

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 ROZHRANÍ

DIPLOMOVÁ PRÁCE

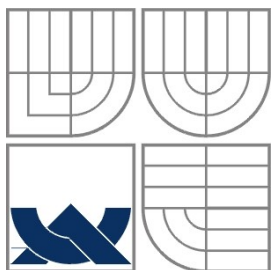
MASTER'S THESIS

AUTOR PRÁCE

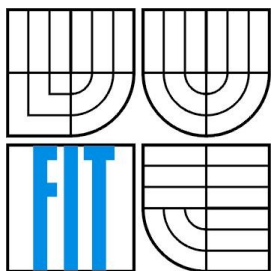
AUTHOR

Bc. JIŘÍ ZAHRÁDKA

BRNO 2011



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

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JIŘÍ ZAHRÁDKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL ŠPANĚL, Ph.D.

Abstrakt

Práce se zabývá problematikou rozhraní mezi člověkem a počítačem. Zaměřuje se na moderní rozhraní, která využívají projektor a kameru pro rozšíření reálných předmětů o virtuální informace. Součástí práce je také popis kalibrace těchto zařízení. Hlavním cílem této práce je implementace rozšířeného uživatelského rozhraní pro správu aplikačních oken. Systém zahrnuje kameru, projektor a uchopitelné předměty např. papírové desky. Předměty jsou označeny výchozími značkami kvůli jejich sledování ve videosekvenci. Projektor promítá prvky uživatelského rozhraní na stůl a uchopitelné předměty např. aplikační okna na papírové desky. Aplikační okna jsou následně přesouvána a otáčena společně s deskami.

Abstract

This thesis falls into a field of user interface design. It focuses on tangible user interfaces which utilize a camera and projector to augment physical objects with a digital information. It also includes description of calibration of those devices. The primary object of this thesis is the implementation of an augmented user interface for application windows management. The system consists of a stationary camera, overhead projector and movable tangible objects - boards. The boards are equipped with fiducial markers, in order to be tracked in a camera image. The projector displays the conventional desktop onto the table and the tangible objects. For example, application windows can be projected onto some boards, while the windows move and rotate simultaneously with the boards.

Klíčová slova

kalibrace kamery a projektoru, správa aplikačních oken, výchozí značky, sledování v obraze, Tangible user interface, Tabletop display, Studierstube, Ubuntu, Compiz

Keywords

camera-projector calibration, application windows management, markers, tracking, Tangible user interface, Tabletop display, Studierstube, Ubuntu, Compiz

Citace

Jiří Zahrádka: Rozšířené uživatelské rozhraní, diplomová práce, Brno, FIT VUT v Brně, 2011

Rozšířené uživatelské rozhraní

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Michala Španěla, Ph.D a MSc Manuely Waldner z TU Graz. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jiří Zahrádka
20.5.2011

Poděkování

Chtěl bych poděkovat vedoucímu mé práce Ing. Michalu Španělovi, Ph.D za užitečné rady a odbornou pomoc v průběhu řešení celého projektu. Dále bych chtěl poděkovat MSc Manuele Waldner za velmi cennou odbornou pomoc při práci na tomto projektu.

© Jiří Zahrádka, 2011

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.....	2
2 Prvky moderních GUI.....	3
2.1 Horizontální stolní displeje (angl. Tabletop displays).....	3
2.2 Hmatová uživatelská rozhraní (angl. Tangible User Interfaces).....	5
2.3 Uchopitelný displej (angl. Tangible Display).....	7
2.4 Správa aplikačních oken.....	11
3 Kalibrace systému kamera–projektor.....	14
3.1 Přední versus zadní projekce.....	14
3.2 Kalibrace kamery.....	15
3.3 Kalibrace projektoru.....	21
4 Detekce a sledování objektů ve videosekvenci.....	24
4.1 Sledování výchozích značek	24
5 Návrh TUI pro správu aplikačních oken.....	28
5.1 Hardwarová sestava.....	28
5.2 Kalibrace kamery a projektoru.....	29
5.3 Řídící program.....	33
6 Implementace.....	37
6.1 Kalibrační program.....	37
6.2 Řídící program.....	38
7 Výsledky.....	41
7.1 Kalibrace projektoru.....	41
7.2 Řídící program pro TUI	45
7.3 Interakce s TUI.....	51
7.4 Diskuse	52
7.5 Návrh dalšího vývoje projektu.....	53
8 Závěr.....	54
Literatura.....	56
Seznam příloh.....	60
Příloha 1. Manuál - Kalibrační program.....	61
Příloha 2. Manuál - Řídící program pro TUI.....	68
Příloha 3. DVD.....	76

1 Úvod

V současné době je rozhraní mezi počítačem a člověkem nejčastěji realizováno pomocí klávesnice, myši a displeje. Uživatel se musí naučit používat tato zařízení, aby mohl komunikovat s virtuálním světem. Nevýhodou je značně omezené spektrum uživatelských vstupů, které vyžaduje mnoho úsilí pro přenos informací mezi reálným a virtuálním světem. Toto úsilí je možné ušetřit lepší integrací virtuálního světa počítače a fyzického světa uživatele. Jednou z možností integrace nabízí tzv. Tangible user interface (dále jen TUI), což lze volně přeložit do češtiny jako Hmatové uživatelské rozhraní viz. kapitola 2.2.

TUI bylo původně prezentováno jako tzv. Graspable user interface a sloužilo pro ovládání počítačových aplikací. Mezi výhody tohoto přístupu patří například interakce pomocí obou rukou, využití motorických návyků a intuitivní manipulace vstupních zařízení [FIB95]. TUI bývá mnohdy úzce provázáno s Rozšířenou realitou [wik10]. Příkladem jsou fyzické objekty z TUI, které jsou rozšířeny o virtuální informace tím, že je na ně promítán obraz. Pak lze toto TUI nazvat také jako Rozšířené uživatelské rozhraní. Téměř neodlučitelnou součástí TUI je stolní displej (viz. Kapitola 2.1). Toto zařízení je používáno zejména kvůli možnosti snadného umísťování dalších předmětů na jeho plochu. Stolní displej je také vhodný pro týmovou spolupráci a snadné sdílení informací.

Hlavním cílem této práce je implementace rozšířeného uživatelského rozhraní pro správu aplikačních oken. Systém bude zahrnovat kameru, projektor a uchopitelné předměty např. papírové desky. Předměty budou označeny značkami kvůli jejich sledování ve video sekvenci. Díky znalosti rozměrů značek bude možné určit jejich prostorovou pozici z obrazu pouze jedné kamery. Projektor bude promítat prvky uživatelského rozhraní na stůl a uchopitelné předměty. Příkladem může být promítání aplikačních oken na papírové desky. Aplikační okna budou následně přesouvána a otáčena společně s deskami.

Následující kapitola obsahuje popis oblastí a zařízení, které jsou důležité vzhledem k cíli této práce. Součástí jsou také příklady konkrétních aplikací. Třetí kapitola se zabývá kalibrací kamery a projektoru. Konkrétně popisuje transformaci mezi prostorem a obrazem. Další kapitola je věnována sledování značek ve videosekvenci. Pátá kapitola zahrnuje návrh kalibračního programu a programu pro řízení TUI, který je cílem této práce. V následující kapitole je popsána implementace obou programů. Poslední kapitola prezentuje dosažené výsledky.

2 Prvky moderních GUI

Téma práce v sobě kombinuje problematiku z několika oblastí, které spolu více či méně souvisí. Tato kapitola obsahuje popis nejdůležitějších oblastí, kam práce zasahuje. Součástí je také přehled vybraných prací z těchto oblastí.

2.1 Horizontální stolní displeje (angl. Tabletop displays)

Tento typ zobrazování virtuální pracovní plochy se často používá v souvislosti s dotykovým ovládáním, které umožňuje společnou interakci i více uživatelů současně. Zařízení je možné realizovat za použití dotykového displeje, který je převrácen do horizontální polohy a zabudován např. do desky stolu. Další možností realizace je využití dataprojektoru. V tomto případě je obraz promítán na stůl, který slouží jako projekční plocha. Je možné využít tzv. přední nebo zadní projekce, více viz. kapitola 3.1. V tomto případě je uživatelská interaktivita zajištěna pomocí kamery a metod pro rozpoznávání obrazu. Nyní následuje popis vybraných zařízení, které fungují na uvedených principech.

DigitalDesk

DigitalDesk [Wel93] je průkopníkem v oblasti „moderních“ uživatelských rozhraní. Hlavním cílem bylo spojit reálnou a virtuální pracovní plochu. Běžný pracovní stůl byl rozšířen o virtuální prvky (např. kalkulačka), které jsou promítány dataprojektorem na jeho plochu. Projektor byl spolu s kamerou umístěn nad stolem. Pro rozpoznání uživatelských gest byla použita jednoduchá metoda sledování pohybu prstů. Zařízení bylo navrženo pro interakci pouze s jedním uživatelem. Kompletní systém je zachycen na obrázku 2.1.



[obr. 2.1] Prototyp rozhraní Digital Desk. [Wel93]

InteracTable

Tento interaktivní stůl byl navržen, aby umožnil spolupráci a sdílení informací skupině uživatelů [STM99]. Předpokládaná velikost skupiny je 2 až 6 lidí. Virtuální plocha je promítána pomocí zadní projekce na průsvitnou desku stolu. Projektor je zabudován přímo ve stole. Horní deska stolu slouží jako dotyková obrazovka. Uživatelé tak mohou interagovat se zařízením, aniž by si zastínili promítaný obraz. Stůl byl navržen jako součást uživatelského prostředí i-LAND, které kombinuje prvky reálného a virtuálního světa. Ukázka stolu je na obrázku 2.2.



[obr. 2.2] InteracTable. [STM99]

2.2 Hmatová uživatelská rozhraní (angl. Tangible User Interfaces)

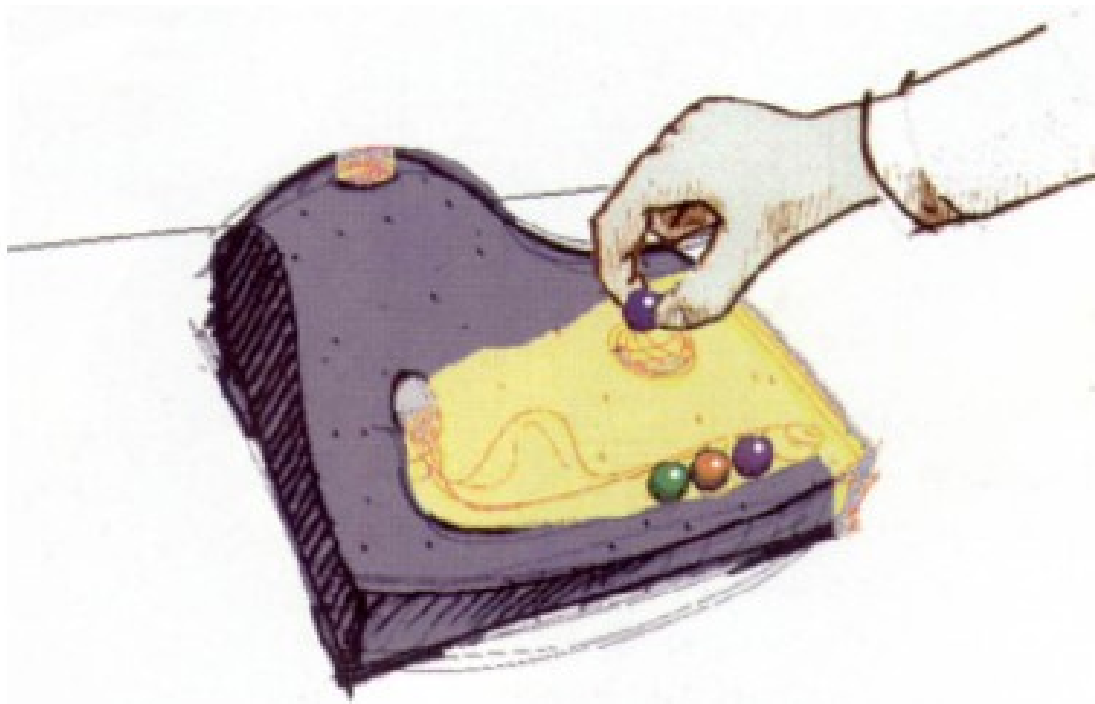
Tzv. Tangible User Interface (dále jen TUI) lze přeložit jako uchopitelné nebo hmatové uživatelské rozhraní. Dříve se tato uživatelská rozhraní nazývala Graspable User Interface. Je to rozhraní mezi člověkem a počítačem, které dovoluje interakci s digitálními informacemi pomocí fyzického prostředí [wik01]. Uživatel tak může zasahovat do virtuálního světa manipulací běžných předmětů z reálného světa. Příkladem takových předmětů může být dřevěná kostka, list papíru nebo psací pero. Těmito předměty však nejsou zařízení, které jsou vytvořena primárně za účelem ovládání počítače, jako je např. počítačová klávesnice nebo myš. Avšak využití těchto zařízení v kombinaci s TUI není vyloučeno.

Charakteristiky TUI [wik01][FIB95] :

1. Fyzický vzhled předmětu doplňuje digitální informaci ,se kterou je vázán.
2. Vzhled předmětu udává způsob použití pro ovládání systému.
3. Aktuální fyzický stav předmětu reprezentuje digitální stav virtuálního systému.

Hlavní výhodou TUI je vytvoření užší vazby mezi virtuálním a reálným světem. Interakce s počítačem se tak stává jednodušší, přirozenější a efektivnější, protože uživatel může aplikovat intuitivní motorické návyky. Mezi výhody patří také fakt, že seznámení neškoleného uživatele s principem ovládání je velmi rychlé.

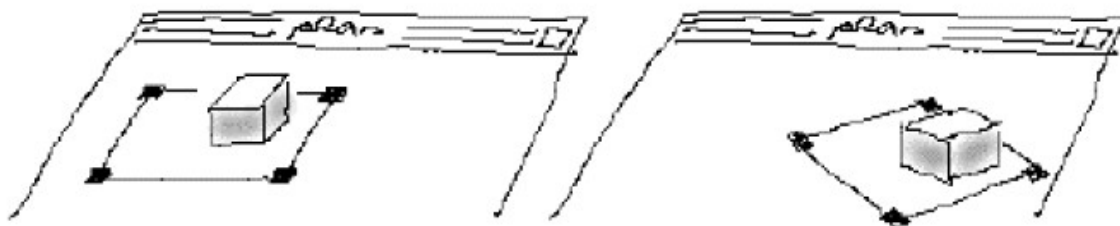
Pravděpodobně prvním systémem s TUI byl prototyp telefonního záznamníku, který vytvořil Durrel Bishop [Bis92]. Záznamník byl ovládán pomocí kuliček, které reprezentovaly přijaté hlasové zprávy. Uživatel tak mohl vložit kuličku do záznamníku pro přehrání příchozí zprávy. Dále mohla být kulička položena na speciální telefonní přístroj pro automatické vytočení čísla volajícího, který vzkaz zanechal. Záznamník je zobrazen na obrázku 2.3 .



[obr. 2.3] Kuličkový záznamník – Marble Answering Machine [Bis92]

Bricks: Graspable User Interfaces

V této práci [FIB95] byl prezentován koncept TUI, který dává možnost manipulovat virtuální objekty pomocí malých bloků/cihel tzv. „bricks“. Virtuálními objekty mohou být například prvky grafického uživatelského rozhraní. Systém zahrnuje horizontální dotykový displej ActiveDesk, na který se bloky pokládají, a tím se napojují na virtuální objekty. To dává možnost virtuální objekty fyzicky „uchopit“. Systém podporuje paralelní manipulaci s více bloky současně a uživatel tak může použít obě ruce. Názorný příklad přesunu a otočení virtuálního objektu manipulací bloku je na obrázku 2.4 .



[obr. 2.4] Posun a rotace virtuálního objektu manipulací fyzického bloku. [FIB95]

BUILD-IT

BUILD-IT [RFK98] je nástroj pro podporu konstruování a tvorbu návrhů ve stavebnictví. Tento nástroj pracuje s malými bloky na podobném principu jako Bricks, ačkoliv byl vytvořen nezávisle. Základní myšlenka je následující. Uživatelé sedí okolo stolu, na který je promítána virtuální stavební plocha (obrázek 2.5). Mají k dispozici bloky, které pokládají na stůl, a tím manipulují s virtuálními objekty. Objekty mohou označovat, mazat a libovolně přesouvat. Těmito akcemi tvoří výsledný stavební návrh. Paralelně může pracovat více uživatelů na jednom zařízení. Poloha a orientace bloků je sledována kamerou, která je umístěna nad stolem.



[obr. 2.5] BUILD-IT, Návrhový stůl se zobrazeným půdorysem. Na zdi je perspektivní pohled na scénu. [RFK98]

2.3 Uchopitelný displej (angl. Tangible Display)

Tzv. Tangible Display (dále jen TD) lze přeložit do češtiny jako uchopitelný respektive pohyblivý displej. Jedná se o obrazovku malých rozměrů, která je předurčena pro častou manipulaci. Obraz,

který se na ni vykresluje, se mění podle její aktuální polohy a rotace. Uživatel tak může měnit zobrazovaný obsah přesouváním a otáčením displeje bez nutnosti použití dalších vstupních zařízení.

V praxi se TD realizuje buď jako aktivní např. LCD displej nebo jako projekční plocha, na kterou je obraz promítán. V případě aktivního displeje se jedná o složité zařízení, které kromě zobrazovací jednotky zahrnuje také systém pro určení vlastní polohy a komunikaci s počítačem.

Pokud je TD realizován pomocí projekční plochy, musí být současně použita kamera a projektor. Jako projekční plocha bývá nejčastěji užívána deska např. z papíru nebo dřeva. Deska je označena značkami, aby mohla být sledována. Scéna je pak snímána běžnou nebo infračervenou kamerou. Ze snímků kamery se vyhodnotí poloha a orientace sledované desky. Tyto výsledky určí, kam je obraz promítnut. TD se často používá v kombinaci s horizontálně položeným displejem. S výhodou to lze využít v následujících aplikacích [IU97],[SST09]:

- Zobrazení prostorových dat. Uživatel pohybuje virtuální kamerou a tím mění zobrazovaný kontext.
- Virtuální lupa. Vzdálenost TD od horizontálního displeje určuje velikost zobrazovaného detailu.
- Zobrazení pohledu do reálného světa rozšířeného o virtuální objekty tzv. rozšířená realita.

