Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Prokládání obrazů pro lentikulární tisk

Plzeň 2016 Jan Albl

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: **2015/2016**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Jan ALBL**
Osobní číslo:	**A13B0256P**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Informatika**
Název tématu:	**Prokládání obrazů pro lentikulární tisk**
Zadávající katedra:	**Katedra informatiky a výpočetní techniky**

Zásady pro vypracování:

1. Prostudujte a popište chování lentikulární desky.
2. Prostudujte techniky prokládání obrazů, soustřeďte se na kompenzace jevů spojených s užitím reálných lentikulárních desek a tiskových technik.
3. Prostudujte metody měření parametrů prokládání obrazů (např. pitchtest).
4. Vybrané techniky prokládání obrazů a měření parametrů implementujte. Pozornost věnujte i práci s velkými výstupními soubory.
5. Srovnejte dosažené výsledky s výstupy dostupných programů pro prokládání.

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 22. června 2016

Jan Albl

Poděkování

Na tomto místě bych rád poděkoval panu Ing. Petru Lobazovi za odborné vedení práce, cenné rady, trpělivost a ochotu, které mi pomohly tuto práci zkompletovat.

Abstract

This thesis is focused on problematics of creating lenticular images in the program Interlacer. Here is also briefly explained the principle of lenticular images, creating these images using different methods of interlacing, the optimalization of interlacing for the big output images. Achieved results are compared to another programs for creating lenticular images. Further, this paper discusses the method for measuring parameters of lenticular sheet (pitch test) and the problematics of ghosting that occur in lenticular print.

Abstrakt

Tato práce se zabývá problematikou vytváření lentikulárních obrázků v programu Interlacer. Je zde stručně popsán princip lentikulárních obrázků. Dále se zabývám vytvářením těchto obrázků pomocí různých metod prokládání a optimalizace prokládání pro velké výstupní soubory. Dosažené výsledky jsou srovnávány s různými programy zabývající se vytvářením proložených obrázků pro lentikulární tisk. Dále je zde rozebrána metoda pro měření parametrů optické desky (pitch test) a problematika přeslechů vznikajících v lentikulárním tisku.

Obsah

Úvod

Cílem této bakalářské práce je rozšíření programu Interlacer. Tento program byl vyvíjen v rámci semestrální práce z předmětu KIV/ZSWI na Fakultě aplikovaných věd Západočeské univerzity v Plzni. Snahou bylo vytvořit aplikaci, která bude sloužit pro studenty a zaměstnance z Fakulty designu a umění Ladislava Sutnara (FDU), kteří díky ní mohou sami vytvářet podklady pro lentikulární tisk. Na vývoji Interlaceru jsme pracovali v čtyřčlenném týmu, kde se mnou spolupracovali Štěpán Baratta, Tomáš Matějka a Lukáš Hruda. Aplikace je napsána v programovacím jazyce C# s využitím grafické knihovny Magick++. Po skončení předmětu ZSWI byl Interlacer plně funkční a dnes se na FDU běžně používá.

Tato práce doplňuje software o možnost výběru dalších metod prokládání, kalibrace pomocí pitch testu, možnosti šikmého prokládání a optimalizuje stávající algoritmus prokládání, zejména s ohledem na velké výstupní soubory. Práce se dále zabývá problematikou přeslechů v lentikulárním tisku; z důvodů uvedených dále v textu nebyla funkcionalita začleněna do Interlaceru.

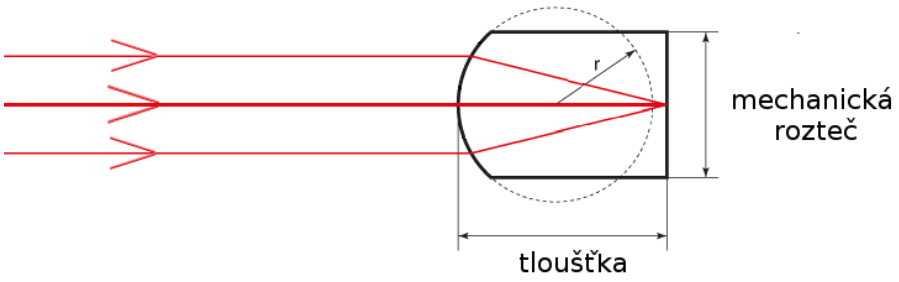
Simulace optické desky, zejména pro testování různých metod prokládání, byly prováděny v programu Lenticular Viewer. Tento program byl vytvořen panem Ing. Petrem Vaněčkem, Ph.D. v interním projektu Západočeské univerzity "Zkvalitnění výuky speciálních grafických technik, ilustrace a fotografie", č. VS-15-022 (vedoucí doc. Barbara Šalamounová, M.A., Fakulta designu a umění Ladislava Sutnara).

1 Lentikulární tisk

Lentikulární tisk je technologie, která slouží k výrobě tištěných obrázků s iluzí hloubky nebo možností animace. K vytvoření těchto lentikulárních obrázků je zapotřebí tzv. lentikulární desky (folie) a speciálně upraveného obrázku. Tento obrázek se buď vytiskne přímo na desku nebo na substrát (nejčastěji papír), který se poté na desku nalepí. Lentikulární deska je sestavena z lineárního pole cylindrických plankonvexních čoček (tzv. lentikulárních čoček), díky kterým má jednu stranu hrbolatou a druhou plochou. Na rovinnou plochu této desky se přikládá upravený obrázek. S lentikulární technologií se také můžeme setkat u autostereoskopických displejů. Poznatky pro tuto kapitolu čerpám především ze zdrojů [4–7, 12–14, 16].

1.1 Lentikulární čočka

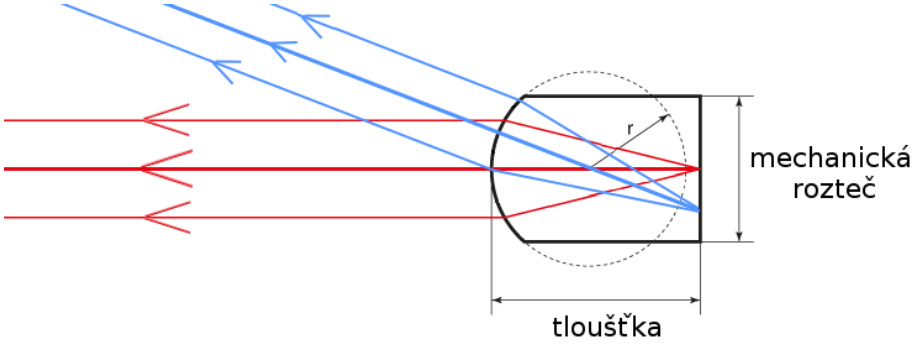
Optická čočka, kterou lentikulární technologie využívá, je plankonvexní spojka. Důležitou vlastností této čočky je, že zaostřuje rovnoběžné paprsky do bodu na ohniskové rovině, která se nachází na rovinné straně čočky viz obrázek 1.1. Ke konstrukci směrů paprsků v obrázku 1.1 jsem využil známé vlastnosti, že paprsky, které dopadají kolmo na povrch čočky, se nelámou.



Obrázek 1.1: Paprsky dopadající na lentikulární čočku

K popisu chování lentikulární čočky (lentikule) při sledování lentikulárního tisku si musíme představit obrácený chod paprsků. Pokud přiložíme na plochou stranu čočky papír s jedním bodem, budou paprsky vycházející z tohoto bodu lámány vždy jen do jednoho směru. Na obrázku 1.2 lze vidět paprsky vycházející ze dvou bodů ležící na ohniskové rovině. Při vysvětlení

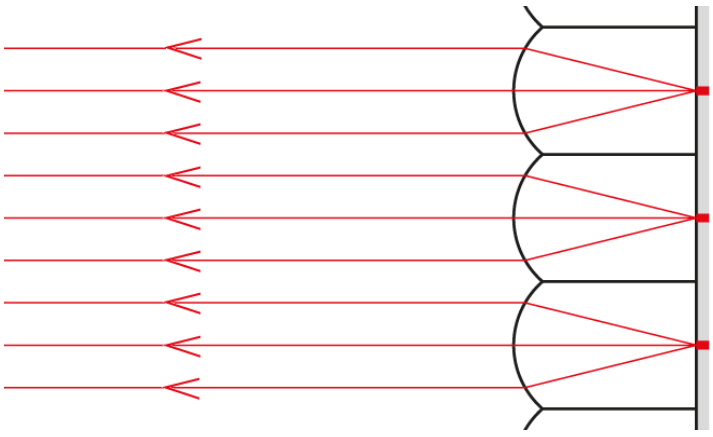
principu lentikulární čočky předpokládáme, že se chová jako ideální tenká čočka.



Obrázek 1.2: Paprsky vycházející ze dvou bodů na ohniskové rovině

1.2 Lentikulární deska

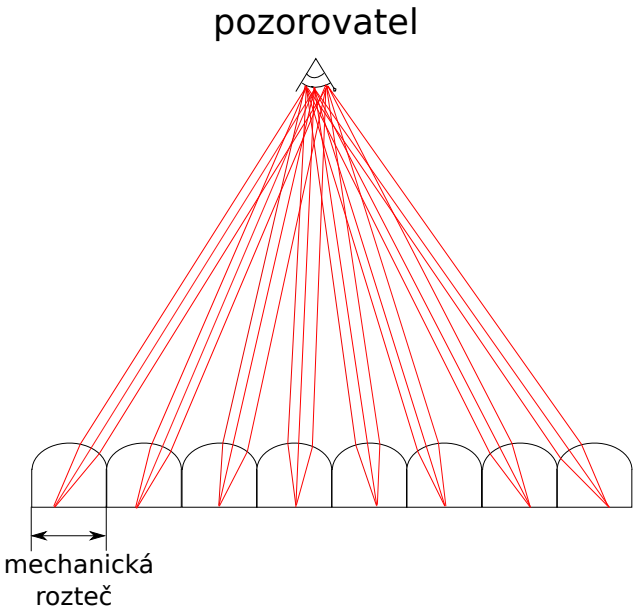
Spojíme-li mnoho takových čoček vedle sebe získáme lentikulární desku. Vlastnost takové desky je, že přicházející paprsky z jednoho směru se koncentrují do několika málo bodů na zadní straně desky v jisté vzdálenosti od sebe. K popisu chování lentikulární desky si můžeme představit, že paprsky vycházející z těchto bodů jsou deskou lámány výhradně do jednoho směru viz obrázek 1.3.



Obrázek 1.3: Rovnoběžné paprsky vycházející z bodů na ohniskové rovině

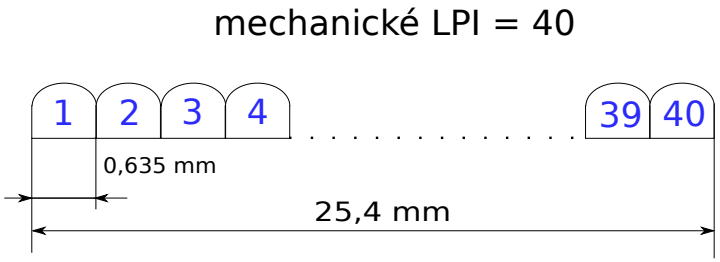
Dejme tomu, že se pozorovatel dívá na lentikulární obrázek z dostatečné vzdálenosti z jednoho bodu. Paprsky, které vychází z tohoto bodu dopadají na jednu lentikuli pod určitým úhlem. V dostatečné vzdálenosti pozorovacího bodu bude tento úhel zanedbatelný a můžeme tyto paprsky považovat za

rovnoběžné. Díky tomu se paprsky jdoucí od pozorovatele koncentrují vždy do jednoho bodu za jednou lentikulí viz obrázek 1.4.



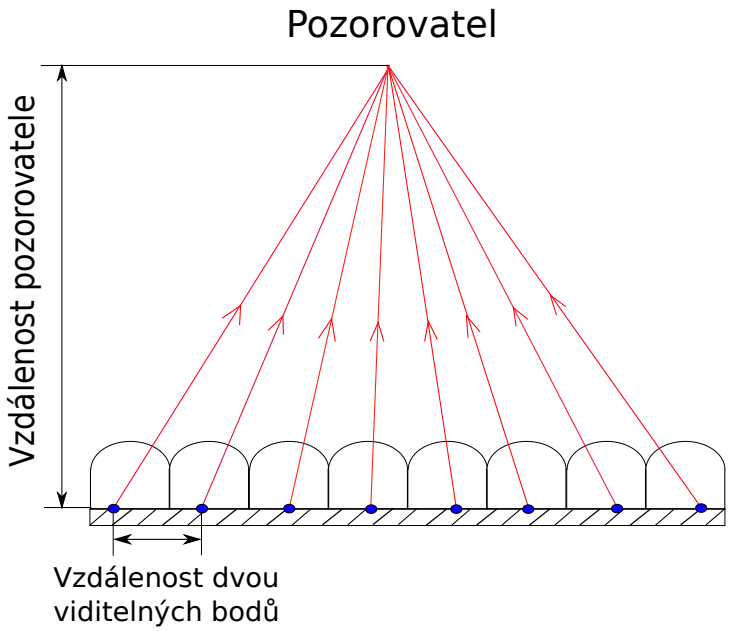
Obrázek 1.4: Pozorovatel dívající se na lentikulární desku

Pro další analýzu bude vhodné seznámit se s běžně používanými parametry lentikulární desky. Rozteč čoček viz obrázek 1.4 se pohybuje od několik desetin milimetru až po několik milimetrů. Tato hodnota se hůře pamatuje, proto je obvyklejší udávat počet čoček na jeden palec (tj. 25,4 mm). Máme-li například rozteč čočky 0,635 mm bude hustota čoček na jeden palec 25,4/0,635 = 40 LPI (Lenses per inch). Tomuto parametru závisejícím na rozteči čoček se říká **mechanické LPI** (viz obrázek 1.5).



