetní náročnost procesu SR může být problémem pro využití algoritmu v reálném čase. Zpracování by částečně urychlilo omezení výpočtů pouze na oblast samotného dokumentu ve snímku. K tomu by bylo potřeba dokument spolehlivě segmentovat a následně pracovat pouze s segmentovanou oblastí. To by bylo paměťově úspornější a některé procesy by zvládly zpracovat menší snímek rychleji. Výpočet by také mohl být rozdělen po částech snímku, což by snížilo požadavky na velikost paměti, kterou výpočet vyžaduje. Umožnilo by to dále využít také vícevláknových výpočtů – paralelizace.

Zajímavým rozšířením by mohla být kombinace navrženého přístupu s SI-SR algoritmy. Zlepšení obrazové informace, které nevedlo k výraznému zlepšení čtení, navrženým algoritmem MI-SR by mohlo mít pozitivní dopady na výraznější zlepšení obrazové informace při následném využití metod SI-SR.



Kapitola 8

Závěr

Cílem teoretické části práce bylo prozkoumat existující metody využití více snímků za účelem zisku snímku o vyšším rozlišení. Práce analyzuje existující metody a porovnává jejich využitelnost a možnost navázat na ně. Dále práce definuje metodiku měření a navrhuje vlastní algoritmy pro zlepšení obrazové informace za využití zvýšení rozlišení snímku algoritmem MI-SR. V praktické části se práce věnuje získání obrazového datasetu účetních dokladů a implementaci navržených algoritmů v programovacím jazyce Python. Algoritmy jsou testované na získaném datasetu a jsou zhodnoceny jejich důsledky na čtení textu pomocí OCR.

Výsledky naznačují, že k mírnému zlepšení došlo. Zlepšení ale není příliš výrazné. V některých konfiguracích došlo dokonce k výraznému zhoršení výsledků. Vliv na to může mít nepřírozenost hran, s čím neumí používané OCR pracovat. Při výrazných tvarových deformacích dokumentu se projevuje rozmazání některých hran textu a artefakty způsobené nepřesným mapováním. Algoritmus není velmi předvídatelný a spolehlivý. Doporučeno ale může být jeho spojení s SI-SR metodami, které by mohlo pozitivně ovlivnit výsledky. Výsledný algoritmus je velmi paměťově a výkonově náročný, proto je jeho využití v reálném čase v této implementaci nemožné.

Vzhledem k ne plně negativním výsledkům a několika navrženým možnostem, jak algoritmus zlepšit nebo rozšířit, je doporučené přístup navržený v této práci zkoumat hlouběji a implementovat navržená vylepšení algoritmu.

