



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

ROZŠÍŘENÉ UŽIVATELSKÉ ROZHRAŇÍ

AUGMENTED USER INTERFACE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VEDOUCÍ PRÁCE
SUPERVISOR

ONDŘEJ VLADÍK

PAVEL NAJMAN, Ing.

BRNO 2016

Abstrakt

Cílem této práce je navrhnout a implementovat Rozšířené uživatelské rozhraní, které bude využívat systému kamera-projektor. Součástí je i knihovna funkcí pro poloautomatickou kalibraci kamery a celého systému. Bylo navrženo rozhraní pro ovládání widgetů pomocí značek. Na pracovní ploše (stůl, případně stěna nebo tabule) jsou rozmístěny různé značky. Kamera tyto značky snímá ve videosekvenci. Ze snímků se určí pozice značky v prostoru. Widgety jsou promítány na tuto pracovní plochu ke značkám. Se značkami může uživatel libovolně hýbat a widgety se budou hýbat se značkami.

Abstract

The purpose of this thesis is to design and implement an augmented user interface for widget management which uses a camera-projector system. Implement also includes a library of functions for semi-automatic calibration of this system. An interface to control the widgets using markers was designed. Various markers are arranged on the desktop (desk, wall or table) . The camera captures these markers in a camera image. The position of the markers in space is then determined from the camera image . Widgets are projected on the desktop next to the markers. User can freely move those markers and the widgets will follow the markers.

Klíčová slova

kamera a projektor, kalibrace, značky, ARToolkit, Compiz, rozšířené uživatelské rozhraní

Keywords

camera and projector, calibration, markers, ARToolkit, Compiz, Augmented user interface

Citace

VLADÍK, Ondřej. *Rozšířené uživatelské rozhraní*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Najman Pavel.

Rozšířené uživatelské rozhraní

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Pavla Najmana. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Ondřej Vladík
17. května 2016

Poděkování

Chtěl bych poděkovat vedoucímu mé práce Ing. Pavlu Najmanovi za ochotnou odbornou pomoc a užitečné rady k řešení projektu.

© Ondřej Vladík, 2016.

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

1	Úvod	3
2	Uživatelská rozhraní	4
2.1	Audiopad	4
2.2	Rozšířený papír	4
2.3	Supply Chain Visualization	5
2.4	Tangible tiles	6
3	Prvky gui	7
3.1	Správce oken	7
3.1.1	KWin	7
3.1.2	Mutter	8
3.1.3	Compiz	8
3.2	Desktopové widgety	8
3.3	Správa widgetů	9
3.3.1	Conky manager	9
3.3.2	Screenlets manager	9
3.4	Značky	9
3.4.1	Artoolkit	10
3.4.2	Detekce značek	11
4	Kalibrace kamery a projektoru	12
4.1	Model dírkové kamery	12
4.2	Kalibrace kamery	13
4.3	Kalibrace projektoru	14
5	Návrh programu	16
5.1	Struktura programu	17
5.2	Kalibrace	17
6	Implementace	19
6.1	OpenCV	19
6.2	Kalibrační program	19
6.3	Hlavní program	20
7	Testování	22
7.1	Projekční plocha	22
7.1.1	Projekční plocha-svislá	22

7.1.2	Projekční plocha-vodorovná	22
7.2	Kalibrační program	23
7.2.1	Kalibrace kamery	23
7.2.2	Kalibrace kamery a projektoru	23
7.3	Osvětlení pracovní plochy	24
7.4	Velikost značek	24
8	Závěr	25
	Literatura	26
	Přílohy	28
	Seznam příloh	29
A	Obsah CD	30
B	Potřebný software pro spuštění programu	31
B.1	Compiz	31
B.2	Screenlets	32
B.3	Artoolkit	32
B.4	OpenCV	32
C	Manuál	33
C.1	Kalibrace kamery a projektoru	33
C.2	Instalace a spuštění hlavního programu	33

Kapitola 1

Úvod

Uživatelé se stále více zajímají o alternativní ovládací prvky počítače, než klávesnice a myš, které jsou pro ně zastaralé. Mezi takové prvky patří například senzor Leap Motion a rozhraní využívající kameru a projektor. Rozhraní využívající tyto prvky se mohou ovládat různými způsoby, pohyby těla uživatele, gesty rukou nebo pomocí různých značek. Označuje se jako TUI¹ (tangible user interface) neboli také jako hmatové uživatelské rozhraní.

Cílem této práce je vytvoření uživatelského rozhraní využívající systém kamera- projektor pro správu widgetů. Na pracovním stole (ploše) budou umístěny značky a ke každé bude projektor promítat nějaký widget. Těmito značkami by uživatel libovolně pohyboval po pracovní ploše a widgety by se pohybovaly s nimi. Widgety by měly být promítány pouze v případě, kdy je detekována určitá značka. Každé značce by odpovídal jeden specifický widget, který by si uživatel sám zvolil, před spuštěním celého programu.

Promítaná pracovní plocha by byla rozšířená plocha počítače, takže by bylo možné s počítačem stále pracovat a zároveň by mohl s widgety pohybovat někdo jiný. Případně rozšířenou plochu přímo upravit, pomocí myši, změnit velikost widgetů, apod.

Účelem tohoto rozhraní je rozšířit možnosti klasického počítače a usnadnit uživateli práci.

První kapitola se věnuje již existujícím rozhraním využívající systém kamera-projektor. Ukazuje jaké další využití tento systém má. Druhá kapitola podrobněji popisuje prvky GUI. Celá třetí kapitola se zabývá kalibrací kamery a celého systému. Kapitoly 5 a 6 se zabývají samotným návrhem a implementací navrženého rozhraní. Poslední kapitola se věnuje testování tohoto rozhraní.

¹https://en.wikipedia.org/wiki/Tangible_user_interface

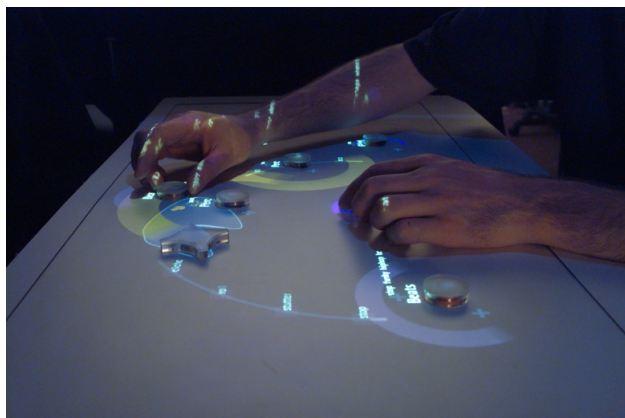
Kapitola 2

Uživatelská rozhraní

2.1 Audiopad

K této práci není využita kamera, ale funguje na podobném principu. Kamera je zde nahrazena elektronikou, pro určení pozice, mezi ostatními objekty. Nástroj pro elektronickou hudbu, který na základě pozice objektů na povrchu, převádí svůj pohyb na hudbu, viz 2.1.

Umožňuje nejen náhodné přehodnocení hudebních skladeb, ale také vytváří vizuální a hmatový dialog mezi uživateli. Audiopad má matici anténních prvků, které sledují pozice elektronicky označených objektů na stolní plochu. Software převádí informace o poloze do hudby a graficky zpětnou vazbou na stůl. Každý objekt představuje hudební skladbu nebo mikrofon[11].