Obrázek 1.5: Znázornění mechanického LPI

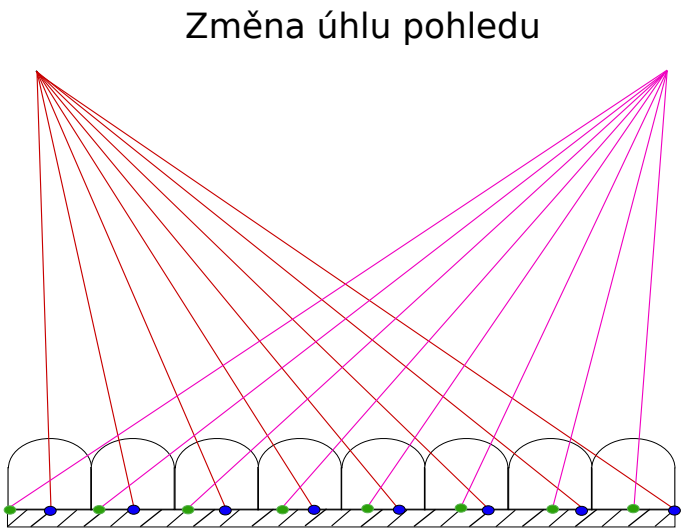
Dalším potřebným parametrem pro vytvoření lentikulárního obrázku je **vizuální LPI**. Pro přiblížení tohoto parametru využijeme obrázku 1.6, kde se pozorovatel dívá na lentikulární obrázek z určité vzdálenosti. Máme-li na rovinné straně optické desky přiložený bílý papír, vidí pozorovatel díky lámání paprsků jen několik málo bodů na tomto papíře. Vzdálenost dvou viditelných sousedních bodů je téměř stejná a je o něco málo větší než rozteč čoček. Tato hodnota se stejně jako rozteč čoček špatně pamatuje, a proto se udává v počtu takto viditelných bodů na jeden palec jako **vizuální LPI**. Například pro desku s hodnotou 40 LPI se vizuální LPI pohybuje v rozmezí 39.9 až 40 LPI.



Obrázek 1.6: Body viditelné pozorovatelem pro výpočet vizuálního LPI

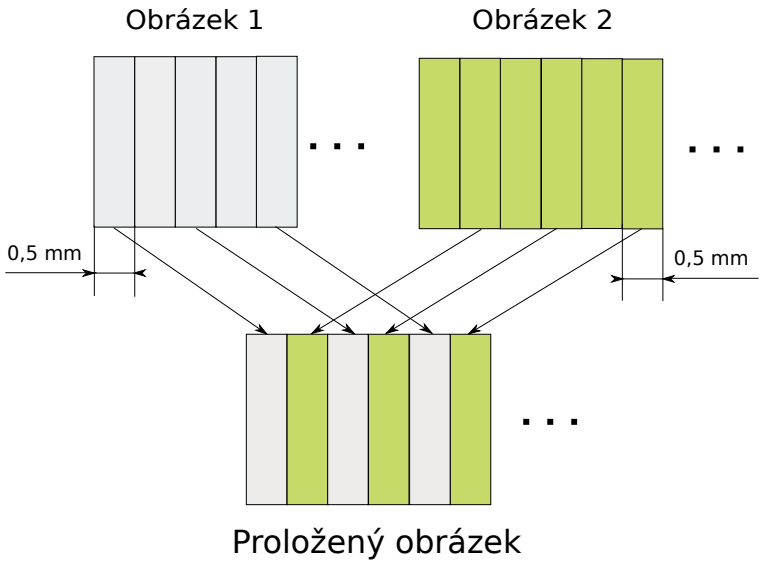
1.3 Prokládání

Pro vytvoření lentikulárního obrázku s iluzí hloubky či animace nestačí pouze lentikulární deska, ale potřebujeme i speciální obrázek (tzv. proložený), který usadíme za desku. Usadíme-li na ploché straně lentikulární desky bílý papír, tak z kapitoly 1.2 víme, že za každou lentikulí uvidí pozorovatel právě jeden bod z bílého papíru. Mění-li se vzdálenost či úhel pohledu pozorovatele, mění se i viditelné body na bílém papíře viz obrázek 1.7.



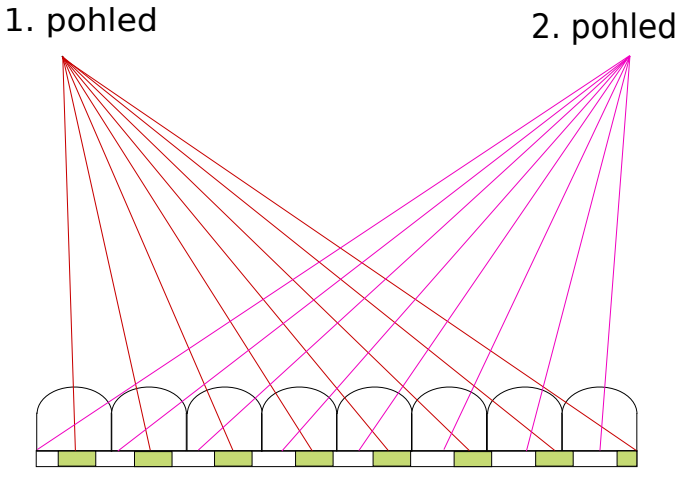
Obrázek 1.7: Znázornění viditelných bodů ze dvou různých úhlů pohledu

Chceme například vytvořit animaci ze dvou obrázků tak, aby pozorovatel ze dvou definovaných úhlů pohledu viděl buď první, nebo druhý obrázek. Pokud pozorovatel zůstává ve stejné vzdálenosti od lentikulárního obrázku a mění pouze úhel pohledu, vzdálenost mezi viditelnými sousedními body na přiloženém papíru bude stejná. To znamená, že se vizuální LPI ze dvou různých úhlů pohledu při stejné vzdálenosti od lentikulárního obrázku nemění. Jak víme z kapitoly 1.2, vizuální LPI vlastně určuje vzdálenost mezi sousedními viditelnými body. Chceme-li například vytvořit lentikulární obrázek pro animaci ze dvou obrázků při 25,4 LPI (vizuální) s rozměrem desky 100 x 100 mm a velikostí obrázků také 100 x 100 mm, musíme tyto obrázky nějakým způsobem spojit do speciálního proloženého obrázku, který se nalepí na lentikulární desku. V tomto případě rozdělíme obrázky na proužky velké 0,5 mm a střídavě vkládáme první proužek z prvního obrázku, druhý proužek z druhého, třetí proužek z prvního atd. viz obrázek 1.8.



Obrázek 1.8: Prokládání pro lentikulární tisk

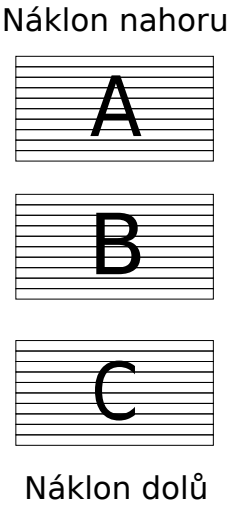
Velikost těchto proužků jsme určili z vizuálního LPI. Protože jsme použili hodnotu 25,4 LPI, každý viditelný bod na papíře je vzdálen od sousedního 25,4/25,4 mm (tj. 1 mm), jelikož jsme prokládali dva obrázky, vydělíme tuto vzdálenost dvěma a velikost jednoho proužku je tedy 0,5 mm. Pokud přiložíme lentikulární desku na proložený obrázek tak, aby lentikule ležely ve směru prokládání obrázku, uvidíme ze dvou různých pohledů dva různé obrázky viz obrázek 1.9.



Obrázek 1.9: Proložený obrázek za lentikulární deskou

1.3.1 Horizontální prokládání

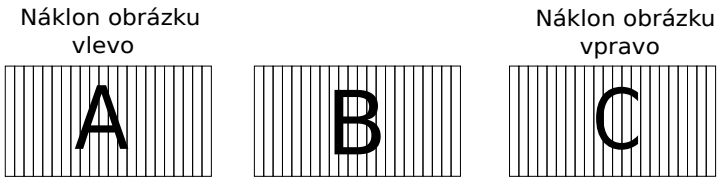
Výsledkem tohoto prokládání je lentikulární obrázek, jehož lentikule jsou orientovány vodorovně. Animace se u tohoto obrázku projevuje nakláníme-li jeho horní a dolní stranu. Na obrázku 1.10 je horizontálně proložený obrázek s lentikulární deskou vytvořený ze tří obrázků s písmeny A, B a C. Výsledný proložený obrázek se skládá z jednotlivých **řádků** vstupních obrázků (A, B a C). Horizontální prokládání se používá zejména pro efekt **flip**, kde se obrázky při změně pozorovacího úhlu postupně střídají. Díky tomu, že obě oči pozorují jednu lentikuli pod stejným pozorovacím úhlem, vidí pozorovatel oběma očima tentýž obrázek.



Obrázek 1.10: Horizontálně orientovaný lentikulární obrázek

1.3.2 Vertikální prokládání

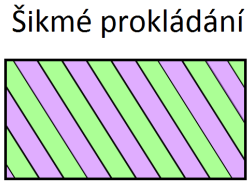
Výsledkem tohoto prokládání je lentikulární obrázek, jehož lentikule jsou orientovány svisle. Animace se u tohoto obrázku projevuje nakláníme-li ho na levou a pravou stranu. Na obrázku 1.11 je vertikálně proložený obrázek také vytvořený ze tří obrázků s písmeny stejně jako v sekci 1.3.1. Výsledný proložený obrázek se skládá z jednotlivých **sloupečků** vstupních obrázků (A, B a C). Vertikální prokládání se používá například u **velkoplošných lentikulárních obrázků**, kde se kolemjdoucím lidem postupně střídají jednotlivé obrázky. Také lze na rozdíl od horizontálního použít pro **3D efekty**, protože každé oko vidí jednu lentikuli z jiných pozorovacích úhlů a tím i jiné obrázky.



Obrázek 1.11: Vertikálně orientovaný lentikulární obrázek

1.3.3 Šikmé prokládání

Obrázky nemusíme prokládat pouze **horizontálně** nebo **vertikálně**, ale i v nějakém sklonu. Výsledný obrázek bude mít jednotlivé sloupečky či řádky vstupních obrázků nakloněné pod určitým úhlem viz obrázek 1.12. Toto prokládání se používá například k redukci nežádoucího jevu **moaré**. Tento jev může nastat například při zvolení nesprávné kalibrace nebo vadou tisku. Více informací o tomto prokládání se můžeme dočíst ve zdrojích [2, 15].



Obrázek 1.12: Ukázka výstupního obrázku po šikmém prokládání

1.3.4 Tiskové rozlišení

Pro výsledný lentikulární obrázek je zapotřebí proložený obrázek vytisknout. Důležitým parametrem tisku obrázků je **PPI** (Pixels per inch). Tento parametr tiskárny určuje kolik pixelů je tiskárna schopna vytisknout na jeden palec. Máme-li například tiskárnu s 300 PPI, je schopna vytisknout 300 pixelů vedle sebe s celkovou velikostí 2,54 cm, neboli hustota, s jakou se obrázek tiskne, je 300 pixelů na jeden palec. Tiskárny většinou nedokáží vytisknout jeden pixel libovolné barvy. Aby korektně zobrazily jeden pixel, musí jeho barvu namíchat z několika bodů barevných inkoustů. Jeden pixel se tedy skládá z několika inkoustových bodů. Parametr **DPI** (Dots per inch) určuje s jakou hustotou je tiskárna schopná stříkat inkoustové body na papír. DPI musí být větší než PPI, aby tiskárna dokázala vytvořit pixel libovolné barvy z několika tiskových bodů. V praxi se tyto veličiny často zaměňují. **V této práci** pro zjednodušení **je DPI rovno PPI**, tedy DPI označuje s jakou hustotou je tiskárna schopna vytisknout pixely na jeden palec.

2 Implementace prokládání

K vytvoření výsledného lentikulárního obrázku musíme sestavit proložený obrázek, který usadíme za optickou desku. Pro jeho výrobu potřebujeme obrázky, ze kterých se bude tento proložený obrázek skládat. Abychom si usnadnili práci, budeme počítat s tím, že vstupní obrázky mají stejné rozměry.

K proložení potřebujeme znát důležité informace:

- **Vizuální LPI (Lenses per inch)** je parametr, který souvisí s jemností lentikulární desky viz kapitola 1.2. Jeho určení je problematické a dále jej rozebírám v kapitole 3.
- **DPI (Dots per inch) tiskárny** je parametr určující jemnost tisku. Je dán výrobcem tiskárny, nejběžnější hodnoty jsou 600 a 720 DPI.
- **Šířku a výšku** výsledného lentikulárního obrázku v palcích.

2.1 Prokládání obrázků v Interlaceru

Na jednoduchém příkladu se pokusím vysvětlit výrobu lentikulárního obrázku v původní verzi Interlaceru. Dejme tomu, že budeme vytvářet lentikulární obrázek z pěti obrázků. Optická deska bude mít vizuální LPI 39,92. Budeme chtít, aby byl tento obrázek vysoký a široký 3 palce. Směr lentikulí bude vertikální a proložený obrázek budeme tisknout na tiskárně s rozlišením 600 DPI.