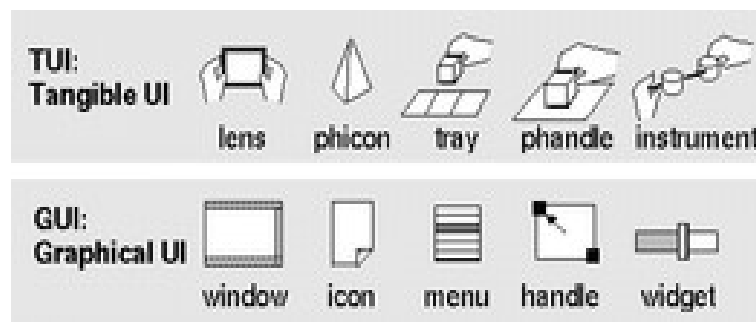
Nyní následují příklady konkrétních systémů, kde se TD prakticky využívá.

metaDESK

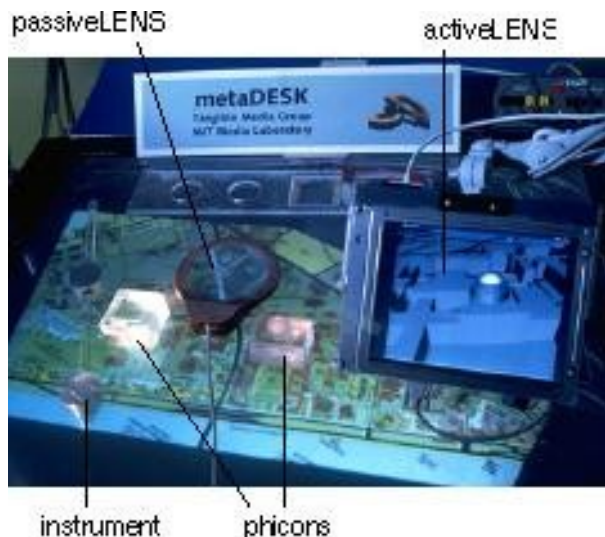
Platforma byla prezentována v [IU97]. Koncept návrhu platformy je na obrázku 2.6c. TUI se v tomto případě skládá ze zařízení:

- Horizontálně položené projekční plochy. Obraz je promítán zadní projekcí.
- „ActiveLens“ - LCD TD, který je připevněn na pohyblivém ramenu. Zprostředkovává perspektivní pohled na scénu.
- „PassiveLens“ – opticky transparentní „čočka“. Podle umístění na projekční ploše je na toto místo promítán opticky modifikovaný (např. zvětšený nebo zmenšený) obraz.
- „phicons“ - Fyzická realizace počítačových ikon.
- Nástroje pro manipulaci virtuálních objektů viz. obrázek 2.6a.

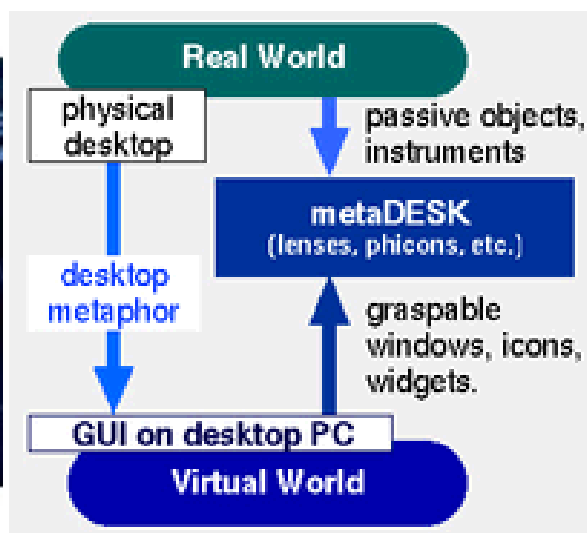
Fyzické objekty a nástroje jsou detekovány na povrchu projekční plochy díky síti optických, mechanických a elektromagnetických senzorů. Příkladem aplikace, která funguje na této platformě, je Tangible Geospace. Aplikace slouží k procházení a modifikaci map. Obrázek 2.6b zachycuje tuto aplikaci na platformě metaDESK.



a) Párování objektů z hmatového(TUI) a grafického(GUI) uživatelského rozhraní.



b) Tangible Geospace



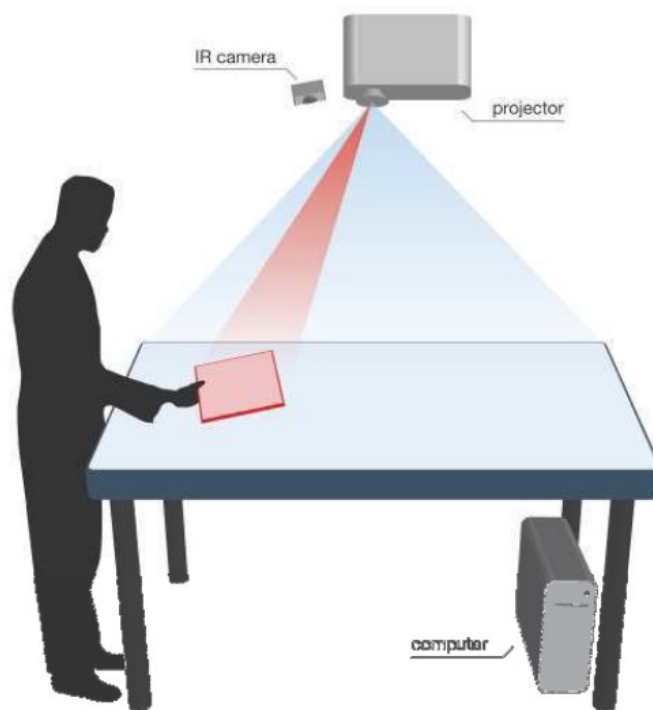
c) Koncept

[obr. 2.6] Ukázka TUI metaDesk [IU97]

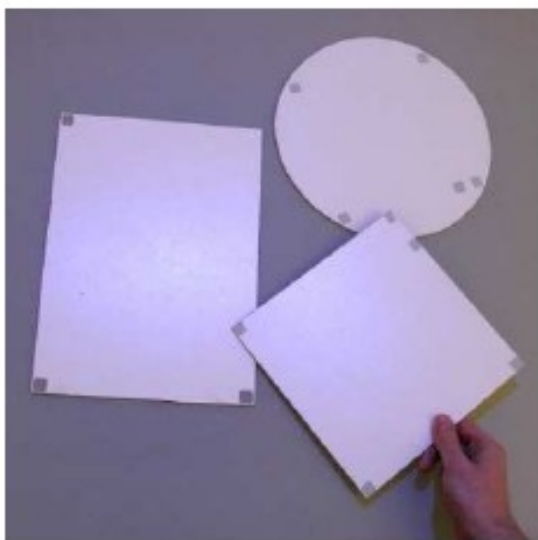
PaperLens

Systém je navržen pro prohlížení prostorových a vrstvených informačních prostorů [SST09]. Rozvržení systému je na obrázku 2.7a. Základní funkcí je sledování tzv. „papírových čoček“ ve video sekvenci a následné promítání obrazu na jejich plochu. Promítaný obraz přímo závisí na aktuální poloze čočky.

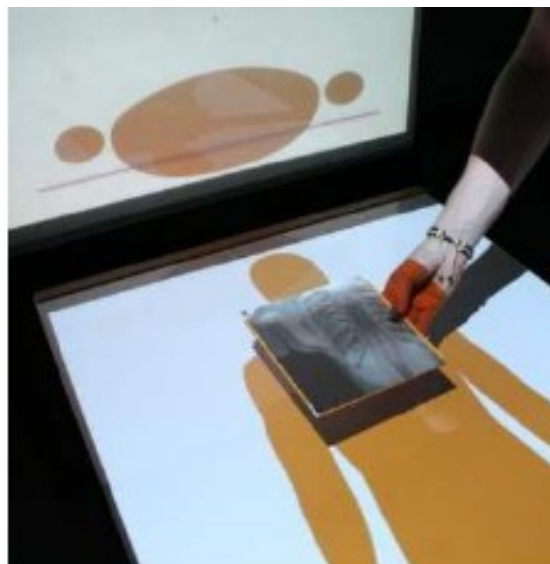
Papírová čočky jsou tvořeny z kartonů, které jsou navíc opatřeny pěti značkami odrážejícími infračervené světlo. Uvedený počet byl zvolen proto, aby bylo možné určit prostorovou orientaci obrazu. Velikost značek je 5x5 mm. Čočky mohou být různých tvarů viz. obrázek 2.7b. Pro sledování čoček v obraze byla použita knihovna ARToolKit. Fyzický prostor nad stolem byl emulován knihovnou OpenGL tak, aby se kompenzovaly chyby projekce. Maximální úhel mezi rovinou čočky a projekční plochy je omezen na 45 stupňů. Při větším náklonu dojde k přílišnému zdeformování promítaného obrazu.



a) Rozvržení systému PaperLens.



b) Papírové čočky.



c) Vizualizace objemových dat.

[obr. 2.7] Ukázka TUI PaperLens. [SST09]

2.4 Správa aplikačních oken

Většina z dosud vytvořených TUI je navržena jako vrstva nad operačním systémem. Všechny události jsou tak zpracovávány nezávisle na OS. Tento přístup s sebou nese problém v podobě nutnosti implementace platformě závislých aplikací. Pro každou platformu tedy musí být implementována vlastní sada aplikací, kterou pak nelze použít na jiném TUI. Uvedenému problému se lze vyhnout integrací subsystému pro zpracování událostí TUI do operačního systému. Uživatelské akce z TUI jsou pak převáděny na vstupní události operačního systému[ChISR09]. Následně je tedy možné přes TUI ovládat aplikace, které běží nad vybraným operačním systémem. Jedním ze základních programů, které lze takto ovládat, je správce aplikačních oken.

Příklad takové architektury můžeme nalézt v [ChISR09]. Ačkoliv prezentovaný systém nelze přímo zařadit do TUI, jedná se o velmi dobrou ukázkou napojení multidotykového rozhraní na správce oken OS Linux. Systém je vytvořen jako zásuvný modul programu Compiz, který je správcem aplikačních oken pod uvedeným operačním systémem. Výsledná víceuživatelská aplikace tak umožňuje interakci s OS Linux pomocí multidotykového displeje (obr. 2.8) .



[obr. 2.8] Víceuživatelská interakce na OS Linux. [ChISR09]

PaperWindows

Jedná se o prototyp TUI pro manipulaci aplikačních oken [HVA+05]. Okna jsou promítána na listy papíru, které se pohybují v rámci světelného kuželu projektoru. Uživatel může dotykem prstu simulovat kliknutí myši. Prst a papíry jsou opatřeny značkami, které odrážejí infračervené světlo. Systém dále obsahuje infračervené kamery pro sledování jejich aktuální polohy. Projektor je umístěn zhruba 1,5 m nad stolem. Umístění bylo zvoleno tak, aby bylo dosaženo přijatelného rozlišení promítaného obrazu a zároveň, aby byla oblast pohybu sledovaných papírů dostatečně velká.

Prototyp TUI dovoluje provádět s okny a jejich obsahem následující operace:

- Aktivace, výběr a deaktivace aplikačního okna.
- Kopírování a vkládání textu.
- Rolovat obsah okna.
- Vyvolávat aplikačně specifické akce pomocí naklánění nebo rotace listů papíru.
- Zobrazit stejné aplikační okno na dva různé listy papíru a paralelně měnit jeho obsah.
- Přidávat poznámky pomocí speciálního pera s infračervenými značkami.



[obr. 2.9] PaperWindows, správa aplikačních oken. [HVA+05]



[obr. 2.10] Vytváření koláže za použití PaperWindows. [HVA+05]

3 Kalibrace systému kamera–projektor

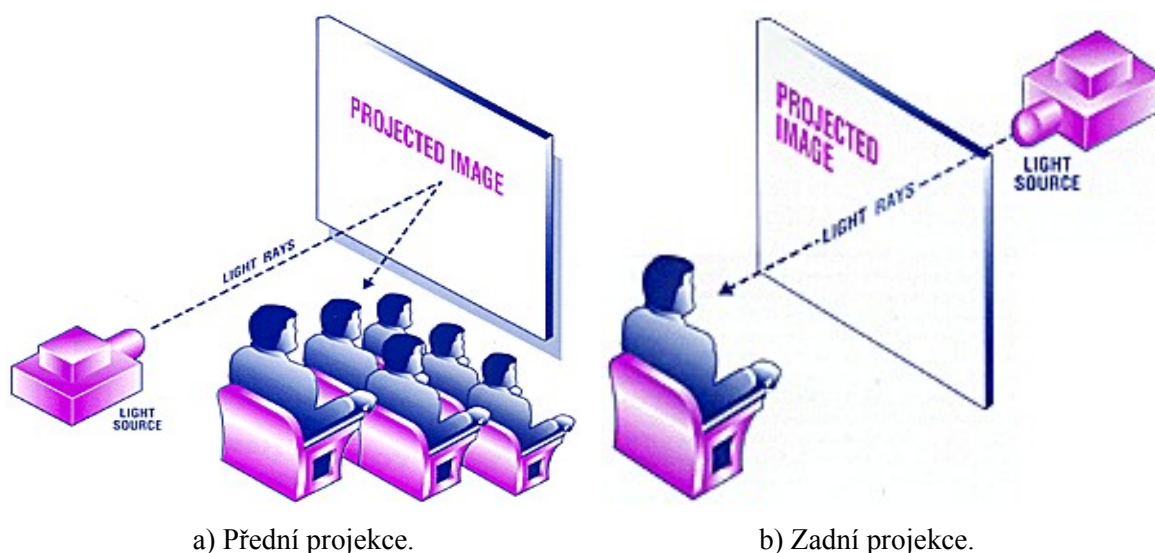
Z ukázek, které byly uvedeny v minulé kapitole, je patrné, že systém kamery a projektoru je častou součástí TUI. Použití tohoto přístupu s sebou nese řadu výhod. V závislosti na softwarovém zpracování může systém sledovat a reagovat na téměř libovolné předměty. Následné přidání interaktivity s dalšími objekty je pak otázkou softwarové implementace. Také lze použít jeden nebo více uchopitelných displejů v kombinaci s přední projekcí. Pokud je celý systém správně nakalibrován, je možné určit 3D polohu rozpoznaného objektu známé velikosti vzhledem ke kameře.

Jak název napovídá, systém v sobě zahrnuje projektor a kameru. Přitom není vyloučeno společné použití více takových zařízení v jednom systému. Ukázka tohoto systému je na obrázku 2.7a. Nejčastěji se zde můžeme setkat s kamerami, které pracují ve spektru infračerveného nebo viditelného světla. Infra-kamery se používají v kombinaci se speciálními značkami, které odrážejí infračervené světlo. Tyto značky jsou v pořízených snímcích velmi dobře rozpoznatelné. V systému se někdy používá více projektorů, zejména kvůli dosažení vyššího rozlišení promítaného obrazu.

3.1 Přední versus zadní projekce

Typ projekce se liší v závislosti na umístění projektoru. Projekce je označována jako přední pokud je projektor umístěn na stejné straně promítacího plátna jako pozorovatel (obr. 3.1a). O zadní projekci se mluví v případě, že projektor je na opačné straně plátna než pozorovatel (obr. 3.1b). Pro většinu výhod přední projekce platí, že jsou nevýhodami zadní projekce a naopak.

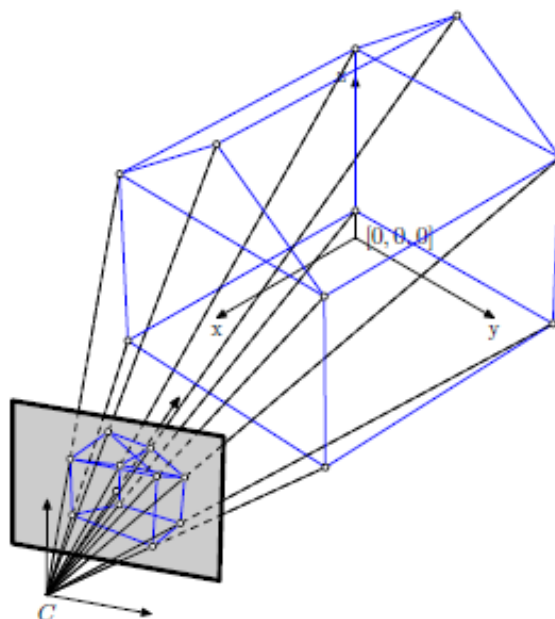
Mezi hlavní výhody přední projekce patří možnost promítání částí obrazu na objekty před plátnem. Toho se využívá právě při tvorbě TUI. Příkladem konkrétní aplikace jsou uchopitelné displeje viz. kapitola 2.3. Nevýhodou tohoto přístupu je, že si uživatel může sám stínit promítaný obraz. Tento problém se zřejmě nevyskytuje u zadní projekce, proto se často používají obě projekce společně.



[obr. 3.1] Ukázka typů projekce. [Ros01]

3.2 Kalibrace kamery

Správná kalibrace kamery dává možnost vypočítat prostorovou pozici pozorovaného objektu známé velikosti vzhledem ke kameře. Kamera provádí mapování mezi 3D světovým metrickým systémem a 2D souřadným systémem obrazu v pixelech [Svo09] (viz. obrázek 3.2). Aby bylo možné provést inverzní proces, tedy spočítat 3D metrické souřadnice z 2D pixelů, je nutné znát prvky interní a externí orientace kamery (dále jen interní resp. externí parametry). Tyto parametry jsou výsledkem kalibračního procesu.



[obr. 3.2] Zobrazení světových bodů do obrazu kamery v pixelech. [Svo09]

Interní parametry

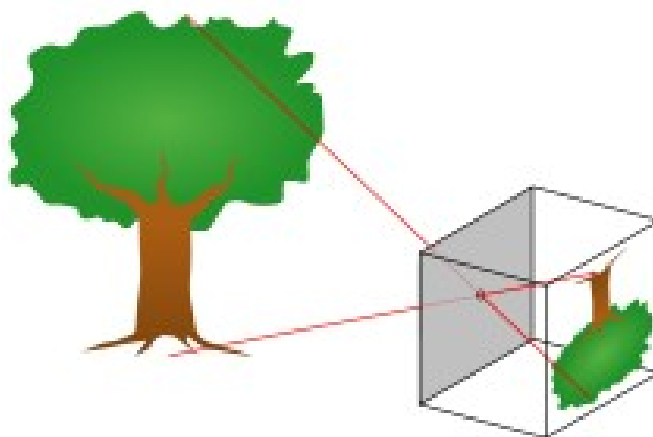
Tyto parametry určují vlastnosti použité optické soustavy kamery, které ovlivňují výsledný pořízený snímek. Parametry slouží k rekonstrukci paprsků středového promítání, transformaci z metrických souřadnic do souřadnic v pixelech a odstranění geometrického zkreslení obrazu. Mezi interní parametry patří [Han10] :

- ohnisková vzdálenost
- poloha hlavního snímkového bodu
- koeficienty radiálního a tangenciálního zkreslení

Dále bude popsáno použití jednotlivých parametrů v procesu pořizování snímku.

Středové promítání

Následující popis vychází z jednoduchého modelu dírkové kamery (angl. pinhole camera) viz. obrázek 3.4 a [Wik02],[Dur09].



