

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## OPRAVA LICHOBĚŽNÍKOVÉHO ZKRESLENÍ PROJEKTORU (KEYSTONING PROBLEM)

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

RADEK VRZAL

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# OPRAVA LICHOBĚŽNÍKOVÉHO ZKRESLENÍ PROJEKTORU (KEYSTONING PROBLEM)

AUTOMATIC KEYSTONE CORRECTION FOR CAMERA-PROJECTOR SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RADEK VRZAL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MAREK ŠOLONY

BRNO 2013

## Abstrakt

Tato práce se zabývá problémem lichobežníkového zkreslení projektoru. Jsou zde popsány příčiny vzniku a také různé metody korekce tohoto jevu. Cílem práce bylo vytvořit aplikaci, která pomocí webkamery připojené k počítači a minimálního vstupu od uživatele zkreslení opraví. Toho bylo docíleno pomocí analýzy obrazu z webkamery a grafických transformací obrazu, které jsou v práci také popsány.

## Abstract

The thesis deals with projector's Keystone effect. Causes of the Keystone effect occurrence and the effect's diverse correction methods are described. The aim of the thesis is to develop the application capable of correcting distortions using a minimum user input and a webcamera connected to a computer. This was achieved through camera image analysis and the graphic transformations of the image as those methods are also described in the elaborate.

## Klíčová slova

lichobežníkové zkreslení, počítačové vidění, homografie, automatická oprava lichoběžníkového zkreslení, automatická kalibrace systému kamera-projektor

## Keywords

keystone effect, computer vision, homography, automatic keystone correction, automatic projector-camera calibration

## Citace

Radek Vrzal: Oprava lichoběžníkového zkreslení projektoru (keystoning problem), bakalářská práce, Brno, FIT VUT v Brně, 2013

# Oprava lichoběžníkového zkreslení projektoru (keystoning problem)

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Marka Šolonyho

.....  
Radek Vrzal  
15. května 2013

## Poděkování

Děkuji vedoucímu své práce, panu Ing. Marku Šolonymu, za rady a čas, který mi věnoval. Také bych rád poděkoval panu Zdeňku Juříčkovi, za zprostředkování vypůjčení projektoru.

© Radek Vrzal, 2013.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Lichoběžníkové zkreslení</b>	<b>4</b>
2.1	Oprava zkreslení . . . . .	4
2.1.1	Optická korekce . . . . .	4
2.1.2	Variable Optical Lens Shift . . . . .	6
2.1.3	Digitální korekce . . . . .	6
2.2	Korekce lichoběžníkového zkreslení v systému kamera-projektor . . . . .	7
<b>3</b>	<b>Homografie a její využití při kalibraci systému</b>	<b>9</b>
3.1	Homogenní souřadnice . . . . .	10
3.2	Výpočet homografie . . . . .	10
3.3	Homografie v systému kamera-projektor . . . . .	11
<b>4</b>	<b>Detekce význačných bodů v obraze kamery</b>	<b>13</b>
4.1	Detekce hran v obraze . . . . .	13
4.1.1	Konvoluce . . . . .	13
4.1.2	Sobelův operátor . . . . .	14
4.1.3	Cannyho hranový detektor . . . . .	15
4.1.4	Houghova transformace . . . . .	15
4.2	Detekce oblastí ve scéně . . . . .	16
4.2.1	Detekce oblasti projekce . . . . .	16
4.2.2	Detekce plátna . . . . .	18
4.2.3	Redukce podobných přímek . . . . .	19
<b>5</b>	<b>Aplikace KeyCorr</b>	<b>20</b>
5.1	Návrh . . . . .	20
5.2	Implementace . . . . .	20
5.2.1	Obrazový výstup projektoru . . . . .	21
5.2.2	Detektor oblastí ve scéně . . . . .	21
5.2.3	Grafické transformace . . . . .	21
5.2.4	Zachytávání obrazu . . . . .	21
5.3	Výsledky . . . . .	22
5.4	Možnosti rozšíření . . . . .	22
<b>6</b>	<b>Závěr</b>	<b>23</b>
<b>A</b>	<b>Fotografie s ukázkou funkce aplikace</b>	<b>25</b>



# Kapitola 1

## Úvod

Při prezentacích se přednášející téměř neobejde bez projekčního systému. Ten posluchačům umožňuje lépe pochopit jádro problému. Možnost zobrazovat na velkém plátně diagramy, obrázky, videa aj. je velice výhodná. V dnešní době se nejčastěji používá kombinace počítače a projektoru. Kvalita takové projekce závisí na mnoha faktorech, jedním z nich je pozice projektoru vzhledem k plátnu.

V případě, že je projektor v místnosti napevno, volba pozice a nastavení se provede při instalaci a není třeba se tímto problémem dále zabývat. Při použití přenosného projektoru ale může nastat situace, při které není možné umístit jej na ideální pozici (omezený prostor v místnosti, projektor brání divákům ve výhledu), pak je obraz na plátně zkreslený. Tento jev se nazývá lichoběžníkové zkreslení projektoru (keystoning effect) a je popsán v kapitole 2. Jsou zde analyzovány různé druhy lichoběžníkového zkreslení a také situace, kdy k němu nastává. Následuje popis metod, které dokážou zkreslení eliminovat. Na konci kapitoly se nachází schéma systému korekce zkreslení, kterým se zabývá tato práce.

Tento systém korekce používá digitální kameru<sup>1</sup> připojenou k systému počítač-projektor, tak může být prozkoumána projekční scéna a následně v ní nalezeno plátno a oblast projekce. K tomu jsou použity algoritmy detekce význačných bodů v obraze. Ty jsou popsány v kapitole 4. Následuje postup detekce v systému. Ten se nachází v sekci 4.2.

Na základě detekce na výstupu kamery je obraz vstupující do projektoru transformován tak, aby na plátně měl obdélníkový tvar. Tyto transformace jsou nalezeny pomocí stanovení homografie mezi jednotlivými projektivními rovinami v systému. Tato problematika je vysvětlena v kapitole 3. Vztahy mezi rovinami v systému kamera-projektor se zabývá sekce 3.3.

Cílem práce bylo vytvořit aplikaci, která pomocí webové kamery a minimálního vstupu od uživatele lichoběžníkové zkreslení opraví. Tuto aplikaci jsem pojmenoval KeyCorr a její popis se nachází v kapitole 5. Tato kapitola obsahuje návrh aplikace, popis její implementace včetně problémů, které byly nutné řešit, dále zhodnocení funkčnosti programu. Na konci této kapitoly se nachází informace o možnostech rozšíření aplikace.

---

<sup>1</sup>Například webkamery

## Kapitola 2

# Lichoběžníkové zkreslení

Lichoběžníkové zkreslení vzniká při projekci obrazu na plátno pod určitým úhlem. Aby vznikl přesně obdélníkový obraz, musí být optická osa projektoru kolmá na střed projekční plochy (obr. 2.1(a)). Je-li vychýlena pouze v jednom směru, promítaný obraz má tvar lichoběžníku, nastává tak vertikální (obr. 2.1(b)) nebo horizontální (obr. 2.1(c)) lichoběžníkové zkreslení. Obraz dopadající na plátno se rozšiřuje směrem k okraji vzdálenějšímu od projektoru. Při kombinaci vertikálního i horizontálního posuvu má obraz tvar obecného čtyřúhelníku (obr. 2.1(d)). Zkreslení je pro diváka nepříjemné a nepřírozené, může změnit význam grafů a technických nákrešů.

### 2.1 Oprava zkreslení

Korekce může být provedena digitální úpravou obrazu nebo pomocí změny polohy optických částí projektoru (tzv. optická korekce). Oprava se provádí manuálně nebo automaticky. Běžné projektory nabízí manuální digitální korekci. Většina zařízení umožňuje korekci vertikálního zkreslení v rozsahu  $\pm 12^\circ$  naklonění [7], sofistikovanější modely nabízí rozsah až  $\pm 35^\circ$  a také korekci horizontálního zkreslení.