Nejdříve si přepočítáme šířku obrázku na počet lentikulí. Předpokládáme-li, že šířka lentikule je $1/LPI_v$, je celkový počet lentikulí roven

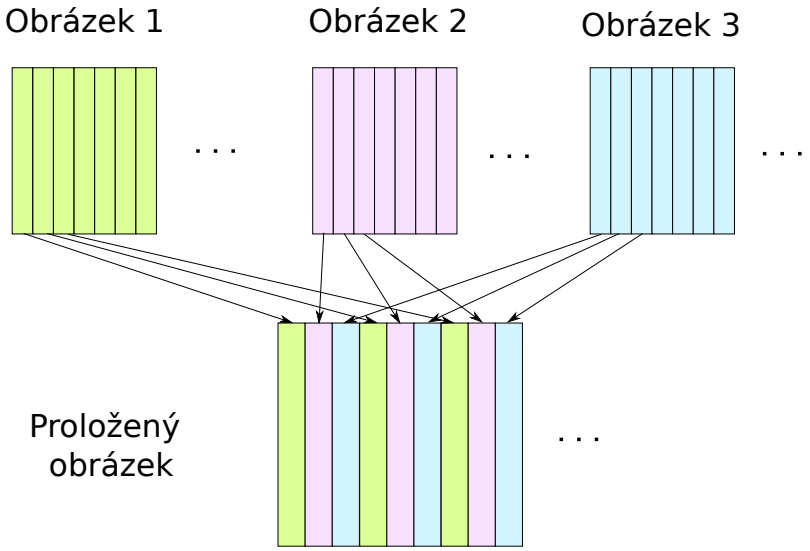
$$lenses = W \times LPI_v, \tag{2.1}$$

kde W je šířka obrázku v palcích. Tato hodnota nemusí být celočíselná, a proto si pro další postup vytvoříme její zaokrouhlenou hodnotu

$$lenses_f = \text{floor}(lenses). \tag{2.2}$$

Vstupní obrázky mají nějakou velikost, tedy nějaký počet sloupečků. Abychom vstupní obrázek správně reprezentovali, musíme vědět jaké sloupečky vybrat. V Interlaceru používáme jednoduchý způsob, jak toho dosáhnout. Vstupní obrázky převzorkujeme na počet lentikulí ($lenses_f$) a díky

tomu každý sloupeček převzorkovaného obrázku odpovídá jednomu viditelnému sloupečku za jednou lentikulí. Ze vstupních obrázků tedy vyjmeme první sloupečky a vložíme je postupně za sebou, poté vyjmeme druhé sloupečky a vložíme je do tzv. předběžně proloženého obrázku viz obrázek 2.1.



Obrázek 2.1: Ukázka skládání předběžně proloženého obrázku v Interlaceru

Máme-li N obrázků bude mít předběžně proložený obrázek šířku W_p rovnou

$$W_p = N \times lenses_f. \tag{2.3}$$

V našem případě tedy šířka předběžně proloženého obrázku v pixelech odpovídá

$$W_p = 5 \times 199 = 995 \text{ pixelů}. \tag{2.4}$$

Pokud bychom chtěli tento předběžně proložený obrázek vytisknout na výsledný lentikulární obrázek, museli bychom použít tiskárnu, kde by se přesně N proložených sloupečků zobrazilo pod jednu lentikuli, tedy

$$N = \frac{DPI'}{LPI_v}. \tag{2.5}$$

DPI této tiskárny by bylo tedy

$$DPI' = N \times LPI_v. \tag{2.6}$$

Protože v našem případě chceme tisknout s rozlišením 600 DPI musíme předběžně proložený obrázek se šířkou W_p pixelů převzorkovat pro toto tiskové rozlišení. Šířka W_c výsledného proloženého obrázku bude tedy

$$W_c = W_p \times \frac{DPI}{DPI'} = lenses_f \times \frac{DPI}{LPI_v}. \tag{2.7}$$

Dosadíme-li naše hodnoty, vyjde nám, že šířka výsledného proloženého obrázku bude přibližně 2988,74 pixelu. Tato hodnota je neceločíselná a v praxi ji zaokrouhlujeme dolů. Důsledkem tohoto zaokrouhlení se může stát, že výsledný proložený obrázek má jiné vizuální LPI, než bychom chtěli. Jelikož se šířka obrázku od ideální liší maximálně o jeden pixel, je tato chyba zanedbatelná.

Následující pseudokód popisuje obecný princip prokládání této metody pro vertikální prokládání v Interlaceru.

Algoritmus 1 Ukázka metody pro vertikální prokládání v Interlaceru

```
1: lenses_f = floor((width in inch ) × (LPI))
2: N = (images count)
3:
4: width of interlaced image = (N × lenses_f)
5: height of interlaced image = height of input images
6:
7: for (i = 0 TO (N - 1)) do
8:     Resize width of picture (i) to (lenses_f)
9:     for (j = 0 TO (lenses_f − 1)) do
10:        Extract column (j) of picture (i)
11:        Insert the extracted column into column (i + j x N) of interlaced image
12:    end for
13: end for
14:
15: Resize interlaced image width to (lenses_f × DPI/LPI_v)
16: Resize interlaced image height to (height in inch × DPI)
```

2.1.1 Výsledky metody

Způsob tohoto prokládání již přes rok využívají studenti a zaměstnanci z Fakulty designu a umění Ladislava Sutnara (FDU) na Západočeské univerzitě v Plzni. Výsledné lentikulární obrázky jsou vystavovány a můžeme tedy prohlásit, že tento způsob prokládání obrázků funguje bez větších problémů.

Při bližším zkoumání však můžeme nalézt některé nedostatky této metody. Například při prokládání vyžaduje plné rozlišení obrázku v operační paměti, a to je při práci s velkými obrázky problematické. Vznikají problémy například při převzorkování těchto obrázků, které rozebírám v kapitole 2.7. Dále při srovnání výsledků s ostatními aplikacemi (například Super Flip nebo Photo Projector) se jeví výstupní obrázky z Interlaceru méně ostré.

Například Photo Projector používá dvě jiné metody pro vytváření proloženého obrázku. Tyto metody jsem se v rámci své bakalářské práce rozhodl implementovat do Interlaceru a porovnat se stávající metodou.

2.2 Ostrá metoda prokládání obrázků

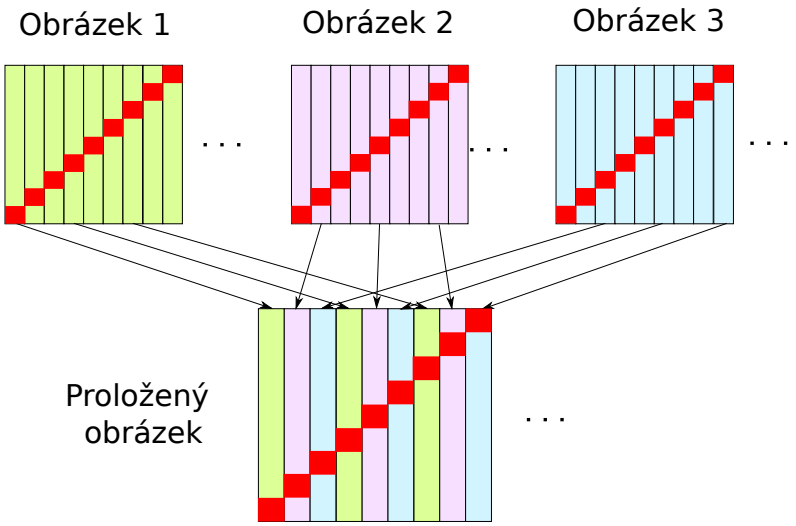
Metoda, kterou jsme v Interlaceru nazvali "Ostrá metoda", se velmi často používá v běžných programech pro lentikulární tisk. Oproti metodě popsané v sekci 2.1 se liší v převzorkování vstupních obrázků a ve výběru sloupečků pro jejich proložení.

Pro lepší pochopení této metody vysvětlím princip také pro vertikální prokládání. Nejdříve si přepočítáme šířku obrázku na počet lentikulí ($lenses_f$) podle vzorečku (2.2). Vstupní obrázky však šířkově nepřevzorkujeme na velikost $lenses_f$, ale na velikost W_{in}, kde

$$W_{in} = N \times lenses_f. \tag{2.8}$$

Vstupní obrázky v tomto kroku budou mít tedy stejnou šířku jako předběžně proložený obrázek. Oproti předchozí metodě z kapitoly 2.1 bude tedy N-krát větší a díky tomu při převzorkování obrázku na menší velikost neztratíme tolik detailů jako v předešlé metodě.

Sloupečky vybíráme tak, že z prvního obrázku začneme na prvním sloupečku a každý *N-tý* sloupeček bude vždy vložen do proloženého obrázku. Z druhého obrázku začneme na druhém sloupečku a každý *N-tý* sloupeček bude vložen do proloženého obrázku viz obrázek 2.2. Tímto postupem nám vznikne předběžně proložený obrázek, který se stejně jako v kapitole 2.1 pak převzorkuje na finální velikost dle tiskového rozlišení.



Obrázek 2.2: Ukázka skládání předběžně proloženého obrázku pomocí Ostré metody prokládání

Následující pseudokód popisuje obecný princip prokládání této metody pro vertikální prokládání.

Algoritmus 2 Ukázka Ostré metody pro vertikální prokládání

$lenses_f = floor((\text{width in inch}) \times (LPI))$
$N =$ (images count)

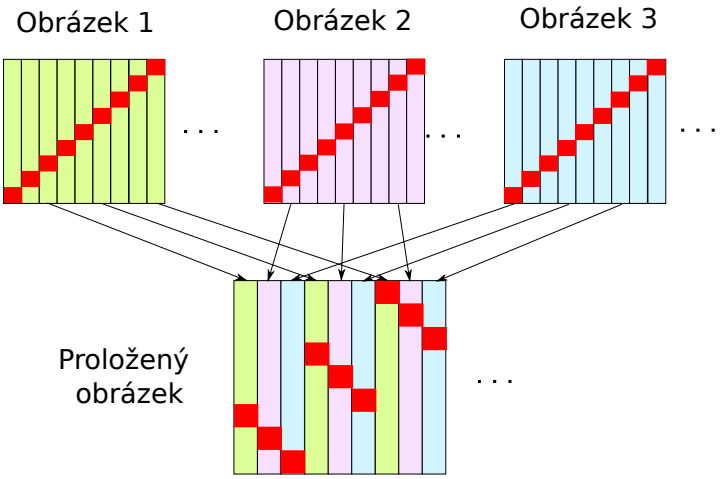
width of interlaced image $= (N \times lenses_f)$
height of interlaced image = height of input images

for (i = 0 TO (N - 1)) **do**
  Resize width of picture (i) to $(N \times lenses_f)$
  for (j = 0 TO $(lenses_f - 1)$) **do**
    Extract column $(i + j \times N)$ of picture (i)
    Insert the extracted column into column $(i + j \text{ x } N)$ of interlaced image
  end for
end for

Resize interlaced image width to $(lenses_f \times \frac{DPI}{LPI_v})$
Resize interlaced image height to $(height\ in\ inch \times DPI)$

2.3 Ostrá metoda se zrcadlením

Další prokládací metoda, se kterou jsme se setkali například ve Photo Projectoru, se od předchozí metody z kapitoly 2.2 liší pouze ve výběru sloupečků vstupních obrázků, které se vkládají do předběžně proloženého obrázku. Tato metoda prohazuje výběr vstupních sloupečků, tedy v prvním obrázku začneme od N-tého sloupečku a v posledním obrázku od prvního viz obrázek 2.3.



Obrázek 2.3: Ukázka skládání předběžně proloženého obrázku pomocí Ostré metody se zrcadlením

Kdybychom měli všechny obrázky stejné, pixely by se v rámci šířky jedné lentikule zrcadlily, proto jsme v Interlaceru tuto metodu pojmenovali "Ostrá metoda se zrcadlením".

Myšlenka této metody spočívá ve vlastnostech lentikulární čočky, která svým způsobem převrací obraz podobně jako objektiv fotoaparátu. Na obrázku 1.7 si můžeme všimnout, že při pohledu zprava jsou vidět sloupečky, které jsou umístěné při levém okraji lentikule a naopak. Budeme-li chtít, aby byl výsledný viditelný obraz nepřevrácený, měli bychom umístit sloupečky za lentikuli v opačném pořadí.

2.4 Implementace Ostrých metod

Při větším počtu vstupních obrázků nebo při výrobě větších lentikulárních obrázků se v Ostrých metodách musí vstupní obrázky značně roztáhnout. Kdybychom měli například 5 vstupních obrázků a celkový počet lentikulí vyráběného lentikulárního obrázku by byl 200, všechny vstupní obrázky by

se musely převzorkovat na šířku rovnou 1000 pixelů. Pokud bychom tento počet vstupních obrázků zvýšili například na 15, museli bychom šířku vstupních obrázků převzorkovat už na 3000 pixelů. Tyto obrázky už můžou mít oproti základní metodě z Interlaceru (viz kapitola 2.1), kde jsme převzorkovali vstupní obrázky pouze na celkový počet lentikulí, v našem případě 200 pixelů, nezanedbatelnou velikost v operační paměti.

Z převzorkovaných obrázků ani nepotřebujeme znát barevné hodnoty všech sloupečků, protože v Ostrých metodách vybíráme pouze některé.

Abychom ušetřili místo v operační paměti při vytváření těchto převzorkovaných obrázků, využili jsme v Interlaceru afinních transformací. Tyto transformace dovolují neceločíselný posun S pixelů při převzorkování. Díky tomu můžeme převzorkovat vstupní obrázky tak, abychom využili všechny sloupečky převzorkovaných vstupních obrázků. Tedy na stejnou velikost jako v základní metodě Interlaceru. Pokud by byl posun S při převzorkování pro všechny obrázky rovný nule, obrázky by se převzorkovaly stejně jako v prvotní metodě Interlaceru. Vhodnou volbou S pro každý vstupní obrázek je možné implementovat i obě Ostré metody. Díky afinní transformaci můžeme tedy vytvořit stejně velké převzorkované obrázky u všech tří metod a využít algoritmu 1 s jedinou změnou u převzorkování vstupních obrázků na řádku 8, tedy

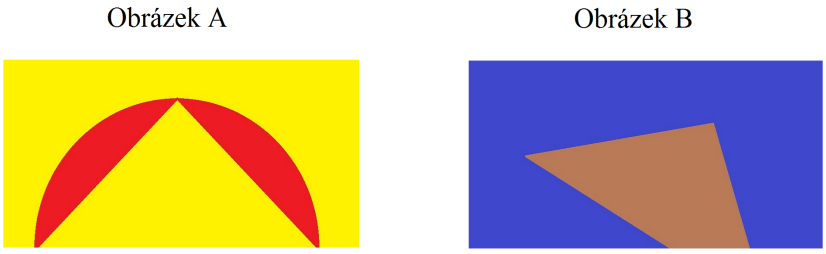
$$\textit{Resize width of picture i to } (lenses_f),\ \textit{apply shift } S_i.$$

Výsledný proložený obrázek, při použití afinních transformací u převzorkování vstupních obrázků na šířku $lenses_f$ pixelů, je srovnatelný s proloženým obrázkem vytvořeným při převzorkování vstupních obrázků na šířku $lenses_f \times N$ pixelů. To platí pouze v případě použije-li se v převzorkování u afinní transformace interpolace metodou Nejbližší soused.