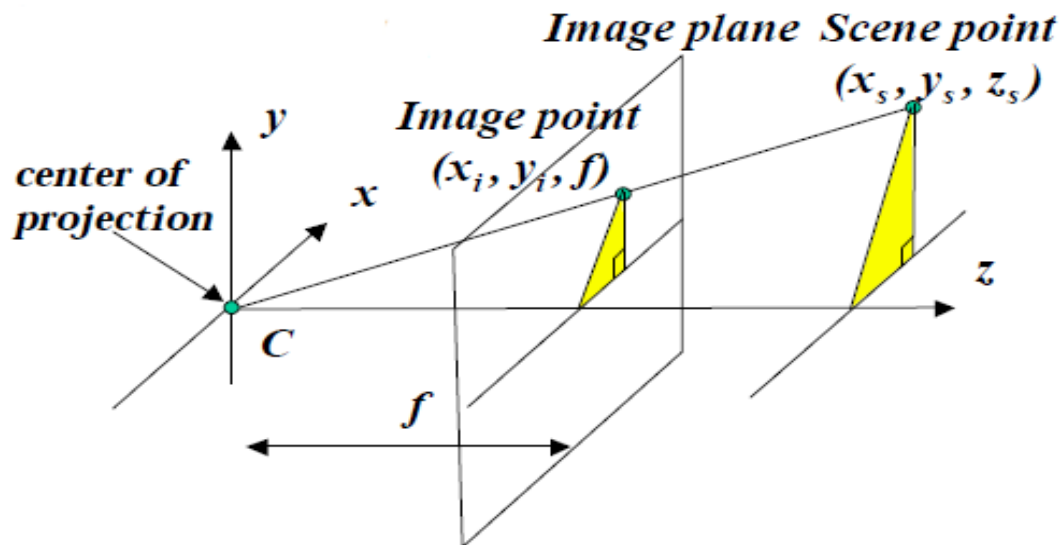
[obr. 3.4] Princip dírkové kamery. [Wik02]

Popis se vztahuje k obrázku 3.5. Střed promítání se nachází ve středu kamery. Tento bod se označuje písmenem C a je výchozím bodem Euklidovského souřadného systému. Pokud jsou obrazové (Image) a světové (World / Scene) body reprezentovány pomocí homogenních vektorů, středové promítání je lineární transformací bodů z prostoru na obrazovou rovinu. Souřadnice obrazového bodu $[x_i, y_i, f]$ lze tedy vypočítat ze souřadnic světového bodu $[x_s, y_s, z_s]$ podle vzorců 3.1 a 3.2. Maticový zápis transformace uvádí vzorec 3.3. Písmeno f zde vyjadřuje ohniskovou vzdálenost, to je vzdálenost mezi středem promítání a obrazovou/ohniskovou rovinou. U digitálních kamer je to vzdálenost mezi ohniskem čočky objektivu a snímacím čipem.

$$x_i = f * \frac{x_s}{z_s} \quad (3.1)$$

$$y_i = f * \frac{y_s}{z_s} \quad (3.2)$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \quad \text{kde} \quad x_i = \frac{u}{w} \quad \text{a} \quad y_i = \frac{v}{w} \quad (3.3)$$



[obr. 3.5] Lineární transformace bodů do obrazové roviny podle paprsků, které procházejí středem promítání a obrazovou rovinou. [Dur09]

Transformace z metrických do pixelových souřadnic

Dochází k převodu souřadnic obrazových bodů z metrického souřadného systému do pixelového.

Interní parametry, které se k tomu používají jsou:

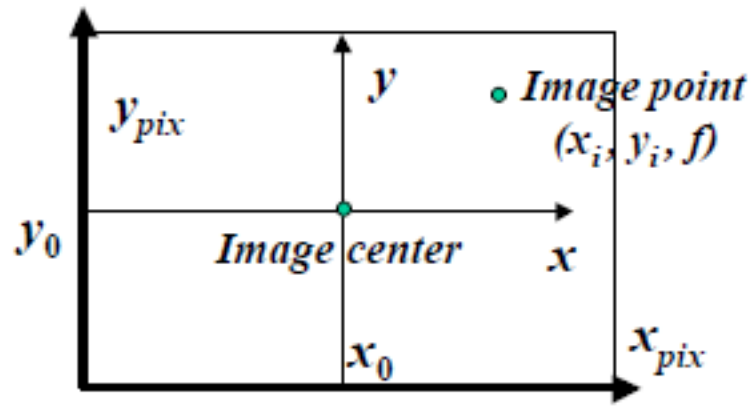
- souřadnice hlavního snímkového bodu kamery
 - x_0 - souřadnice na ose x
 - y_0 - souřadnice na ose y
- míra zvětšení/zmenšení – Vyjadřuje počet pixelů na jednotku ohniskové vzdálenosti v obrazových souřadnicích. Používá se v souvislosti s digitálními kamerami, kde se udávají rozměry obrazu v pixelech s různým měřítkem pro osu x a y, protože pixely snímacího senzoru nemusejí být vždy přesně čtvercové [HZ04].
 - m_x - Počet pixelů na délkovou jednotku na ose x.
 - m_y - Počet pixelů na délkovou jednotku na ose y.

- úhel mezi osou x a y snímacího senzoru – Vyjadřuje se pomocí koeficientu s , pro většinu kamer je nulový [HZ04].

Obrazové souřadnice $[x_i, y_i, f]$ jsou přepočítány na souřadnice v pixelech $[x_{pix}, y_{pix}]$ podle vzorců 3.4 a 3.5. Souřadnice jsou znázorněny na obrázku 3.6.

$$x_{pix} = m_x * x_i + x_0 \quad \text{kde} \quad x_i = f * \frac{x_s}{z_s} \quad (3.4)$$

$$y_{pix} = m_y * y_i + y_0 \quad \text{kde} \quad y_i = f * \frac{y_s}{z_s} \quad (3.5)$$



[obr. 3.6] Převod mezi metrickým a pixelovým systémem. [Dur09]

Celkově lze maticově zapsat zobrazení bodu X umístěného ve scéně, který má metrické 3D souřadnice vztažené k bodu C, do obrazové roviny a souřadnic v pixelech podle vztahu 3.6. Kde K vyjadřuje matici interních parametrů kamery.

$$u \simeq K X \Rightarrow \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} a_x & s & x_0 & 0 \\ 0 & a_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \quad \text{kde} \quad a_x = f * m_x \quad \text{a} \quad a_y = f * m_y \quad (3.6)$$

$$\text{pak} \quad x_{pix} = \frac{u'}{w'} \quad \text{a} \quad y_{pix} = \frac{v'}{w'} \quad (3.7)$$

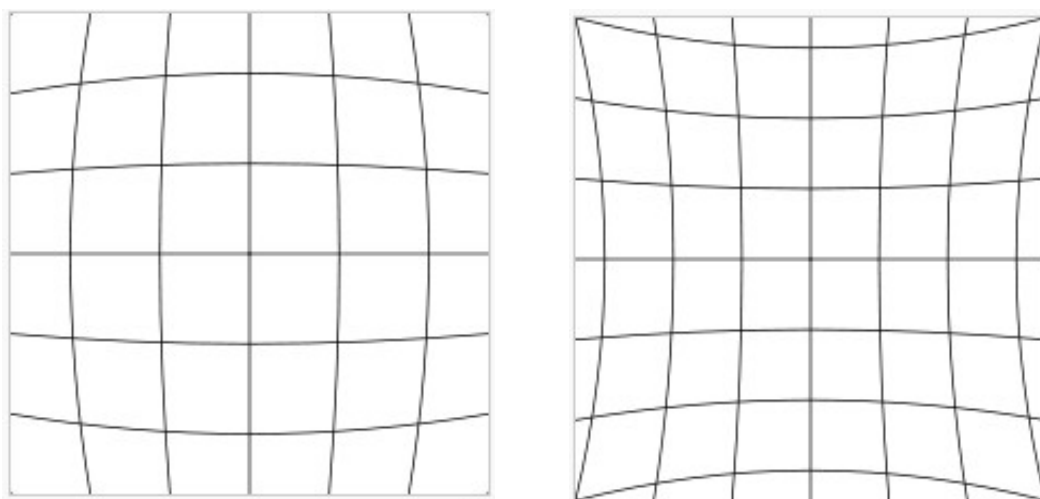
Zkreslení obrazu

Při průchodu světla ideální čočkou nedochází k žádnému zkreslení obrazu. Bohužel při použití reálných čoček je obraz více či méně zkreslen. To se projevuje tím, že se přímky ve scéně zobrazují jako křivky. Největší podíl na zkreslení má tzv. radiální zkreslení, v menší míře se na něm podílí tzv. tangenciální zkreslení.

Tangenciální zkreslení je způsobeno nedokonalým vycentrováním plochy čočky [Kra06]. Jeho koeficienty jsou označovány jako p_1 a p_2 .

Radiální zkreslení způsobuje radiální posunutí bodů v obrazové rovině. Velikost posunutí stoupá se vzdáleností od středu snímku – hlavního snímkového bodu. Příklad tzv. polštářkového a soudkovitého radiálního zkreslení je na obrázku 3.7. Koeficienty tohoto zkreslení jsou

$k_1, k_2, k_3, \dots$. Výsledkem kalibrace bývají pouze koeficienty s indexem 1 až 3. Zkreslení obrazu je možné softwarově odstranit pokud známe uvedené koeficienty [wik03].



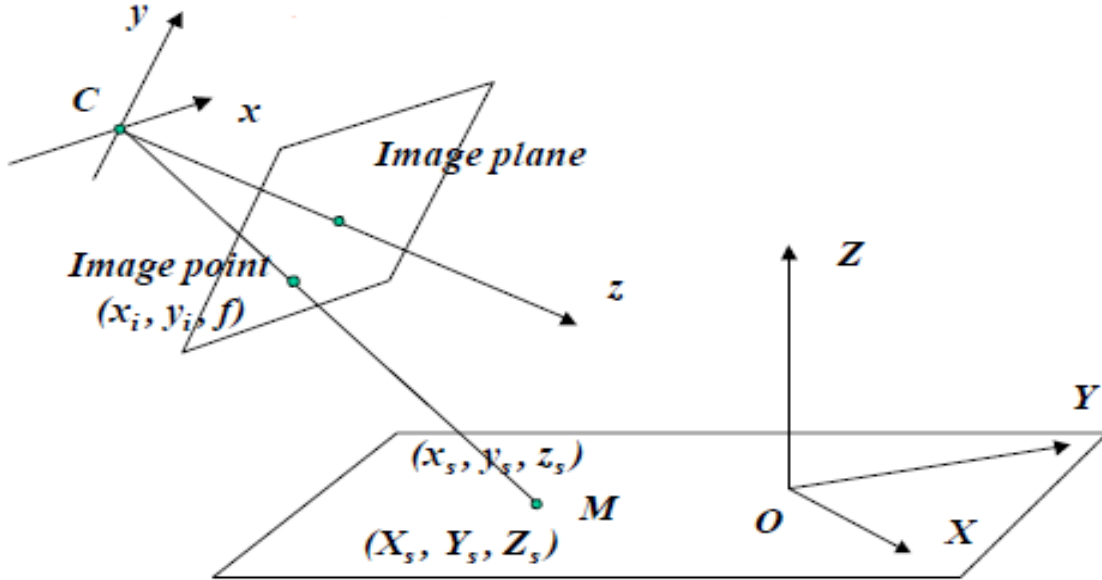
[obr. 3.7] Vlevo polštářkové, vpravo soudkovité radiální zkreslení čtvercové mřížky. [wik03]

Externí parametry

Tyto parametry slouží k určení polohy a natočení kamery v prostoru vzhledem k nějakému bodu. Parametry konkrétně vyjadřují Euklidovskou transformaci mezi souřadným systémem kamery a světovým souřadným systémem proti, kterému je kamera kalibrována. Na obrázku 3.8 mají tyto systémy výchozí body C a O. Transformace se skládá z rotace R a posunutí t a je vyjádřena pomocí matice v rovnici 3.8. Vztah vyjadřuje transformaci bodu M ze světového do kamerového systému. Kvůli maticovému násobení jsou body vyjádřeny v homogenních souřadnicích. R a t jsou výsledky kalibračního procesu.

$$X \simeq [R \ t] M \Rightarrow \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix} \quad \text{Kde } [x_s, y_s, z_s] \text{ resp. } [X_s, Y_s, Z_s] \text{ vyjadřují}$$

souřadnice téhož bodu v kamerovém resp. světovém souřadném systému. (3.8)



[obr. 3.8] Transformace ze světových souřadnic do obrazové roviny. [Dur09]

Celkově lze zapsat transformaci bodu scény, který je zadán ve světových souřadnicích, do obrazové roviny a pixelových souřadnic spojením vzorců 3.7 a 3.8 následovně podle 3.9.

$$u \simeq K [R \ t] M \Rightarrow \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} a_x & s & x_0 & 0 \\ 0 & a_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix} \quad (3.9)$$

Výsledná 3x4 projekční matice P vznikne následovně:

$$P = K [R \ t] \Rightarrow u \simeq P X_o \quad (3.10)$$

Popis kalibrace

Úplná kalibrace znamená výpočet externích R, t a interních K parametrů a také koeficientů zkreslení k, p . Kameru je třeba kalibrovat pokud:

- Nebyla nikdy dříve kalibrována.

- Její optická soustava byla modifikována.
- Výsledky kalibrace nejsou dostačující.

Postup kalibrace je možné rozdělit do dvou fází [Ste06]:

1. Snímkování

Do této fáze patří volba kalibračního vzoru (dále jen vzor) vzhledem ke vzdálenosti snímkování a použitému objektivu. Rozměry vzoru musejí být známy pro následné zpracování. Během snímkování nesmí být manipulováno zaostření kamery. Je nutné pořídit snímky s různým natočením kamery nebo vzoru pro fixaci hlavního snímkového bodu.

2. Zpracování

V této fázi jsou nejdříve předzpracovány pořízené snímky tzn., že jsou např. označeny kalibrační body v obraze kamery. Dále jsou vypočteny samotné kalibrační parametry. Vstupem výpočtu je množina dvojic 2D a 3D souřadnic kalibračních bodů. 2D souřadnice jsou vyjádřeny v rámci obrazové roviny v pixelech. 3D souřadnice bodů jsou určeny ze známé geometrie kalibračního vzoru. Podrobnější informace o metodách výpočtu parametrů lze nalézt v [RF06].

3.3 Kalibrace projektoru

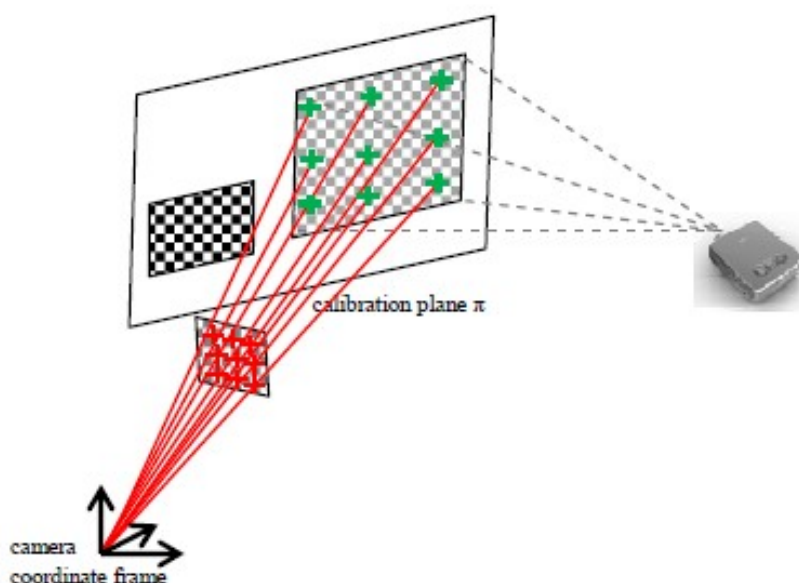
Projektor je třeba kalibrovat, aby bylo možné určit jeho polohu vzhledem ke kameře. Díky tomu lze následně určit polohu objektů ve scéně pozorované kamerou vzhledem k projektoru. Na základě těchto informací může být modifikován promítaný obraz.

V systému kamera-projektor lze teoreticky považovat projektor za druhou kameru. V praxi jsou dva hlavní důvody, které však činí kalibraci projektoru komplikovanější než kalibraci kamery [FHM09].

1. Projektor nemůže snímat povrch, na který promítá obraz. Získání korespondence mezi obrazovými/promítanými 2D body a osvětlenými 3D body není možné bez použití kamery. Správná kalibrace projektoru je tedy možná pouze s kvalitně kalibrovanou kamerou.
2. Získání světových metrických souřadnic bodů promítaného kalibračního vzoru je obtížné, protože pole není vázáno k žádnému světovému souřadnému systému.

Postup kalibrace projektoru je v principu téměř stejný jako v případě kamery. Je zřejmé, že se liší ve fázi Snímkování. V této fázi se provádí získávání obrazových 2D souřadnic kalibračních bodů a jím odpovídajících 3D souřadnic ve scéně za pomoci obrazu z kamery. Po získání vstupních dat lze vypočítat kalibrační parametry stejnou metodou jako v případě kalibrace kamery. Výsledkem kalibrace projektoru jsou interní/externí parametry a koeficienty zkreslení, stejně jako v případě kamery.

Jednou z publikací, kde je prezentována kalibrace systému kamera-projektor je [FHM09]. Kalibrační vzor je zde promítán na plátno vedle reálného vzoru. Externí parametry kamery jsou vypočteny podle rozpoznání reálného vzoru. Podle těchto parametrů je určena rovnice projekční roviny. Ze snímku pořízeného kamerou jsou spočteny vektory paprsků jdoucích ze středu kamery do vnitřních rohů promítaného vzoru. Výpočet je proveden násobením 2D bodů obrazu inverzní projekční maticí kamery. Výsledné 3D pozice vnitřních rohů promítaného vzoru jsou určeny jako průnik vržených paprsků s projekční rovinou viz. obrázek 3.9.