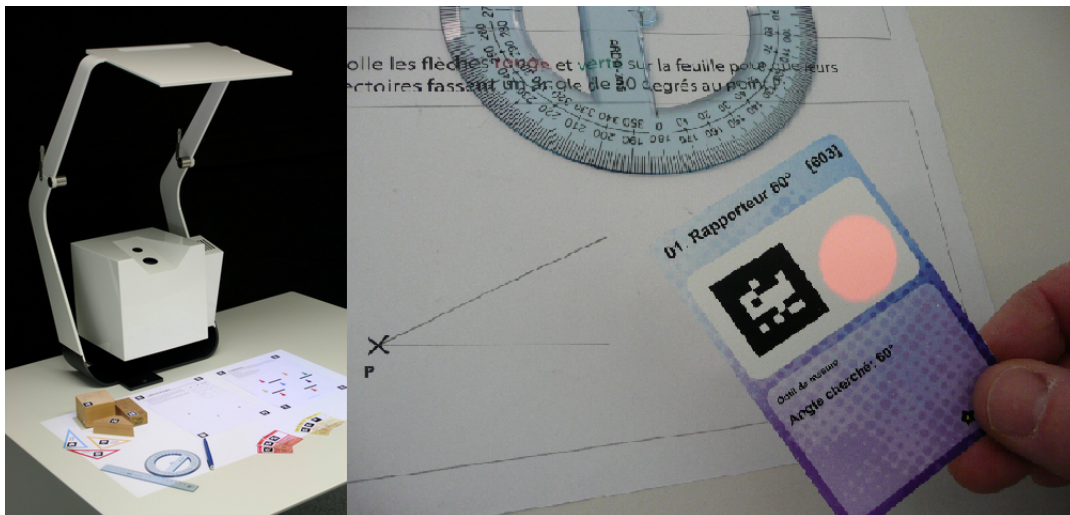


Obrázek 2.1: Ukázka, jak Audiopad funguje[11].

2.2 Rozšířený papír

Tento projekt je především do školního prostředí. Využívá systému kamera-projektor. Kamera snímá značky na pracovní ploše, podle druhu značky a její pozice mezi ostatními značkami se promítají různé geometrické obrazce a hodnoty. Kamera i projektor jsou umístěny do jednoho zařízení a snímání a promítání je umožněno přes zrcadlo. Díky tomu je celé zařízení velice dobře přenositelné, například mezi učebnami školy, viz 2.2.

Značky jsou ve formě papírových kartiček a pracovních listů určených pro práci. Kartičky jsou určeny pro zadání úkolů, zobrazení obrazců, měření různých přímků a zobrazení těchto hodnot. Žáci mohou pomocí nich rýsovat různé geometrické obrazce nebo je mohou narysovat sami a učitelé mohou pomocí kartiček zadané úkoly kontrolovat.[5]



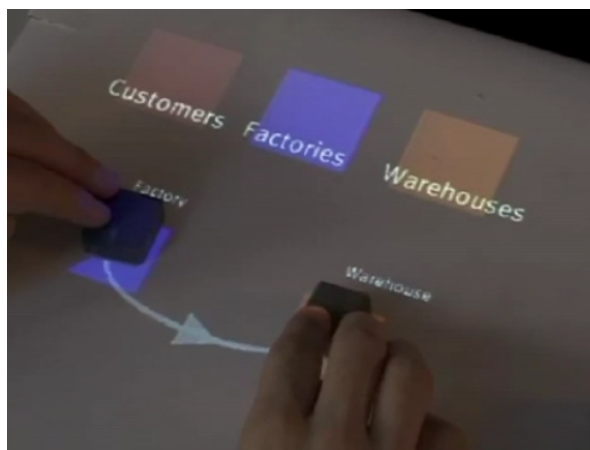
Obrázek 2.2: Vlevo celé zařízení s pracovní plochou. Vpravo ukázka kontroly zadaného úkolu[5].

2.3 Supply Chain Visualization

Tato práce nabízí velké možnosti manažerům. Umožňuje jim fyzicky pracovat s virtuálními modely struktury firmy. Jako zaměstnanci, dodavatelé, cesta zboží, zákazníci. Také dovoluje simulovat různé situace "co kdyby", přidávání dalších budov, rozšíření výroby, atp. Ukázka této simulace je na obrázku 2.3.

Rozhraní má dvě části. První dovoluje vytvářet modely svých dodavatelských řetězců od nuly. Vytváří je pomocí manipulace s fyzickými objekty představující továrny, dodavatele. Počítač na základě pohybů objektů dává uživateli zpětnou vazbu formou projekce.

Druhá část rozhraní dokáže simulovat současný model dodavatelského řetězce.[12]



Obrázek 2.3: Ukázka simulace pohybu zboží [12].

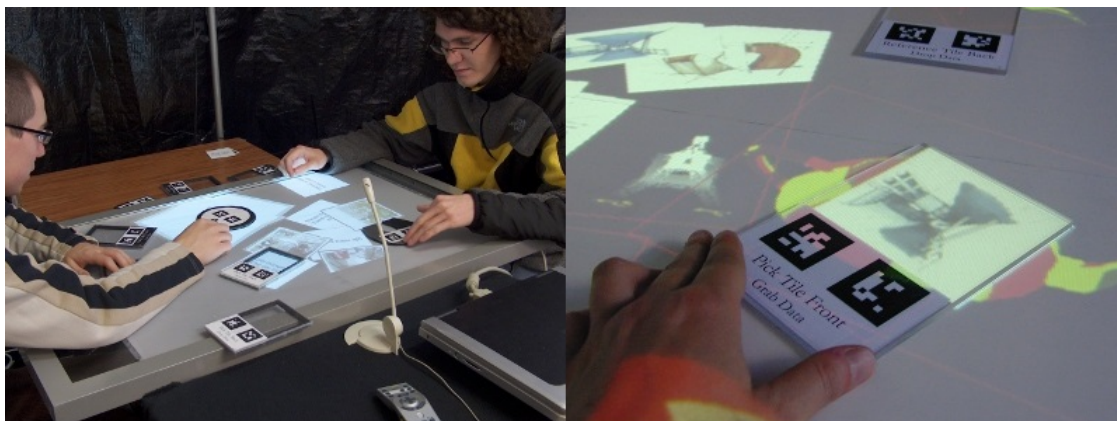
2.4 Tangible tiles

Tangible tiles (dále jen dlaždice) kombinují výhody gui a hmotných uživatelských rozhraní. Dlaždice jsou z průhledného plexiskla s rozměry 15x10cm. Skládají se ze značek na bázi papíru, z obou stran a transparentní plochy.

Stropní kamera rozpoznává a sleduje tyto značky, což umožňuje systému vědět, které dlaždice jsou na stole, kterou stranu nahoru, a kde se nacházejí. Používají se pro pohyb a komplexní manipulaci dat promítaných na stůl. Transparentní plocha je použita jako plátno pro zobrazování náhledu obsažených dat nebo pro manipulační techniku poskytovanou konkrétní dlaždicí, viz 2.4.

Dlaždice z jedné strany může být použita pro virtuální "výběr" promítaných dat, vytváření odkazů, nebo kopií souboru dat. Druhá strana poskytuje pokročilé manipulace pro promítaná data, například mazání.

Vzhledem k tomu, že jsou data v podstatě "obsažena" v dlaždici, jsou data uchopitelná, a tak umožňují snadnější manipulaci pro uživatele. Na rozdíl od standardních vstupních zařízení jako jsou myši nebo laserové ukazovátka [8].



Obrázek 2.4: Vlevo ukázka práce s dlaždicemi, vpravo detail na samotnou dlaždici [8].

Kapitola 3

Prvky gui

V této kapitole jsou popsány jednotlivé prvky gui. Pro přístup k widgetům v operačním systému (dále jen OS) je vhodné použít správce oken, který převádí uživatelské akce na vstupní události operačního systému. Díky tomu je možné ovládat libovolné aplikace na OS. Více je popsán v následující podkapitole.

V další podkapitole jsou popsány widgety a jejich správci. Tito správci umožňují přístup k jednotlivým widgetům, jejich vypínání, zapínání. Také většinou zprostředkovávají uživateli jednoduchý náhled pro jejich výběr. Proto je vhodné nějakého správce použít, aby uživatel mohl mít přehled o množství widgetů, které může použít.

Další kapitola je věnována značkám. Jak značky vypadají, jaké lze použít, případně jak probíhá jejich detekce.

3.1 Správce oken

Správce oken (anglicky window manager) je software v OS, který se stará o umístění a vzhled oken programů běžících uvnitř grafického uživatelského rozhraní. Zajišťuje běh a základní nastavení grafického vzhledu, vykreslování jednotlivých oken a jejich ovládacích prvků.