2.5 Srovnání metod prokládání

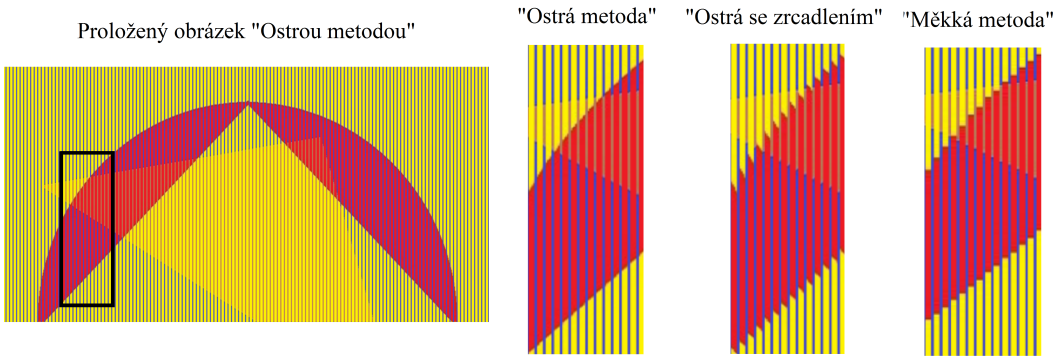
V první verzi Interlaceru byla implementována pouze základní metoda prokládání obrázků (viz kapitola 2.1). Tuto metodu jsme pojmenovali jako "Měkkou metodu", protože se zdálo, že při převzorkování vstupních obrázků jsou obrázky méně ostré než při využití ostatních metod.

Abychom mohli prokládací metody srovnat, vytvořili jsme tři proložené obrázky se stejnými parametry a porovnali je mezi sebou. Nejdříve jsme vytvořili proložené obrázky z uměle vyrobených vektorových obrázků viz obrázek 2.4.



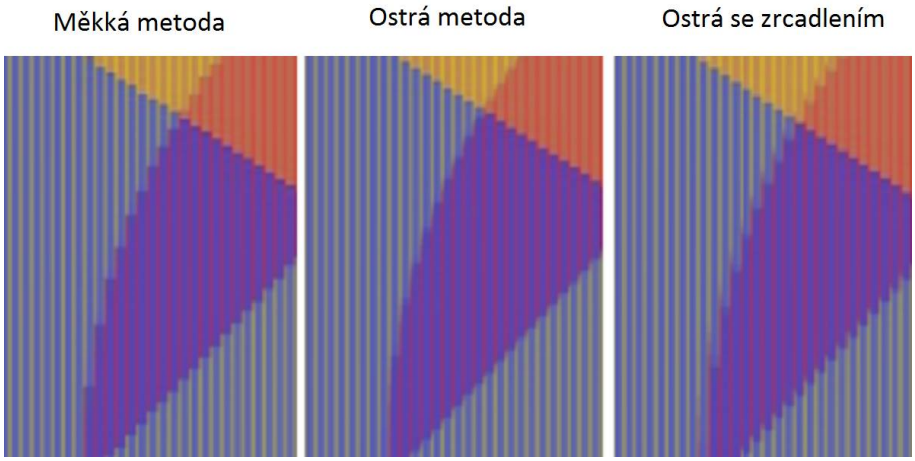
Obrázek 2.4: Vstupní obrázky pro prokládání

Na obrázku 2.5 si můžeme všimnout rozdílu těchto metod prokládání ve výstupním proloženém obrázku.



Obrázek 2.5: Ukázka rozdílu metod na vybrané části proloženého obrázku

Výsledné proložené obrázky jsme testovali v programu Lenticular Viewer. Na obrázku 2.6 můžeme vidět všechny tři proložené obrázky se simulací optické desky, kde jsme pro porovnání přiblížili detail na levé straně při výměně dvou obrázků. Při prvním pohledu vidíme značný rozdíl zobrazení kruhu na jeho hraně s pozadím. Můžeme si všimnout, že v tomto případě má Ostrá metoda oproti ostatním metodám hladší hrany. Nejhorší zobrazení hran kruhu má Ostrá metoda se zrcadlením, kde si můžeme všimnout značně zoubkovaných hran.



Obrázek 2.6: Ukázka proložených obrázků v Lenticular Viewer při přiblížení

Jako další test jsme provedli srovnání lentikulárních obrázků vytvořených z dvou rastrových obrázků 2.7.

Obrázek 2.7: Vstupní rastrové obrázky pro prokládání [9]

Na obrázku 2.8 je vidět srovnání těchto metod v Lenticular Viewer s přiblížením na detail hlavy.

Obrázek 2.8: Ukázka proložených obrázků v Lenticular Viewer

Na obrázku 2.8 si můžeme všimnout zásadního rozdílu v očích kočky v Ostré metodě se zrcadlením oproti ostatním metodám. Měkká metoda a Ostrá metoda bez zrcadlení jsou velmi podobné a žádný větší rozdíl jsme nenašli.

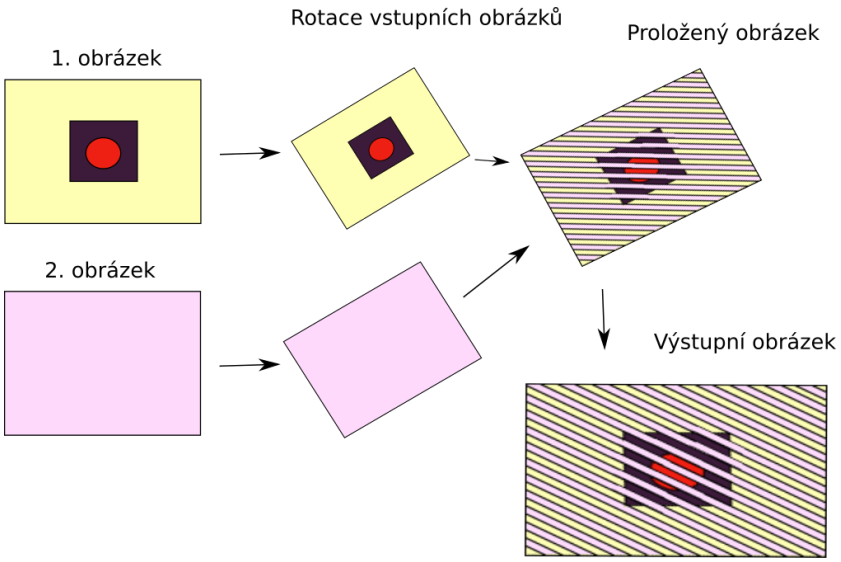
Dále jsme testovali i obrázky s 3D efektem. Můžeme si všimnout, že výsledky těchto metod na obrázcích 2.6 a 2.8 jsou srovnatelné a rozdíly nejsou tak zásadní.

Ostrá metoda se zrcadlením se jeví oproti ostatním metodám jako nejhorší. Jeden z autorů programu Photo Projectoru však tvrdí, že použití Ostré metody se zrcadlením je občas výhodnější než použití Ostré metody bez zrcadlení. Z tohoto důvodu jsme v Interlaceru ponechali obě varianty Ostrých metod.

Výsledky jednotlivých metod prokládání závisí z podstatné části na interpolačních filtrech, které se používají k převzorkování všech vstupních obrázků a výstupního proloženého obrázku. Použijeme-li u Měkké metody například interpolační filtr Nejbližší soused, dostaneme ostřejší lentikulární obrázek, který je srovnatelný s lentikulárním obrázkem vytvořeným Ostrou metodou.

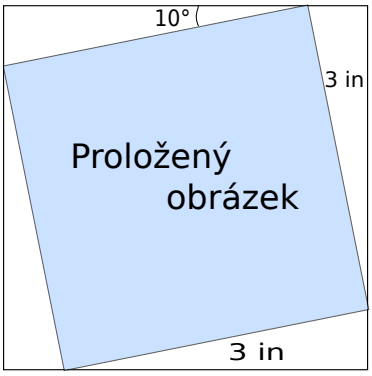
2.6 Šikmé prokládání

Toto prokládání bylo popsáno v sekci 1.3.3. Předchozí verze Interlaceru šikmé prokládání nepodporovala, a proto jsem tuto funkcionalitu doplnil. Možným způsobem, jak toto prokládání realizovat, je pootočení vstupních obrázků a následné proložení klasickým způsobem. Na obrázku 2.9 je znázorněný postup tohoto prokládání v Interlaceru.



Obrázek 2.9: Šikmé prokládání v Interlaceru

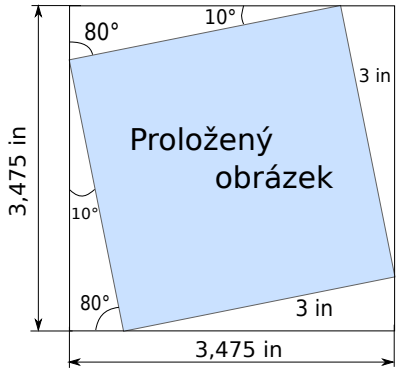
Toto prokládání se realizuje pomocí horizontálního prokládání, kde se však výška a šířka prokládaného obrázku musí přepočítat, aby samotný výstupní obrázek zabíral požadovanou výšku a šířku. Chceme například proložit obrázek pod úhlem 10° s vizuálním LPI 40, tiskovým rozlišením 600 DPI, šířkou a výškou výsledného lentikulárního obrázku 3 palce. Otočíme-li vstupní obrázky o 10°, změní se jejich výška a šířka. Požadujeme-li, aby výsledný proložený obrázek měl na šířku a výšku 3 palce, musíme počítat s tím, že po proložení bude také otočený o 10° viz obrázek 2.10.



Obrázek 2.10: Ukázka otočeného proloženého obrázku

Celková výška a šířka tohoto obrázku nebude 3 palce, ale 3,475 palce. Tuto hodnotu spočítáme díky znalosti úhlů a známých stran pomocí goniometrických funkcí viz obrázek 2.11. Známe-li výslednou velikost pootočeného

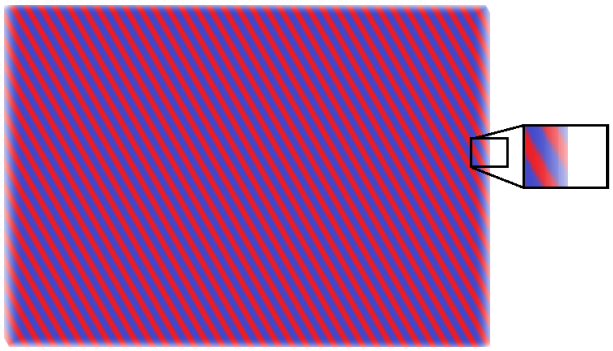
obrázku, můžeme postupovat stejným způsobem jako při horizontálním prokládání.



Obrázek 2.11: Ukázka otočeného proloženého obrázku s dopočítanými hodnotami

Výsledkem tohoto postupu vznikne natočený obrázek s horizontálním prokládáním. Po otočení zpět (v našem příkladě o -10°) se horizontálně proložené řádky jednotlivých obrázků nakloní a tím dostaneme zešikmeně proložený obrázek.

Při realizaci tohoto prokládání v Interlaceru vznikl při převzorkování na výslednou velikost obrázku nechtěný defekt. Kraje proloženého obrázku se při interpolaci dopočítávají nejen z pixelů náležících proloženému obrázku, ale i s bílou barvou pozadí. Díky tomu jsou kraje obrázku rozmazané. Například při prokládání červené a modré barvy s náklonem prokládání 60° viz obrázek 2.12. Tomuto problému se při této metodě prokládání nevyhneme.



Obrázek 2.12: Ukázka proloženého obrázku z modré a červené barvy při náklonu prokládání 60°

V této bakalářské práci neuvádím praktickou ukázku redukce moaré, protože jsme moaré nedokázali na naší tiskárně nasimulovat.

2.7 Práce s velkými obrázky

Kvalita lentikulárního obrázku závisí z podstatné části na tiskových možnostech uživatele. Přesná tiskárna s velkým tiskovým rozlišením DPI vytiskne kvalitnější obrázek než tiskárna s nízkým DPI. Pro vytištění obrázku s lepším rozlišením, však potřebujeme více barevné informace o obrázku. Tím se jeho velikost podstatně zvyšuje.

Kdybychom chtěli například vytisknout lentikulární obrázek 20 x 20 cm při tiskovém rozlišení 600 DPI, bude mít výsledný proložený obrázek 4724 x 4724 pixelů (tj. 22 MPx). Při použití 24 bitové hloubky (16 777 216 barev tj. 2^{24}) by tento obrázek v nekomprimované podobě zabíral skoro 67 MB. Budeme-li chtít vytvořit lentikulární obrázek, který má 100 x 100 cm, tak při stejném tiskovém rozlišení, bude jeho velikost 23 622 x 23 622 pixelů (tj. 558 MPx). Při stejné barevné hloubce by tento obrázek zabíral v operační paměti téměř 2,128 GB.