[obr. 3.9] Průnik paprsků s rovinou určuje 3D pozice vnitřních rohů promítaného vzoru. [FHM09]

Zjednodušená kalibrace

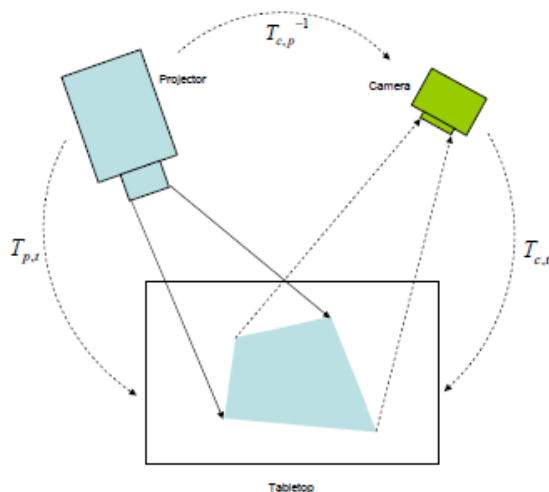
Pokud není v systému kamera-projektor vyžadována znalost prostorového umístění projektoru vzhledem ke kameře, je možné zjednodušit proces kalibrace na získání homografie H mezi obrazovými rovinami kamery a projektoru. H je matice, která vyjadřuje projektivní transformaci bodů mezi rovinami. Díky ní je možné vypočítat souřadnice nějakého bodu v obraze projektoru ze souřadnic téhož bodu v obraze kamery viz. vzorec 4 [wik04]. Následně tedy není problém určit pozici sledovaného objektu v rámci obrazové roviny projektoru. Plocha kam je obraz promítán musí být zcela rovná. Tento přístup byl prezentován např. v [SWGZ07] viz. obrázek [obr. 3.10] .

Mějme $p_a = \begin{bmatrix} x_a \\ y_a \\ 1 \end{bmatrix}$, $p_b' = \begin{bmatrix} x_b * w' \\ y_b * w' \\ w' \end{bmatrix}$, $H_{ab} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix}$ pak $p_b' = H_{ab} * p_a$ kde $H_{ba} = H_{ab}^{-1}$

dále $p_b = \frac{p_b'}{w'} = \begin{bmatrix} x_b \\ y_b \\ 1 \end{bmatrix}$, p_a a p_b jsou body v rovinách a respektive b zadané v

homogenních souřadnicích kvůli maticovému násobení (4)

Pro výpočet homografie existují různé metody. Vstupem pro tento výpočet je množina bodů. Přičemž pro každý bod jsou zadány jeho souřadnice v rámci první a druhé roviny. Více informací viz. [Ben08].



[obr. 3.10] Homografie mezi obrazem kamery a projektoru. [SWGZ07]

4 Detekce a sledování objektů ve videosekvenci

Oblast detekce a sledování objektů je velmi rozsáhlá. V této kapitole jsou popsány pouze metody pro sledování značek, které se bezprostředně týkají tématu této práce.

4.1 Sledování výchozích značek

Výchozími značkami jsou myšleny černobílé značky předurčené pro rozpoznávání pomocí metod počítačového vidění. Jsou vytvořeny tak, aby byli snadno a rychle rozpoznatelné. Při sledování se využívá jejich apriorní znalosti. Pokud jsou známy rozměry značky, lze určit její polohu a orientaci vzhledem ke kameře. Proto se sledování značek často používá v souvislosti s rozšířenou realitou.



[obr. 4.1] Ukázka výchozí značky.

ARToolKit

ARToolKit [Kat00] je programová knihovna určená pro sledování známých značek v obraze. Navíc umožňuje rozšířit snímáný obraz přidáním virtuálních objektů. Nabízí nástroje pro určení pozice a orientace kamery vzhledem k rozpoznávaným značkám. Řeší klíčové problémy rozšířené reality jako sledování pohledové kamery a interakce s virtuálními objekty. Uvedená knihovna patří mezi hlavní zástupce ve své oblasti. Na jejím základě bylo vytvořeno mnoho dalších nástrojů viz. [wik05].

Další vlastnosti podle [Kat00]:

- Jednoduchý nástroj pro tvorbu aplikací rozšířené reality běžících v reálném čase.
- Obsahuje jednoduchou kalibraci kamery.
- Možnost použití čtvercových značek.
- Zdrojové kódy jsou volně přístupné.
- Detekce více značek současně.



[4.2] Příklad výchozí značky, který lze rozpoznat pomocí ARToolKit. [Kat00]

Studierstube tracker

Studierstube tracker [Sch07] (dále jen Studierstube) je knihovna počítačového vidění určená pro detekci a určení prostorové pozice výchozích značek. Je následníkem ARToolKitPlus, což je rozšíření ARToolKit. Konceptuálně je těmto knihovnám velmi podobná, ale implementačně se zcela liší. Studierstube byla vytvořena pro dosažení nejvyššího výkonu na osobních počítačích a mobilních telefonech. Mezi její hlavní vlastnosti kromě již zmíněných patří:

- Detekce velkého množství značek různých typů v jednom snímku současně.
- Robustní detekce značek za nevyvážených světelných podmínek. Obrázek 4.3.
- Velmi kvalitní určení prostorové polohy a orientace značek.
- Nezávislost na externích knihovnách.
- Nízká systémová náročnost.



[obr. 4.3] Značka v extrémně nevyvážených světelných podmínkách. [Sch07]

Sekvence zpracování obrazu touto knihovnou je následující [Sch07]:

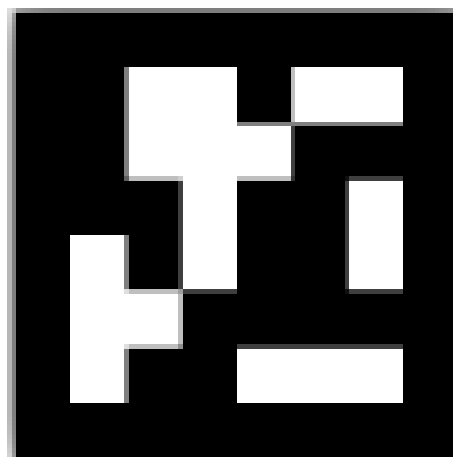
1. Automatické adaptivní prahování obrazu z kamery.
2. Detekce čtverců nebo obdélníků v obraze.
3. Detekce značek.
4. Filtrace pozic rohů značek.
5. Určení prostorové polohy a orientace značek vzhledem ke kameře.

Typy sledovaných značek (viz. obrázek 4.4)

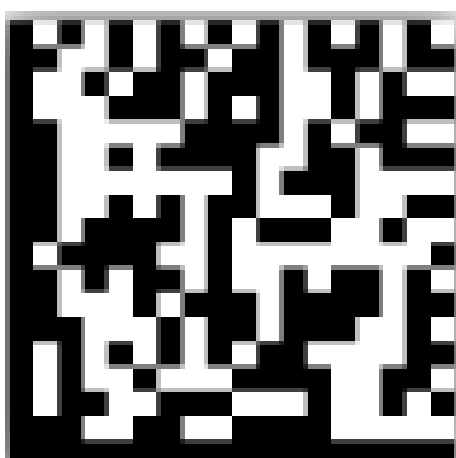
- Template marker – Jedná se o stejné značky, které umí rozpoznávat knihovna ARToolKit.
- ID-marker – Jednoduché značky, které mají přiřazeno unikátní identifikační číslo v rámci knihovny Studierstube.
- DataMatrix marker – Datová matice je ISO standard pro 2D čárové kódy.
- Frame marker – Značka ve tvaru čtverce. Střed může být libovolně vyplněn.
- Split marker – Obdélníková značka, která je rozpoznávána pouze podle dvou oblastí na dvou okrajích.
- Dot marker – Značka tvořená body rozmístěnými v pravidelné čtvercové síti.



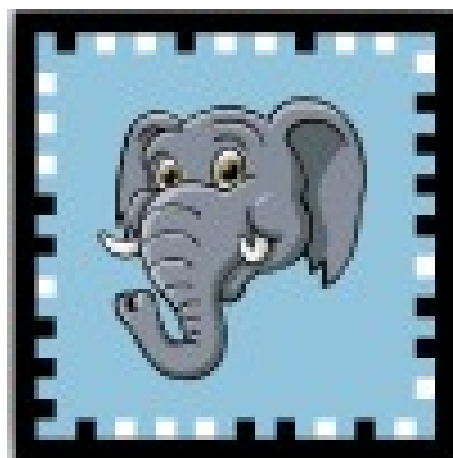
a) Template marker.



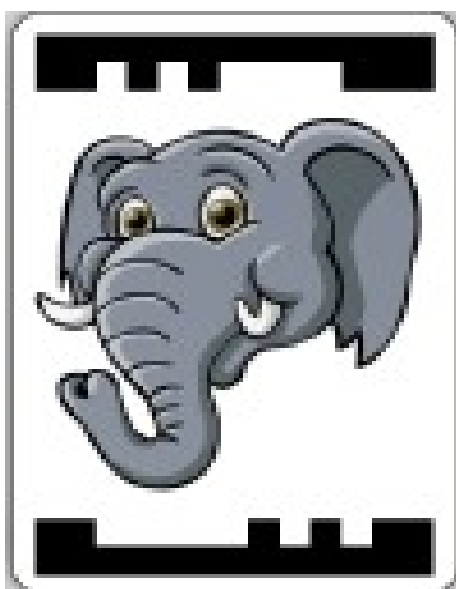
b) ID marker.



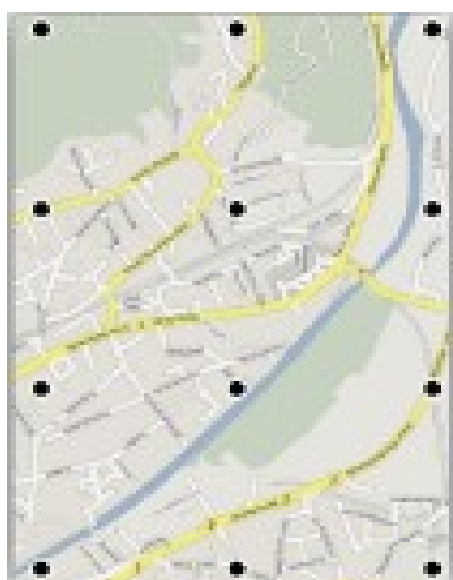
c) DataMetrix marker.



d) Frame marker.



e) Split marker.



f) Dot marker.

[obr. 4.4] Přehled typů značek, které umí rozpoznávat knihovna Studierstube tracker [Sch07].

5 **Návrh TUI pro správu aplikačních oken**

Kapitola obsahuje návrh systému pro správu aplikačních oken pomocí TUI. Nejprve jsou popsána zařízení, ze kterých se systém skládá. Dále následuje návrh kalibračního a řídicího programu systému.

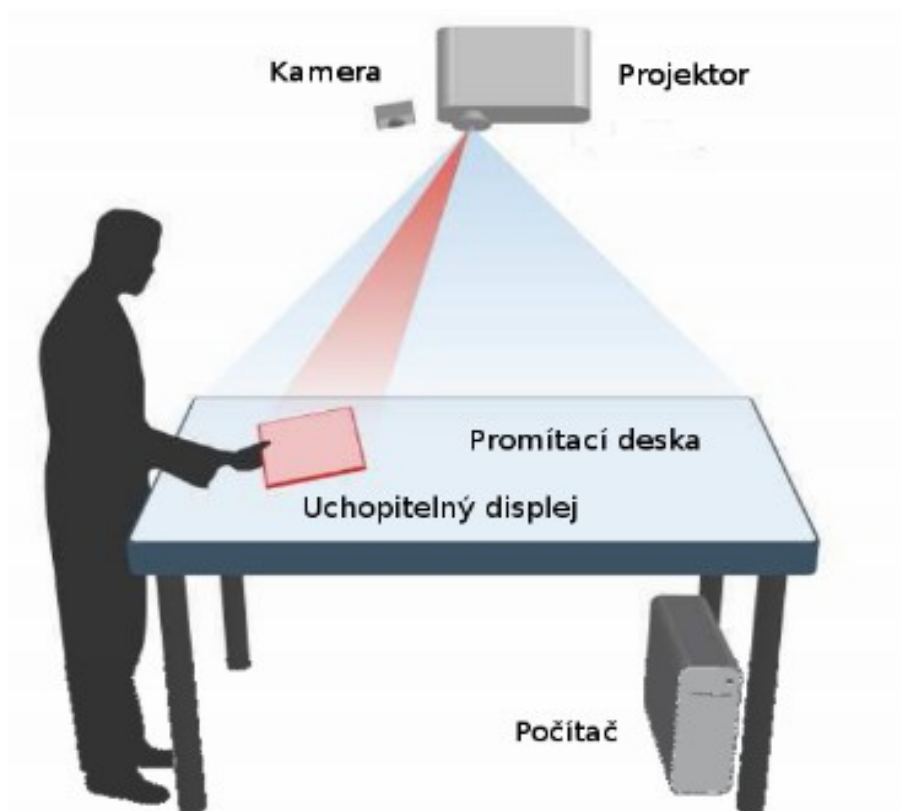
5.1 **Hardwarová sestava**

Složení systému je znázorněno na obrázku 5.1. Hlavní součásti jsou:

- Kamera - Snímá obraz scény ve spektru viditelného světla.
- Projektor - Promítá prvky grafického uživatelského rozhraní.
- Počítač - Hlavní řídicí jednotka celého systému. Zpracovává obraz z kamery, podle kterého upravuje promítaný obraz.

Další součásti systému jsou:

- Stůl - Jeho vrchní deska slouží jako promítací plátno. Kamera i projektor jsou umístěny nad stolem.
- Obdélníkové desky - Desky budou realizací uchopitelných displejů viz kapitola 2.3. Mohou být vyrobeny z různých materiálů jako např. papír, karton nebo polystyren. Jsou opatřeny značkami, aby je bylo možné sledovat v obraze. Budou sloužit k manipulaci aplikačních oken. Aplikační okno bude spojeno s deskou tak, že na něj bude celé promítnuto. Následně bude toto okno přemísťováno a otáčeno společně s deskou.



[obr. 5.1] Správa aplikací oken pomocí TUI. Upravený obrázek z [SST09].

5.2 Kalibrace kamery a projektoru

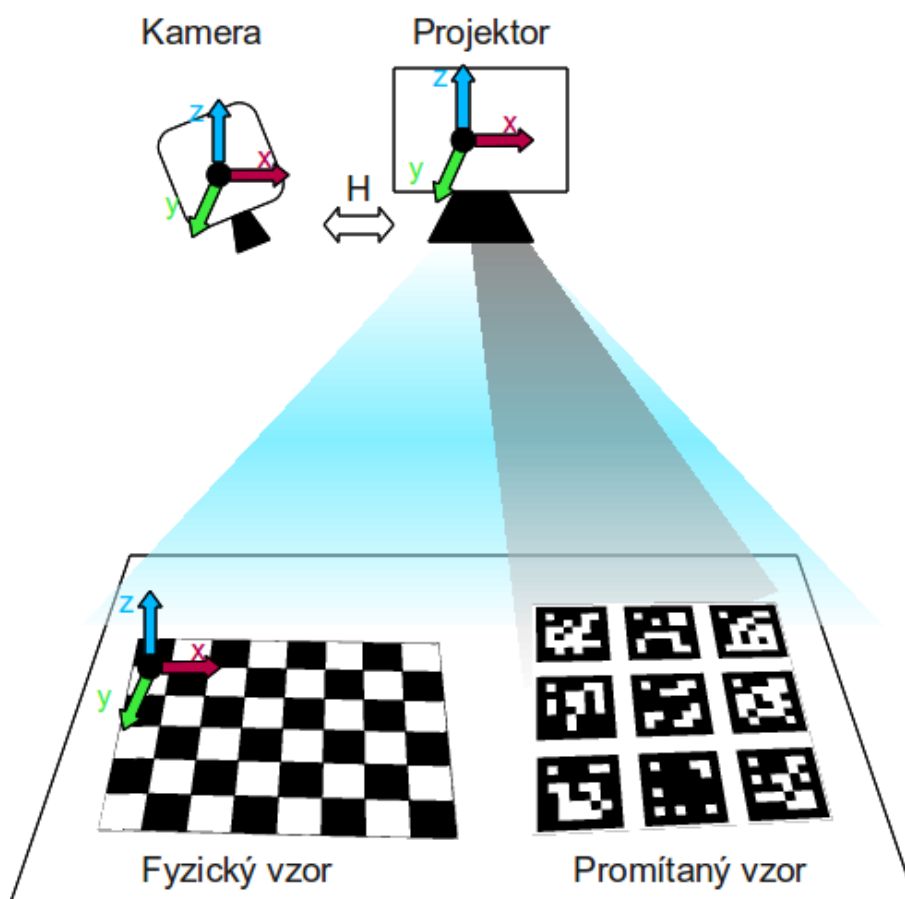
V cílovém systému bude třeba znát vzájemnou prostorovou pozici kamery a projektoru. Tyto informace jsou nutné především pro správné určování prostorových pozic značek vzhledem k projektoru kvůli transformacím a promítání aplikačních oken. Protože desky nebudou ležet pouze v projekční rovině, ale budou se pohybovat v prostoru scény, nelze použít jednoduchou kalibraci založenou na homografii mezi obrazem projektoru a kamery.

Kalibraci je možné provést např. pomocí programu Matlab viz. [FHM09] nebo jiných podobných nástrojů. Tyto nástroje však vyžadují zdlouhavý postup, při kterém musí uživatel zadávat do počítače dodatečné informace o kalibračních snímcích. Lepším řešením z pohledu uživatele je poloautomatická kalibrace. Kalibrační proces se tak urychlí a zjednoduší.

Pro kalibraci kamery bude použit jednoduchý program, který byl již dříve vytvořen v rámci projektu Deskothèque[SWS08] na TU Graz. Dále tedy bude popsán návrh programu na poloautomatickou kalibraci projektoru za předpokladu, že interní parametry kamery jsou již známy.

Následující návrh předpokládá přítomnost fyzického a promítaného kalibračního vzoru v jedné rovině před kamerou. Pro další popis bude nutné jednoznačné vymezení některých pojmů. Popisované pojmy jsou znázorněny na obrázku 5.2.

- **fyzický kalibrační vzor** - Bude realizován jako šachovnice vytištěná na papír. Má vlastní metrický souřadný systém s výchozím bodem v levém horním rohu. Prostorové souřadnice ostatních rohů lze určit z rozměrů vzoru.
- **promítaný kalibrační vzor** – Bude realizován sadou promítaných značek. Program si bude promítat vlastní vzor tzn. že jeho 2D souřadnice v obrazu projektoru budou vždy známé.



[obr. 5.2] Kalibrace systému kamera-projektor. Na obrázku jsou označeny výchozí body souřadných systémů kamery, projektoru a fyzického kalibračního vzoru. H ilustruje homografii mezi obrazem projektoru a kamery.

Kalibrace projektoru

Snímání

V tomto kroku je třeba získat souřadnice všech rohových bodů fyzického vzoru v obraze projektoru. Tyto souřadnice je nutné určit pro různé polohy kalibračního vzoru (vzhledem k projektoru), aby byl výsledek kalibrace co nejpřesnější. Ve scéně tedy bude prováděna opakovaná detekce fyzického a promítaného vzoru. Pokud budou oba vzory rozpoznány, zahájí se proces popsáný následujícím algoritmem č.1.

Algoritmus 1

1. Vypočti homografii mezi obrazem kamery a projektoru. Výpočet je proveden na základě souřadnic rohových bodů promítaného vzoru v obrazech kamery a projektoru.
2. Pomocí homografie vypočti souřadnice rohových bodů fyzického vzoru v obraze projektoru z rozpoznáných souřadnic v obraze kamery.
3. Výsledek ulož pro finální výpočet kalibračních parametrů.