Jedním z možných rozšíření klasického správce oken je kompozitní správce. Hlavní rozdíl oproti klasickému správci oken je, že nepřevádí svůj výstup rovnou na obrazovku. Obrazy jednotlivých oken i pozadí jsou vytvářeny v paměti zvlášť a grafický procesor je pouze instruován, jak je má na monitoru zobrazit. Díky tomu může poskytnout různé grafické efekty (průhlednost, vlnící se okna, 3D pracovní plochu). Také je mnohem lepší přístup k samotným úpravám plochy [18]. Proto je tento typ správce vhodné použít.

Správce oken jako takových je velice mnoho na různých OS. Zde jsou popisovány pouze někteří kompozitní správci pro OS linux. Vzhledem k tomu, že se předpokládá volba tohoto typu správce a implementace celé práce na tomto systému.

3.1.1 KWin

KWin je kompozitní správce pro KDE. Může být ale použit i v jiných desktopových prostředích. V mnoha ohledech se neliší od správce oken Compiz. Umožňuje stejné nebo podobné 3D efekty pro manipulaci s okny. Jeho hardwarové požadavky jsou velmi nízké. Je celý napsán v C++ [6].

3.1.2 Mutter

Mutter je kompozitní správce oken pro prostředí GNOME , který nahradil staršího správce oken Metacity. Je naprogramován v jazyce C a používá knihovnu Clutter a tedy i OpenGL, což mu umožňuje využívat při provádění třírozměrné grafiky hardwarovou podporu v grafické kartě. Obsahuje také stručnou dokumentaci s popisem, pro případné vytváření vlastních pluginů.[17]

3.1.3 Compiz

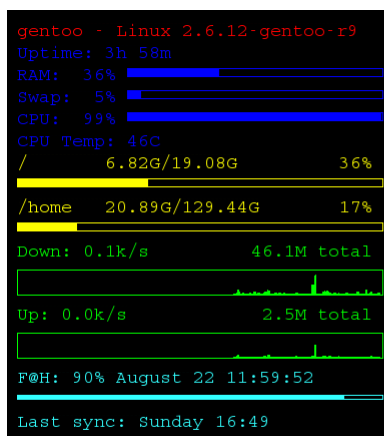
Compiz je typickým zástupcem kompozitního správce pod OS Linux. To jakým způsobem se budou okna zobrazovat zajišťují pluginy.

Pluginy umožňují mnoho grafických efektů jako průhledná nebo vlnící se okna, různé světelné efekty při otevírání nebo zavírání oken. Nejčastěji je Compiz používán pro plugin, který vytváří 3D plochu ve tvaru kostky. Uživatel přepíná mezi plochami, které vypadají jako by byli součástí krychle, kde stěny krychle představují samotné plochy.

Compiz umožňuje také vytvářet vlastní pluginy. Kde uživatel může vytvářet vlastní vizuální efekty pro okna. Součástí je stručná dokumentace pro jejich implementaci.[2]

3.2 Desktopové widgety

Desktopové widgety (dále jen widgety) jsou aplikace reprezentující věci, které se většinou nachází na psacím stole, jako jsou kalendář, hodiny, pravítko, apod. Kromě nich jsou to i aplikace zobrazující stav a informace o systému, počasí nebo právě přehrávanou hudbu. Jsou zobrazovány v pozadí na ploše a je většinou možné s nimi libovolně hýbat podle potřeb uživatele. Jejich velkou výhodou je mít tyto věci neustále po ruce a také ušetření místa na pracovním stole [16].



a)



b)

Obrázek 3.1: Vlevo ukázka a) widgetu z Conky manageru[13]. Vpravo ukázka b) 4 widgety ze Screenlets manageru

3.3 Správa widgetů

Programů pro správu widgetů je více. Jsou zde popsány v současné době ty nejpoužívanější a ty které jsou podporovány OS linux.

3.3.1 Conky manager

Je aplikace zobrazující na plochu informace o vytížení systému, zaplnění disku, zatížení sítě, apod. Na rozdíl od ostatních aplikací zobrazuje informace pouze v textové podobě s velmi jednoduchou grafikou. To mu sice nedovoluje zobrazovat různé grafické prvky, tak jako u ostatních aplikací, ale jeho velkou výhodou je možnost provozu i na velmi slabých počítačích[13].

3.3.2 Screenlets manager

Je jednoduchý nástroj pro jejich zobrazení a práci s nimi. Obsahuje jednoduché menu pro jejich úpravy a přehledný náhled všech nainstalovaných widgetů, viz 3.2. Nastavit se dají jednotlivě nebo je možné je upravit hromadně. Jednotlivé widgety se dají přepínat z pozadí do popředí a je s nimi možné pracovat stejně jako s aplikačními okny. Mohou se také například nastavit, aby se spouštěly samostatně po přihlášení [9].

Nabízí velké množství předpřipravených widgetů. Pokud uživateli nestačí je možné další widgety importovat z jiných systémů.



Obrázek 3.2: Ukázka spuštěného správce widgetů Screenlets.

3.4 Značky

Značkami (nebo také markery) jsou myšleny černobílé značky určené pro rozpoznání pomocí počítačového vidění. Jejich design je jednoduchý, aby byly snadno rozpoznatelné. Používají se často v souvislosti s rozšířenou realitou. Značek jako takových je více druhů, zde jsou popsány značky typu tzv. template marker. Tento typ používá knihovna ARToolkit. Ostatní typy značek jsou podrobně popsány zde. Jsou to černobílé čtverce se specifickými rozměry

a určitým obrazcem uvnitř. Tyto značky jsou používány kvůli jednoduché detekci i při zhoršené kvalitě obrazu.

Tento typ značek má několik pravidel pro případné vytváření vlastních [1]:

- Musí to být čtverec.
- Musí mít celý spojený okraj (většinou bílý nebo černý) a musí být kontrastní podklad.
- Toto omezení se týká obrazce uvnitř, ten nesmí být rotačně symetrický.



Obrázek 3.3: Vzorová značka knihovny Artoolkit [1].

3.4.1 Artoolkit

Artoolkit je otevřená knihovna pro aplikace s rozšířenou realitou. Využívá algoritmů počítačového vidění k určení pozice a natočení kamery, vzhledem k daným objektům. Dokáže snímat obraz z videokamery v reálném čase a na něm detekovat různé objekty, nejčastěji značky, viz 3.3. Následně na nich umožňuje vytvářet různé 3D virtuální objekty [1].

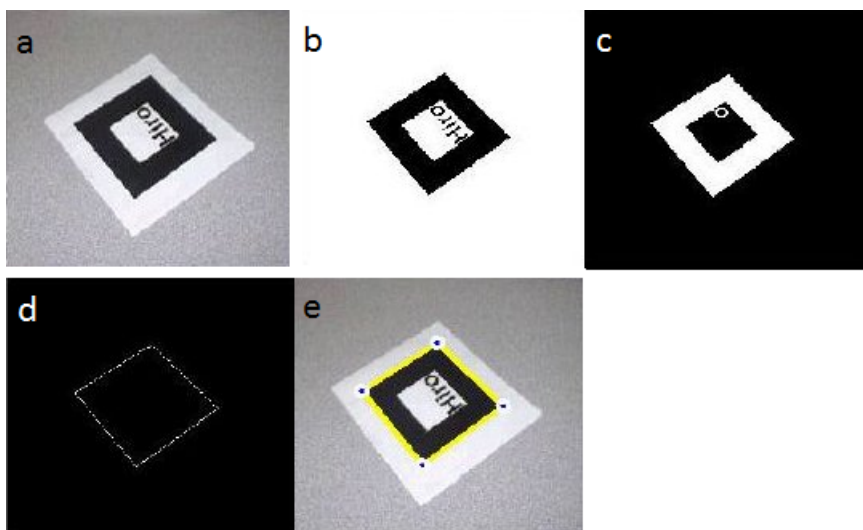
Samotná knihovna obsahuje několik ukázkových příkladů a je k dispozici také přehledná dokumentace. Součástí je také nástroj pro tvorbu vlastních značek.

V současné době je vývoj této knihovny pozastaven. Ale na jejím základě je vytvořeno mnoho knihoven, které nesou podobné názvy.

Mezi další vlastnosti knihovny patří [1]:

- Zajišťuje zpracování videa v reálném čase.
- Je distribuován kompletní zdrojový kód se vzorovými příklady.
- Podpora více systémů (Linux, Mac OS X, Windows).
- Obsahuje jednoduchou kalibraci kamery.