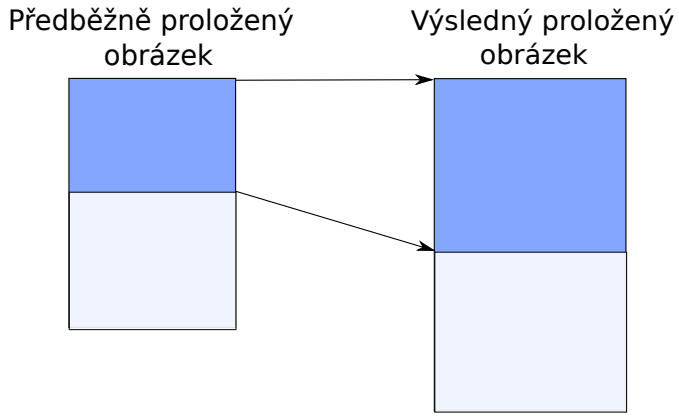
Pro tak velké obrázky a jejich vytvoření potřebujeme tedy značné množství v operační paměti.

2.7.1 Implementace v Interlaceru

Při vytváření velkých lentikulárních obrázků je většinou výsledný proložený obrázek několikanásobně větší než vstupní obrázky. Postup vytváření byl popsán v kapitolách 2.1, 2.2 a 2.3. Jako poslední krok se v metodách proložený obrázek převzorkuje pro tiskové rozlišení. V předchozí verzi Interlaceru byl problém toto finální převzorkování realizovat. Využívaná grafická knihovna Magick++ si alokovala pro výpočet velké množství paměti. Naším úkolem proto bylo, upravit finální převzorkování, aby při výpočtu knihovna nezabírala tolik paměťového prostoru. Zároveň počítáme s tím, že se obrázky vejdou v nekomprimované podobě do operační paměti a systém je nebude žádným způsobem omezovat (max. souvislá alokace atd.).

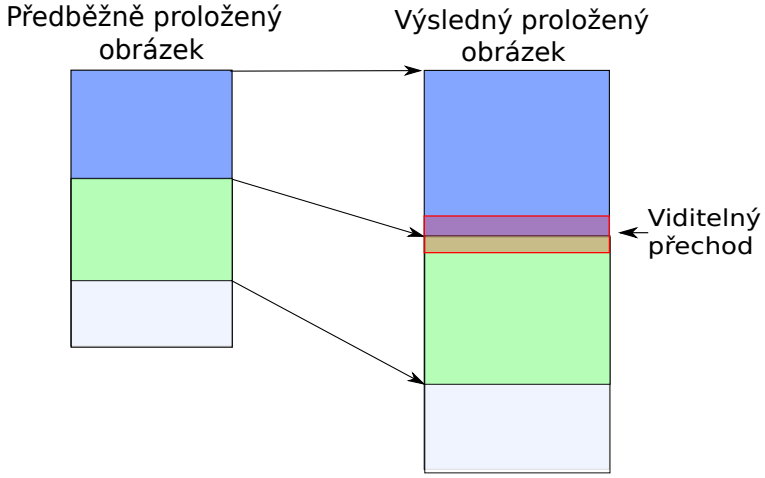
Po zamyšlení jsme uvažovali o dvou možných řešeních tohoto problému. Buď převzorkování obrázků budeme řešit jinými či vlastními interpolacemi nebo budeme převzorkovávat proložený obrázek postupně po jednotlivých částech a tyto části ukládat do finálního výstupního obrázku. Pro realizaci jsme zvolili druhou variantu, a to tedy postupné převzorkování na výsledný obrázek.

Nejdříve si vypočítáme velikost jednotlivých částí (bloků), které budeme zvětšovat. V nastavení Interlaceru můžeme definovat jakou maximální paměť může převzorkovaný blok zabírat. Známe-li velikost těchto bloků, můžeme realizovat první převzorkování viz obrázek 2.13.



Obrázek 2.13: Zobrazení bloku z předběžně proloženého obrázku do výsledného

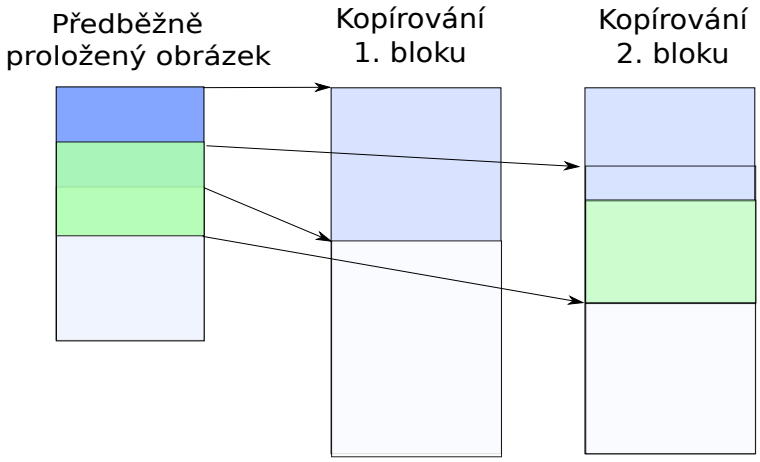
Pokud bychom brali bloky postupně za sebou, mohly by vznikat ve výsledném obrázku viditelné přechody mezi jednotlivými bloky viz obrázek 2.14.



Obrázek 2.14: Postupné vkládání převzorkovaných bloků do výsledného obrázku

Důvodem těchto viditelných přechodů je, že dopočítávané pixely by se interpolovaly jen z příslušných bloků. Abychom získali stejnou barevnou hodnotu těchto pixelů, jako při převzorkování celého obrázku, musíme do interpolace následujícího bloku zahrnout i několik koncových řádků pixelů předešlého bloku. Tyto řádky se musí vybírat s ohledem na použitý interpolační filtr. Čím více řádků pixelů interpolační filtr potřebuje k určení dopočítávaného řádku tím více řádků předešlého bloku musíme do následujícího bloku zahrnout.

Pokud bychom však do výsledného obrázku nakopírovali celý tento blok, došlo by opět ke stejnému problému jako na obrázku 2.14, protože by se horní pixely opět dopočítávaly jen z pixelů daného bloku. Proto do výsledného obrázku nebudeme kopírovat několik řádků horních pixelů bloku. Začátek kopírování posuneme na polovinu vzdálenosti začátku tohoto bloku a konce předešlého bloku viz obrázek 2.15.



Obrázek 2.15: Ukázka kopírování převzorkovaných bloků v Interlaceru

Při převzorkování bloků si musíme dát pozor ještě na jeden problém, který může při převzorkování nastat, a to zaokrouhlovací chyby. Pokud bychom měli například obrázek o velikosti 100 x 100 pixelů a zvětšovali ho na velikost 100 x 101 pixelů, barevná hodnota z původního obrázku na pozici y by se měla zobrazit na y' = 1,01×y. U velikosti bloků například 10 pixelů na výšku by se tento blok měl výškově převzorkovat na 10,1 pixelů. Protože obrázky mohou mít jen celočíselnou hodnotu pixelů, musíme tuto hodnotu zaokrouhlit. Pokud bychom tuto zaokrouhlovací chybu ignorovali, převzorkování po blocích by se lišilo oproti normálnímu celému převzorkování obrázku. V našem případě by se například pixely na pozici y = 80 zobrazily na pozici y' = 80 (oříznutí desetinné části), ale korektně by se měly zobrazit až na pozici y' = 80,8.

Jako příklad této zaokrouhlovací chyby si ukážeme lentikulární obrázek, který byl vytvořený ze dvou obrázků zelené a červené barvy. Při ignorování této zaokrouhlovací chyby vznikly, po přiložení optické desky (simulace v Lenticular Viewer), místy červené a místy zelené oblasti viz obrázek 2.16. Přerušovaná červená či zelená barva je odrazem zaokrouhlovací chyby a nepřesnosti v pozici pixelů jednotlivých barev.

Obrázek 2.16: Ukázka lentikulárního obrázku sestaveného po blocích se zaokrouhlovací chybou (simulace v Lenticular Viewer)

Pro vyřešení tohoto problému musíme nějak posunout pozice x' či y' v převzorkovaném bloku o desetinnou část. V používané knihovně Magick++ jsme využili afinních transformací. Tyto transformace přepočítávají pozice pixelů x a y z původního obrázku na pozici x' a y' v převzorkovaném obrázku vztahem

$$\begin{pmatrix} x & y & 1 \end{pmatrix} = \begin{pmatrix} x & y & 1 \end{pmatrix} \times \begin{pmatrix} s_x & r_x & 0 \\ r_y & s_y & 0 \\ t_x & t_y & 1 \end{pmatrix}. \tag{2.9}$$

Pozice x' a y' se tedy rovnají

$$(x', y') = (x \times s_x + y \times r_y + t_x, x \times r_x + y \times s_y + t_y). \tag{2.10}$$

Kde s_x a s_y určují zvětšení či zmenšení obrázku, t_x a t_y posun hodnot pixelů, r_x a r_y rotaci obrázku. Protože rotaci zde nevyužijeme, můžeme za r_x a r_y dosadit nulu a dostaneme rovnici

$$(x', y') = (x \times s_x + t_x, y \times s_y + t_y). \tag{2.11}$$

V našem příkladu, kde jsme zvětšovali obrázek ze 100 pixelů na 101 pixelů, bychom vypočítali hodnoty pro y' vztahem

$$y' = y \times 1.01 + t_y. \tag{2.12}$$

Bez použití bloků by t_y bylo rovné nule, protože zde žádný posun není. S použitím však musíme zobrazení posunout o desetinnou část, kterou jsme byli nuceni pro celočíselné hodnoty pixelů obrázků zaokrouhlit. V našem případě jsme například blok začínající na pozici y = 80 v původním obrázku zaokrouhlili na y' = 80 v převzorkovaném (oříznutí desetinné části). Tento blok

v převzorkovaném obrázku by měl správně začínat na pozici y' = 80.8, dosadíme tedy za t_y hodnotu 0.8. Tím vykompenzujeme chybu po zaokrouhlení, protože se barevné hodnoty pixelů posunou na příslušnou pozici.

Pro testování funkčnosti jsme porovnávali rozdílnost převzorkovaných obrázků s použitím afinní transformace bez použití bloků a s použitím bloků. Tyto obrázky se od sebe vůbec nelišily a díky tomu jsme tedy převzorkování po blocích s využitím afinní transformace mohli realizovat.

V knihovně Magick++ je různá implementace filtrace pro metodu Resize (kterou jsme využívali pro převzorkování v první verzi Interlaceru) a Distort, která je používaná pro realizaci afinních transformací. Tyto metody však pro stejnou volbu interpolačních filtrů generují obrázky, kde chyba neroste a je vlevo, vpravo i uprostřed stejná. Metody tedy poskytují srovnatelné výsledky, ale ne úplně stejné.

Po implementaci postupného převzorkování po blocích se prokázalo, že skutečně nevyčerpáme tolik paměťového prostoru jako při klasickém převzorkování celého obrázku. Proces převzorkování je však pomalejší než v klasickém případě a záleží na velikosti využitých bloků při převzorkování.

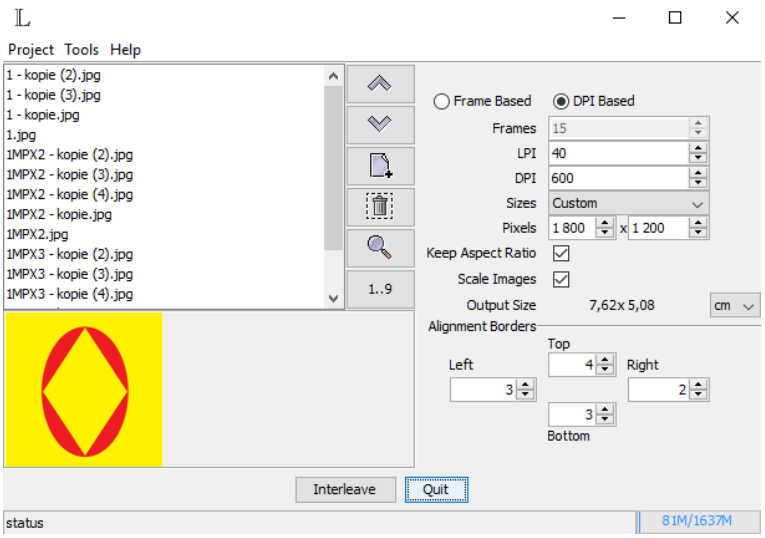
2.8 Programy pro lentikulární tisk

Lentikulární obrázky mohou být vytvořeny v běžných grafických editorech jako je Adobe Photoshop nebo Gimp. Nejčastěji se však používají specializované softwarové programy, které práci výrazně urychlují. Výstup této práce budeme porovnávat se třemi programy Photo Projector, Lentikit a Super Flip.

2.8.1 Lentikit

Lentikit je volně dostupný program napsaný v jazyce Java. Je snadno ovladatelný a nabízí pouze základní proložení, kde uživatel nemá možnost zvolit interpolaci pro převzorkování obrázku. Tento program prokládá vstupní obrázky pomocí **Měkké metody** prokládání popsané v sekci 2.1. Na obrázku 2.17 lze vidět uživatelské rozhraní tohoto programu.

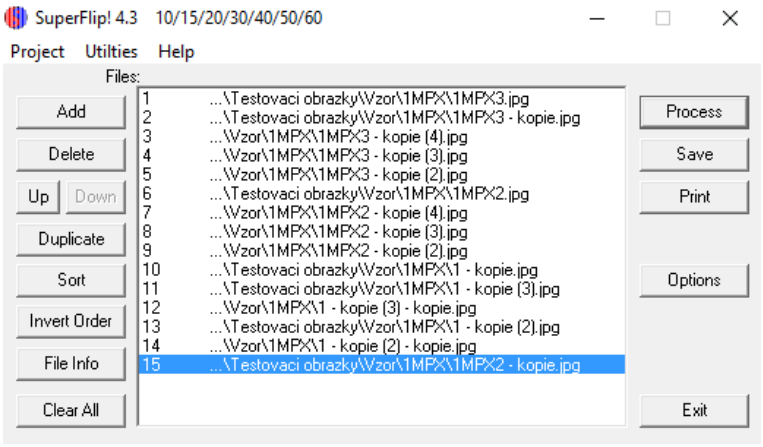
Oproti Interlaceru je velmi jednoduchý a nabízí pouze základní proložení obrázků. Nevýhodou tohoto programu je velká paměťová náročnost při prokládání i u malých obrázků, to však kompenzuje svou rychlostí při prokládání, která převyšuje Interlacer.