Bibliografie

1. WANG, Zhaozhi; LIN, Shoufu; CHEN, Yang; LYULYOV, Oleksii; PIMONENKO, Tetyana. Digitalization Effect on Business Performance: Role of Business Model Innovation. *Sustainability*. 2023, roč. 15, č. 11. ISSN 2071-1050. Dostupné z DOI: 10.3390/su15119020.
2. CHRAMOSTA, Tomáš. Digitalizace účetnictví V česku Pokulháva. *ISVS.CZ / Aktuálně to nejdůležitější o ISVS a eGovernmentu zde na jednom místě*. [online]. 2023 [cit. 2024-04-19]. Dostupné z: <https://www.isvs.cz/digitalizace-ucetnictvi-v-cesku-pokulhava/>.
3. IGNATYEVA, Diana. *Digitalizace účetnictví v praxi*. 2023. Diplomová práce. Vysoká škola ekonomická v Praze.
4. SCHERMELLEH, Lothar; FERRAND, Alexia; HUSER, Thomas; EGGELING, Christian; SAUER, Markus; BIEHLMAIER, Oliver; DRUMMEN, Gregor. Super-resolution microscopy demystified. *Nature Cell Biology*. 2019, roč. 21. Dostupné z DOI: 10.1038/s41556-018-0251-8.
5. TSAI, Roger Y.; HUANG, Thomas S. Multiframe image restoration and registration. In: 1984. Dostupné také z: <https://api.semanticscholar.org/CorpusID:59796060>.
6. KIM, S.P.; BOSE, N.K.; VALENZUELA, H.M. Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1990, roč. 38, č. 6, s. 1013–1027. Dostupné z DOI: 10.1109/29.56062.
7. UR, Hanoch; GROSS, Daniel. Improved resolution from subpixel shifted pictures. *CVGIP: Graphical Models and Image Processing*. 1992, roč. 54, č. 2, s. 181–186. ISSN 1049-9652. Dostupné z DOI: [https://doi.org/10.1016/1049-9652\(92\)90065-6](https://doi.org/10.1016/1049-9652(92)90065-6).
8. FRUCHTER, Andrew; HOOK, R. Drizzle: A Method for the Linear Reconstruction of Undersampled Images. *Publications of The Astronomical Society of The Pacific - PUBL ASTRON SOC PAC*. 1998, roč. 114. Dostupné z DOI: 10.1086/338393.
9. FRUCHTER, Andrew; MUTCHLER, Max. Drizzling Singly-Dithered Hubble Space Telescope Images: A Demonstration. *Space Telescope Science Institute*. 1998.
10. FREEMAN, William; PASZTOR, Egon; CARMICHAEL, Owen. Learning Low-Level Vision. *International Journal of Computer Vision*. 2000, roč. 40, s. 25–47. Dostupné z DOI: 10.1023/A:1026501619075.
11. LEDIG, Christian; THEIS, Lucas; HUSZAR, Ferenc; CABALLERO, Jose; CUNNINGHAM, Andrew; ACOSTA, Alejandro; AITKEN, Andrew; TEJANI, Alykhan; TOTZ, Johannes; WANG, Zehan; SHI, Wenzhe. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In: 2017, s. 105–114. Dostupné z DOI: 10.1109/CVPR.2017.19.

12. ZIHAN, Wang; CAO, Jie; HAO, Qun; ZHANG, Fanghua; CHENG, Yang; KONG, Xianyue. Super-resolution imaging and field of view extension using a single camera with Risley prisms. *Review of Scientific Instruments*. 2019, roč. 90. Dostupné z DOI: 10.1063/1.5050833.
13. *Basic concepts of the homography explained with code* [online]. [cit. 2024-04-19]. Dostupné z: https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html.
14. HUAMÁN, Ana. *Feature Detection* [online]. [cit. 2024-04-19]. Dostupné z: https://docs.opencv.org/4.x/d7/d66/tutorial_feature_detection.html.
15. BOJANIC, David; BARTOL, Kristijan; PRIBANIC, Tomislav; PETKOVIC, Tomislav; DONOSO, Yago Diez; MAS, Joaquim Salvi. On the Comparison of Classic and Deep Key-point Detector and Descriptor Methods. In: *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*. IEEE, 2019. Dostupné z DOI: 10.1109/ispa.2019.8868792.
16. LOWE, David G. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*. 1999, roč. 2, s. 1150–1157. Dostupné také z: <https://api.semanticscholar.org/CorpusID:5258236>.
17. MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2005, roč. 27, č. 10, s. 1615–1630. Dostupné z DOI: 10.1109/TPAMI.2005.188.
18. LOWE, David. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. 2004, roč. 60, s. 91–. Dostupné z DOI: 10.1023/B:VISI.0000029664.99615.94.
19. YU, Guoshen; MOREL, Jean-Michel. ASIFT: An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line*. 2011, roč. 1. Dostupné z DOI: 10.5201/ipol.2011.my-asift.
20. CMGLEE. *File:Comparison of 1D and 2D interpolation.svg* — *Wikimedia Commons, the free media repository* [online]. 2020-06-03. [cit. 2024-04-19]. Dostupné z: https://commons.wikimedia.org/w/index.php?title=File:Comparison_of_1D_and_2D_interpolation.svg&oldid=866704366.
21. OTSU, Nobuyuki. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*. 1979, roč. 9, č. 1, s. 62–66. Dostupné z DOI: 10.1109/TSMC.1979.4310076.
22. IMMERSKÆR, John. Fast Noise Variance Estimation. *Computer Vision and Image Understanding*. 1996, roč. 64, č. 2, s. 300–302. ISSN 1077-3142. Dostupné z DOI: 10.1006/cviu.1996.0060.
23. BEN-EZRA, Moshe; NAYAR, Shree. Motion-based motion deblurring. *IEEE transactions on pattern analysis and machine intelligence*. 2004, roč. 26. Dostupné z DOI: 10.1109/TPAMI.2004.1.
24. PERTUZ, Said; PUIG, Domenec; GARCIA, Miguel Angel. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*. 2013, roč. 46, č. 5, s. 1415–1432. ISSN 0031-3203. Dostupné z DOI: 10.1016/j.patcog.2012.11.011.
25. *Smartphones vs cameras: Closing the gap on Image Quality* [online]. DXOMARK, © 2008-2024 [cit. 2024-04-19]. Dostupné z: <https://www.dxomark.com/smartphones-vs-cameras-closing-the-gap-on-image-quality/>.
26. A Practical Guide to Machine Vision Lighting. *National Instruments* [online]. 2023 [cit. 2024-04-19]. Dostupné z: <https://www.ni.com/en/shop/choosing-the-right-hardware-for-your-vision-applications/a-practical-guide-to-machine-vision-lighting.html>.