Zpracování

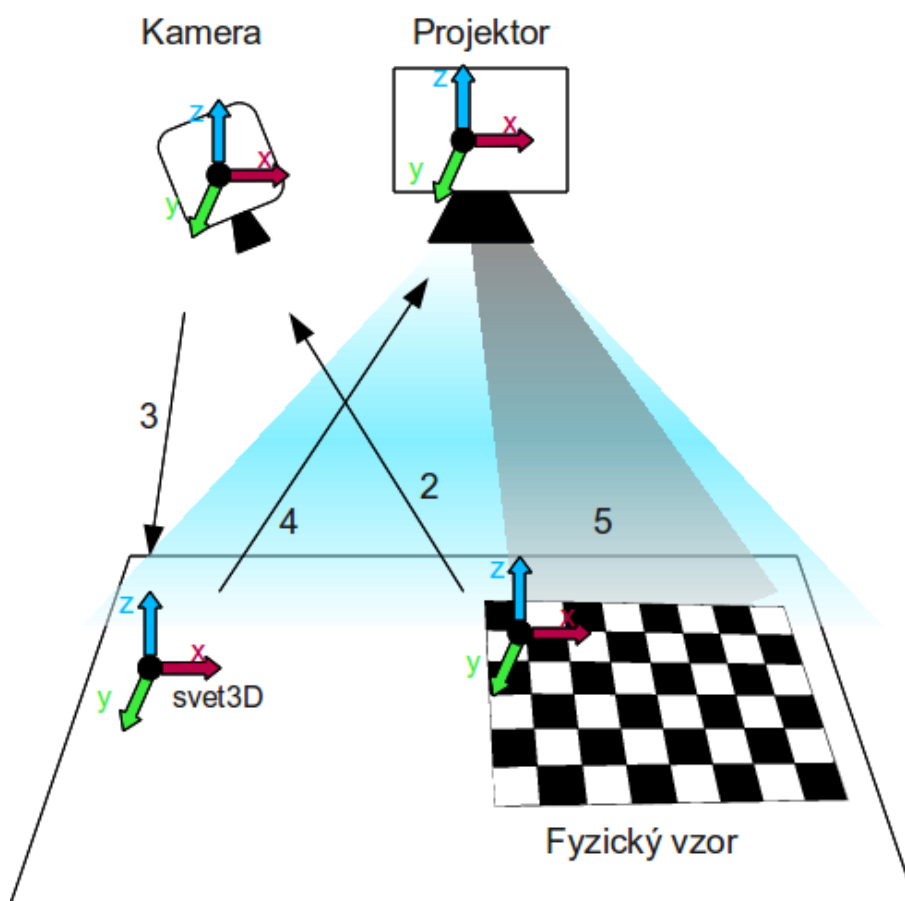
V této fázi budou vypočteny interní a externí kalibrační parametry projektoru na základě získaných 2D souřadnic rohových bodů v obraze projektoru a známých 3D pozic těchto bodů v souřadnicovém systému použitého fyzického vzoru. Pro samotný výpočet parametrů bude použita některá z metod popsáných v [RF06]. Externí parametry budou počítány vzhledem k poslední rozpoznané pozici fyzického vzoru v obraze projektoru. Současně dojde k přepočítání externích parametrů kamery podle stejné polohy fyzického vzoru. Kamera i projektor tak budou kalibrovány vzhledem k výchozímu bodu respektive k jeho souřadnému systému, který je definován poslední rozpoznanou polohou a orientací fyzického vzoru.

Validace výsledku

V této části bude probíhat detekce fyzického vzoru. Podle jeho aktuální prostorové pozice vzhledem ke kameře budou vypočteny souřadnice jeho rohových bodů v obraze projektoru (algoritmus 2). Podle získaných souřadnic budou vykresleny speciální značky. Upravený obraz bude následně promítnut do scény. V případě úspěšné kalibrace, budou značky promítnuty přesně na rohové body fyzického kalibračního vzoru. Chybu kalibrace pak bude možné vyhodnotit jako odchylku promítaných bodů od rohů sledovaného vzoru.

Algoritmus 2

1. Vypočti transformační matici mezi souřadným systémem kamery a fyzického vzoru. Výpočet bude proveden na základě známé geometrie fyzického vzoru a jeho pozice v obraze kamery.
2. Pomocí získané matice transformuj 3D body fyzického kalibračního vzoru(dále jen body) do souřadnicového systému kamery (z souřadnicového systému vzoru).
3. Transformuj body do společného světového souřadnicového systému kamery a projektoru. K tomu použij inverzní matici k matici externích parametrů kamery.
4. Transformuj body do souřadnicového systému projektoru pomocí matice externích parametrů projektoru.
5. Transformuj body do obrazové roviny projektoru. Tato středové promítání bude založeno na matici interních parametrů projektoru.



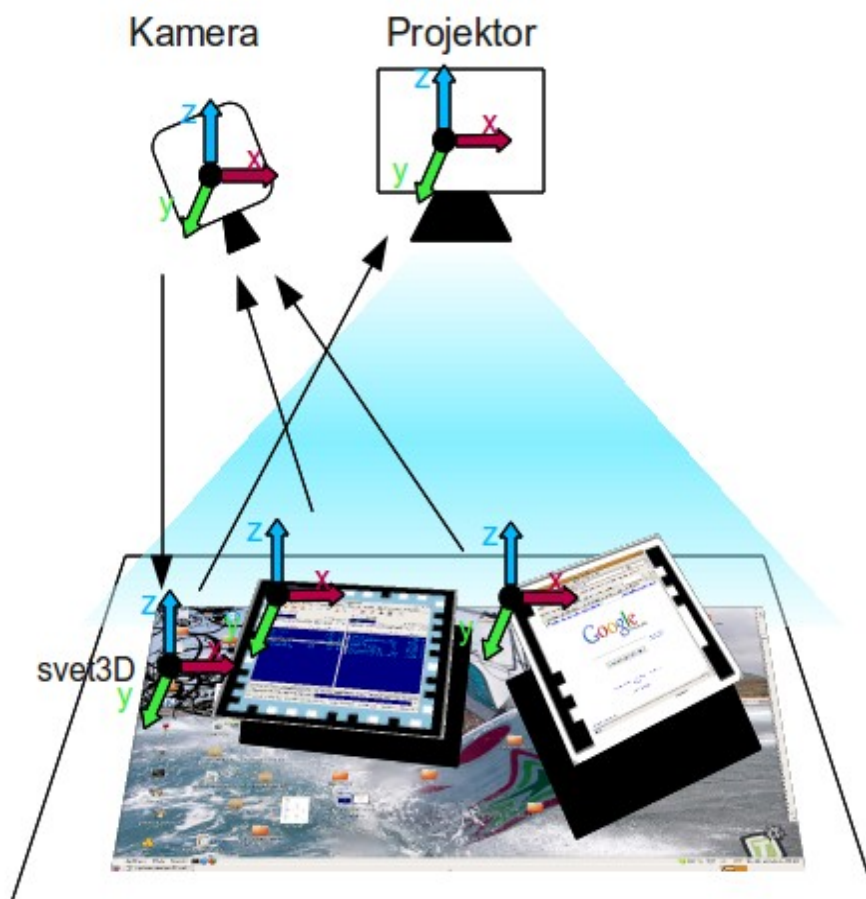
[obr. 5.3] Návrh kalibrace systému kamera-projektor. Šipky znázorňují transformace bodů mezi souřadnými systémy podle algoritmu č. 2. Čísla označují jednotlivé kroky tohoto algoritmu. Svet3D znázorňuje výchozí bod společného světového souřadnicového systému kamery a projektoru

5.3 Řídící program

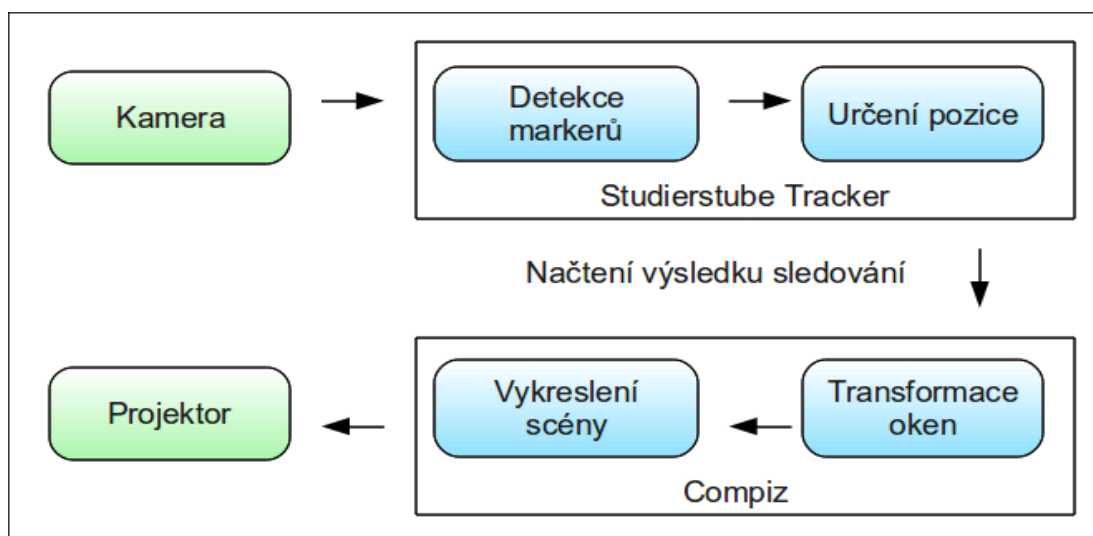
Cílem řídicího programu je zajistit sledování značek ve video sekvenci a transformovat aplikační okna v závislosti na zjištěné pozici značek ve scéně (viz. obr. 5.4). Pro programový přístup k aplikačním oknům je nutné napojení na správce oken operačního systému (dále jen OS). Jedním z vhodných řešení je vytvořit program jako zásuvný modul (dále jen plugin) pro Compiz. Compiz je open-source správce oken pro OS Linux. Více informací [wik06]. Řídící program je tedy dále navržen jako zásuvný modul – plugin pro Compiz.

Základní struktura

Hlavní proces řídicího systému je znázorněn na diagramu 5.1. Systém bude přijímat a zpracovávat snímky z kamery. S pomocí knihovny Studierstube bude probíhat detekce značek a určování jejich prostorové pozice vzhledem ke kameře a projektoru. Výsledkem budou transformační matice, které budou aplikovány při transformaci příslušných aplikačních oken. Plugin bude načítat a nastavovat matice pro jednotlivá okna, samotné transformace a vykreslování pak bude provádět přímo Compiz.

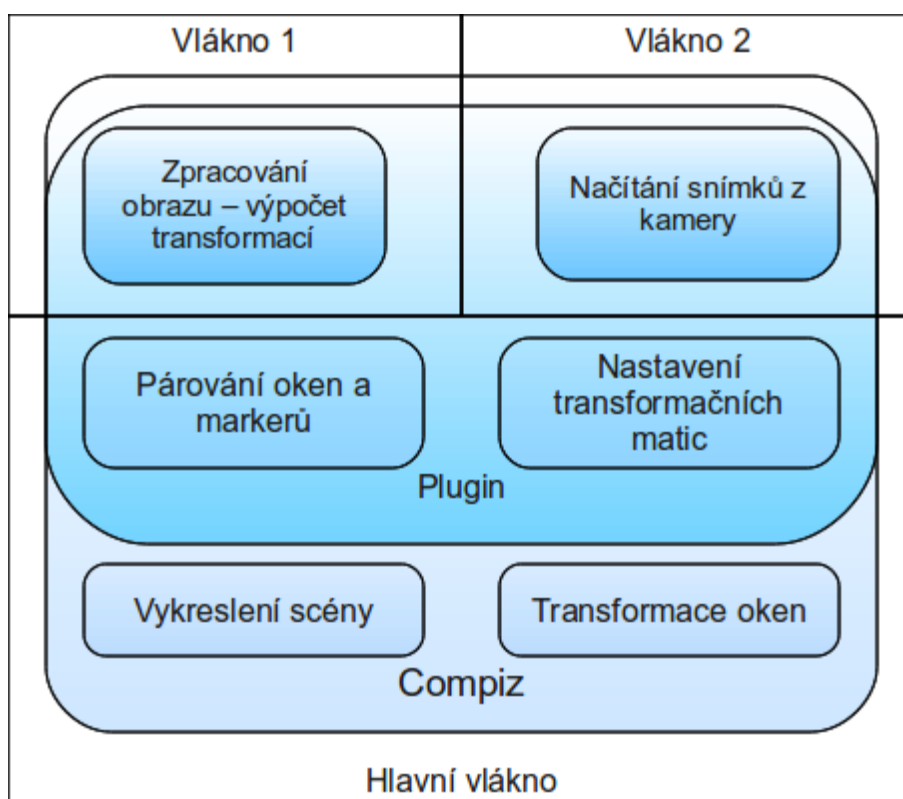


[obr. 5.4] Promítání aplikačních oken. Šipky ukazují transformace mezi souřadnými systémy značek a projektoru. Svet3D je střed světového souřadnicového systému kamery a projektoru.



[diag. 5.1] Přehled hlavního procesu řídicího systému.

Řídicí systém musí pracovat v reálném čase, aby odezva na uživatelské akce byla dostatečně rychlá. Proto byl vytvořen návrh, který separuje časově nejnáročnější úlohy do oddělených vláken viz. diagram 5.2. Odděleně poběží načítání snímků z kamery, aby byl pokaždé ihned k dispozici snímek pro zpracování. Dále bude odděleně běžet proces zpracování obrazu, který zahrnuje výpočet transformačních matic. Podrobnější popis úloh z diagramu 5.2 bude následovat.



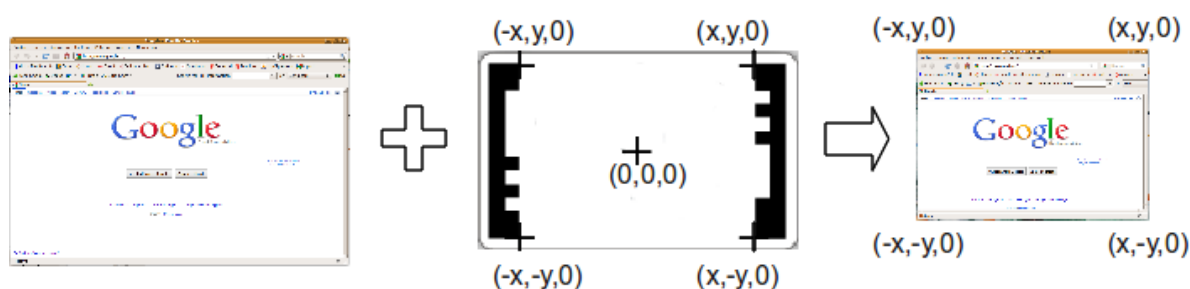
[diag. 5.2] Použití Multi-threadingu v řídicím programu. Oddělená vlákna pro načítání snímků a zpracování obrazu/výpočet transformačních matic.

Zpracování obrazu

Nejprve je pomocí knihovny Studierstube provedena detekce značek a určení jejich prostorové pozice vzhledem ke kameře. Přitom budou použity interní kalibrační parametry kamery a znalost rozměrů jednotlivých značek. Pro každou rozpoznanou značku bude výsledkem transformační matice mezi vlastním a kamerovým souřadným systémem. Následným násobením získané matice s inverzní maticí externích parametrů kamery a maticí externích parametrů projektoru bude získána transformační matice mezi souřadným systémem značky a projektoru. Díky této matici je možné transformovat rohové body značky do souřadnic projektoru stejně jak je popsáno v algoritmu č. 2 při validaci kalibračního výsledku viz. kapitola 5.2.

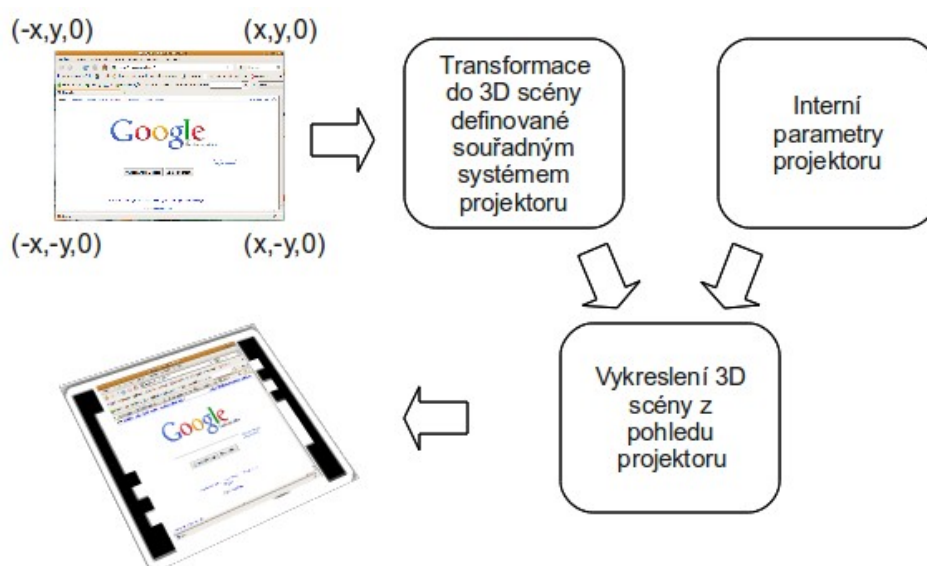
Transformace a vykreslování aplikačních oken

V prvním kroku jsou jednotlivá okna upravena podle rozměrů značek, ke kterým jsou přiřazena (bude popsáno dále). Oknům jsou tak nastaveny x, y souřadnice rohových bodů značek tak, že ve středu okna je výchozí bod $(0,0,0)$ viz. obrázek 5.5. Souřadnice z je pro všechny body rovna nule. Nastavení souřadnic je provedeno nastavením šířky ($2 \cdot x$) a výšky ($2 \cdot y$) okna v pixelech.



[obr . 5.5] Úprava velikosti aplikačního okna podle velikosti značky

Následně je každé upravené okno transformováno do 3D scény definované souřadným systémem projektoru pomocí příslušné matice získané v kroku Zpracování obrazu. Výsledná scéna je nakonec vykreslena z pohledu projektoru (viz. diagram 5.3). Přitom jsou použity interní kalibrační parametry projektoru pro nastavení pohledové kamery [Lee06].



[diag. 5.3] Transformace a vykreslení aplikačního okna

Párování aplikačních oken a značek

Vazby mezi okny a značkami (resp. deskami, které jsou značkami označeny) jsou vytvořeny při překrytí značky a okna v obrazové rovině projektoru. Souřadnice značky v obrazové rovině je možné určit transformací rohových bodů stejným způsobem jako při výše popsané transformaci oken.