Obrázek 2.17: Ukázka uživatelského rozhraní v programu Lentikit

2.8.2 Super Flip

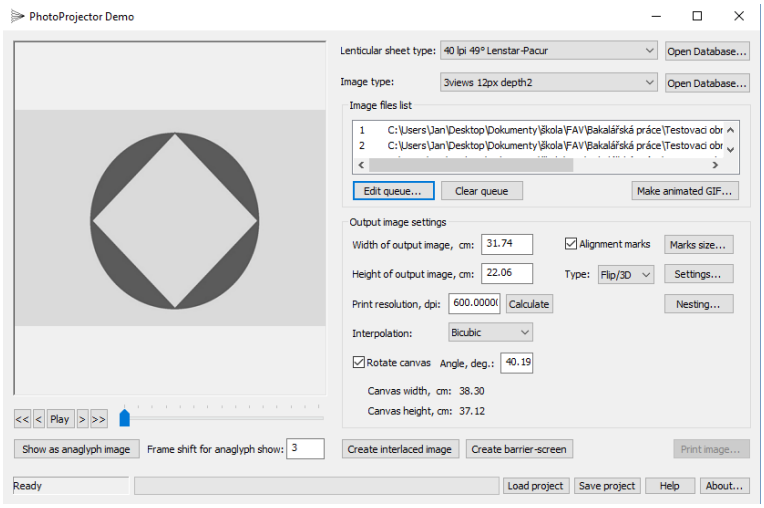
Tento program pro tvorbu lentikulárních obrázků je o něco sofistikovanější než Lentikit. Na obrázku 2.18 můžeme vidět ukázku uživatelského rozhraní tohoto programu. Super Flip prokládá obrázky pouze pomocí **Ostré metody** prokládání popsané v sekci 2.2. Také zde nemáme žádnou kontrolu nad interpolačními filtry, které jsou potřebné při převzorkování obrázků. Za pomocí tohoto programu však můžeme vytvořit jednoduché pitch testy. Volba rozdílu LPI sousedních proužků zde však není libovolná jako v Interlaceru. Paměťová náročnost při tvorbě proloženého obrázku je úměrná jeho velikosti. Super Flip při prokládání pracuje i s pevným diskem počítače a ukládá na něj mezivýsledky prokládání, tím je schopen efektivně vytvořit velké proložené obrázky. Rychlost vytvoření proloženého obrázku je pomalejší než u programu Lentikit, ale rychlejší než Interlacer.



Obrázek 2.18: Ukázka uživatelského rozhraní v programu Super Flip

2.8.3 Photo Projector Demo

Photo Projector je jedním z profesionálnějších programů pro vytváření lentikulárních obrázků. Je to komerční software, který disponuje několika verzemi jako Photo Projector Easy, Photo Projector a Photo Projector Plus lišící se například v maximální velikosti výstupního obrázku. V této práci využívám demo verzi 1.5, která dovoluje pouze černobílý výstup proloženého obrázku. Na obrázku 2.19 lze vidět uživatelské rozhraní toho programu.



Obrázek 2.19: Ukázka uživatelského rozhraní v programu Photo Projector Demo 1.5

Srovnání Photo Projectoru a Interlaceru je velmi zajímavé jak z pohledu prokládání tak z pohledu uživatelského rozhraní. Photo Projector dovoluje prokládat obrázky pomocí obou **Ostrých metod** prokládání (viz kapitola 2.2 a 2.3). Rychlostně je Photo Projector srovnatelný s Interlacerem, paměťově je však daleko úspornější při práci s většími obrázky. Photo Projector nepodporuje jako Interlacer šikmé prokládání, ale umí výstupní proložený obrázek orotovat kolem svého středu.

3 Kalibrace

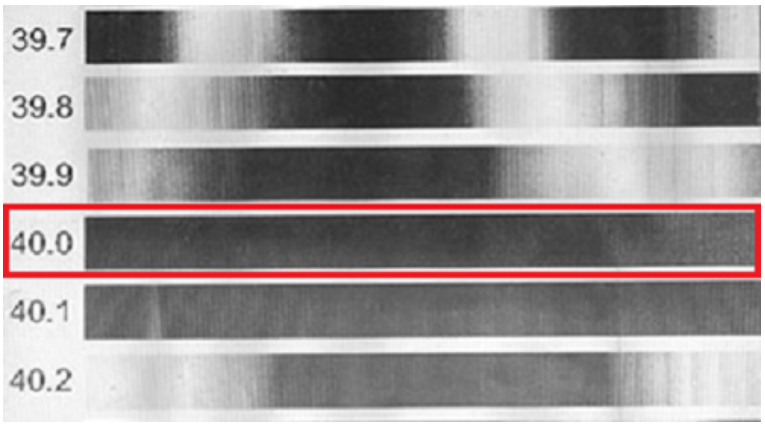
Pro dokonalý lentikulární obrázek potřebujeme znát přesné **vizuální LPI**. Volba tohoto vizuálního LPI není tak triviální, jak se na první pohled zdá. Pro správné proložení obrázku musíme vědět v jaké vzdálenosti se bude pravděpodobně pozorovatel nacházet. Také však musíme brát v potaz parametry optické desky (rozteče čoček, výrobní nepřesnost, atd.) a vady tiskárny při samotném tisku (rozpíjení inkoustu, prokluz papíru atd.). Metoda, která nám lépe pomůže určit vizuální LPI je **pitch test**. [1, 8]

3.1 Pitch test

Tato jednoduchá kalibrační metoda nám pomůže při vhodném výběru vizuálního LPI. Je založena na testování proložených obrázků s různou volbou LPI. Mějme například lentikulární desku s 60 LPI (mechanické). Vizuální LPI bude zřejmě blízko této hodnoty, zpravidla bývá o něco málo menší. Víme, že budeme chtít proložit například 10 bílých a 1 černý obrázek. Proložíme tedy tyto obrázky s hodnotou 60 LPI. Po přiložení na optickou desku zjistíme, že černý proužek je vidět ve všech pozorovacích úhlech přerušovaně (viz obrázek 3.1). Tím jsme ověřili, že jsme špatně volili vizuální LPI, proces tedy budeme opakovat například s hodnotou 59,95.

Obrázek 3.1: Obrázek s lentikulární deskou při špatné volbě vizuálního LPI

Pitch test je založen na sloučení více proložených obrázků s různým vizuálním LPI, abychom rychle a efektivně zjistili ideální LPI obrázku. Ideální LPI zjistíme tak, že při napasování pitch testu na folii uvidíme v některém z úhlů pohledu jeden černý proužek nepřerušovaně, tato hodnota LPI je pro nás nejvhodnější viz obrázek 3.2.



Obrázek 3.2: Ukázka pitch testu s lentikulární deskou [11]

Lidské oko je citlivé především na vnímání **kontrastu**. Čím menší kontrast mezi barvami, tím lze hůře určit, zda jsou barvy různé či stejné. Pro správnou volbu vizuálního LPI je nutné, aby kontrast mezi zvolenými barvami byl co největší. Tím usnadníme pozorovateli výběr vhodného LPI, protože lépe pozná celistvost barvy jednoho proužku pitch testu.

Klasický pitch test se obvykle tiskne na bílý papír. Chceme-li vytvořit pitch test z černých a bílých proužků (největší kontrast), bude tiskárna na papír tisknout pouze černou barvu. Inkoust se však na papíře rozpíjí a tím může dojít k tomu, že černé proužky budou o něco málo širší než bychom chtěli. Tím dochází k nepřesnosti pitch testu. Vytvoříme-li ten samý pitch test z černé a méně kontrastní barvy, bude tiskárna nucena tisknout i tuto barvu. Vlivem druhé barvy by se teoreticky mělo omezit rozpíjení černé. Vytvořený pitch test bude přesnější, ale pozorovatel hůře rozpozná ideální vizuální LPI, protože kontrast mezi barvami bude menší.

Možným způsobem jak zamezit rozpíjení inkoustu a usnadnit volbu vizuálního LPI je vytvořit pitch test z **několika kontrastních barev**. S tímto pitch testem jsme se zatím nesetkali v žádném testovaném programu (viz sekce 2.8). Lidské oko je velmi citlivé na barevný odstín. Použijeme-li tedy několik navzájem kontrastních barev, usnadníme pozorovateli výběr vhodného LPI a zamezíme rozpíjení inkoustu.

Nevýhodou této metody je potřeba jemnějšího tisku než u klasického dvoubarevného pitch testu. Vytváříme-li například pitch test z pěti barev, musí být každá barva zastoupena "pod jednou lentikulí". To znamená, že potřebujeme tisknout s rozlišením, při kterém se vejde minimálně pět pixelů za jednu lentikuli, abychom dosáhli výsledného efektu.

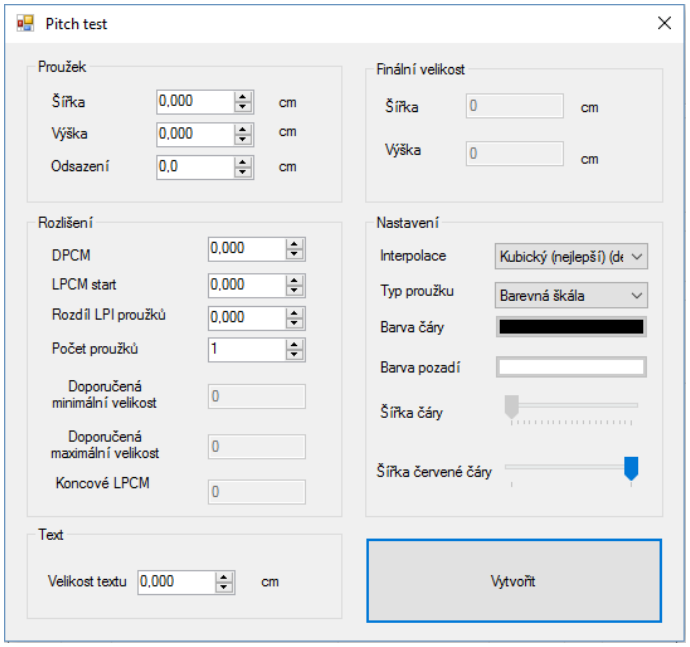
Většina lentikulárních obrázků se skládá z většího počtu proložených obrázků. Pro jejich zobrazení potřebujeme odpovídající tiskové rozlišení

a to odpovídá i více pixelům za jednou lentikulí. Vytváření barevného pitch testu z více barev, pro klasické lentikulární obrázky, by proto neměl být problém.

3.2 Implementace v Interlaceru

Předchozí verze Interlaceru uměla pouze vytvářet proložené obrázky pro lentikulární tisk. Před samotným proložením obrázků jsme však museli zjistit, pro jaké vizuální LPI obrázek prokládáme. To se zjišťovalo pomocí pitch testů z jiných specializovaných programů. Mým úkolem proto bylo přidat do Interlaceru nástroj pro vytváření pitch testů.

Rozhodli jsme se realizovat klasický dvoubarevný i několika barevný pitch test. Využili jsme algoritmů pro prokládání, které již byly v Interlaceru implementovány. Díky tomu jsme zvýšili přesnost výběru vhodného LPI, protože výstupní proložený obrázek a pitch test se vytváří stejným způsobem. Některé specializované programy pro lentikulární tisk vytváří pitch test staticky (tj. pouze zabarvují příslušné pixely na určitých pozicích). To však může vést k nevhodné volbě prokládaného LPI, protože lentikulární obrázek vytvořený při volbě tohoto LPI se vytváří jiným způsobem. Důsledkem toho je, že LPI vytvořeného obrázku může být odlišné od vybraného LPI na pitch testu. Na obrázku 3.3 je vidět přidaný nástroj pro výrobu pitch testu v Interlaceru.



Obrázek 3.3: Ukázka nástroje pro vytvoření pitch testu v Interlaceru

Při vytváření pitch testu v Interlaceru můžeme měnit běžné parametry jako velikost jednotlivých proužků, odsazení, velikost textu, LPI, DPI atd. jako u běžných nástrojů v konkurenčních programech. Dobrou pomůckou při výběru velikosti rozdílu LPI mezi jednotlivými proužky je nápověda, která říká jaký minimální nebo maximální rozdíl po sobě jdoucích proužků je vhodné pro pitch test zvolit.

3.2.1 Doporučená volba rozdílu LPI

Známe-li hodnotu mechanického LPI, tak víme, že vizuální LPI bude o něco málo menší (viz sekce 1.2). Díky tomu můžeme přibližně odhadnout v jakých hodnotách se bude vizuální LPI nacházet. Například pro desku s 40 LPI (mechanické) bude vizuální LPI ležet přibližně v intervalu od 39,0 až 40,0 LPI. Tím jsme si určili počáteční hodnotu LPI pro první proužek pitch testu. Otázkou je, jaký rozdíl LPI mezi ostatními proužky je vhodné zvolit, abychom po vyzkoušení pitch testu získali bližší informaci o hodnotě vizuálního LPI.