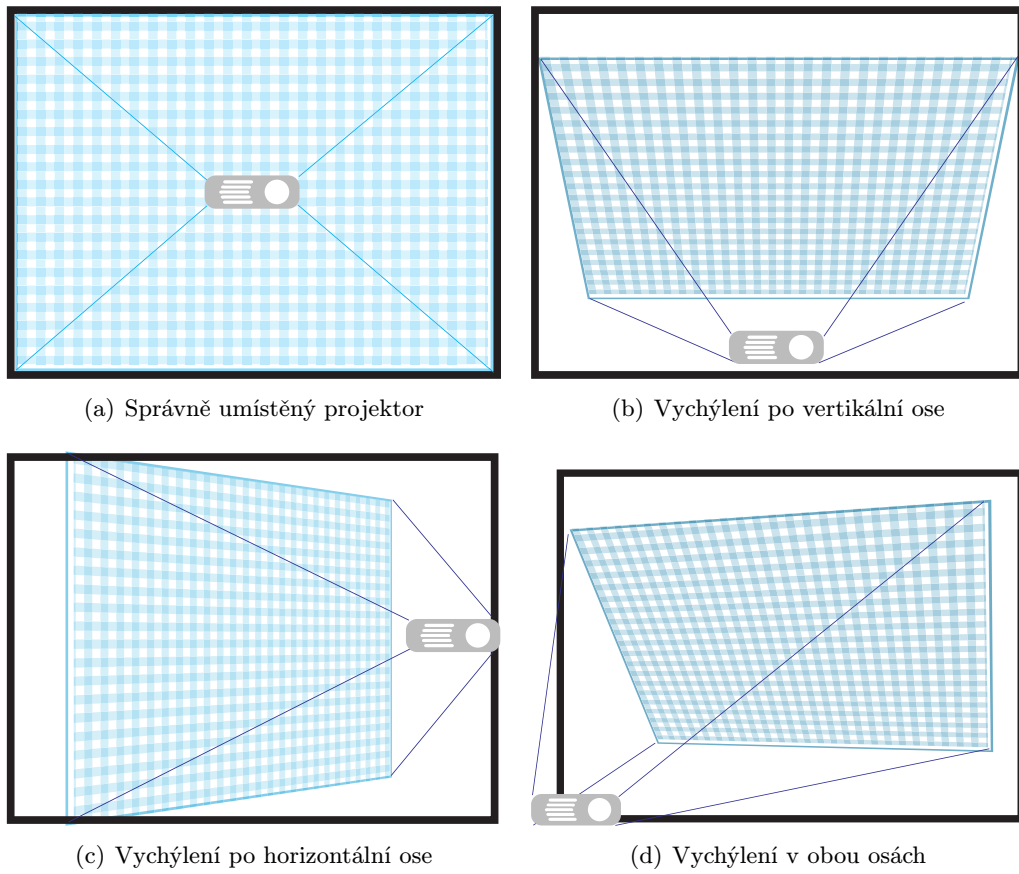
Při manuální opravě uživatel nastaví korekci pomocí ovládacích prvků projektoru. Stupeň korekce volí dle vlastního uvážení, tak může nastat situace, že uživatel korekci nenastaví správně a obraz zůstane deformován.

Automatické opravy zkreslení nabízí projektory se zabudovaným senzorem sklonu. Ten určí pod jakým úhlem zařízení promítá a údaje předá k úpravě výstupního obrazu. Tato zařízení však nejsou schopna detekovat horizontální natočení projektoru. Korekce také nefunguje správně v situaci, kdy plátno není ve svislé poloze. Byl vyvinut systém automatické detekce pozice projektoru vůči plátneu, kde jsou v plátneu i projektoru instalovány senzory náklonu a digitální kompasu. Díky nim je možné určit jejich vzájemnou polohu. [10] Tento systém se však v praxi nevyužívá.

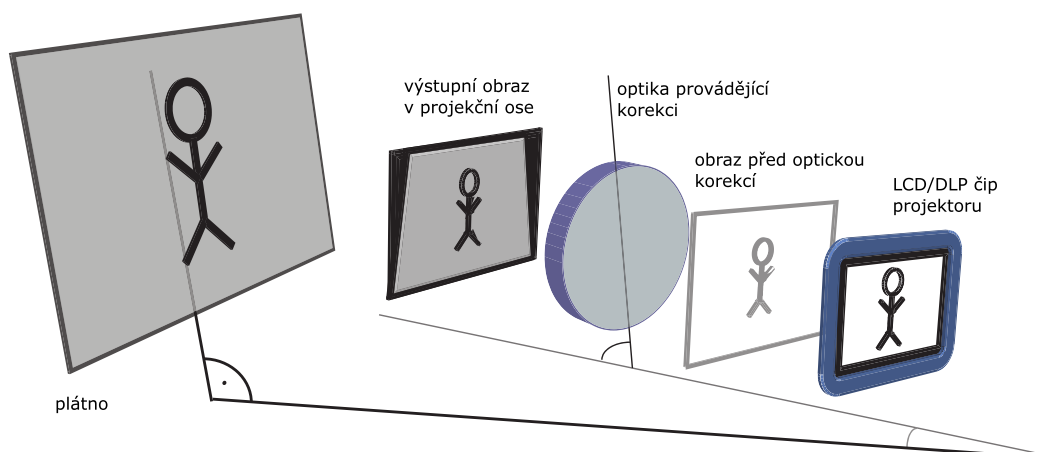
#### 2.1.1 Optická korekce

Optická korekce lichoběžníkového zkreslení není v současných projektorech příliš obvyklá. Existují ale adaptéry obsahující optický systém, které po nasazení na objektiv projektoru zkreslení opraví (obr. 2.2). Cena takového zařízení se ale pohybuje kolem \$ 1800. Což je vzhledem k ceně projektorů zbytečná investice. Výhodou této metody je menší ztráta kvality obrazu než u digitální korekce. Nevýhodou je její cena.





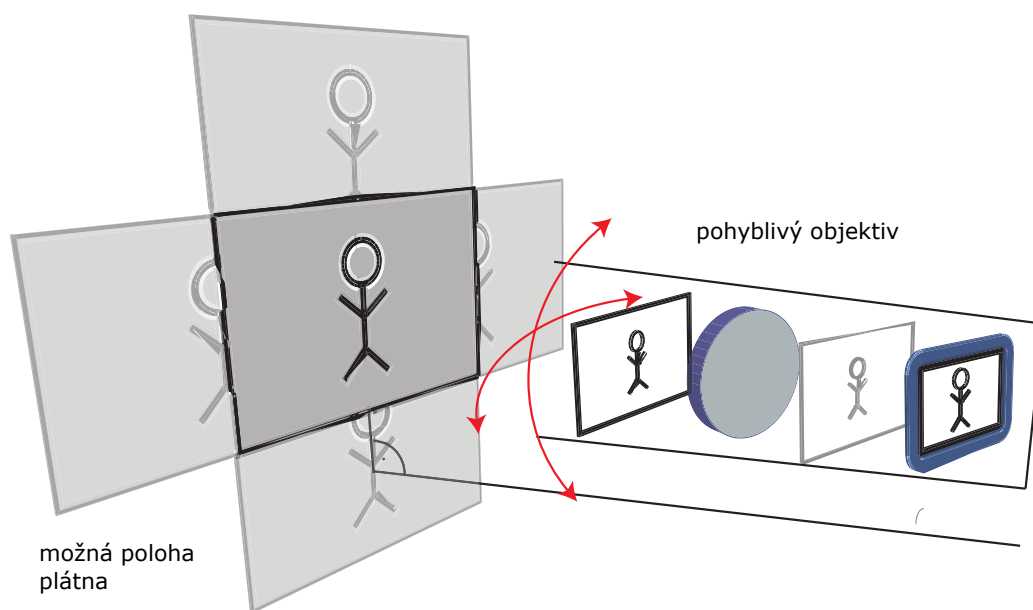
Obrázek 2.1: Ukázky zkreslení projektoru



Obrázek 2.2: Schéma optické korekce

### 2.1.2 Variable Optical Lens Shift

Zajímavou možností jak vůbec předejít lichoběžníkovému zkreslení je funkce *Variable Optical Lens Shift* (dále jen Lens Shift) (obr. 2.3), kterou mají sofistikovanější projektor. V nich není objektiv napevno, ale je možné ho oboustranně vychýlit. Díky tomu jde do určité úrovně posunout promítaný obraz horizontálně i vertikálně. Promítací osa projektoru musí být stále kolmá k rovině plátna, nemusí však být v jeho středu. Rozsah funkce Lens shift je na všech zařízeních ve vertikálním směru větší, než v horizontálním.