3.4.2 Detekce značek



Obrázek 3.4: Ukázka zpracování snímku z kamery [1].

Postup zpracování obrazu [14]

1. Kamera získá snímek, který posílá počítači.
2. Počítač zpracuje snímek, převede jej pomocí prahování na černobílý snímek.
3. V obraze jsou nalezeny všechny čtyřhranné objekty, které odpovídají vzorům značek.
4. Pokud jsou nějaké značky detekovány, je vypočítána jejich pozice ve snímku z kamery.

Při detekci je nutné znát fyzickou velikost značky. Ta je nutná pro výpočet pozice a rotace kamery v obraze vůči značkám. Poloha kamery je důležitá pro případné vykreslení 3D grafiky, pro kterou se tato knihovna nejčastěji používá. Při tvorbě této práce není potřeba velikost značky znát.

Knihovna má i jistá omezení pro detekci značek [1]:

- Detekce je velice závislá na světelných podmínkách. Při horším osvětlení nebo jakémkoliv odlesku nemusí být značka detekována.
- Značky jsou detekovány jen pokud jsou z pohledu kamery celé vidět. Pokud je část zakryta, značka není vůbec detekována.
- Při tvorbě vlastních značek je dobré volit jednoduché symboly. Komplexnější vzory jsou z větší dálky hůře detekovatelné.

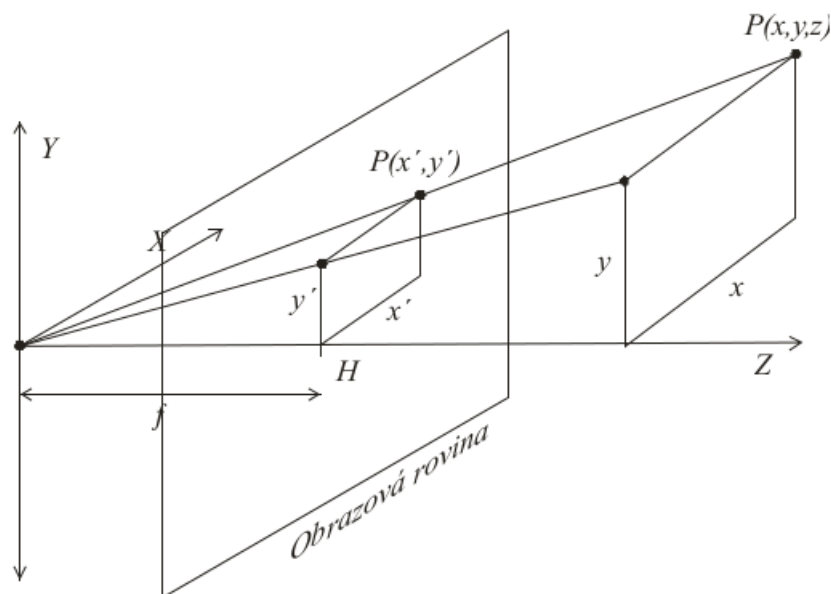
Kapitola 4

Kalibrace kamery a projektoru

Pro rozhraní využívající systému kamera-projektor je potřeba mít správně nakalibrované jednotlivé komponenty. Proto se tato kapitola zabývá jejich kalibrací.

4.1 Model dírkové kamery

Kamera se definuje jako matematický model, který se nazývá „model dírkové kamery (pinhole camera model)“. Tento model je definován pomocí vnitřních a vnějších parametrů, které popisují deformaci obrazu a pozici kamery ve scéně. Výpočet těchto parametrů nazýváme kalibrací kamery. Pomocí tohoto modelu tedy můžeme pomocí lineární transformace promítnout bod z prostorové roviny do obrazové roviny [7]. Souřadnice bodu $P'(x', y')$ lze získat z $P(x, y, z)$ v prostorové rovině pomocí vzorce (4.1). Přepočítání získaných koeficientů již je možné provést pomocí výrazů (4.2).



Obrázek 4.1: Princip modelu dírkové kamery [7].

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (4.1)$$

$$x' = \frac{u}{v}, y' = \frac{v}{w} \quad (4.2)$$

4.2 Kalibrace kamery

Tyto parametry jsou nutné pro výpočet prostorové pozice objektu, o známé velikosti, vzhledem ke kameře., [19] Kalibrace probíhá tak, že kamera provádí mapování mezi 3D světovým metrickým systémem a 2D souřadným systémem obrazu v pixelech. Aby bylo možné provést inverzní proces, tedy spočítat 3D metrické souřadnice z 2D pixelů, je tedy nutné znát interní a externí parametry této kamery."

Interní parametry

Následující rovnice ukazuje matici interních parametrů [15]:

$$K = \begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

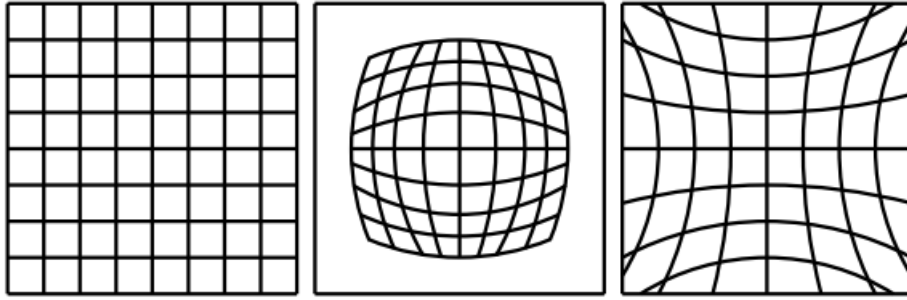
Znázorňuje tyto parametry:

- α_x, α_y - Ohnisková vzdálenost vyjádřená v pixelech.
- u_0, v_0 - Repräsentují polohu hlavního snímkového bodu.

Tyto parametry jsou unikátní pro každou kameru. Je možné je uložit a znovu použít, pokud by byla používána stále stejná kamera. Při dalším spuštění aplikace by, už kalibrace kamery nebyla potřeba.

Zkreslení obrazu

Pro vstupní obraz z kamery, dle optických parametrů objektivu mohou v získaném obraze nastat geometrická zkreslení způsobená jeho optikou. To se projevuje tím, že se přímky v vstupním obraze zobrazují jako křivky. Mezi takové zkreslení patří například typ polštář, barel nebo fisheye. Polštář a barel patří mezi nejběžnější typy zkreslení, jsou označovány jako tzv. radiální zkreslení. To způsobuje posunutí bodů v obrazové rovině. Velikost tohoto posunutí závisí na typu zkreslení, kde se od středu do stran zvyšuje nebo snižuje [4]. Ukázka takových zkreslení je na obrázku 4.2.



Obrázek 4.2: Nezkreslený obraz, zkreslení obrazu typu barel, zkreslení typu polštář [4].

Externí parametry

Těmito parametry se vyjadřuje transformace mezi souřadným systémem kamery a světovým souřadným systémem, vůči němuž je kamera kalibrována. Vztah pro vyjádření pozice kamery, kde C označuje pozici kamery, R vyjadřuje rotaci a T označuje pozici počátku souřadnic reálného systému vyjádřené v souřadnicích souřadného systému kamery.

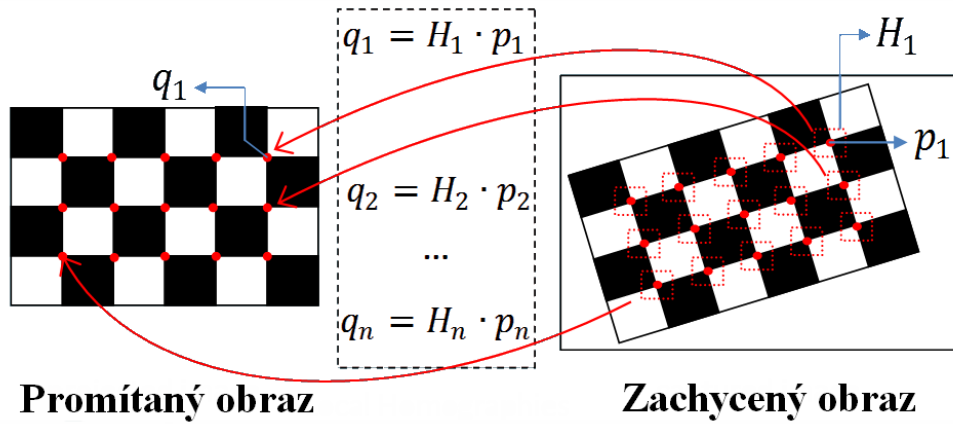
$$C = -R^{-1}T \quad (4.4)$$

4.3 Kalibrace projektoru

Tato kalibrace se provádí, protože se souřadnice objektu v obou rovinách liší. Aby je bylo možné převádět, je nutné znát polohu kamery i projektoru. Díky tomu, že se ví, že plocha kam se bude obraz promítat je zcela rovná, není nutné znát prostorové umístění vzhledem ke kameře a je možné provést kalibraci projektoru pomocí homografie.