Určíme si délku jednoho proužku pitch testu, který obsahuje vzor pro LPI, jako W. Počet lentikulí N, pod který se tento vzor vejde je

$$N = W \times LPI. \tag{3.1}$$

Pro zobrazení vzoru v pixelech při tiskovém rozlišení (DPI) tento vzor odpovídá počtu pixelů N_p, kde

$$N_p = \frac{DPI}{LPI} \times N. \tag{3.2}$$

Rozdíl délky vzoru Δ (v pixelech) dvou po sobě jdoucích pruhů pitch testu je roven

$$\Delta = \frac{DPI}{LPI_1} \times N - \frac{DPI}{LPI_2} \times N. \tag{3.3}$$

Převodem získáme tvar

$$\Delta = DPI \times N \times \frac{LPI_2 - LPI_1}{LPI_1 \times LPI_2}. \tag{3.4}$$

Pokud jsou hodnoty LPI_1 a LPI_2 velmi blízké původní hodnotě LPI, tj. platí $\mathrm{LPI}_1 \approx \mathrm{LPI}_2 \approx \mathrm{LPI}$, a zavedeme-li rozdíl $\delta = \mathrm{LPI}_2 - \mathrm{LPI}_1$, můžeme psát

$$\Delta = DPI \times (W \times LPI) \times \frac{\delta}{LPI^2}. \tag{3.5}$$

Konečným převodem získáme tvar

$$\Delta = DPI \times W \times \frac{\delta}{LPI}, \tag{3.6}$$

který nám určuje rozdíl dvou vzorů LPI v pixelech.

Při tisku pitch testu musíme zvolit δ (tj. rozdíl dvou vizuálních LPI). Pokud bychom zvolili příliš malou hodnotu, rozdíl dvou vzorů by mohl být menší něž jeden pixel. To by znamenalo, že by se proužky na pitch testu od sebe nelišily téměř vůbec. Pro zvolení δ tedy chceme, aby platilo

$$\delta > \frac{LPI}{DPI \times W}. \tag{3.7}$$

Máme-li například tiskárnu s tiskovým rozlišením 600 DPI, lentikulární desku s 40 LPI (mechanické) a šířka proužků pitch testu bude 3,2 palce, musí platit že $\delta > 0{,}0208$.

Ze vzorce (3.7) je patrné, že zvětšováním délky proužku W (při stejném LPI a DPI) klesá hodnota δ. Kdybychom v předchozím příkladu použili délku proužku rovnou velikosti papíru formátu A4, tedy přibližně 8,26 palce, bude δ rovna přibližně 0,0081. Takto malý rozdíl LPI se nám například nemusí hodit, pokud neznáme ani přibližnou hodnotu vizuálního LPI. Protože rozdíl proužků je velmi malý, uživatel by musel vytisknout mnoho proužků pitch testu, než by přišel na to, které LPI je nejvhodnější. Z tohoto důvodu je tedy dobré určit maximální rozdíl LPI dvou sousedních proužků, při kterém pozorovatel dostane přibližnou hodnotu vizuálního LPI.

Pokud by byl tento rozdíl větší než šířka jedné lentikule, jednotlivé proužky pitch testu by se od sebe velmi lišily a pozorovatel by měl opět problémy s určením vizuálního LPI. Budeme tedy požadovat, aby rozdíl dvou sousedních proužků byl menší než šířka jedné lentikule.

$$\frac{DPI}{LPI} > \Delta \tag{3.8}$$

Určíme-li si proměnou K, která nám udává procentuální šířku jedné lentikule, dostaneme

$$K \times \frac{DPI}{LPI} > \delta \times \frac{DPI}{LPI} \times W \tag{3.9}$$

Pro určení δ upravíme rovnici na tvar

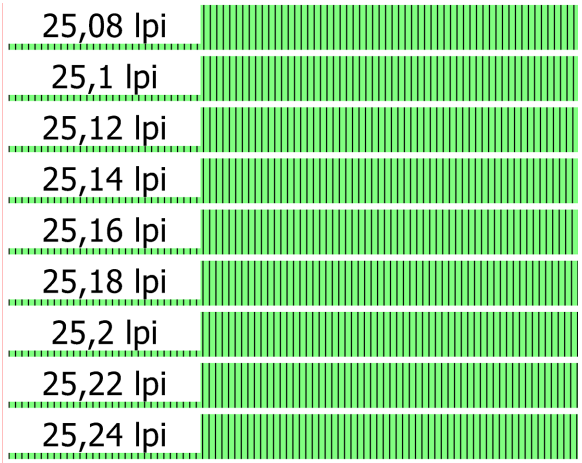
$$\delta < \frac{K}{W} \tag{3.10}$$

Vyžadujeme-li například, abychom na levé straně dvou sousedních proužků viděli jiný obrázek než na straně pravé, bude posun vzorů o polovinu lentikule tedy o K rovno 0,5. Dosadíme-li tuto hodnotu do předchozího příkladu, maximální doporučený rozdíl dvou sousedních proužků pitch testu pro formát papíru A4 by měl být menší než 0,06 LPI.

V Interlaceru jsme zvolili dvě doporučené hodnoty označené jako doporučená minimální a maximální velikost rozdílu LPI. Minimální doporučený rozdíl je rozdíl vzorů pitch testu o jeden pixel a maximální doporučený rozdíl je právě o polovinu jedné lentikule.

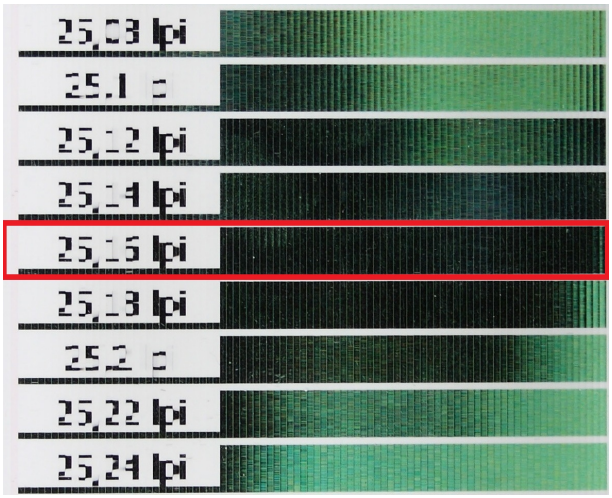
3.2.2 Dvoubarevný pitch test

Klasický dvoubarevný pitch test v Interlaceru můžeme vytvořit z dvou libovolných barev. Jak již bylo zmíněno na začátku této kapitoly, je rozumné zvolit dvě navzájem kontrastní barvy. Na obrázku 3.4 je vytvořený pitch test s černými proužky a zelenou barvou pozadí. Na levé straně obrázku je vidět červená čára, která slouží k usazení lentikulární folie nad pitch testem.



Obrázek 3.4: Ukázka standardního dvoubarevného pitch testu

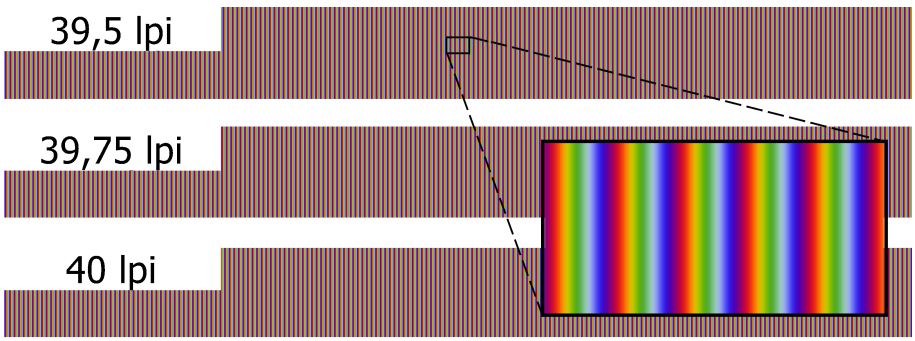
Po usazení lentikulární fólie na tomto pitch testu můžeme pozorovat jaké vizuální LPI je pro vytvoření proloženého obrázku v Interlaceru nejvhodnější. Na obrázku 3.5 je pitch test s lentikulární deskou. Jako nejlépe se nám jeví proužek s vizuálním LPI rovno 25,16. Tuto hodnotu bychom tedy použili pro vytváření proloženého obrázku.



Obrázek 3.5: Ukázka pitch testu s lentikulární deskou

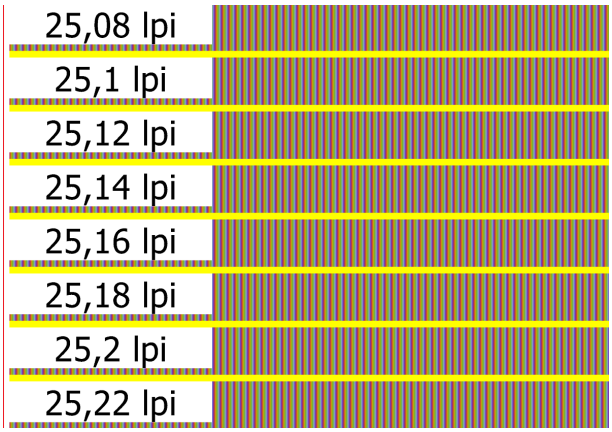
3.2.3 Několika barevný pitch test

Další pitch test, který lze v Interlaceru vytvořit, se skládá z několika barev. Zvolili jsme základní barvy jako červenou, žlutou, zelenou, azurovou a purpurovou. Na obrázku 3.6 je vytvořený pitch test v Interlaceru s přiblížením na barvy pro lepší pozorování detailu pitch testu.



Obrázek 3.6: Ukázka několika barevného pitch testu v Interlaceru

Stejně tak jako v sekci 3.2.2 jsme vytvořili pitch test pro stejnou lentikulární desku viz obrázek 3.7.



Obrázek 3.7: Ukázka několika barevného pitch testu

Usazením lentikulární desky na tento pitch test také zjistíme vhodné LPI pro vytvoření proloženého obrázku. Snažíme se najít proužek, který má po celé jeho délce stejnou barvu.

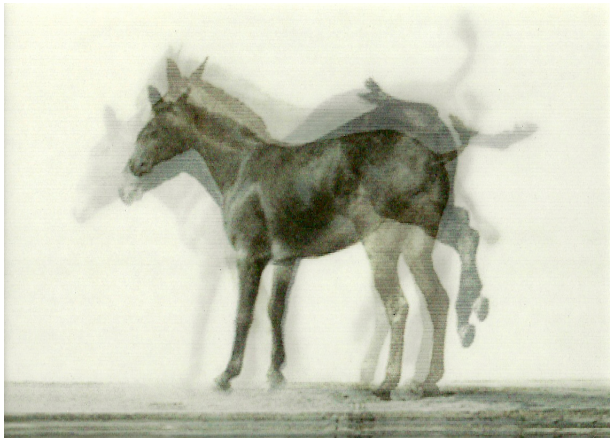
Obrázek 3.8: Ukázka několika barevného pitch testu s lentikulární deskou

Na obrázku 3.8 si můžeme všimnout, že výběr vhodného LPI u více barevného pitch testu není tak triviální, jak bychom čekali. Vhodné LPI se v tomto případě pohybuje mezi hodnotami 25,12 až 25,16 LPI. Záleží také na citlivost oka pozorovatele na barevný tón. V klasickém pitch testu byl rozdíl proužků lépe poznat, protože jsme rozlišovali rozdíl dvou kontrastních barev, kde jsme jasně věděli v jaké barvě má být proužek s vhodným LPI. Ve více barevném pitch testu s malým rozdílem LPI se hůře určuje celistvost jedné barvy proužku pitch testu.

4 Přeslechy

V reálném světě není vše tak jednoduché. Při výrobě lentikulárního obrázku může dojít k mnoha vadám, které způsobí zhoršení kvality lentikulárního obrazu. Nelze například vyrobit dokonalou optickou desku, kde by všechny čočky měly naprosto stejnou rozteč a světlo dopadající na tyto čočky se lámalo v požadovaném směru. Samotná tiskárna, která tiskne výsledný proložený obrázek, může vlivem prokluzu papíru či rozpíjení inkoustu na papíře zhoršit kvalitu obrázku. Také jsme při vysvětlování principu lentikulární desky v kapitole 1.2 předpokládali, že se lentikulární čočka chová jako ideální tenká čočka a paprsky vycházející z jednoho bodu (oko pozorovatele) jsou lámány do jednoho bodu na zadní straně ohniskové roviny. To však není zcela korektní. Lentikulární čočka není zcela dokonalá.

Vlivem všech těchto problémů dochází ke zhoršení kvality výsledného lentikulárního obrázku. Oddělení pravého snímku a levého snímku nemusí být, a většinou nikdy není, zcela dokonalé. Levé oko může částečně vidět pravý snímek a naopak. Technicky se tento nežádoucí jev nazývá **přeslech**. Nejčastěji si můžeme přeslechu všimnout u lentikulárního obrázku vytvořeného z proložených obrázků s vysokým kontrastem. Na obrázku 4.1 je lentikulární obrázek, kde dochází k přeslechu na několika místech, například mezi bílou barvou pozadí a hlavou koně. Poznatky pro tuto kapitolu byly čerpány především ze zdroje [3].

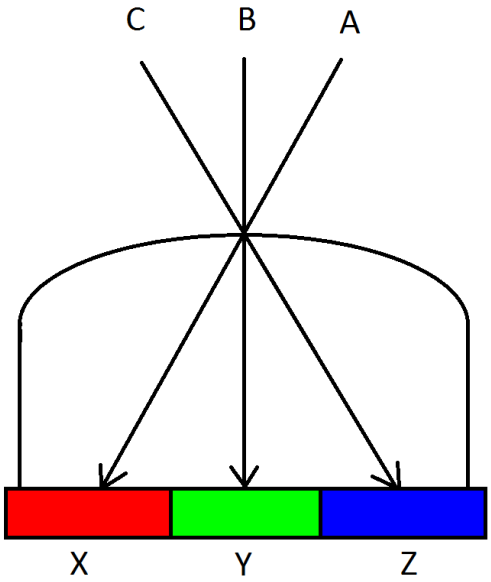


Obrázek 4.1: Ukázka přeslechů na lentikulárním obrázku [10]

4.1 Odstranění přeslechu

U lentikulárních obrázků jsou přeslechy nežádoucí a pro zvýšení kvality obrázku se jich potřebujeme zbavit či je eliminovat. Nejvíce se přeslechy projevují u lentikulárních obrázků vytvořených z kontrastních barev například z černé a bílé. Pro eliminaci přeslechů můžeme například sestavit lentikulární obrázek z obrázků se stejným barevným tónem.