Obrázek 2.3: Schéma funkce Lens shift

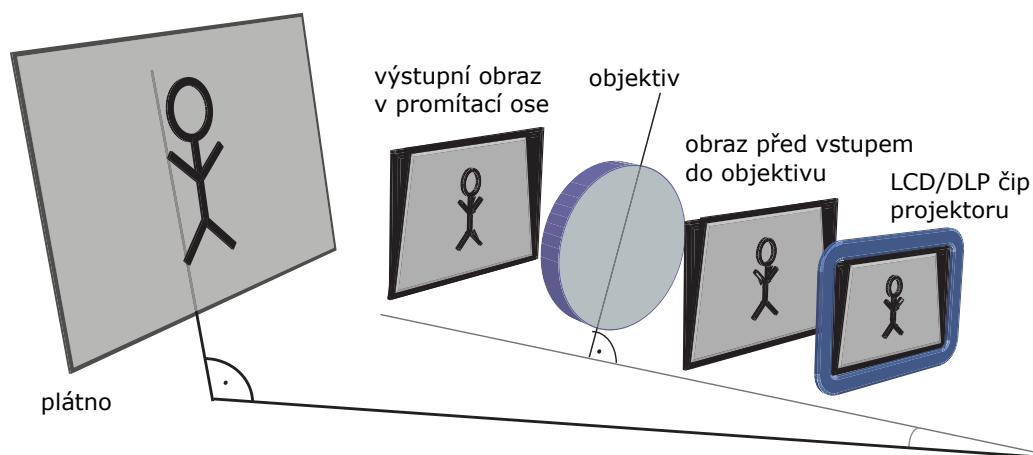
Výhodou této funkce je zachování kvality promítaného obrazu a také mnohem lepší cenová dostupnost než u optické korekce lichoběžníkového zkreslení. Nevýhodou je vyšší cena než u digitální korekce a také omezený rozsah posunu obrazu. Průměrný projektor s funkcí Lens shift je schopen posunout obraz o celou výšku projekce ve vertikálním směru a o polovinu šířky projekce ve směru horizontálním.

### 2.1.3 Digitální korekce

Digitální korekce zkreslení se na rozdíl od optické provádí ještě před průchodem optickým systémem projektoru. Obraz je na LCD nebo DLP panelu upraven tak, aby měl na plátně obdélníkový tvar (obr. 2.4). Tímto způsobem je možno eliminovat horizontální i vertikální zkreslení, ne všechna zařízení však nabízí eliminaci horizontálního zkreslení.

Úprava obrazu funguje na principu změny šířky jednotlivých řádků. Na zobrazovacím čipu projektoru má obraz tvar přesně opačný než je tvar obrazu na plátně u nekalibrovaného projektoru. To znamená, že když se neopravený obraz na plátně rozšiřuje směrem nahoru, na zobrazovacím čipu se obraz musí směrem nahoru zužovat.

Největší výhodou tohoto způsobu korekce je dostupnost, podporuje ho každý současný projektor. Jako u každé digitální úpravy však dochází k degradaci obrazu. Při digitální korekci jsou na zobrazovacím čipu oblasti, které nejsou využité, mohou také vzniknout obrazové artefakty a zubaté okraje. Obraz je na okrajích mírně rozmazaný, protože projektor



Obrázek 2.4: Schéma digitální korekce

je možné zaostřit pouze na určitou vzdálenost a při vzniku lichoběžníkového zkreslení se na různé části plátna promítá z různé vzdálenosti.

## 2.2 Korekce lichoběžníkového zkreslení v systému kamera-projektor

Systém korekce zkreslení, kterým se tato práce zabývá, lze zařadit do kategorie automatické digitální korekce a byl inspirován [7] a [8]. Korekce nebude probíhat v projektoru ale ještě před výstupem z počítače. Systém je složen z následujících prvků:

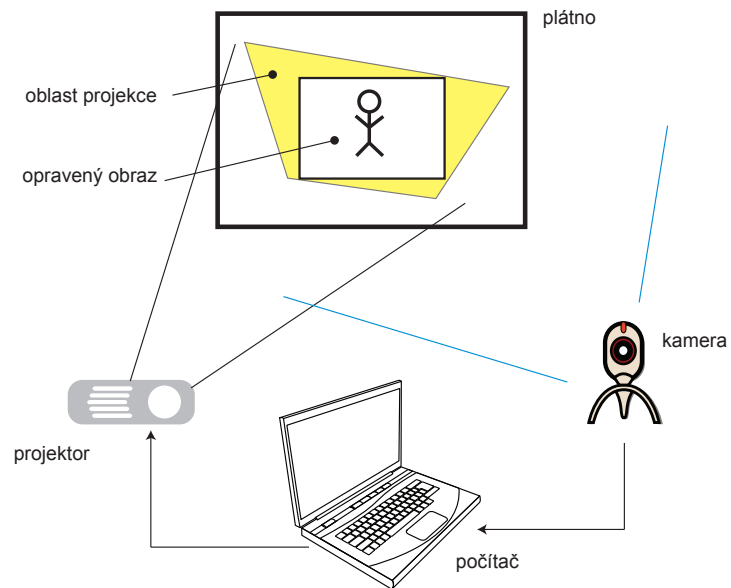
- osobní počítač nebo notebook
- webová kamera (např. Logitech HD webcam)
- standardní LCD projektor

Projektor se může nacházet kdekoliv v místnosti a musí být namířen na projekční plátno. Pozice kamery je nezávislá na projektoru, je nutné aby se v zorném poli kamery nacházela oblast projekce. Projekční plátno musí být rovné a ohraničené okrajem s jinou barvou, než je barva plátna.

Tato metoda by se dala shrnout do následujících kroků:

1. Nalezení oblasti projekce a vztahu mezi body na monitoru počítače (resp. projektoru) a odpovídajícími body v obrazu kamery.
2. Nalezení obdélníkové oblasti v obrazu kamery, kde by se měl obraz objevit.
3. Pomocí vztahu z 1 promítnout rohy této obdélníkové oblasti do souřadnic projektoru.
4. Vytvořit projektivní transformaci, která zdeformuje obraz projektoru aby měl na plátně tvar obdélníku.

Metody analýzy obrazu z kamery jsou blíže popsány v kapitole 4. Nalezení cílové oblasti pro promítání probíhá v několika krocích. Pokud projektor promítá přes celé plátno, tak



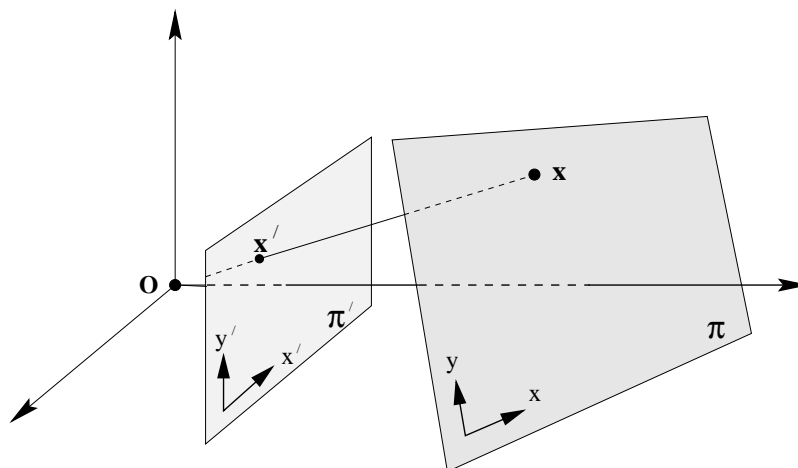
Obrázek 2.5: Schéma systému

je cílová oblast totožná s oblastí promítacího plátna. V opačném případě je hledán v oblasti projekce co největší obdélník stejného tvaru a orientace, jako má promítací plátno. K nalezení jeho souřadnic v oblasti kamery nám slouží stanovení homografie mezi obrazovou rovinou kamery a plátnem. Výpočet homografie je popsán v kapitole 3.