Homografie

Homografie se znázorňuje maticí H . Tato matice vyjadřuje projektivní transformaci bodů mezi rovinami kamery a projektoru. Díky tomu je možné vypočítat souřadnice nějakého bodu v obraze projektoru ze souřadnic téhož bodu v obraze kamery. Výpočet probíhá s množinou bodů, u kterých jsou zadány souřadnice z jedné i druhé roviny.



Obrázek 4.3: Promítnutí bodů mezi rovinami [3].

Kapitola 5

Návrh programu

V celém návrhu se předpokládá rozvržení celého systému ve vodorovné poloze. Kamera a projektor jsou pevně umístěny nad stolem. Na kterém leží značky, ke kterým se projektorem promítají widgety.

Před prvním spuštěním se provede kalibrace celého systému, která se už opakovat nemusí, pokud se nebude hýbat s projektorem nebo kamerou.

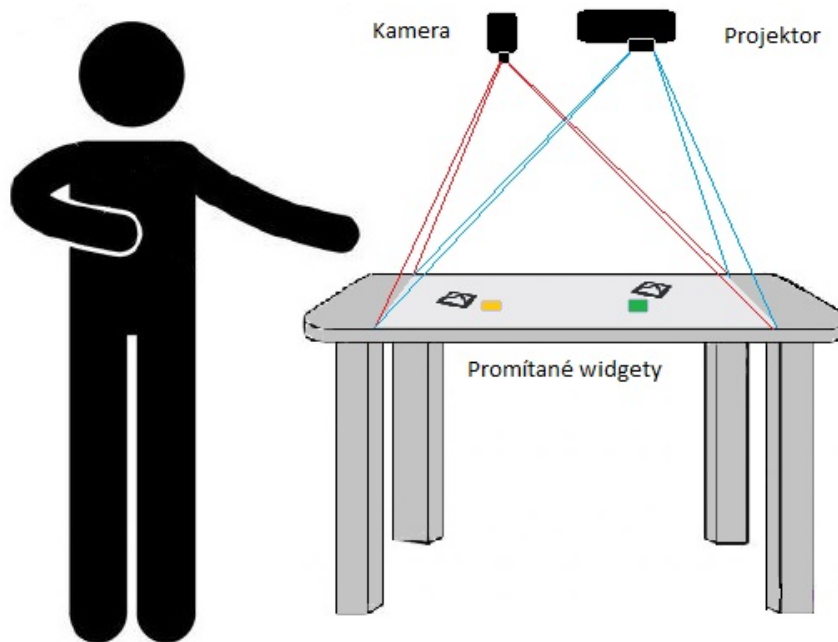
Potom si uživatel zvolí jaké widgety mohou být promítány. Jakou značku bude každý widget představovat. Poté se aplikace spustí.

Uživatel může se značkami libovolně hýbat po pracovní ploše a widgety se budou pohybovat s nimi. Pokud značka není na pracovní ploše nebo je nějakým způsobem částečně zakryta widget, přestane být promítán.

Hardwarová sestava

Hlavní části systému:

- Kamera - Snímkuje pracovní plochu.
- Projektor - Promítá prvky rozhraní na pracovní plochu.
- Počítač - Zpracovává snímky z kamery, pohyb prvků rozhraní, generuje výsledný promítaný obraz na pracovní plochu.
- Pracovní plocha - Stůl, stěna, případně jiná větší plocha, která je zcela rovná.



Obrázek 5.1: Model navrhnutého rozhraní.

5.1 Struktura programu

Hlavní cíl programu je zajistit detekci značek ve video sekvenci a předat jednotlivé souřadnice značek, z obrazu kamery, promítaným widgetům. Díky tomu, že projektor bude umístěn kolmo nad pracovní plochou, nemusí se řešit případné transformace widgetů.

Pro přístup k widgetům je nutné napojení na správce oken OS. Vybrán byl Compiz?? kvůli jeho velké rozšířitelnosti, dobrým možnostem pro vytváření nových pluginů a jeho dobré dokumentaci.

Vhodné by bylo celou práci implementovat jako plugin do výše zmiňovaného kompozitního správce.

Pro správu widgetů byl zvolen manager Screenlets ??, kvůli jednoduchému přístupu k widgetům a jeho přímé podpoře ovládání správcem oken Compiz.

5.2 Kalibrace

Vzhledem k tomu, že nepotřebujeme znát prostorovou pozici kamery a projektoru, můžeme použít zjednodušenou kalibraci projektoru a kamery pomocí homografie. Pro kalibraci se většinou používá kalibrační vzor. Jedná se o šachovnici, pro kalibraci kamery je vytištěná na papír. Pro kalibraci projektoru a kamery je promítána na pracovní plochu.

Kalibrace kamery a projektoru probíhá v následujících krocích:

1. Projektorem je promítán šachovnicový vzor, o předem známe velikosti, na pracovní plochu, zároveň jsou u promítaného vzoru, z obrazu projektoru, detekovány vnitřní rohy šachovnice, které jsou uloženy.

2. Kamera snímá ve video sekvenci promítanou šachovnici, ve které detekuje její vnitřní rohy.
3. Jakmile jsou získány vnitřní rohy šachovnice v obrazové rovině kamery a rovině projektoru, je pomocí homografie získána transformační matice. S touto maticí je možné převést souřadnice z jedné roviny do druhé.
4. Pracovní plocha - Stůl, stěna, případně jiná větší plocha, která je zcela rovná.

Poloautomatická kalibrace celého systému je úplně oddělena od hlavního programu. Výhodou takového rozdělení je samostatné použití i jiné programy. A zároveň není potřeba, kvůli kalibraci kamery a celého systému, spouštět hlavní program.

Kapitola 6

Implementace

V této kapitole je popsán způsob implementace kalibračního a hlavního programu. Pro zpracování obrazu v obou programech bylo využito knihovny OpenCV.

6.1 OpenCV

OpenCV (Open-source Computer Vision) [10] je otevřená knihovna počítačového vidění pro zpracování obrazového vstupu v reálném čase. Je napsána v C/C++ a lze ji používat na systémech Linux, Windows, Mac. Podporuje jazyky C/C++ tak i Python a Java. Jeho otevřenost a multiplatformnost, spolu s velmi dobře napsanou dokumentací, zajišťuje jeho neustále větší používání pro projekty s počítačovým viděním.

6.2 Kalibrační program

Kalibrační program je implementován v jazyce C++. Vypočty potřebné pro kalibraci jsou prováděny pomocí knihovny OpenCV. Celý program se skládá ze dvou částí:

- Kalibrace kamery.
- Kalibrace projektoru.

Kalibrace kamery

Ke kalibraci se využívá vytištěný kalibrační vzor šachovnice s vnitřními body 9x6. Tento vzor je potřeba mít na pevném a rovném podkladu. Aby nedocházelo ke špatným výsledkům kalibrace, zapříčiněných různými deformacemi tohoto vzoru. Pro tuto práci je z kalibrace potřebné pouze získání vnitřních parametrů.

Nejdříve je nutné získat snímky kalibračního vzoru v různých pozicích. S tímto vzorem se různě pohybuje před kamerou, která je pevně umístěna. Snímky jsou zachycovány v reálném čase. Po získání snímků kalibračního vzoru se provádí jeho detekce pomocí funkce `findChessboardCorners()`. Detekce spočívá v nalezení vnitřních rohů šachovnice, které jsou uloženy jako matice bodů. Aby je bylo možné vůči něčemu kalibrovat, je třeba mít vlastní ideální šachovnicový vzor, který bude sloužit jako předpis pro nalezený vzor. Po získání dostatečného počtu snímků je možné přejít ke kalibraci.