V této práci jsme se rozhodli vyzkoušet eliminaci přeslechů drobnou změnou vstupních obrázků.



Obrázek 4.2: Lentikulární čočka s proloženým obrázkem

Na obrázku 4.2 je znázorněna jedna lentikulární čočka se třemi proloženými obrázky X, Y a Z. Tyto obrázky jsou nejlépe pozorovány z bodů A, B, C. Dejme tomu, že chceme z pohledu A vidět pouze obrázek reprezentující písmeno X. Vlivem nedokonalostí, však uvidíme částečně obrázek Y a v krajním případě i obrázek Z. Z pohledu A tedy nevidíme pouze X, ale kombinaci barev aX + bY + cZ (kde a, b, c jsou procenta viditelnosti). Z pohledů A, B a C vidíme kombinace určené vztahem

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \times \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{4.1}$$

Převodem získáme rovnici

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}^{-1} \times \begin{pmatrix} A \\ B \\ C \end{pmatrix}, \tag{4.2}$$

po dosazení a rozepsání

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \times \left(\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}^{-1} \times \begin{pmatrix} A \\ B \\ C \end{pmatrix} \right). \tag{4.3}$$

Z rovnice 4.3 vyplývá, že pro eliminaci přeslechů stačí znát pouze procenta viditelnosti jednotlivých obrázků z různých úhlů pohledu. Z těchto procent pak sestavíme inverzní matici, kterou vynásobíme se vstupními obrázky. Tyto přepočítané obrázky pak standardně proložíme a po přiložení lentikulární folie bychom měli pozorovat eliminaci přeslechu.

Stěžejní část této metody spočívá v určení viditelnosti jednotlivých snímků z různých úhlů pohledu a v počtu proložených snímků. Máme-li N proložených snímků označené $A_1, A_2, ..A_n$, bude mít matice procentuální viditelnosti snímků rozměr N×N. Obecně platí

$$\begin{pmatrix} A_1 \\ A_2 \\ . \\ . \\ A_n \end{pmatrix} = \begin{pmatrix} n_{11} & n_{12} & . & . & n_{1n} \\ n_{21} & n_{22} & . & . & n_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ n_{n1} & n_{n2} & . & . & n_{nn} \end{pmatrix} \times \left(\begin{pmatrix} n_{11} & n_{12} & . & . & n_{1n} \\ n_{21} & n_{22} & . & . & n_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ n_{n1} & n_{n2} & . & . & n_{nn} \end{pmatrix}^{-1} \times \begin{pmatrix} A_1 \\ A_2 \\ . \\ . \\ A_n \end{pmatrix} \right). \tag{4.4}$$

Pokud bychom chtěli vytvořit lentikulární obrázek například ze sedmi obrázků, museli bychom pro výpočet určit čtyřicet devět hodnot. Určení takového počtu hodnot je celkem vyčerpávající, nejprve jsme však museli určit, zda tato metoda opravdu funguje.

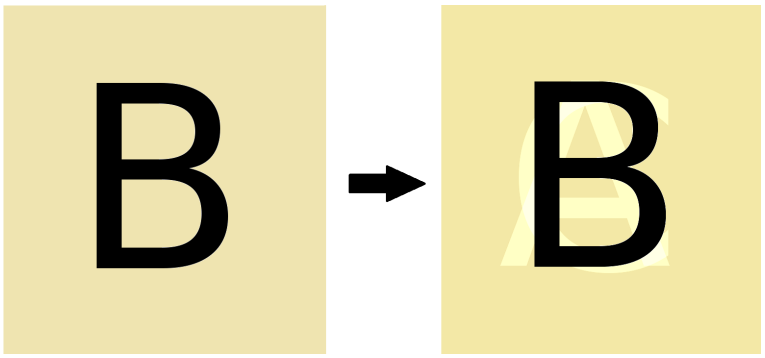
Pro testování jsme zvolili tři obrázky s písmeny A, B a C. Určili jsme si z kolika procent je vidět hlavní snímek, který má být vidět, a z kolika procent vedlejší nežádoucí snímky. Do vzorce

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \times \left(\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}^{-1} \times \begin{pmatrix} A \\ B \\ C \end{pmatrix} \right) \tag{4.5}$$

jsme dosadili vymyšlené hodnoty. Například pro viditelnost 90 % hlavního snímku jsme dosadili

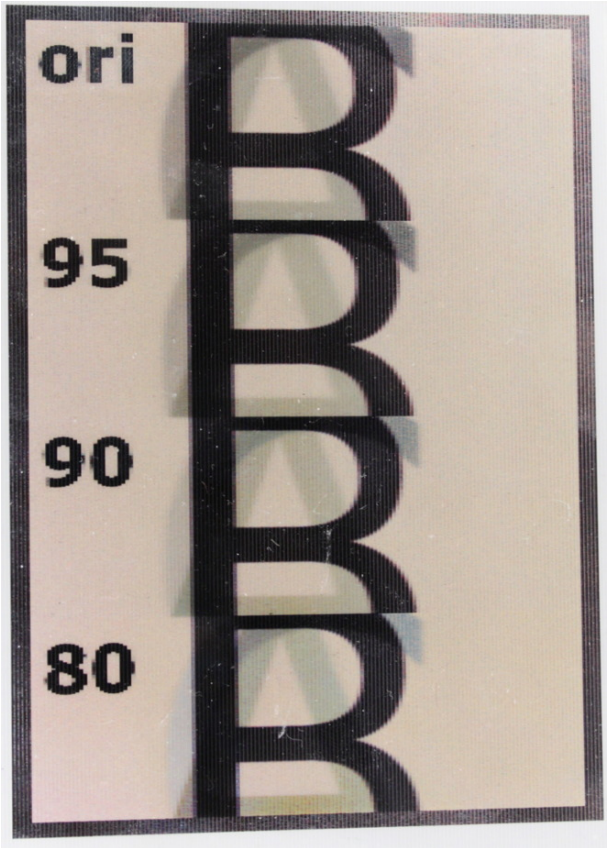
$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} 0.90 & 0.07 & 0.03 \\ 0.05 & 0.90 & 0.5 \\ 0.03 & 0.07 & 0.90 \end{pmatrix} \times \left(\begin{pmatrix} 0.90 & 0.07 & 0.03 \\ 0.05 & 0.90 & 0.5 \\ 0.03 & 0.07 & 0.90 \end{pmatrix}^{-1} \times \begin{pmatrix} A \\ B \\ C \end{pmatrix} \right). \quad (4.6)$$

Vstupní snímky jsem pomocí předešlého vzorce upravil. Ve výsledných snímcích byla vidět z malé části i ostatní písmena. Na obrázku 4.3 vlevo je vidět neupravený snímek s písmenem B, pravý obrázek je již upravený pomocí rovnice.



Obrázek 4.3: Ukázka normálního a upraveného snímku

Pro srovnání jsem vytvořil vstupní snímky s viditelností hlavního obrázku z 95 %, 90 % a 80 %. Odpovídající snímky jsem proložil a srovnal s originálem. Na obrázku 4.4 je vidět srovnání těchto proložených obrázků při pohledu na snímek B.



Obrázek 4.4: Ukázka lentikulárního obrázku při pohledu na snímek B

Můžeme si všimnout, že v originálním proloženém obrázku nahoře jsou písmena A a C vidět o něco více než v upravených obrázcích při 80 % a 90 %. Tím se potvrdilo, že tato metoda částečně eliminuje přeslechy. V případě, kdy je hlavní obrázek vidět z 80 % je již vidět písmeno C příliš světlou barvou. Testování tímto způsobem není dokonalé, protože procentuální viditelnost snímků jsme si vymysleli a pouze otestovali, zda tato metoda funguje. Také jsme použili pouze tři vstupní obrázky pro proložení. Kdybychom zvýšili počet vstupních obrázků, zvětšil by se také řád matice a k výpočtu bychom potřebovali více informací o viditelnosti dalších snímků.

Pro přesnější výpočty bychom museli vymyslet metodu pro měření viditelnosti okolních snímků. To můžeme realizovat například fotoaparátem, který by snímal viditelnost snímků v jednotlivých pozorovacích úhlech a tyto snímky pořízené fotoaparátem bychom zpracovávali v počítači.

Tato metoda eliminace přeslechů a přeslechy samotné nejsou v této práci pro nedostatek času dále rozvíjeny.

Závěr

Existuje mnoho způsobů, jak můžeme vytvořit lentikulární obrázek. V rámci této bakalářské práce byly popsány některé z nich. Při srovnání těchto metod jsme došli k závěru, že kvalita lentikulárních obrázků závisí z podstatné části na interpolačních filtrech použitých u převzorkování obrázků.

Prokládání obrázků v prvotní verzi Interlaceru má srovnatelné výsledky s nově implementovanými Ostrými metodami. Použití Ostré metody se zrcadlením při vytváření lentikulárního obrázku přináší oproti ostatním metodám nepatrně horší výsledky. Také v této metodě velmi záleží na přesnosti napasované optické desky oproti normální Ostré metodě.

Při práci s velkými obrázky jsme využili postupné převzorkování po částech. Díky tomu jsme ušetřili místo v operační paměti, ale rychlost vytváření výsledného proloženého obrázku se snížila.

Dále jsme v této práci zkoumali výsledky klasického a několika barevného pitch testu. Z klasického pitch testu jsme dokázali lépe a rychleji určit ideální vizuální LPI desky. Několika barevný pitch test se tedy v praxi příliš neosvědčil.

Další prací v tomto projektu byla redukce přeslechů mezi proloženými snímky. Uvedená metoda přeslechy při testování nepatrně redukovala, ale požadovaný efekt nebyl tak značný, jak jsme čekali. Testování jsme však prováděli s nenaměřenými hodnotami, které lze obtížněji určit, a proto se ve zkoumání této metody může pokračovat.

Literatura

[1] Andrew Rowbottom, *Lentikit - Calibration.* Sourceforge [online]. 2004 [cit. 2016-04-13]. Dostupné z: <http://lentikit.sourceforge.net/calibration.html>

[2] BERKEL, Van. *CHARACTERISATION AND OPTIMISATION OF 3D-LCD MODULE DESIGN.* Petr Lobaz – domácí stránka. [online]. [cit. 2016-01-02]. Dostupné z: <http://www.kiv.zcu.cz/~lobaz/lentikular/clanky/Berkel97.pdf>

[3] CG Sheng. *Ghosting – How to Detect and Avoid it when Designing Animation-effect Artwork For Lenticular Printing.* Lenticular Printing by ViCGI. [online]. [cit. 2016-01-02]. Dostupné z: <http://www.vicgi.com/How_to_Prevent_Ghosting_20120113.pdf>

[4] DAVID, E. Roberts. *History of Lenticular and Related Autostereoscopic Methods.* Petr Lobaz – domácí stránka. [online]. [cit. 2016-01-02]. Dostupné z: <http://www.kiv.zcu.cz/~lobaz/lentikular/clanky/History-of-Lenticular.pdf>

[5] DAVID E. Roberts, TREBOR Smith. *The History of Integral Print Methods.* Petr Lobaz – domácí stránka. [online]. [cit. 2016-01-02]. Dostupné z: <http://www.kiv.zcu.cz/~lobaz/lentikular/clanky/Integral-History.pdf>

[6] GALI-3D: *3D Technologická knihovna [online].* © 2005-2011 [cit. 2015-12-07]. <http://cs.gali-3d.com/stereoskopie-3d/>

[7] *How does lenticular printing work?* Explain that Stuff. [online]. [cit. 2016-01-02]. Dostupné z: <http://www.explainthatstuff.com/lenticularprinting.html>

[8] Jameson Bannett, *Lenticular Image Creator* [online]. 2006 [cit. 2016-04-13]. Dostupné z: <http://lenticularimagecreator.com/docs/calibration.html>

[9] Kitty Explosion. *LoveThisGif* [online]. [cit. 2016-06-18]. Dostupné z: <http://www.lovethisgif.com/image/676/kitty-explosion->

[10] Lenticular Printing – An Amazing Technique for 3D and Motion Effects. *TaPhotos* [online]. [cit. 2016-06-18]. Dostupné z: <http://taphotos.com/photo-to-art/lenticular-printing-an-amazing-technique-for-3d-and-motion-effects/>

[11] Lenticular Printing. *Lenticular Printing by ViCGI* [online]. [cit. 2016-06-18]. Dostupné z: <http://www.vicgi.com/blog/>

[12] LOBAZ, Petr. *KVU / UF3D 3D fotografie a alternativní techniky ve fotografii* Lobaz Petr. Petr Lobaz – domácí stránka. [online]. [cit. 2016-01-02]. Dostupné z: <http://www.kiv.zcu.cz/~lobaz/uf3d/04_multiview/text09_autostereo.html>

[13] LOBAZ, Petr. *KVU / UF3D 3D fotografie a alternativní techniky ve fotografii.* Petr Lobaz – domácí stránka. [online]. [cit. 2016-01-02]. Dostupné z: <http://www.kiv.zcu.cz/~lobaz/uf3d/04_multiview/text10_lentikular.html>

[14] OKOSHI, Takanori. *Three-Dimensional Imaging Techniques.* 2. vydání. Los Angeles: Atara Press, 2011. ISBN 978-0-09822251-4-1

[15] RAYMOND Mark A., HECTOR Andres Porras Soto. *Slanted lens interlacing* [online]. [cit. 2016-01-02]. Dostupné z: <http://www.google.com/patents/US20140153007>

[16] URAY, H.; CHELLAPPAN, K.V.; ERDEN, E.; SURMAN, P. *State of the Art in Stereoscopic and Autostereoscopic Displays* Proceedings of the IEEE , vol.99, no.4, pp.540,555, [online]. Duben 2011 [cit. 2016-01-02]. Dostupné z: <http://www2.units.it/ramponi/teaching/DIP/materiale/z04_3D_Urey11.pdf>