## Kapitola 3

# Homografie a její využití při kalibraci systému

Homografie je invertibilní zobrazení  $h$  mezi dvěma projektivními rovinami tak, že tři body  $x_1, x_2$  a  $x_3$  leží v jedné přímce pouze tehdy, když  $h(x_1), h(x_2)$  a  $h(x_3)$  také leží v jedné přímce. Dalšími názvy pro homografii, se kterými je možné se v literatuře setkat, jsou kolinearita a projektivní transformace.[5]



Obrázek 3.1: Ukázka homografie

Příkladem homografie je středové promítání (obr. 3.1<sup>1</sup>) se středem v bodě  $O$  z roviny  $\pi$  do roviny  $\pi'$ . Pokud mají obě roviny definován souřadnicový systém, střed  $O$  definuje zobrazení každého bodu  $x$  z roviny  $\pi$  do bodu  $x'$  z roviny  $\pi'$ . Z toho je zřejmé, že přímky jedné plochy jsou mapovány na přímky druhé plochy.[5]

Pokud jsou body  $x$  a  $x'$  vyjádřeny v homogenních souřadnicích, můžeme zapsat homografii jako:  $x' = Hx$ , kde  $H$  je matice homografie o velikosti  $3 \times 3$ . Tato matice je regulární.<sup>2</sup>

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$$

<sup>1</sup>Obrázek převzat z [5]

<sup>2</sup>Regulární (invertibilní) matice je taková čtvercová matice, jejíž determinant je různý od 0.

### 3.1 Homogenní souřadnice

Homogenní souřadnice jsou způsob vyjádření polohy bodu v projektivní geometrii. Jejich výhodou je možnost reprezentovat body v nekonečnu. Projektivní transformaci homogenních souřadnic lze provádět násobením s transformační maticí, což u kartézských souřadnic není možné.

Bod v 2D kartézském souřadnicovém systému  $[x, y]$  lze vyjádřit homogenními souřadnicemi  $(x, y, 1)^T$ , nebo ještě obecněji:  $(w.x, w.y, w)^T, w \neq 0$ . Při násobení homogenních souřadnic jakýmkoli nenulovým číslem se nezmění bod který reprezentují v kartézském systému, z toho vyplývá, že jeden bod může být v homogenních souřadnicích vyjádřen nekonečně mnoha způsoby. Převod z homogenních do kartézských souřadnic se provádí opačným způsobem:  $(X, Y, Z)^T \Rightarrow [X/Z, Y/Z]$ .

### 3.2 Výpočet homografie

K výpočtu matice homografie  $H$  se používá algoritmus DLT (Direct Linear Transform). Pro nalezení matice homografie je nutno znát souřadnice dostatečného počtu korespondujících bodů  $x$  a  $x'$ .

Pokud pracujeme s homogenními souřadnicemi bodu 3.1, vztah mezi dvěma korespondujícími body  $x$  a  $x'$  může být napsán jako:

$$w \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (3.1)$$

kde  $w$  je libovolná nenulová konstanta,  $(u \ v \ 1)^T$  odpovídá  $x'$ ,  $(x \ y \ 1)^T$  odpovídá  $x$  a  $H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$ .

Dělením prvního řádku rovnice 3.1 třetím řádkem a druhého řádku rovnice třetím řádkem dostaneme následující dvě rovnice:

$$-h_1x - h_2y - h_3 + (h_7x + h_8y + h_9)u = 0 \quad (3.2)$$

$$-h_4x - h_5y - h_6 + (h_7x + h_8y + h_9)v = 0 \quad (3.3)$$

Rovnice 3.2 a 3.3 mohou být zapsány ve formě matice:

$$A_i h = 0$$

$$\text{kde } A_i = \begin{pmatrix} -x & -y & -1 & 0 & 0 & 0 & ux & uy & u \\ 0 & 0 & 0 & -x & -y & -1 & vx & vy & v \end{pmatrix}$$

$$\text{a } h = (h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7 \ h_8 \ h_9)^T$$

Vytvořením rovnic pro 4 korespondující body vzniká soustava 8 rovnic o 9 neznámých 3.4. Podmínkou pro správný výpočet homogenní matice je, že žádné 3 korespondující body nesmí být v přímce. Pak soustava obsahuje 8 lineárně nezávislých řádků a je možné nalézt řešení.

$$Ah = \begin{pmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & u_1x_1 & u_1y_1 & u_1 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & v_1x_1 & v_1y_1 & v_1 \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & u_2x_2 & u_2y_2 & u_2 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & v_2x_2 & v_2y_2 & v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_n & -y_n & -1 & 0 & 0 & 0 & u_nx_n & u_ny_n & u_n \\ 0 & 0 & 0 & -x_n & -y_n & -1 & v_nx_n & v_ny_n & v_n \end{pmatrix} = 0 \quad (3.4)$$

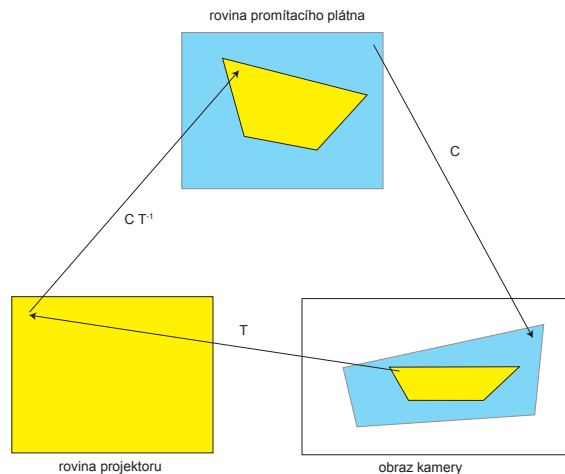
I když použijeme více než 4 korespondence, matice A bude mít stále hodnotu 8 a bude mít stále stejné řešení. Tento postup byl převzat z [4], kapitola 2.1.

### 3.3 Homografie v systému kamera-projektor

V našem systému pracujeme s několika projektivními rovinami:

- rovina projektoru
- rovina kamery
- rovina promítacího plátna

Nalezení vztahu mezi nimi je pro úspěšné provedení korekce velmi důležité. Poloha, orientace a optické vlastnosti kamery ani projektoru nejsou známy. Optika projektoru a kamery může být modelována pomocí projektivních transformací [8]. Transformace jsou vypočteny pomocí homografie mezi těmito rovinami. Na obrázku 3.2 jsou zobrazeny roviny v systému a vztahy mezi nimi.



Obrázek 3.2: Vztahy mezi rovinami v systému

Po nalezení souřadnic vrcholů čtyřúhelníkové oblasti projekce v obrazu kamery jsme schopni vypočítat transformační matici  $T$  mezi kamerou a projektorem. Odpovídající body v rovině projektoru jsou souřadnice krajních bodů promítaného obrazu. Díky této transformaci je možno zobrazit ve zdrojovém obrazu body, jejichž souřadnice známe v rovině

kamery. Zobrazení  $x \rightarrow x'$  se provede násobením homogenních souřadnic bodu v obrazu kamery s invertovanou transformační maticí  $T^{-1}$ :

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = T^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Analogicky nalezneme transformaci  $C$  z roviny plátna do souřadnic v obrazu kamery. Jelikož plátno je hmotná rovina a nemá pevně daný souřadnicový systém, zvolíme stejný, jako má projektor.

Pokud projektor nepromítá přes celé plátno, je nutné vypočítat cílovou oblast projekce na promítacím plátně. Souřadnice oblasti promítání projektoru jsou vynásobeny s maticí  $C^{-1}$ , tak jsou převedeny do roviny promítacího plátna. Následně nalezneme obdélník ohraničený touto oblastí, který má stejný poměr stran jako promítací plátno. Souřadnice nalezené cílové oblasti převedeme do souřadnic v rovině projektoru tak, že je vynásobíme součinem matic  $CT^{-1}$ .