Pomocí funkce `calibrateCamera()` jsou získány vnitřní parametry kamery. Výsledkem je získaná matice kamery a matice s koeficienty zkreslení. Návratovou hodnotou je průměrná

chyba reprojekce. Tato hodnota udává přesnost nalezených parametrů. Měla by být co nejbližší 0. Nyní je možné pomocí získaných parametrů provést korekci snímku.

Pro implementaci této kalibrace byl použit vzorový příklad z dokumentace ke knihovně OpenCV¹, který byl upraven pro potřeby tohoto zadání.

Kalibrace projektoru

Kalibrace projektoru vůči kameře využívá metody pro získání homografie mezi rovinou projektoru a rovinou kamery. Homografie se získá na základě použití šachovnicového vzoru. U tohoto vzoru je nutné znát velikost čtverce a šířku bílých okrajů okolo vzoru. Pro byl navrhnout vlastní vzor, který odpovídá rozlišení monitoru, tak aby pokryl obsah celé projekce.

Velikost celého vzoru je 1200x800 pixelů. Počítá se do toho i bílý okraj okolo šachovnice, který je nutný pro detekci šachovnice. Okraj je široký 20 pixelů z levé i horní strany. Pro kalibraci je nutné znát pouze tyto strany. Ostatní strany by měli být alespoň stejně velké nebo větší. Velikost jednoho čtverce je 150 pixelů.

Protože je známa jeho velikost, mohou se uložit jeho vnitřní rohy šachovnice. Jakmile je tento vzor promítán na plochu, kamera získá snímek. V tomto snímku je nalezen promítaný vzor. Po jeho nalezení jsou uloženy jeho vnitřní body šachovnice.

Po získání vnitřních bodů šachovnice z obou rovin, je možné provést výpočet homografie pomocí funkce `findHomography()`. Výsledkem je transformační matice, kterou lze převést souřadnice bodu z jedné projekční roviny do druhé.

6.3 Hlavní program

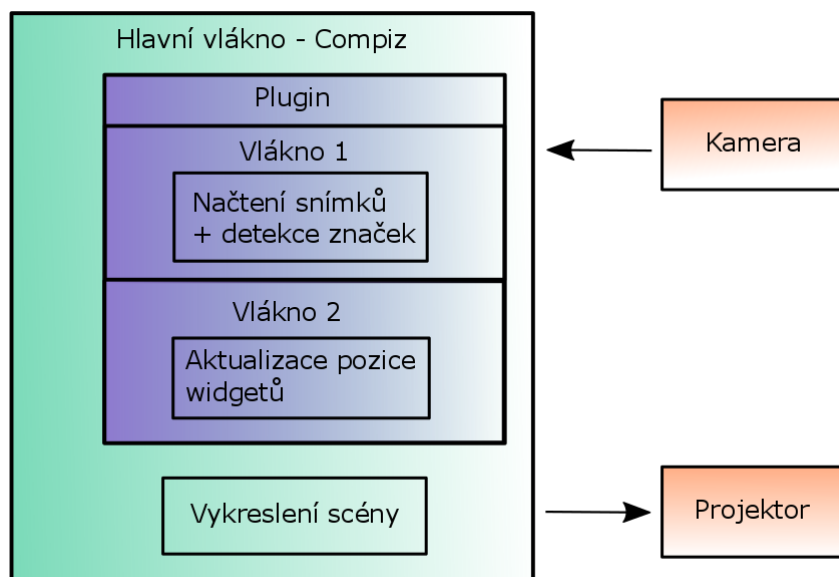
Hlavní program byl vytvořen podle návrhu jako plugin do správce oken Compiz. Plugin je implementován v jazyce C++. Přepnutí správce oken se provádí pomocí scriptu `RunSystem.sh`. Z tohoto scriptu se také pouští všechny widgety vybrané uživatelem.

Pluginy v Compizu se obecně liší od většiny programů implementovaných v jazyce C/C++. Především proto, že nemají například funkci `Main()`. Většinou jsou pluginy jednou funkcí inicializovány a nastaveny. Potom jsou jednotlivé funkce volány na základě různých událostí jako stisk klávesy, pohyb nebo kliknutí myši.

Plugin pro tuto práci byl implementován na základě vzorového příkladu `Move`. Ten zajišťuje pohyb všech oken po monitoru pomocí myši.

Jelikož program musí pracovat v reálném čase, musí být odezva při pohybu značek co nejmenší. Proto bylo navrženo oddělení nejnáročnější části do samostatného vlákna. Samostatně tedy běží získávání snímků z kamery a detekce značek v tomto snímku, viz. 6.1.

¹http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html



Obrázek 6.1: Přehled oddělených vláken a struktura celého programu.

Při detekci je také získána pozice značky, tyto souřadnice jsou dále zpracovány už samotným pluginem v hlavním vláknu. Ten předá souřadnice widgetům, které odpovídají jednotlivým značkám.

Aktualizace pozice widgetů, je implementována ve funkci `updatePosition()`, která je volána funkcí `Compizu Paint()`. Tato funkce zajišťuje vykreslování všech oken na monitoru a je volána v pravidelných intervalech v řádu milisekund. Tímto způsobem je zajištěno pravidelné aktualizování souřadnic.

Kapitola 7

Testování

Testování bylo zaměřeno primárně na funkčnost celého rozhraní, na kvalitu poloautomatické kalibrace celého systému. Také bylo testováno více druhů projekční plochy.

7.1 Projekční plocha

Při testování byla brána i možnost, kdy prostory pro práci, nedávají možnost sestavit celý systém nad stolem. Kde by projektor a kamera museli být umístěny téměř u stropu. Uživatel také nemusí mít potřebné vybavení pro uchycení kamery a projektoru. Proto bylo rozhraní testováno na 2 projekčních plochách. Testování probíhalo v seminární místnosti v knihovně školy a v laboratoři UPGM.

7.1.1 Projekční plocha-svislá

Takové to rozložení se využívalo při prvních testováních a dále po zprovoznění celého programu. Projekční plocha byla v tomto případě školní tabule, která je pro toto rozložení nejvhodnější. Značky byly drženy na tabuli pomocí magnetu. Kamera i projektor byli postaveny na stole kolmo k tabuli.

Mezi další možnosti projekční plochy by se dala zařadit nástěnka. Značky by byli v tomto případě připnuty na nástěnce. To by bylo velkou nevýhodou pro každý pohyb se značkou. Pokud by ale uživatel nepotřeboval s nimi pravidelně hýbat a využíval ji jen jako dodatečnou pracovní plochu, byla by plocha dostačující.

Výhodou svislé plochy je, že není potřeba dodatečné vybavení pro uchycení kamery a projektoru nad pracovní plochu. Umožňuje větší promítanou pracovní plochu.

Nevýhodou je komplikovanější pohyb značek po ploše, pokud se nejedná o magnetickou tabuli. Mezi další nevýhody patří možnost clonění uživatelem nebo někým jiným.

7.1.2 Projekční plocha-vodorovná

V tomto případě kamera a projektor jsou uchyceny na stojanu, na stole. Pro testování bylo toto rozložení nejvhodnější. Značky byly položeny na stole a byl k nim okamžitý přístup. Promítaná pracovní plocha byla sice menší ale plně dostačující, viz 7.1. Pro případnou větší promítanou pracovní plochu by byla potřeba větší konstrukce, která by projektor a kameru uchytila co nejlépe stropu.



Obrázek 7.1: Ukázka vodorovné projekční plochy.

Mezi výhody takové plochy patří:

- Okamžitý přístup ke značkám, pokud uživatel pracuje na stejném stole.
- Značky není potřeba nijak uchycovat.

Mezi nevýhody patří:

- Nutnost mít vybavení pro uchycení kamery a projektoru, které zabírá dost místa.
- Promítaná pracovní plocha je menší.
- Větší pravděpodobnost rozhození značek na pracovní ploše.