27. HAVLE, Otto. Strojové vidění IV: Osvětlovače. *Automa*. 2008, roč. 2008, č. 4. ISSN 1210-9592. Dostupné také z: https://www.automa.cz/cz/casopis-clanky/strojove-videni-iv-osvetlovace-2008_04_36988_5853/.
28. GRAY, Elizabeth. Understanding Depth of Field – A Beginner’s Guide. *photographylife* [online]. 2021 [cit. 2024-04-19]. Dostupné z: <https://photographylife.com/what-is-depth-of-field>.
29. BEN-EZRA, M.; ZOMET, A.; NAYAR, S.K. Video super-resolution using controlled subpixel detector shifts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2005, roč. 27, č. 6, s. 977–987. Dostupné z DOI: 10.1109/TPAMI.2005.129.
30. GONZALEZ, Rafael C.; WOODS, Richard E. *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN 9780131687288.
31. PERTUZ, Said; PUIG, Domenec; GARCÍA, Miguel Ángel. Analysis of focus measure operators for shape-from-focus. *Pattern Recognit.* 2013, roč. 46, s. 1415–1432. Dostupné také z: <https://api.semanticscholar.org/CorpusID:17364877>.
32. *Samsung Galaxy A34 5G* [online]. [cit. 2024-04-19]. Dostupné z: https://www.camerav5.com/devices/manufacturers/samsung/sm-a346m_a34x_0/.
33. *Samsung Galaxy A34 5G Camera test* [online]. 2023-03-21. [cit. 2024-04-19]. Dostupné z: <https://www.dxomark.com/samsung-galaxy-a34-5g-camera-test/>.
34. BUSSCHAERT, Pieter-Jan. *Smoothing with a mask* [online]. 2013-03-06. [cit. 2024-04-19]. Dostupné z: <https://answers.opencv.org/question/3031/smoothing-with-a-mask/>.

Obsah příloh

readme.txt	stručný popis obsahu média
data	soubor s datasetem
src		
_ impl	zdrojové kódy implementace
_ lib.ipynb	Jupyter notebook navržených algoritmů
_ lib.py	Python modul navržených algoritmů
_ thesis	zdrojová forma práce ve formátu \LaTeX
text	text práce
_ thesis.pdf	text práce ve formátu PDF