Nyní systém disponuje dvěma sadami souřadnic v rovině projektoru. První sada čtyř bodů ohraničuje promítací oblast projektoru. Jsou to 4 krajní body jejichž souřadnice jsou dány rozlišením projektoru. Druhá sada čtyř bodů ohraničuje oblast, kam by měl být výsledný obraz promítán, aby bylo opraveno lichoběžníkové zkreslení. Pokud projektor svítí přes celou oblast plátna, jde o souřadnice vrcholů obdélníkového plátna. V opačném případě jsou to souřadnice obdélníku, který má stejný tvar a orientaci jako plátno, ale "vejde se" do promítací oblasti projektoru. Při výpočtu homografie mezi těmito rovinami dostaneme matici, kterou aplikujeme na každý obrázek vstupující do projektoru. Tak bude obraz na plátně obdélníkový a lichoběžníkové zkreslení bude opraveno.



## Kapitola 4

# Detekce význačných bodů v obraze kamery

Pro správné nalezení grafických transformací v předchozí kapitole je nutné detekovat v obraze kamery několik význačných bodů. Jsou to vrcholy čtyřúhelníkové oblasti projekce a vrcholy promítacího plátna. Řešení tohoto problému není triviální a skládá se z několika částí. Před začátkem detekce se promítne černá a následně bílá plocha na projektoru, kamera zaznamená obě tyto situace. Následně se tyto obrazy mezi sebou porovnávají a aplikují se na ně různé detekční algoritmy. Postup je popsán v sekci 4.2.

### 4.1 Detekce hran v obraze

První metoda která je použita při hledání bodů v obraze je detekce hran. Hrana je oblast, kde se v obraze významně mění hodnota bodů (u černobílého obrázku jas), jsou to místa s vysokou frekvencí obrazu. Například černá čára na bílém pozadí vytvoří dvě hrany. Jednu na přechodu z pozadí (bílé) na čáru (černou) a druhou na přechodu z čáry na pozadí. Na detekci hran se často používají lineární filtry, které fungují na principu konvoluce s určitým jádrem. V této práci byl použit Cannyho hranový detektor, který využívá Sobelův operátor.

#### 4.1.1 Konvoluce

Konvoluce je základem mnoha grafických transformací, je aplikována na každý bod obrázku. To, co konvoluce s obrazem dělá, je dáno *Konvolučním jádrem* které je použito. Příklad konvolučního jádra je matice 4.1. Jádro je matice pevných rozměrů obsahující numerické koeficienty a kotevní bod. Na pozici cílového (kotevního) bodu se dosadí výsledek konvoluce.

Výpočet probíhá "přiložením" kotvy konvolučního jádra na místo výsledného bodu. Zbytek konvolučního jádra překrývá okolní pixely. Výsledek dostaneme vynásobením koeficientů

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & \mathbf{1/9} & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad (4.1)$$

Obrázek 4.1: Příklad konvoluční matice  $3 \times 3$  pro rozmazání obrazu, tučně je zobrazen kotevní bod

jádra s odpovídajícími hodnotami bodů, které jádro překrývá, a sečtením těchto násobků. Tento proces se opakuje pro každý bod ve zdrojovém obrázku.

Tento postup je možno vyjádřit rovnicí. Vstupní obraz je definován jako  $I(x, y)$ , konvoluční matice je  $G(i, j)$ , (kde  $0 < i < M_i - 1$  a  $0 < j < M_j - 1$  a  $M_i, M_j$  je počet řádků, resp. sloupců konvoluční matice), kotva odpovídá souřadnicím  $(a_i, a_j)$  v konvolučním jádru a výsledek konvoluce odpovídá  $H(x, y)$ . Rovnice potom vypadá takto:

$$H(x, y) = \sum_{i=0}^{M_i-1} \sum_{j=0}^{M_j-1} I(x+i-a_i, y+j-a_j)G(i, j) \quad (4.2)$$

Jediným problémem při výpočtu konvoluce jsou okraje zdrojového souboru. Při výpočtu konvoluce okrajových bodů je jádro určitou částí nad neexistujícími body. Tento problém se řeší rozšířením zdrojového obrázku o potřebný počet bodů. Jejich hodnota je pak stejná jako hodnota krajních bodů obrázku. Druhou možností je nastavit hodnotu rozšířených bodů na nějakou konstantu. [3]

#### 4.1.2 Sobelův operátor

Sobelův operátor patří mezi často používané lineární filtry určené k detekci hran. Operátor používá dvě  $3 \times 3$  konvoluční jádra, které spočítají odhad změn intenzity obrazu (parciální derivaci), jedno v horizontálním směru, druhé ve vertikálním:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Pokud na obrázek nahlédneme jako na dvourozměrnou funkci intenzity obrazu, Sobelův filtr umožňuje vypočítat gradient této funkce. Ten je definován jako dvourozměrný vektor vytvořený z prvních derivací funkce ve dvou kolmých směrech.

$$grad(I) = \left[ \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]^T$$

Po aplikaci filtru vznikne matice vektorů (hodnot gradientu) se stejnou velikostí jako filtrovaný obraz. Jelikož je gradient vektor, má směr a délku. Délku je možno vypočítat pomocí Euklidovské normy (také nazývána L2 norma):

$$|grad(I)| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

V počítačovém zpracování obrazu se však používá spíše součet absolutních hodnot těchto derivací, tzv. L1 norma. Její hodnoty jsou totiž velice podobné jako L2, ale rychlost výpočtu je mnohem vyšší. [6]

Pro detekci hran se nejčastěji používá pouze velikost gradientu. Při vykreslení těchto hodnot dostaneme obraz o velikosti zdrojového obrázku, kde pixely s vyššími hodnotami odpovídají významnějším hranám a pixely s nižšími hodnotami odpovídají hranám méně významným. Pro zobrazení významnějších hran je nutno výsledek prahovat.

### 4.1.3 Cannyho hranový detektor

Použití hranového detektoru, jako je například Sobelův operátor, má jisté nevýhody. První nevýhodou je fakt, že hrany na výsledném obrazu jsou tlusté, jinými slovy zasahují přes několik pixelů. Nejvýhodnější by bylo mít na místě s největším gradientem binární informaci o tom, že zde se hrana nachází a v na okolních pixelech informaci opačnou. Další nevýhoda je obtížnost nalezení ideálního prahu. Pokud bude nízký, ve výsledném obraze se budou nacházet i pro detekci nevýznamné hrany. Pokud bude moc vysoký, nemusí být zobrazeny důležité hrany. Cannyho hranový detektor částečně eliminuje tyto nevýhody.

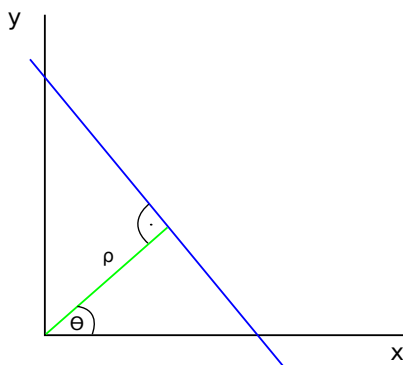
Cannyho detektor je založen na Sobelově operátoru a používá dva stupně prahování jeho výstupu, dolní a horní práh. Dolní práh je zvolen tak, aby byly nalezeny všechny hrany důležité pro detekci. Samozřejmě jsou nalezeny i hrany, které potřebné nejsou. Pro jejich eliminaci je stanoven horní práh. Ten je nastaven tak, aby byly zobrazeny body náležící všem důležitým hranám. U horního prahu není nutné nalézt celé hrany, dostačující je nalezení části všech významných hran.

Algoritmus kombinuje tyto dvě hranové mapy. Všechny nalezené hrany v horním prahu jsou zachovány. Hrany z dolního prahu jsou do výsledku přidány pouze v tom případě, že navazují na hrany z horního prahu (mají stejný směr gradientu). Hrany z nízkého prahu, které nenavazují na hrany z vysokého nejsou zachovány. Díky tomuto způsobu vyhodnocení je možno získat co nejlepší obrysy významných objektů v obraze. I v případě, že je v původním obraze intenzita hrany nestálá, je hrana detekována.

Hrany jsou na výstupu z Cannyho detektoru zobrazeny jen na místě s jejich největší hodnotou gradientu. Každá hrana nalezená Cannyho detektorem má tedy šířku pouze 1 pixel, což je žádaná vlastnost.