7.2 Kalibrační program

7.2.1 Kalibrace kamery

Testování probíhalo na různých počtech snímků (25, 40, 65, 90). Pokaždé byla kalibrace spuštěna 3x. Z jednotlivých výsledků byla vypočítána průměrná hodnota. Získané reprojekční chyby a jejich doba zpracování jsou uvedeny v tabulce 7.1. Nejnižší reprojekční chyba byla získána u 40 snímků. S rostoucím počtem snímků stoupala i jejich doba zpracování. Proto bylo rozhodnuto použití 40 snímků.

Tabulka 7.1: Závislost reprojekční chyby a doby zpracování na počtu snímků.

	25 snímků	40 snímků	65 snímků	90 snímků
Doba zpracování [s]	2	10	39	152
Reprojekční chyba	1,1882	0,9395	1,3106	1,5663

7.2.2 Kalibrace kamery a projektoru

Kalibrace projektoru vůči kameře je velice důležitá. Testování probíhalo formou projekce bodů z obrazu kamery do obrazu projektoru. Naměřený rozdíl mezi promítaným bodem a rohem čtverce je v rozmezí 2-5 pixelů. Z toho jde určit, že dosažená přesnost je pro tento program dostačující.

7.3 Osvětlení pracovní plochy

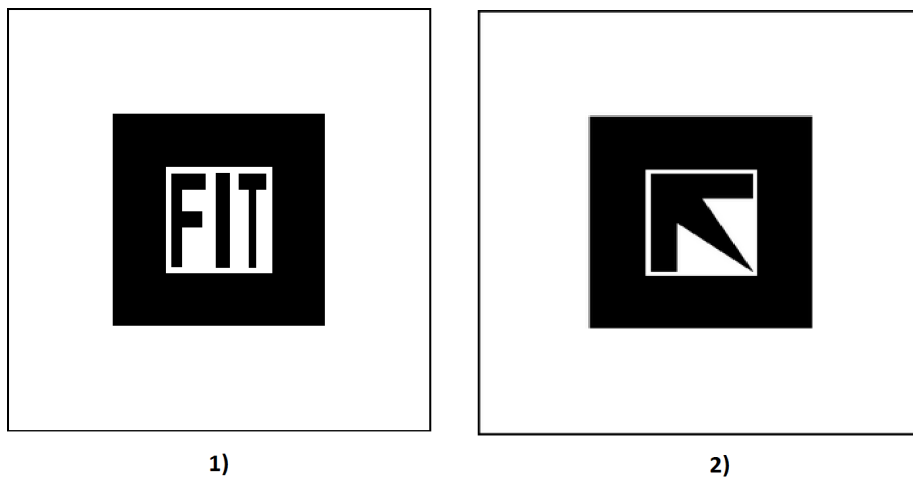
Při testování bylo zjištěno, že detekce značek a následné promítání widgetů je velice závislé na okolním osvětlení.

Při testování svislé pracovní plochy byli prostory špatně osvětlené. Detekce značek probíhala velice špatně. Řešením tohoto problému byla změna barvy pozadí rozšířené plochy, která se promítala. Nejvhodnější byla bílá barva, která osvětlila značky velice dobře. Vzdálenost projektoru od pracovní plochy byla dostačující na to, aby nedocházelo k odlesku od značky. Takto byly značky opět dobře detekovatelné. Nevýhodou tohoto řešení je zviditelnění celé promítané plochy.

Při testování vodorovné pracovní plochy naopak nastala situace, kdy osvětlení celého prostoru bylo až moc velké. Značky byly sice velice dobře detekovatelné, ale promítané widgety nebyly téměř vidět. Tento problém je komplikovanější, protože každý prostor je osvětlen trochu jinak. Nejjednodušším řešením je libovolným způsobem snížit intenzitu osvětlení této plochy. Další ale dražší možností je pořízení nového projektoru s vyšší svítivostí.

7.4 Velikost značek

Velikost značek je závislá na více faktorech. Nejdůležitější je rozlišení kamery, která značky detekuje. Za minimální potřebné rozlišení se považuje 640x480 pixelů. Pro kameru je také důležitá její vzdálenost od pracovní plochy, na které jsou samotné značky. Při příliš velké vzdálenosti nedochází k detekci značek. Testování probíhalo ve více vzdálenostech od 50- 120 cm od pracovní plochy. Při těchto vzdálenostech se pracovalo se značkami o velikostech 4x4 cm a 8x8 cm. S tím, že u velikosti 4x4 cm probíhala detekce pouze některých jednodušších značek. Komplexnější vzory značek i při kratší vzdálenosti nebyly detekovány, viz 7.2.



Obrázek 7.2: Vlevo komplexnější vzor značky, který byl špatně detekovatelný. Vpravo značka, která byla dobře detekovatelná.

Kapitola 8

Závěr

Tato práce se zabývá promítanými uživatelskými rozhraními, které podporují interakci s uživatelem pomocí různých objektů. Je zde uváděn přehled existujících řešení, se zaměřením na rozhraní využívající kameru a projektor. Součástí je popis kalibrace těchto zařízení, který je důležitý pro správnou funkčnost celého programu. Uvedený je také přehled prvků gui, které byly použity při implementaci této práce.

Cílem práce bylo navrhnout uživatelské rozhraní s využitím projektoru a kamery. Navrhnuté rozhraní realizovat a vytvořit aplikaci demonstrující jeho funkčnost. Bylo navrženo uživatelské rozhraní pro správu widgetů pomocí značek. Cíl se podařilo splnit. Bylo implementováno rozhraní, ve kterém pomocí papírových značek, rozmístěných po pracovní ploše (stůl, stěna), lze ovládat promítané widgety. Každý jednotlivý widget odpovídá jedné značce a je k ní neustále promítán. Takto může uživatel libovolně pohybovat s widgety pomocí těchto značek po pracovní ploše. Ukázka funkčnosti celého rozhraní je zachycena na videu, které je součástí příloženého DVD.

Součástí je knihovna pro poloautomatickou kalibraci kamery a projektoru. Kalibrace kamery byla testována na různých počtech snímků a každý se 3x opakoval. Cílem bylo zjištění co nejmenší možné reprojekční chyby. Ta byla zjištěna při získání 40 snímků, kde byla 0,9395.

Testování se zaměřilo na primárně samotnou funkčnost rozhraní. Byly porovnávány projekční plochy. Jak důležité je osvětlení celého prostoru, pro detekci značek. Jaká je ideální velikost značky a na čem její velikost záleží.

Výsledný program ulehčuje uživateli práci. Zobrazuje mu potřebné aplikace na promítané ploše a šetří místo na ploše počítače. Promítané prvky dotváří prostředí kolem počítače. Nabízí další způsob interakce mezi člověkem a počítačem. Celá tato práce byla prezentována formou plakátu na studentské konferenci Excel@FIT 2016.

Možnosti rozšíření tohoto rozhraní jsou veliké. Jedním z nich je odstranění nutnosti kolmé projekce. Takto by kamera a projektor mohli být na libovolném místě a promítat prvky na různé roviny v prostoru. Další možností je podpora více druhů značek, kde by mohli být prvky promítány například přímo na značku. Zajímavým rozšířením by bylo větší využití správce Compiz, který by mohl například promítat 3D plochu na stěnu a uživatel by ji mohl ovládat. K tomuto řešení by bylo vhodné odstranit značky a přidat detekci rukou a gest.