6 Implementace

Kalibrační program byl implementován odděleně od řídicího programu. Může tak být použit nezávisle pro kalibraci libovolného systému kamery a projektoru.

6.1 Kalibrační program

Program byl vytvořen podle návrhu z předešlé kapitoly. V programu jsou použita dvě aplikační okna viz obrázek 6.1. V prvním jsou zobrazovány snímky z kamery a v druhém jsou vykreslovány značky pro vlastní detekci. Při kalibraci je použita tištěná šachovnice a promítaný kalibrační vzor složený z jednoduchých ID markerů. Během kalibrace, která probíhá v reálném čase, uživatel pohybuje s deskou opatřenou šachovnicí před projektorem a kamerou. Podrobný popis kalibračního procesu včetně popisu výstupního formátu kalibračních parametrů viz. Příloha 1. Manuál - Kalibrační program. Validace výsledku je prováděna podle návrhu z kapitoly 5.2. Navíc je automaticky počítána chyba reprojekce (pomocí knihovny OpenCV [Ocv09]), se kterou lze také vyhodnotit správnost výsledku. Chyba reprojekce odpovídá obrazové vzdálenosti mezi vypočtenou a změřenou polohou bodů v obraze kamery nebo projektoru[HZ04] [wik09].

Program je implementován v programovacím jazyce C++. Vykreslování obsahu aplikačních oken je zajištěno díky knihovně OpenGL [wik08]. Pro rozpoznávání značek je použita knihovna Studierstube Tracker. Knihovna OpenCV[wik07] implementuje stěžejní funkčnost programu, kterou je rozpoznávání šachovnice, výpočet homografie a interních resp. externích parametrů kamery a projektoru [Ocv09].



[obr. 6.1] Kalibrační program. Vlevo je okno zobrazující obraz z kamery. Vpravo je okno, které je promítáno projektorem.

Nejdůležitější třídy programu:

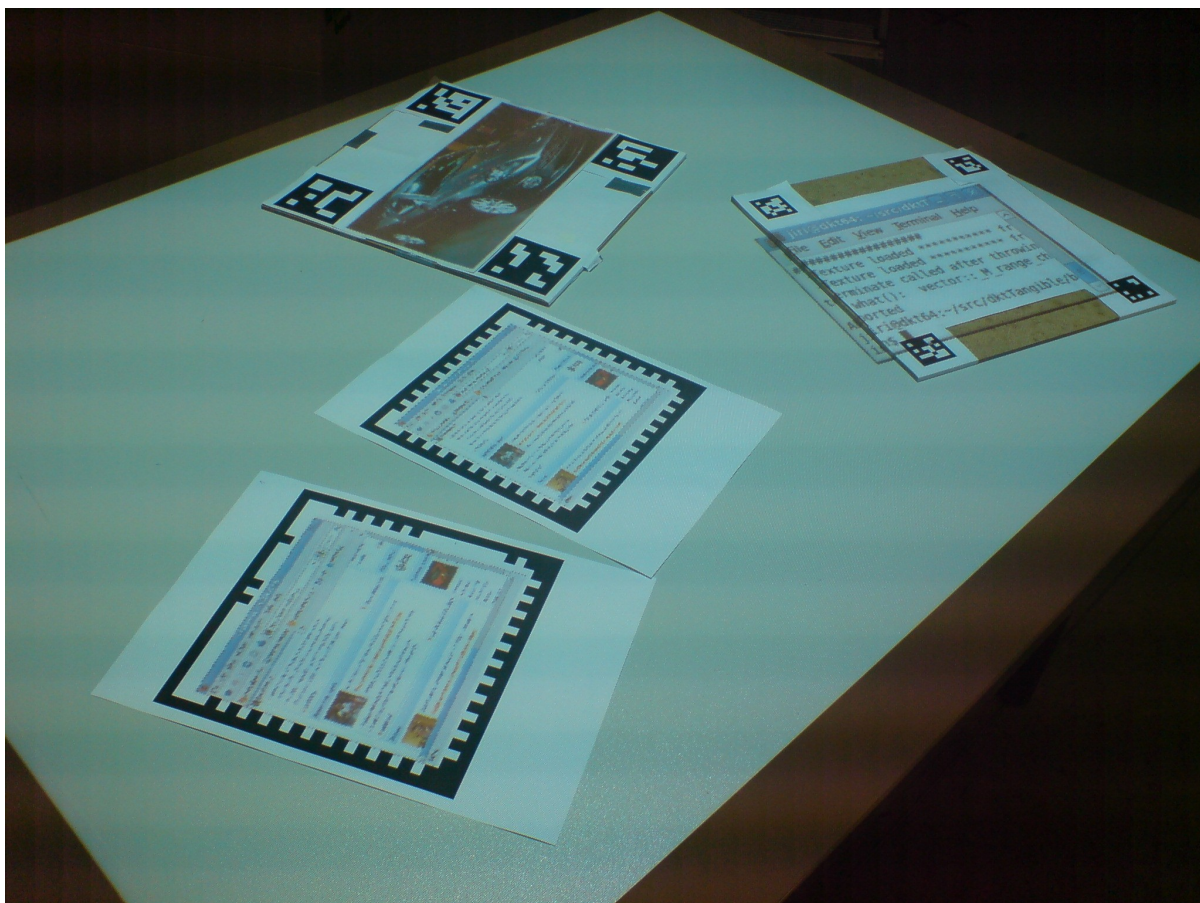
- CalibrationRoutine - Hlavní třída programu. Implementuje jednotlivé fáze programu dle návrhu. Obsahuje volání funkcí pro nejdůležitější výpočty a detekci kalibračních vzorů v obraze. Nabízí aplikační rozhraní, kterým je ovládána z hlavní smyčky programu.
- CodedMarkerDetector – Třída zajišťuje detekci značek v obraze.

6.2 Řídící program

Řídící program byl vytvořen podle návrhu jako plugin pro Compiz verze 0.8.4. Samotný plugin byl implementován v jazyce C. Subsystem pro sledování značek a výpočet transformačních matic byl implementován jako knihovna v jazyce C++. Knihovna byla původně implementována a testována odděleně od pluginu (viz. obrázek 6.2). Při implementaci byli použity knihovny OpenGL, OpenCV a již zmíněná knihovna Studierstube. Knihovna OpenGL byla použita pro transformace aplikačních oken a vykreslování scény. Knihovna OpenCV bylo použita pro transformace rohových bodů značek do obrazové roviny projektoru.

Nejdůležitější třídy programu:

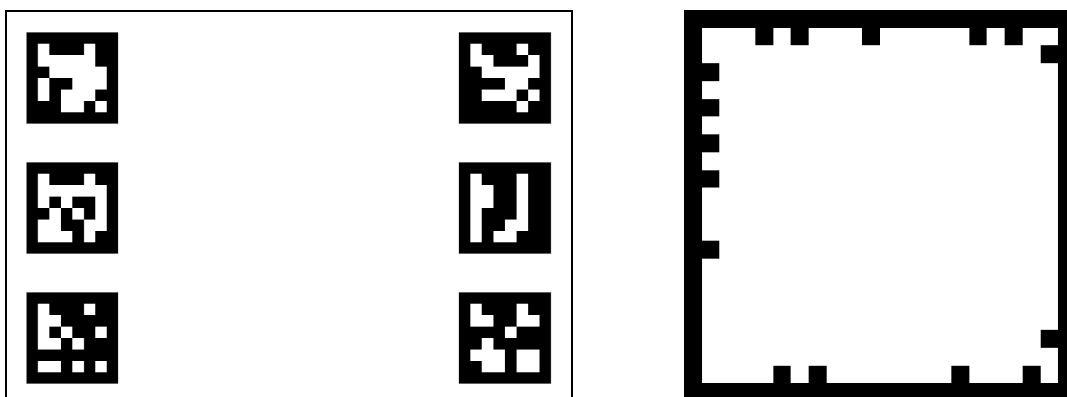
- TangibleRoutine - Hlavní třída knihovny pro sledování značek. Spravuje vlákna pro načítání snímků a zpracování obrazu. Implementuje aplikační rozhraní knihovny v jazyce C++.
- CTangibleRoutine – Třída implementuje aplikační rozhraní knihovny v jazyce C. Obaluje třídy TangibleRoutine a TangibleManager.
- TangibleObject – Instance této třídy reprezentuje objekt označený značkami, který se může vyskytovat ve scéně. Může jím být např. deska nebo dotekové pero, o kterém bude následovat více informací v sekci výsledky.
- TangibleManager – Třída pro správu oken. Udržuje informace o vazbách mezi okny a deskami. Vyhodnocuje vzájemnou polohu desek, podle které provádí řazení oken pro vykreslování.
- MarkerDetector – Třída implementuje zpracování obrazu a výpočet transformací viz. diagram 5.2. Tyto úlohy běží v odděleném vlákně.



[obr. 6.2] Ukázka promítání obrázků testovacím programem knihovny pro sledování značek a výpočet transformačních matic.

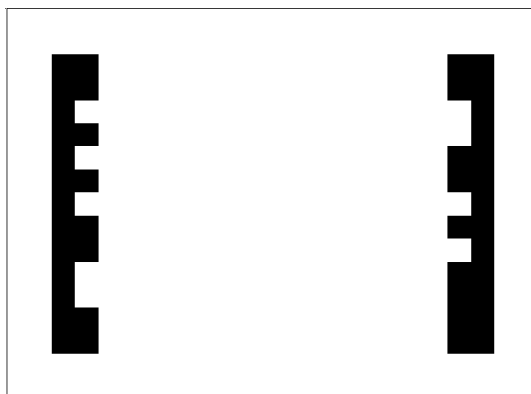
Sledování značek

Pro sledování značek v obraze byla použita knihovna Studierstube. Během implementace bylo experimentováno s různými druhy značek, jejichž sledování zmíněná knihovna nabízí (viz. kapitola 4.1). Jako nejvhodnější se ukázalo, použít sady jednoduchých značek tzv. multimarkery a tzv. frame markery viz. obrázek 6.3.



[obr. 6.3] Vlevo tzv. multi-marker. Vpravo tzv. frame marker.

Původně se jevilo jako nejvhodnější použití tzv. split markerů kvůli možnosti prázdné výplně a okrajům pouze ze dvou stran (viz. obrázek 6.4). Bohužel je nebylo možné použít, protože jejich detekce byla příliš špatná. Navíc bylo zjištěno, že kvůli chybě v knihovně Studierstube nemohou být tyto značky použity při snímání kamerami s rozlišením větším než 320x240 pixelů. U snímků s vyšším rozlišením docházelo k detekci těchto značek pouze v levém horním rohu. Tento poznatek byl zjištěn experimentálně během vývoje, protože byla použita kamera s rozlišením 1024x768 pixelů.



[obr. 6.4] Split marker

7 Výsledky

Výsledky kalibračního programu byly vyhodnocovány oddělené od výsledného TUI. V kalibračním programu jsem se zaměřil na testování přesnosti kalibračních parametrů. Řídící program TUI byl testována především na rychlost a přesnost sledování výchozích značek.

7.1 Kalibrace projektoru

Vývoj kalibračního programu byl časově náročnější než se původně předpokládalo. Největší problémy byly s fází ověřování výsledku kalibrace. Bylo totiž těžké určit jestli je správný výsledek a chybný proces ověření nebo naopak. Nakonec se podařilo program úspěšně implementovat.

Vytvořený program byl testován na přesnost vypočtených kalibračních parametrů projektoru s následujícím nastavením:

- Kamera a projektor umístěny vedle sebe ve vzdálenosti cca 10 cm. Obě zařízení byla otočená stejným směrem viz obrázek 7.1. Obě zařízení měla rozlišení 1024x768 pixelů.
- Vzdálenost mezi projektorem a projekční plochou/stolem byla 240 cm.



[obr. 7.1] Kamera(uprostřed vlevo) a projektor umístěny na kovové konstrukci nad stolem.

Přesnost vypočtených parametrů byla vyhodnocována podle odchylky značek promítaných na vnitřní rohy šachovnice viz. validace výsledků v kapitole 5.2 nebo obrázek 7.3. Nejlepších výsledků bylo dosaženo při výpočtu kalibračních parametrů na základě 100 zachycených snímků a při použití čočky s ohniskovou vzdáleností 8 mm (u kamery). Tyto výsledky měli při validaci přesnost promítání s odchylkou 0 - 4 mm, která se lineárně měnila v závislosti na vertikální vzdálenosti sledované šachovnice od projekční roviny viz. graf 1.



[graf 1] Graf vyjadřuje velikost odchylky při promítání značek závislou na vertikální vzdálenosti sledované šachovnice od projekční roviny. Značky jsou promítány na vnitřní rohy sledované šachovnice viz. obrázek 7.3.

Přesnost byla dále vyhodnocována podle chyby reprojekce s ohledem na jednotlivé interní kalibrační parametry. Průměrné hodnoty spolu se směrodatnými odchylkami pro chyby parametrů ohniskové vzdálenosti (f_x , f_y) a posunu základního bodu (c_x , c_y) jsou zachyceny v tabulce č.1. Hodnoty byly vypočteny použitím funkce `ProjectPoints2` z knihovny OpenCV (více informací viz. [Ocv09]). Naměřené hodnoty ukazují, že kalibrace projektoru dosahuje téměř stejně dobré přesnosti jako kalibrace kamery. V tabulce nejsou uvedena data týkající se parametrů geometrického zkreslení, protože tyto parametry nebyly v případě projektoru správně vypočteny. Podrobnější vysvětlení následuje na konci kapitoly.

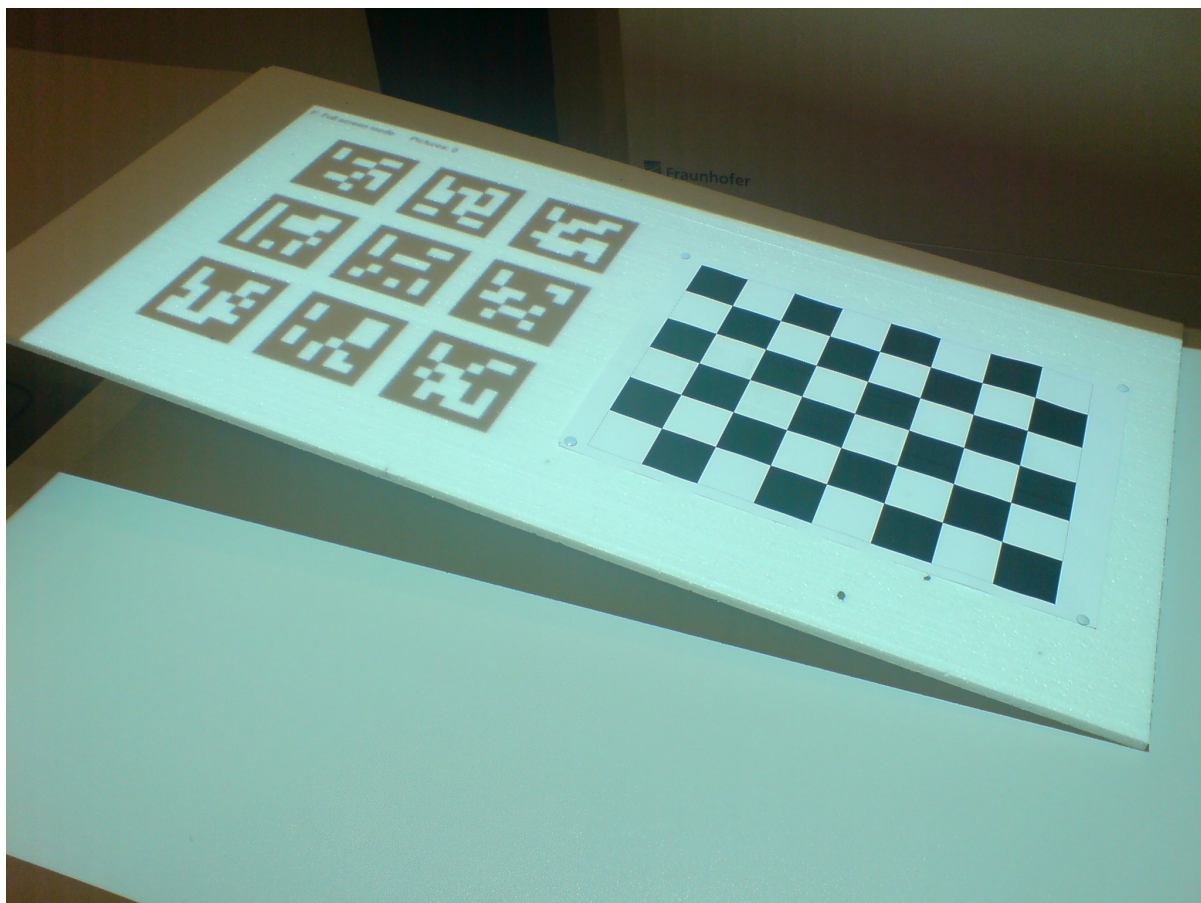
Parametr	f_x		f_y		c_x		c_y	
	$\bar{f}_x$	σ	$\bar{f}_y$	σ	$\bar{c}_x$	σ	$\bar{c}_y$	σ
Kamera	-0,003	0,025	0,022	0,030	0,333	0,001	0,667	0,001
Projektor	0,015	0,022	-0,047	0,041	0,333	0,001	0,667	0,001

$\bar{a}$ - průměr

σ - směrodatná odchylka

[tabulka 1] Chyba reprojekce pro parametry ohniskové vzdálenosti (f_x , f_y) a posunu základního bodu (c_x , c_y). Tabulka obsahuje hodnoty pro kalibrovanou kameru a projektor.

Při testování byly také použity čočky s ohniskovou vzdáleností 4.8, 6. Nebylo však dosaženo tak dobrých výsledků, protože promítaný obraz byl menší v obraze kamery. Proto byly výpočty homografie a souvisejících kalibračních parametrů méně přesné.