### 4.1.4 Houghova transformace

Po detekci pomocí Cannyho detektoru získáme binární obraz na kterém jsou zobrazeny významné hrany obrazu. Tyto body ale nejčastěji tvoří linie. Body které náležejí hranám nejsou do linií sjednoceny, jinak řečeno nelze zatím určit, které hranové body společně tvoří jednu přímku. Houghova transformace dokáže v binárním obraze detekovat všechny linie, které prochází určitým počtem bodů. [6]



Obrázek 4.2: Vyjádření přímky v polárních souřadnicích

V Houghově transformaci je přímka definována pomocí bodu v polárních souřadnicích

$(\rho, \theta)$ , kde  $\rho$  je vzdálenost bodu od počátku souřadnicového systému, a  $\theta$  je úhel, který svírá spojnice bodu s počátkem souřadnicového systému a přímkou s úhlem 0 (obr. 4.2). Tato spojnice je pak normálou dané přímky a dá se vyjádřit následující rovnicí:

$$\rho = x \cos \theta + y \sin \theta \quad \theta \in (0, \pi), \rho > 0$$

Pro každý zvýrazněný pixel v obraze s detekovanými hranami jsou nalezeny všechny možné přímky, které přes něj mohou procházet. Jejich souřadnice se vykreslí do dvou-rozměrného akumulátoru (osy  $\theta$  a  $\rho$ ), vykreslená křivka má tvar sinusoidy. Pro každý další zvýrazněný pixel se postup opakuje. Ukázka takové ho akumulátoru je na obr. 4.3. V případě že několik bodů patří do jedné přímky, vznikne průsečík těchto sinusoid na souřadnicích  $(\rho, \theta)$ . Ty jsou vyjádřením této přímky. Při vyhodnocení Houghovy transformace se volí počet bodů, kterými musí přímka procházet aby byla uznána za přímku. Ty které tento počet překročí jsou výsledkem transformace. [1]



(a) Vyjádření všech úseček procházejících jedním bodem

(b) Tři body které leží na společné přímce

Obrázek 4.3: Ukázky akumulátoru

Tato metoda transformace hran na přímky často na jedné hraně vytvoří větší počet přímek. Eliminace jejich počtu je vysvětlena v sekci 4.2.3.

## 4.2 Detekce oblastí ve scéně

Jak už bylo řečeno v úvodu kapitoly, k detekci jsou použity dva obrazy z kamery, jeden s bílým obrazem na projektoru a druhý s černým. Oba obrazy jsou převedeny do odstínů šedi.

### 4.2.1 Detekce oblasti projekce

Nejprve probíhá detekce oblasti projekce. Nejideálnější situace z pohledu detekce by byla získat binární obraz, ve kterém by byly označeny pouze body v místech, kam projektor promítá. Podobného obrazu lze dosáhnout při odečtení hodnot projektorem neosvětlené scény od osvětlené. V ideálním případě by se měla intenzita změnit pouze v oblasti projekce.

## Předpříprava obrazu

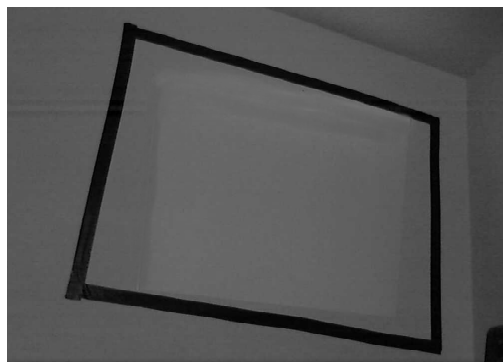
Webkamery disponují funkcí automatického nastavení doby expozice. Ta tento postup mírně komplikuje. Jas obrazu v jednom bodě je proměnlivý v závislosti na intenzitě ostatních bodů. Když porovnáme jas určitého bodu, který není v místě projekce, na projektorem osvětlené a neosvětlené scéně, zjistíme, že na neosvětlené scéně je jas vyšší. Kamera se snaží v obraze zajistit co největší kontrast.<sup>1</sup> Neosvětlená scéna má přibližně stejný jas v celém obraze, naopak osvětlená scéna má v oblasti projekce vysoký jas a neosvětlená místa jsou mnohem tmavější.

Tento problém je vyřešen pomocí výpočtu průměrného rozdílu mezi okrajovými body obou scén. Krajiní body jsou zvoleny z důvodu, že se na těchto místech nevyskytuje oblast projekce. Výsledný rozdíl je odečten od všech bodů neosvětleného obrazu. Neosvětlené hodnoty v obou obrazech by měly mít přibližně stejný jas.

Nyní je možno tyto obrázky od sebe odečíst, tak že od hodnoty bodu na osvětleném obrázku odečteme hodnotu korespondujícího bodu na neosvětleném a uložíme do nového obrázku. Tento výsledek by měl odpovídat binárnímu obrazu, který je potřebný pro detekci. Ukázka na obr. 4.5



(a) Projektorem osvětlená plocha



(b) Projektorem neosvětlená plocha



(c) Rozdíl dvou předchozích obrázků

Obrázek 4.4: Ukázky vstupu z kamery

<sup>1</sup>Rodíl mezi nejsvětlejším a nejtmačejším místem.

## Detekce hran a přímek

Detekce hran se provádí pomocí Cannyho detektoru (viz. 4.1.3). Jeho prahy byly nalezeny empiricky testováním detekce s různými prahy v různých scénách. Výsledkem bude binární obraz, který by měl obsahovat pouze hrany odpovídající okrajům projekční plochy. Pokud projektor promítá přes okraje plátna vzniknou navíc hrany i na těchto místech.

Pro jejich eliminaci je nutno detekovat hrany i na původním neosvětleném obrázku a porovnat výstupy těchto dvou hranových detekcí. Hrany které se nacházejí na výstupu z první detekce a zároveň na výstupu z druhé jsou z prvního výstupu odstraněny. Tak budou odebrány hrany odpovídající okrajům plátna.

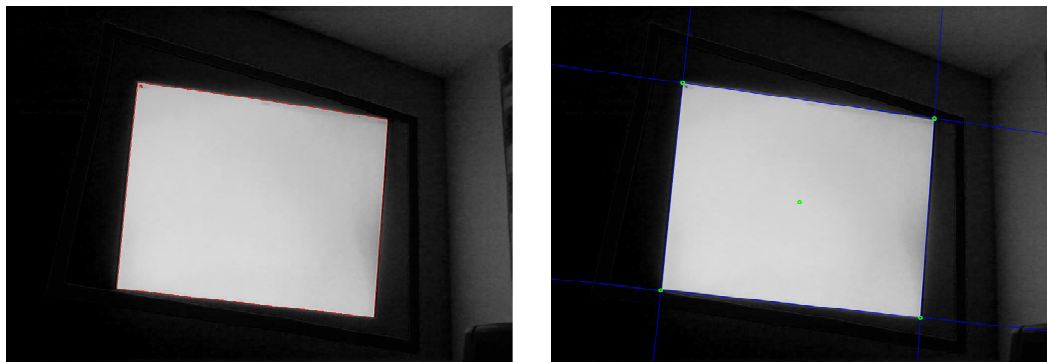
Z důvodu omezeného rozlišení webkamery nemusí být hrany nalezeny dostatečně přesně. Proto jsou převedeny na přímky pomocí Houghovy transformace (viz. 4.1.4). Výstupem jsou analyticky vyjádřené přímky, které leží na nalezených hranách. V závislosti na kroku nastavení parametrů této metody může na jedné hraně vzniknout více přímek, eliminace jejich počtu je vysvětlena v kapitole 4.2.3.

Po eliminaci máme rovnice 4 přímek, které odpovídají okrajům oblasti projekce. Pro získání souřadnic vrcholů je nutno vypočítat průsečíky. Průsečík dvou přímek se dá v polárních souřadnicích vyjádřit soustavou rovnic o dvou neznámých:

$$\rho_1 = x \cos \theta_1 + y \sin \theta_1$$

$$\rho_2 = x \cos \theta_2 + y \sin \theta_2$$

Pokud se výsledný bod nachází v obrazu kamery, jedná se o vrchol hledané oblasti. Hledáme průsečíky všech těchto přímek mezi sebou. Výsledkem jsou souřadnice 4 vrcholů oblasti projekce v souřadnicovém systému kamery.