Literatura

- [1] ARToolKit [online]. Dostupné na URL:
<https://www.hitl.washington.edu/artoolkit>, 2007 [cit. 2016-03-16].
- [2] Compiz: Web portal [online]. Dostupné na URL: <http://wiki.compiz.org/>, 2016 [cit. 2016-04-10].
- [3] Daniel Moreno and Gabriel Taubin: Simple, Accurate, and Robust Projector-Camera Calibration [online]. 2012 [cit. 2016-04-12].
- [4] Gaura J.: Odstranění geometrických zkreslení [online]. Dostupné na URL:
http://mrl.cs.vsb.cz/people/gaura/dzo/geom_distortion_cz.pdf, Ostrava 2012 [cit. 2016-04-16].
- [5] James Patten: Web portal [online]. Dostupné na URL:
<http://craft.epfl.ch/lang/en/PaperTangibleInterface>, 2016 [cit. 2016-04-10].
- [6] KDE UserBase Wiki: KWin. 2015.
URL <https://userbase.kde.org/KWin>
- [7] Král, E.: Analýza obrazové informace kamerového systému. 2006.
URL http://digilib.k.utb.cz/bitstream/handle/10563/1370/kr%C3%A1l_2006_dp.pdf?sequence=1&isAllowed=y
- [8] Manuela Waldner, Jörg Hauber, Jürgen Zauner, Michael Haller, Mark Billinghurst: Tangible Tiles: Design and Evaluation of a Tangible User Interface in a Collaborative Tabletop Setup [online]. Dostupné na URL: http://manuela-waldner.net/download/Waldner_etal_TangibleTiles_OZCHI06.pdf, 2006 [cit. 2016-04-16].
- [9] O., S.: Screenlety [online]. Dostupné na URL:
<http://www.linux-mint-czech.cz/screenlety>, 2012 [cit. 2016-04-16].
- [10] OpenCV: OpenCV [online]. Dostupné na URL: <http://opencv.org/>, 2016 [cit. 2016-04-16].
- [11] Tangible Media Group: Web portal [online]. Dostupné na URL:
<http://tangible.media.mit.edu/project/audiopad/>, 2016 [cit. 2016-04-16].
- [12] Tangible Media Group: Web portal [online]. Dostupné na URL:
<http://tangible.media.mit.edu/project/supply-chain-visualization/>, 2016 [cit. 2016-04-16].

- [13] Wiki Ubuntu: Conky [online]. Dostupné na URL: <http://wiki.ubuntu.cz/conky>, 2012 [cit. 2016-04-16].
- [14] Wikipedia: ARToolKit — Wikipedia, The Free Encyclopedia. 2016.
URL <https://en.wikipedia.org/w/index.php?title=ARToolKit&oldid=714794302>
- [15] Wikipedia: Camera resectioning — Wikipedia, The Free Encyclopedia. 2016, [Online; accessed 15-May-2016].
URL https://en.wikipedia.org/w/index.php?title=Camera_resectioning&oldid=710606262
- [16] Wikipedia: Software widget — Wikipedia, The Free Encyclopedia. 2016.
URL https://en.wikipedia.org/w/index.php?title=Software_widget&oldid=715311560
- [17] Wikipedie: Mutter (správce oken) — Wikipedie: Otevřená encyklopedie. 2014.
URL [https://cs.wikipedia.org/w/index.php?title=Mutter_\(spr%C3%A1vce_oken\)&oldid=11608327](https://cs.wikipedia.org/w/index.php?title=Mutter_(spr%C3%A1vce_oken)&oldid=11608327)
- [18] Wikipedie: Správce oken — Wikipedie: Otevřená encyklopedie. 2014.
URL https://cs.wikipedia.org/w/index.php?title=Spr%C3%A1vce_oken&oldid=11883040
- [19] Zahradka, J.: Rozšířené uživatelské rozhraní [online]. Dostupné na URL: <http://www.fit.vutbr.cz/study/DP/DP.php?id=11321&file=t>, Brno 2011 [cit. 2016-04-16].

Přílohy

Seznam příloh

A	Obsah CD	30
B	Potřebný software pro spuštění programu	31
B.1	Compiz	31
B.2	Screenlets	32
B.3	Artoolkit	32
B.4	OpenCV	32
C	Manuál	33
C.1	Kalibrace kamery a projektoru	33
C.2	Instalace a spuštění hlavního programu	33

Příloha A

Obsah CD

Na přiloženém DVD je přiloženo:

- Zdrojové kódy projektu.
- Plakát prezentující práci.
- Video prezentující realizované rozhraní.
- Text této práce.

Příloha B

Potřebný software pro spuštění programu

Celá práce je implementována na operačním systému Linux Mint Xfce 32bit. Pro správný postup instalace (překlad) by neměly být žádné složky ani soubory, kde bude práce instalována, pojmenovány za použití diakritiky.

B.1 Compiz

Některé balíky jsou již součástí systému, jiné je se musí stáhnout a nainstalovat. Jejich úplný přehled je vypsán níže.

- compiz
- compizconfig-settings-manager
- fusion-icon
- compiz-plugins
- compiz-core
- compiz-gnome
- compiz-dev
- compiz-fusion-bcop
- compiz-plugins-default
- libdecoration0
- libdecoration0-dev
- python-compizconfig

B.2 Screenlets

Je potřeba mít nainstalované tyto balíky.

- screenlets
- screenlets-pack-all

B.3 Artoolkit

Je nutné mít nainstalovanou knihovnu ARToolkit¹, byla použita verze 2.72.1.

B.4 Opencv

Při implementaci byla použita knihovna OpenCV verze 3.1.0. Na jiné verzi nebyla práce testována.

¹<https://sourceforge.net/projects/artoolkit/files/artoolkit/>

Příloha C

Manuál

C.1 Kalibrace kamery a projektoru

Pokud je nainstalována knihovna OpenCV stačí program přeložit pomocí `sudo make`. Před samotným spuštěním programu je nutné mít zapojený projektor a mít nastavený displej jako rozšířený.

Po spuštění programu se spustí aplikační okno s obrazem z kamery. Teď je nastaven režim pro kalibraci kamery, stiskem klávesy 'g' se spustí snímkování kalibračního vzoru z obrazu kamery. Ovládání kalibrace kamery funguje stejným způsobem jako vzorový příklad, podle kterého byl program implementován.

Pro kalibraci projektoru stisknete klávesu 'p'. Ta přepne režim kalibrace kamery na kalibraci projektoru. Je vhodné přesunout okno s obrazem z kamery do rozšířeného displeje, který se nyní promítá na určitou plochu. Nyní se stiskne klávesa 'c' pro zobrazení šachovnice, která se zobrazí pře celý obsah projekce rozšířené plochy. Aplikační okno s obrazem z kamery je možné nyní přesunout zpátky na hlavní displej, aby byl vidět. Nyní můžeme v okně vidět jestli byl kalibrační vzor v obraze kamery nalezen. Jakmile je nalezen stiskneme klávesu 's' pro výpočet transformační matice a její uložení. Po uložení se zobrazí nové aplikační okno po transformaci. Stiskem 'esc' se ukončí celý program.

C.2 Instalace a spuštění hlavního programu

Nejdříve je nutné v souboru `/MoveMarkers/MoveMarkers/CMakeLists.txt` doplnit úplné cesty do několika adresářů:

- INCDIRS
 - cesta k opencv, většinou ve tvaru `/usr/local/include/opencv`
 - ke knihovně ARToolKit, bude končit `artoolkit/include`
 - také k adresáři `src`, která je na tomto DVD, je nutné doplnit úplnou cestu
- LIBDIRS
 - ke knihovně ARToolKit, bude končit `artoolkit/lib`
 - také k adresáři `/MoveMarkers/MoveMarkers/src/`, která je na tomto DVD, je nutné doplnit opravdu úplnou cestu

Ve skriptu `RunSystem.sh` je nutné doplnit úplnou cestu k jednotlivým widgetům. POZOR: Jejich pořadí určuje jaké značky jsou přiřazeny. Ukázka této cesty je již v tomto skriptu.

V souboru `MoveMarkers/Data/objectdata2` je nutné přidat úplnou cestu ke konfiguračnímu souboru pro každou značku. Tyto soubory jsou přiloženy na DVD ve složce `MoveMarkers/Data/Marker-config`.

Teď jen otevřít terminál ve složce a zadat `cmake ..` a `make instal`. Nyní je plugin nainstalován. Zapneme Compiz manager a v náhledu najdeme tento plugin, bude pojmenován jako `movemarkers`. Klikneme u něj aktivovat.

Nyní můžeme výše uvedený skript zapnout. Spustí se jednotlivé widgety a přepne se správce oken. Teď je možné změnit motiv nebo velikost jednotlivých widgetů. Ukážeme na ně kurzorem myši a klikneme pravým tlačítkem. Ukáže se nám nabídka jednotlivých úprav. Pokud se nebude nic měnit je možné zapnout plugin klávesou `'s'`. Tuto klávesu je možné změnit v nastavení tohoto pluginu. POZOR: klávesa nesmí být používána jiným pluginem v Compiz manageru.