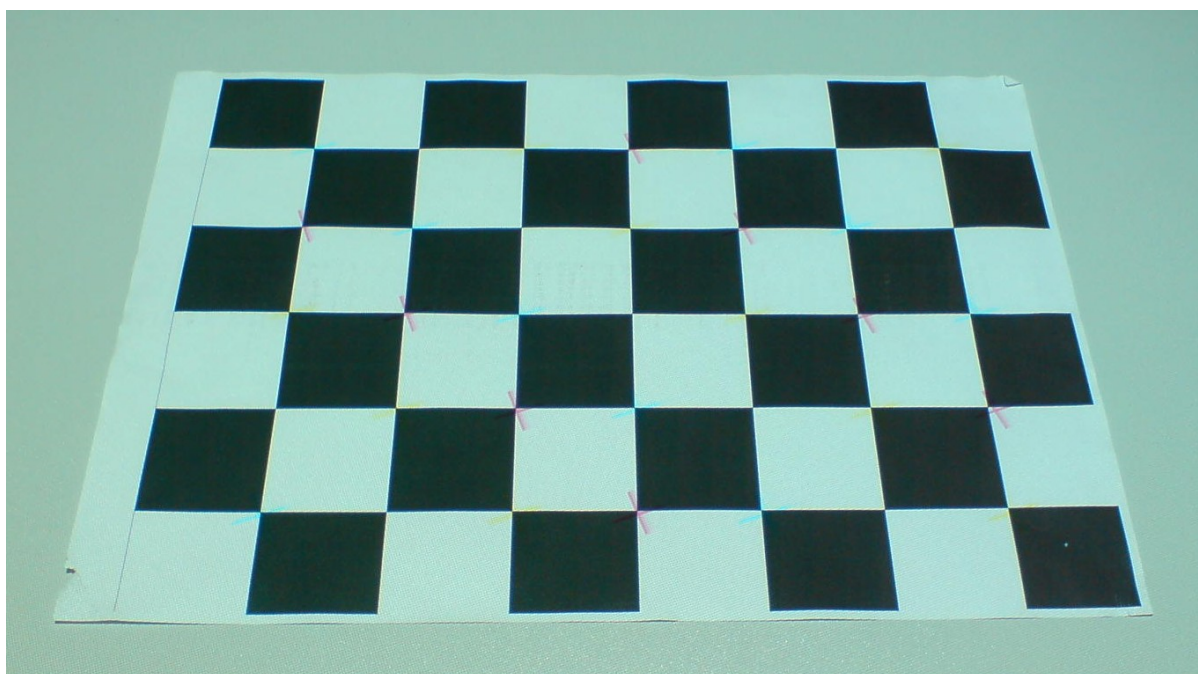


[obr.7.2] Ukázka poloautomatické kalibrace projektoru vytvořeným programem. Vlevo promítaný kalibrační vzor. Vpravo šachovnice = fyzický kalibrační vzor.

Dále byla prováděna kalibrace s různými počty kalibračních snímků. Počty byly v rozmezí 50 – 200 snímků, na kterých byly rozpoznány oba kalibrační vzory. Velmi kvalitních výsledků bylo dosaženo právě při zmíněných 100 snímcích. Při vyšších počtech snímků se výsledek mírně zlepšoval. Doba, která byla potřeba pro výpočet kalibračních parametrů, však s počtem snímků výrazně narůstala. Optimální počet snímků pro kvalitní kalibraci byl experimentálně zvolen na 90 až 100. Ačkoliv teoreticky pro vyřešení kalibračních rovnic (výpočet kalibračních parametrů) stačí znát světové 3D a odpovídající obrazové 2D souřadnice pouze 6ti bodů [HZ04].

Experimentálně byly zjištěny následující faktory, které výrazně ovlivňují kalibraci.

- Poloha čočky projektoru - Čočka musí být na středu, jinak je homografie nepřesná. Tím je ovlivněn celkový výsledek.
- Vzájemná poloha promítaného vzoru a šachovnice při výpočtu externích parametrů – Šachovnice musí být umístěna těsně vedle promítaného vzoru, aby byl výpočet externích parametrů co nejpřesnější (viz. Příloha 1. Manuál - Kalibrační program). Jinak můžou být vypočteny nepřesné externí parametry i v případě, že interní parametry jsou správné.
- Nastavení clony a expozice kamery – Promítaný obraz je za běžných podmínek velmi světlý v obraze kamery. V takto přесvíceném obraze není možné rozpoznávat promítané vzory. Nejvhodnější úpravou je nastavení clony čočky. Expozici kamery je vhodné ponechat na standardní hodnotě. Experimentálně bylo zjištěno, že při krátké expozici obraz blikal a ačkoliv nebyl příliš světlý, rozpoznávání promítaného vzoru téměř nefungovalo.



[obr. 7.3] Barevné křížky promítnuté na vnitřní rohy šachovnice.

Při testování nebyl kladen důraz na rychlost zpracování. Doba zpracování ve fázi snímání byla dostačující pro reálné použití, protože odezva systému měla zpoždění maximálně 1 s. Časově nejnáročnější fází je výpočet kalibračních parametrů. Časová spotřeba tohoto procesu je přímo závislá na množství pořízených snímků. Např. pro 80-100 snímků se doba zpracování pohybuje okolo 2 minut. Program byl testován na počítači s 4 jádrovým procesorem (frekvence 2,66GHz) od společnosti AMD. Empiricky bylo zjištěno, že celková doba kalibrace kamery a projektoru s vytvořeným programem zabere v průměru 5-10 minut. Uvedený čas nezahrnuje vyladění kamery na aktuální světelné podmínky a přípravu desky s šachovnicí.

Po dokončení kalibračního programu bylo zjištěno, že získané geometrické zkreslení projektoru nemůže být správné. Problém je v získávání souřadnic kalibračních bodů v obrazu projektoru pomocí projektivní transformace (homografie) z obrazu kamery, ze kterého jsou odstraněna geometrická zkreslení. Souřadnice kalibračních bodů tak nejsou zkresleny čočkou projektoru. Z obrazu kamery se odstraňují geometrická zkreslení proto, aby bylo možné lépe rozpoznat sledované objekty. Problém by mohl být vyřešen použitím snímků s ponechaným geometrickým zkreslením. Pak ovšem nastanou komplikace se zhoršenou detekcí objektů. Problém by také mohl být odstraněn kompletní změnou postupu získávání 2D a 3D souřadnic kalibračních bodů. Nicméně po konzultaci s vedoucí této práce na TU Graz a s přihlédnutím k dosaženým výsledkům jsem se rozhodl ponechat stávající implementaci. Geometrické zkreslení projektoru bývá velmi malé a výrazně neovlivňuje transformace prováděné se získanými kalibračními parametry.

7.2 Řídící program pro TUI

Řídící program byl vytvořen bez větší komplikací podle návrhu. Navíc bylo vyřešeno vzájemné překrývání aplikačních oken, se kterým se v návrhu nepočítalo. Aplikační okna se vykreslují v pořadí daném aktuální výškou jejich desek nad stolem od nejnižší po nejvyšší viz. obrázek 7.5. Okna, která nejsou napojena na žádnou desku jsou vykreslována jako první, aby nikdy nepřekrývala ostatní napojená okna. Tím je zajištěno, že okno na nejvýše položené desce je vždy promítnuto celé. Vzájemná poloha desek je určována podle aktuální vzdálenosti od kamery.

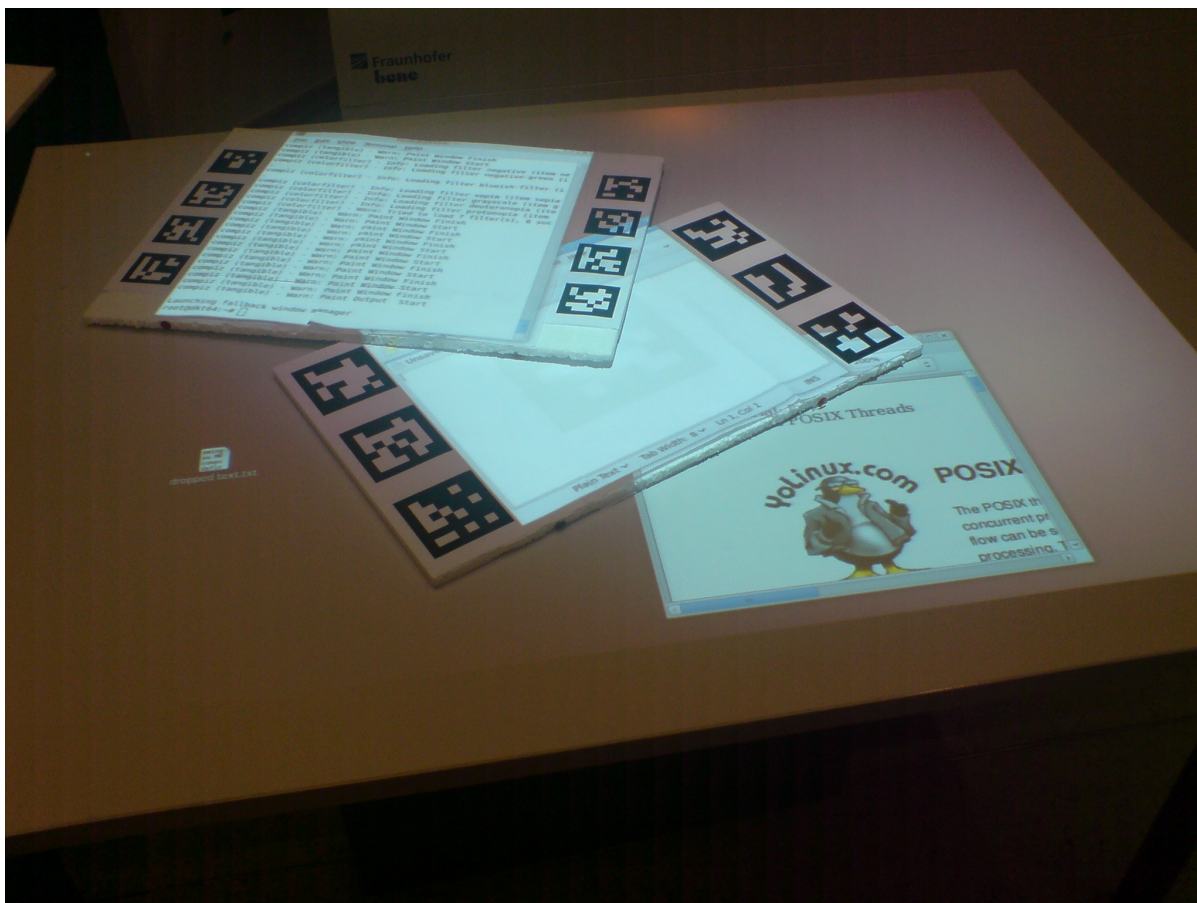


[obr. 7.4] Promítání aplikačních oken nově vytvořeným systémem s podporou Compiz. Desky nahoře jsou označeny tzv. multimarkery. Deska uprostřed je označena tzv. frame markerem.

Vytvořený program byl testován při stejném nastavení kamery a projektoru jako v případě kalibračního programu viz. kapitola 7.1. Byly provedeny testy na rychlost odezvy, stabilitu a spolehlivost při sledování značek. Testy na přesnost promítání aplikačních oken nebyly uskutečněny, protože tento atribut je přímo závislý na kvalitě provedené kalibrace.

Při experimentování byl použit projektor s rozlišením 1024x768 pixelů. Velikost promítaného obrazu na desce stolu byla 104x79 cm. Pro desku s prostorem pro promítání o rozměrech 30x25 cm to znamená relativně malé rozlišení 295x243 pixelů. Kvůli možnosti promítání aplikačních oken ve vyšším rozlišení byla přidána možnost nastavit jejich měřítko za běhu programu.

Uvedené rozlišení 295x243pixelů platí pokud je deska umístěna přímo na stole. Rozlišení stoupá s výškou sledované desky nad stolem. Přitom však klesá ostrost promítaného obrazu, protože projektor je zaostřen vzdálenost desky stolu. V testovaném nastavení došlo k přílišnému rozmazání obrazu ve výšce okolo 0.5 m nad stolem.

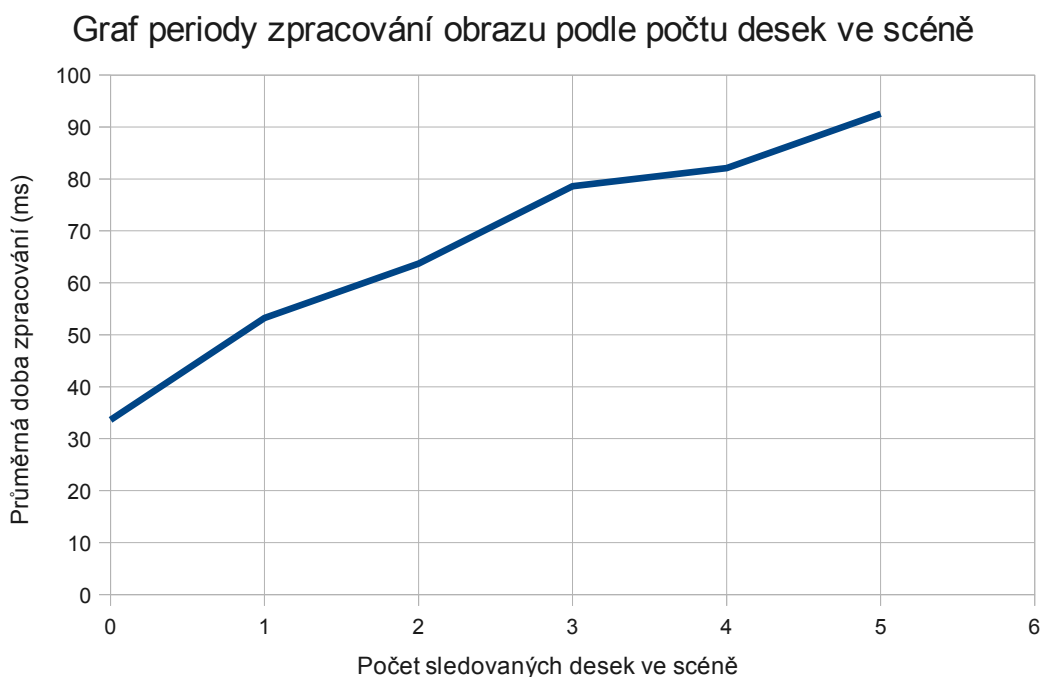


[obr. 7.5] Ukázka překrývání aplikačních oken připojených k deskám.

Rychlost odezvy

Při experimentování s řídicím programem byla testována odezva na uživatelské akce s deskami. Přitom bylo pozorováno zpoždění odezvy, které se pohybovalo mezi 0.1 až 0.3s. Hlavními příčinami zpoždění bylo načítání snímků z kamery, rozpoznávání značek a výpočty transformací. Zpoždění také bylo ovlivněno aktuálním vytížením procesoru v závislosti na ostatních spuštěných aplikacích. Testovaný program byl dostatečně rychlý pro plynulé přehrávání videosekvence v okně, kterým bylo pohybováno pomocí desky (viz. video na přiloženém DVD). Při sledování výkonu pomocí Compiz Benchmark (vysoce závislé na grafické kartě) byl v zaznamenán průměrný úbytek 20% z hodnoty snímků/s (FPS) při zapnutí pluginu oproti vypnutému stavu.

Testování rychlosti bylo dále zaměřeno na subsystém pro zpracování obrazu a výpočet transformací, který je výkonnostně nejnáročnější část programu. Při testech bylo současně sledováno až 5 desek označených celkem 20ti značkami. Deskami bylo paralelně pohybováno, přičemž na ně byla napojena/promítána aplikační okna. Při testování byla měřena perioda potřebná pro zpracování jednoho snímku a výpočet transformací podle nalezených značek. Výsledky testování pro různé počty sledovaných desek ve scéně jsou zobrazeny v grafu číslo 2.



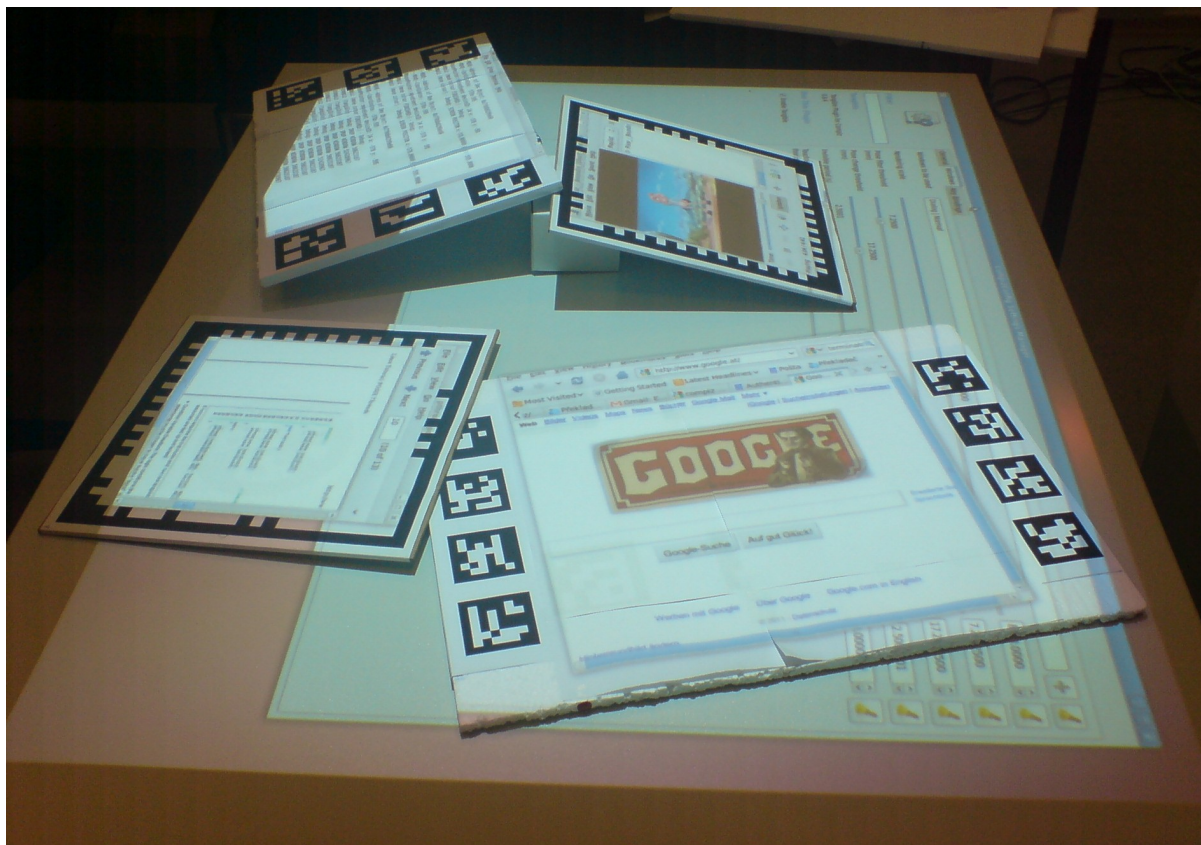
[graf 2] Graf zobrazuje průměrnou dobu zpracování jednoho snímku závislou na počtu sledovaných desek ve scéně viz. např. obrázek 7.4.