(a) Výstup cannyho detektoru vložený do souboru s rozdílem intezit

(b) Přímky detekované houghovým algoritmem a nalezené průsečíky

Obrázek 4.5: Ukázky výstupu detekce

### 4.2.2 Detekce plátna

Pro nalezení plátna v obrazu kamery využíváme skutečnost, že má jednotnou barvu (nejčastěji bílou), také se v něm nevyskytují žádné ostré barevné přechody, které by mohly být klasifikovány jako hrany. Tyto skutečnosti umožňují použití algoritmu semínkového vyplňování, který po zasazení *semínka*, vyplní oblast s podobnou hodnotou bodů danou barvou.

Algoritmus funguje tak, že po zvolení semínka prozkoumá body v jeho nejbližším okolí. Pokud je hodnota zkoumaného bodu stejná nebo v určitých mezích podobná, bod je obarven a stává se novým semínkem. Pokud je hodnota zkoumaného bodu jiná, než hodnota semínka, daný bod se nevybarví. Tato situace nastane na přechodu z plátna do jeho okraje, kde je rozdíl mezi bílou a černou velký. Většinou je tento algoritmus implementován rekurzivně.[2]

Jako pozice semínka je zvolen střed oblasti projekce vypočtený z bodů získaných v předchozí podkapitole. Další postup je stejný, jako při hledání oblasti projekce. Na obraz s plátnem vyplněným jednotnou barvou je aplikován Cannyho hranový detektor, následně jsou hrany převedeny na přímky Houghovou transformací a jsou spočítány průsečíky těchto přímek. Ty odpovídají poloze vrcholů promítacího plátna v obraze kamery.

### 4.2.3 Redukce podobných přímek

Houghův transformační algoritmus má jednu nepříjemnou vlastnost. Na místě jedné hrany totiž může vzniknout více přímek. To se dá eliminovat správným nastavením jeho parametrů, ale nastávají situace, kdy není možné se tomuto jevu vyhnout. Je potřeba zjistit, které přímky jsou si podobné (polohou a směrem) a určit kterou z nich použít.

Nalezení přímek, které náležejí jedné hraně probíhá porovnáním jejich souřadnic. Pokud jsou jejich souřadnice v určitém rozsahu podobné ( $\rho$  a  $\theta$  mají přibližně stejnou hodnotu), jsou prohlášeny za přímky patřící k jedné hraně. Tak vzniknou množiny přímek, které patří ke stejným hranám. Nyní se přímky ohodnotí, aby bylo možné rozhodnout, která bude použita.

Každá přímka je ohodnocena následujícím způsobem:

1. Přímka je vykreslena do obrázku o rozměrech vstupního obrazu z kamery.
2. Je proveden logický součin s výstupem Cannyho detektoru. Vznikne obraz kde jsou vykresleny body, které odpovídají společným bodům detekované hrany a přímky.
3. Počet těchto bodů je stanoven jako hodnocení přímky.

Následně se z každé skupiny přímek vybere ta s největším hodnocením. Z každé skupiny tedy vzejde přímka která nejlépe odpovídá vykreslené hraně. Tyto přímky jsou výsledkem redukce.

## Kapitola 5

# Aplikace KeyCorr

### 5.1 Návrh

Na základě požadavků v zadání a vlastních zkušeností s úspěšností detekce ve scéně jsem vytvořil návrh výsledné aplikace. Ta by měla být určena pro běžného uživatele bez větších znalostí zpracování obrazu.

Výsledná aplikace by měla fungovat následujícím způsobem:

1. Po spuštění otevřít obraz webkamery a rozsvítit projektor aby bylo možné tyto prvky nasměrovat.
2. Detekovat v kameře scénu a následně zkalibrovat obraz projektoru. Pokud detekce neproběhne úspěšně, může ji uživatel provést manuálně.
3. Po kalibraci zobrazovat na projektoru stejný výstup jako na displeji počítače transformovaný na základě kalibrace.

Uživatelské rozhraní aplikace bude velice jednoduché, jelikož pro jeho používání nebude potřeba velkého počtu ovládacích prvků. V případě že by detekce byla vždy provedena správně, stačilo by pouze tlačítko pro provedení kalibrace a několik polí pro nastavení vstupních údajů programu.

Detekce a kalibrace bude provedena pouze jednou, jejím výsledkem bude transformační matice, která se bude aplikovat na všechny snímky zobrazené na projektoru.

### 5.2 Implementace

K implementaci aplikace KeyCorr byl zvolen programovací jazyk C++. Důvodem pro jeho výběr byla knihovna OpenCv, která je v tomto jazyce také implementována, a je v programu často využita. Obsahuje funkce pro zpracování obrazu a implementaci algoritmů hojně využívaných v počítačovém vidění. Knihovna umožňuje jednoduše pracovat s obrázky v reálném čase. Uživatelské rozhraní je vytvořeno v prostředí QT.

Strukturu aplikace tvoří dva moduly. První s názvem keyCorr zajišťuje detekci ve scéně, kalibraci kamery a transformaci výstupního obrazu. Druhý s názvem scrCap zajišťuje snímání primární obrazovky počítače. Nad těmito moduly je postaveno uživatelské rozhraní.



### 5.2.1 Obrazový výstup projektoru

Aby bylo možné zobrazit na projektoru transformovaný obraz z primárního displeje počítače, je nutné mít obrazové výstupy rozdělené. Řešením bylo nastavit v operačním systému režim více obrazovek s rozšířenou plochou, tak že projektor je napravo od hlavní obrazovky.

Aplikace na rozšířenou plochu umístí celoobrazovkové okno, na kterém je zobrazen výstup mířící do projektoru. Aby aplikace věděla kam umístit výstupní okno programu, je nutno po spuštění nastavit rozlišení obou obrazovek.

### 5.2.2 Detektor oblastí ve scéně

Detektor je nejrozsáhlejší část celé aplikace. Postup detekce byl popsán v sekci 4.2, v kódu je zapsán analogicky. Největším problémem této části bylo najít postup, jehož výsledky budou co nejméně závislé na změně osvětlení scény a vzhledu okolí plátna. Při nízkém osvětlení se sníží kontrast obrazu a může nastat situace, že v důsledku prahu nastaveného na jiné osvětlení nebyly důležité hrany nalezeny. Postup popsáný v této práci se jevil jako nejméně náchylný k těmto negativním vlivům.

V této části byly použity detekční algoritmy implementované v knihovně OpenCV, jsou to funkce `cv::Canny` a `cv::HoughLines`, které odpovídají algoritmům popsáných v kapitole 4.

### 5.2.3 Grafické transformace

Implementace této části odpovídá postupu v sekci 3.3. Jelikož může mít obrazovka počítače a projektor různé rozlišení, je nutné jej přepočítávat a také hlídat poměr stran obrazu. Pokud nastane změna poměru stran, jsou do cílového obrazu přidány černé pruhy na prázdných místech.

### 5.2.4 Zachytávání obrazu

OpenCV samo o sobě je schopno pouze zobrazovat obrázky a videa. Při prezentacích a projekcích se často otevírají média, která knihovna OpenCV nedokáže zobrazit (Flash, Youtube videa, Java applety, pdf, Powerpoint,...). Pro ideální funkci aplikace bylo tedy potřeba získat obraz plochy a zobrazovat ho také na projektoru. Tak by se tento problém eliminoval.

Řešení jsem našel díky knihovně X11. Ta umožňuje manipulaci se zobrazovacími zařízeními v operačním systému Linux. Její funkce dokážou vytvořit snímky určité části obrazovky v relativně rychlém čase. Ty jsou pak předány k transformaci a zobrazeny na projektoru. Obrázek je ale v knihovně X11 reprezentován jiným způsobem než v knihovně OpenCV a je nutné ho převést. Kód pro převod byl částečně převzat z příkladů použití OpenCV<sup>1</sup>. [9]

Jelikož postup zachytávání obrazu a jeho distribuce do projektoru není triviální a provádí se během něj několik transformací, dochází ke zpoždění obrazu na projektoru vůči obrazu na displeji. Toto zpoždění je ale menší než 1 sekunda a pro účely projekce je zanedbatelné.

---

<sup>1</sup><http://opencv.willowgarage.com/wiki/ximage2opencvimage>

## 5.3 Výsledky

Úspěšnost automatické kalibrace souvisí pouze s úspěšností detekce oblastí v obraze. Ta je závislá na několika faktorech:

- **Osvětlení plátna** - Pokud je plátno nerovnoměrně nasvíceno může být nalezeno špatně. K tomu dochází v případech kdy na plátno svítí bodové světlo, pak světelný kužel na plátně může narušit detekci.
- **Předměty v okolí promítacího plátna** - Různé předměty umístěné na zdi mohou narušit detekci projekční oblasti.
- **Osvětlení projekční scény** - Pokud je scéna moc tmavá, může se stát, že detekce také neproběhne úspěšně.

Systém byl testován na 5 různých místech za různých světelných podmínek. Celkově byl testován na vzorku 17 různých kombinací osvětlení, místa a polohy projektoru a kamery. Ve dvou případech bylo nutné automatickou korekci opakovat, než byla provedena správně. Ve dvou případech proběhla detekce neúspěšně a bylo nutno manuálně v obraze označit oblasti projekce a plátna. Ve zbytku případů byla automatická korekce provedena úspěšně.

Při testování nastala situace, kdy kalibrace nebyla provedena dokonale. Došlo k tomu za podmínek, kde projektor promítal přes oblast celého plátna, ale to bylo umístěno asi 10 cm před zdí. Plátno a oblast projekce tak nebyly v jedné rovině a obraz pak nebyl na plátně přesně umístěn. Výsledný obraz byl sice správně tvarován, ale byl na každé straně o 2 cm menší, než by mohl být. Vzhledem k velikosti plátna se tento problém zdá zanedbatelný.

Projekt byl vyvíjen a testován na počítači s operačním systémem *Arch Linux* s knihovnou *OpenCV* a *QT*. Byla použita kamera *Logitech HD Webcam C270* a projektor *NEC MT-850* s rozlišením 1024x768.

## 5.4 Možnosti rozšíření

Aplikace byla implementována pro operační systém Linux. Modul *scrCap* je pevně vázán na knihovnu X11, která je obsažena pouze v Linuxu. Pro rozšíření podpory na OS Windows by stačilo vytvořit knihovnu obdobnou jako *scrCap* která by byla vázaná na knihovny Windows. Implementaci modulu *keyCorr* by nebylo třeba měnit.

Další směr kterým by se mohl aplikace vydat je využít vypočtené transformační matice mezi obrazem kamery a plátnem. Aplikace by umnožňovala porovnávat obraz vstupující do projektoru a obraz plátna v kameře a analyzovat změny mezi nimi. Tak by bylo například možné nalézt oblast na kterou přednášející svítí laserovým ukazovátkem.

Díky sledování pohybu laserového ukazovátka by se s využitím gest dala prezentace ovládat, popřípadě do ní i kreslit.

## Kapitola 6

# Závěr

Cílem této práce bylo vytvořit aplikaci, která by v systému projektor-počítač-kamera prováděla automatickou korekci lichoběžníkového zkreslení. Při hledání postupu detekce a transformace obrazu, jsem se inspiroval systémy popsanými v [7] a [8]. Detekce zde byla popsána velice zevrubným způsobem a bylo nutné algoritmus dosti upravit a zkonkretizovat. Aplikace je nyní schopna na rozdíl od vzorového řešení kalibrovat i systém, kde je oblast projekce větší než plátno. Jelikož ve zmíněných článcích nebyly zveřejněny žádné výsledky, nejsem schopen porovnat úspěšnost této aplikace s jiným existujícím systémem.

Mezi největší výhody této aplikace patří jednoduchost použití. Projektor a kameru je možno umístit kamkoliv v místnosti nezávisle na sobě, stačí je pouze nasměrovat na projekční plátno. Nastavení počítače, které je nutné pro správnou funkci aplikace, také není náročné.

Je nutné si uvědomit, že jde stále o metodu založenou na digitální korekci obrazu. To znamená že při ní dochází ke zhoršení kvality obrazu a také není využita celá plocha zobrazovacího čipu projektoru. Čím je zkreslení obrazu větší, tím větší pak bude snížení kvality kalibrovaného obrazu. Při natočení projektoru také občas vznikají zubaté okraje a nerovné čáry. V určitých situacích je však uživatel raději schopen oželeť kvalitu obrazu kvůli možnosti umístit projektor téměř kamkoliv.

Úspěšnost provedení automatické korekce je uspokojivá. Pokud při detekci oblastí ve scéně nastane chyba, uživatel může zvolit opakování detekce nebo provést kalibraci manuálně, tak že označí důležité oblasti v obrazu kamery. Tento postup není uživatelsky náročný.

Snímání hlavní obrazovky počítače, transformace tohoto obrazu a zobrazení na projektoru nějakou dobu trvá, takže při přehrávání videí bude obraz mírně zpožděn. Při větším vytížení počítače nemusí být výsledný obraz plynulý, pro promítání prezentací a obrázků je rychlost dostatečná.

Jako možnost rozšíření aplikace se jeví dvě možnosti. První je možnost rozšířit podporu programu i na operační systém Windows vytvořením modulu na snímání obrazu pod Windows. Druhou možností je sledování promítacího plátna pomocí kamery a možnost ovládat prezentace pomocí gest laserovým ukazovátkem. Tím se zabývají v [8].

# Literatura

- [1] Hough Line Transform. online, 2013.  
URL [http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough\\_lines/hough\\_lines.html](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html)
- [2] Miscellaneous Image Transformations. online, 2013.  
URL [http://opencv.willowgarage.com/documentation/cpp/miscellaneous\\_image\\_transformations.html](http://opencv.willowgarage.com/documentation/cpp/miscellaneous_image_transformations.html)
- [3] Bradski, G.; Kaehler, A.: *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 2008, ISBN 9780596554040.  
URL <http://books.google.cz/books?id=seAgiOfu2EIC>
- [4] Dubrofsky, E.: *Homography Estimation*. [diplomová práce], The University of British Columbia, Duben 2009.
- [5] Hartley, R.; Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, druhé vydání, 2004, ISBN 0521540518.
- [6] Laganier, R.: *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing, Limited, 2011, ISBN 9781849513258.  
URL <http://books.google.cz/books?id=OC7jc8zWjlkC>
- [7] Sukthankar, R.; Stockton, R.; Mullin, M.: Automatic Keystone Correction for Camera-assisted Presentation Interfaces. In *Proceedings of International Conference on Multimedia Interfaces*, Říjen 2000.
- [8] Sukthankar, R.; Stockton, R.; Mullin, M.: Smarter presentations: exploiting homography in camera-projector systems. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, ročník 1, 2001, s. 247–253 vol.1, doi:10.1109/ICCV.2001.937525.
- [9] Sunesson, C.: Introduction to the X11 library. online, Duben 1999.  
URL <http://www.codersger.de/mags/cscene/CS8/CS8-04.html>
- [10] Wood, J. S.; Taber, R. L.; Ramachandran, G.: Automatic keystone correction system and method. Patent Application, Únor 2004, uS 2004/0036844 A1.  
URL [http://www.patentlens.net/patentlens/patent/US\\_2004\\_0036844\\_A1/en/](http://www.patentlens.net/patentlens/patent/US_2004_0036844_A1/en/)

## Příloha A

# Fotografie s ukázkou funkce aplikace



Obrázek A.1: Příklad projekce dovnitř plátna. Nahoře se nachází nezkalibrovaný obraz. Vlevo je obraz z webkamery, vpravo z digitálního fotoaparátu. Uprostřed je obraz již kalibrovaný. Dole se nachází porovnání výstupu displeje a projektoru.



Obrázek A.2: Příklad projekce vně plátna. Nahoře se nachází nezkalibrovaný obraz. Vlevo je obraz z webkamery, vpravo z digitálního fotoaparátu. Uprostřed je obraz již kalibrován. Dole se nachází porovnání výstupu displeje a projektoru.

## Příloha B

### Obsah CD

- **Zdrojové soubory** -složka KeyCorr, uvnitř je uložen instalační skript, *readme* soubor a zdrojové soubory aplikace KeyCorr
- **Videa** -složka videa, obsahuje 2 videa s ukázkou automatické korekce
- **Text** -složka text, obsahuje tuto práci ve formátu PDF