Sledování značek

Kvalita sledování byla testována se značkami různých velikostí. Pokud byly použity příliš malé značky, výpočet prostorových pozic ve scéně byl nestabilní což způsobovalo chvění promítaných oken. V případě použití příliš velkých značek bylo chvění odstraněno, ale již nezbyvalo mnoho místa na promítaná okna. Pro testované nastavení TUI (viz. kapitola 7.1) byla experimentálně vyhodnocena optimální velikost jednotlivých značek, ze kterých jsou poskládány multimarkery, v rozmezí od 50x50 do 70x70 mm. Tyto velikosti dávají vhodný kompromis mezi chvěním a prostorem pro zobrazování oken. Pro úplné odstranění chvění oken, bylo nutné implementovat jednoduchou filtraci prostorových souřadnic značek vypočtených knihovnou Studierstube. Filtrace je založena na průměrné hodnotě souřadnic vypočtené z posledních 5-ti vzorků. Souřadnice, které se liší minimálně o prahovou hodnotu od průměru nejsou dále použity. Dále bylo nutné odstranit chvění v rozsahu 1-2 pixelů, které bylo způsobené použitím nových, ale přitom téměř totožných souřadnic. Problém byl vyřešen nastavením prahu, o který musejí být nové souřadnice odlišné od posledně použitých, aby mohly být použity. Ačkoliv se tím systém mírně zpomalil, uživatel má k dispozici stabilní obraz. Prahy je možné upravovat za běhu programu.

Téměř žádné chvění oken nebylo zaznamenáno při použití tzv. frame markerů kvůli dostatečné velikosti od 20x20 do 28x28cm. Tyto značky však nabízejí relativně malý prostor pro promítání oken, který musí být vždy čtvercový viz. obrázek 7.6. Hlavní problém byl ve snadném zneviditelnění

značky již při jejím mírném překrytí např. rukou uživatele. Tento problém nebyl zaznamenán v případě multimarkerů, protože ty jsou viditelné pokud alespoň jeden jejich jednotlivý ID marker je viditelný.



[obr. 7.6] Promítání aplikačních oken. Velikosti jednotlivých značek jsou 50x50mm (pro multimarker vpravo dole) a 70x70mm (pro multimarker vlevo nahoře). Velikost frame markerů je 28x28 cm.

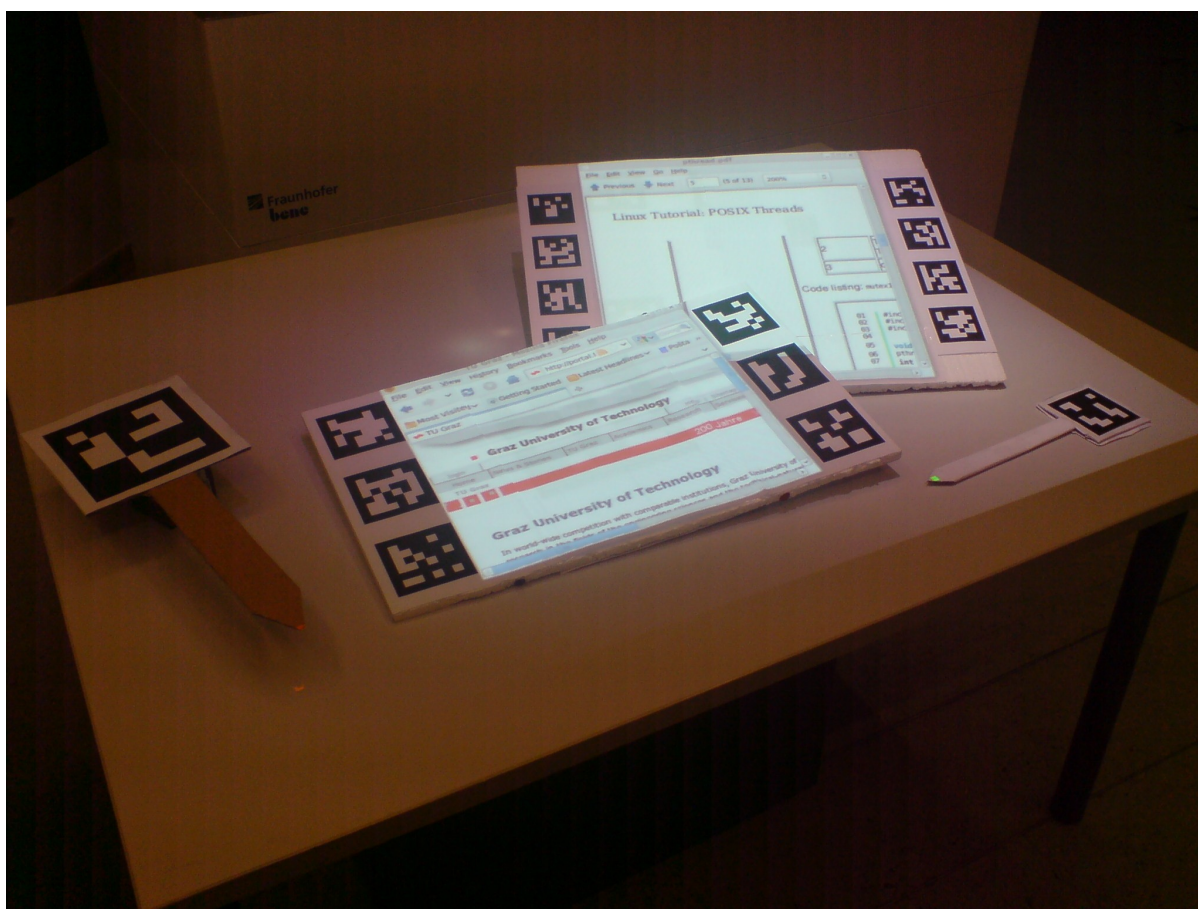
Při experimentování se stávalo, že promítaný obraz zakrýval značky, které se tak staly neviditelnými v obraze kamery. Tento problém by téměř odstraněn vhodným nastavením doby expozice a clony tak, že obraz z kamery byl hodně přesvícený. Promítaný obsah se tak stal neviditelným na rozdíl od značek, které byly stále rozpoznatelné.

Dotekové pero

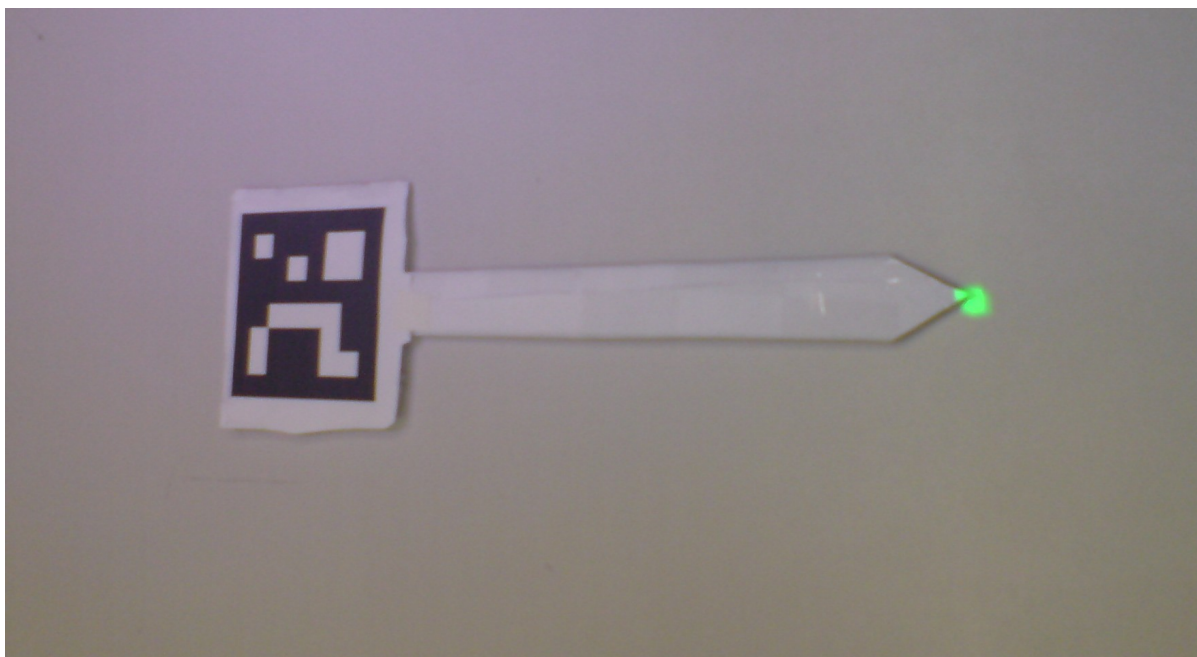
Po dokončení části pro manipulaci s aplikačními okny bylo navíc navrženo a částečně implementováno dotekové pero (viz. obrázek 7.8 a 7.9). Pero mělo sloužit jako plnohodnotná náhrada počítačové myši. Uživatel by tak mohl interagovat s promítaným obsahem pomocí dotyků tímto perem. Prostorovou pozici špičky pera je možné určit z jeho známé geometrie a polohy značky,

kterým je pero opatřeno. Na základě toho lze určit průnik s deskami nebo projekční plochou. Pero tedy mělo fungovat tak, že podle jeho pozice je programově přesouván kurzor myši a v případě průniku s jiným objektem, je vyvoláno myši „kliknutí“.

Pro programové volání událostí myši byla použita knihovna, kterou implementoval Mark Dokter z TU Graz (dokter@icg.tu-graz.ac.at). Knihovna však obsahovala chyby, kvůli kterým program často havaroval. Z toho důvodu a z důvodu nedostatku času nebylo dotekové pero dokončeno. Celkově tedy byla implementována pouze část pro detekci polohy pera a jeho průniku s projekční rovinou nebo deskami.



[obr. 7.8] TUI pro správu aplikačních oken. Vlevo je dotekové pero, kde červená tečka na špičce signalizuje, že není žádný průnik s jiným objektem. Dotekové pero vpravo je osvětleno zelenou tečkou signalizující průnik s projekční plochou.



[obr. 7.9] Ukázka dotekového pera. Zelená tečka na špičce signalizuje průnik s projekční rovinou.

7.3 Interakce s TUI

Vytvořený řídicí program nabízí některá nastavení pro snadnější použití celého TUI. Kromě již zmíněných možností nastavení citlivosti sledování značek a měřítka promítaných oken, lze také upravovat periodu pro zneviditelnění nebo vybírat typy oken, která mají být napojována na desky. Okna mohou být např. typu: dialogové, normální, nabídka nebo upozornění. Perioda pro zneviditelnění určuje po jaké době se má přestat promítat aplikační okno pokud jeho deska není rozpoznána.

Navázání okna na desku

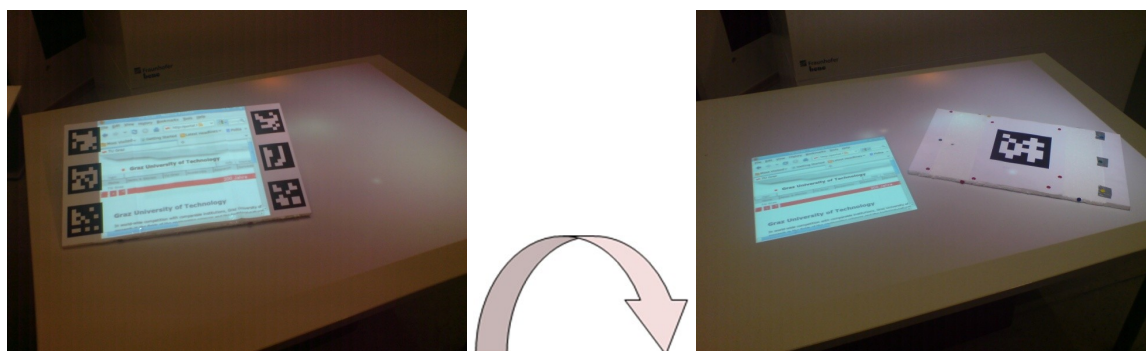
Tvoření vazeb mezi aplikačními okny a deskami označenými značkami je založeno na jejich průniku v obraze projektoru. Pokud se střed desky dostane do oblasti kam je promítáno okno, zahájí se proces napojení okna na desku (viz. obrázek 7.10). Následně je okno přesouváno a rotováno společně s deskou. Pokud není deska rozpoznána v obraze po určitou časovou periodu, dojde k zneviditelnění jejího okna. Když je deska navrácena do scény, napojené okno je na ní opět promítáno. Uživatel tak může „skladovat“ aplikační okna mimo prostor kam dopadá světlo projektoru a v případě potřeby vrátit požadované okno do scény.



[obr. 7.10] Ukázka vytvoření vazby mezi deskou a aplikačním oknem

Odpojení okna od desky

Vazba mezi aplikačním oknem a deskou je zrušena pokud je rozpoznána značka, který je umístěn na druhé straně desky. Takže uživatel musí převrátit desku spodní stranou nahoru (viz. obrázek 7.11), aby od ní odpojil aplikační okno. Aplikační okno je následně umístěno na souřadnice v obraze projektoru, kde byla horní strana desky naposledy rozpoznána. Tento přístup byl prezentován v [Wal06] .



[obr. 7.11] Ukázka převrácení desky => zrušení vazby mezi deskou a aplikačním oknem.

7.4 Diskuse

Vytvořený systém TUI pro správu aplikačních oken nabízí rozhraní pro přesouvání aplikačních oken, které může současně využívat více uživatelů nezávisle na pozici u stolu. Odezva systému je dostatečně rychlá a přitom je zobrazování oken stabilní, což je vhodné pro čtení/sledování promítaného obsahu. Systém nabízí některá nastavení pro snadnější instalaci a optimalizaci funkčnosti. Dále také nabízí velmi intuitivní ovládání, takže uživatel se s ním naučí pracovat během několika vteřin.

Systém TUI poskytuje jen omezenou funkčnost v porovnání s podobným uživatelským rozhraním PaperWindows [HVA+05]. Toto rozhraní dává uživateli možnost větší interakce s aplikačními okny (viz. kapitola 2.4). Nevýhodou je nutnost použít infračervenou kameru a značky,

které odrážejí světlo v tomto spektru. Oproti tomu je v této práci použita normální kamera a značky vytištěné na papír, čímž se snižuje pořizovací cena celého rozhraní. Další výhodou je také napojení na správce oken Compiz, které umožňuje rychlý programový přístup k „živým“ aplikačním oknům. Příkladem je plynulé přehrávání videa v okně promítaném na pohybující se desku.

Rozhraní prezentované v této práci by také mohlo být vytvořeno např. použitím několika zařízení iPad [Ipa10]. Přitom by nebylo nutné řešit sledování značek v obraze, dotykové ovládání, chvění zobrazovaného kontextu nebo omezení zobrazování na prostoru kam dopadá světlo z projektoru. Tento přístup by však sebou nesl některé nevýhody v podobě nutné síťové komunikace/sdílení dat mezi zařízeními nebo vysoké pořizovací ceny v závislosti na počtu použitých zařízení.

7.5 Návrh dalšího vývoje projektu

V dalších fázích projektu by bylo vhodné zaměřit se na následující úlohy:

- Implementace interakce dotekového pera s aplikačními okny.
- Rozpoznávání gest prováděných s deskami jako např. otáčení, překrývání, skládání na sebe, naklánění.
- Sledování lidské ruky v obraze - reakce na gesta a dotekové ovládání.
- Přidání dalších objektů pro interakci s počítačem např.:
 - papírovou „lupu“ pro zvětšování obrazu v místě kde je aktuálně umístěna
 - kostku pro přecházení mezi virtuálními pracovními plochami, aktuální pracovní plocha bude vybrána podle horní strany kostky
 - objekt (kostka, válec) pro listování/rolování obsahu aplikačních oken
- Použití více projektorů současně pro zvětšení rozlišení a pracovní/promítací plochy.

8 Závěr

Práce se zabývá tzv. Tangible user interface(TUI), což je „hmatové“ uživatelské rozhraní mezi člověkem a počítačem, které podporuje interakci pomocí běžných předmětů. Uvádí přehled některých uživatelských rozhraní na bázi TUI se zaměřením na ta rozhraní, ve kterých se využívá kamera a projektor. Dále obsahuje popis kalibrace těchto zařízení, která je nezbytná pro správnou funkčnost vybraných TUI. Součástí práce je také návrh a implementace programu pro poloautomatickou kalibraci kamery a projektoru. Stěžejní část se zabývá návrhem a implementací TUI pro správu aplikačních oken.

Cílem práce bylo vytvořit TUI pro správu aplikačních oken zahrnující kameru, projektor a výchozí značky. Součástí měla být i kalibrace použité kamery a projektoru. Cíl se podařilo splnit. Bylo vytvořeno rozhraní, kde jsou použity desky označené výchozími značkami pro přesouvání aplikačních oken. Uživatel může jednoduše „nabrat“ okno na desku, libovolně s ním manipulovat a nakonec jej opět „setřást“ z desky. Aplikační okna jsou transformována a promítána podle aktuální situace ve scéně, která je sledována kamerou. Interakce s rozhraním je zachycena na videu, které je uloženo na příloženém DVD.

Součástí řešení je program pro poloautomatickou kalibraci kamery a projektoru, jehož výstupem jsou interní a externí parametry obou zařízení. Kalibrace dosahuje velmi dobrých výsledků s odchylkou 0 – 4 mm při přepočtu pozice sledovaného objektu do obrazu projektoru. Při použití vytvořeného programu se čas kalibrace pohybuje v rozmezí 5 - 10 minut. Během výpočtu kalibračních parametrů nejsou správně vypočtena geometrická zkruslení čočky projektoru, protože jsou použity snímky z kamery, ze kterých jsou tato zkruslení odstraněna.

Vytvořené TUI nabízí rychlou interakci s reakčním zpožděním v rozmezí 0.1- 0.3s, které je tvořeno především načítáním a zpracováním snímků z kamery, a také výpočty jednotlivých transformací. S rozhraním může současně pracovat více uživatelů aniž by se reakční zpoždění výrazně prodloužilo. Přesnost promítání aplikačních oken je přímo závislá na kvalitě provedené kalibrace. Při testování byly zaznamenány problémy s chvěním promítaných oken a jejich nízkým rozlišením. Chvění bylo odstraněno nalezením optimální velikosti značek a filtrací souřadnic vypočtených knihovnou Studierstube. Velikost značek byla zvolena tak, aby bylo minimalizováno chvění oken, ale přitom byl na deskách dostatek místa pro jejich promítání. Zvýšení rozlišení by bylo možné použitím více projektorů současně nebo přiblížením projektoru k projekční rovině, čímž by se projekční oblast zmenšila.

Vytvořené rozhraní může sloužit pro sdílení informací a týmovou spolupráci lidí sedících okolo stolu. Mohou si tak aplikační okna podávat, přesouvat a otáčet tak, aby mohli pohodlně číst jejich obsah. Systém také může být použit v kombinaci se standardním uživatelským rozhraním např.

tak, že se si uživatel přesune některá okna (např. slovník, přehrávač hudby nebo textový dokument) na desky, které si následně podle potřeby bere, rovná nebo odkládá. Projektor přitom může promítat obraz přímo do oblasti, kde je umístěno běžné pracoviště v podobě displeje, klávesnice a myši.

V porovnání s předchozími pracemi z oblasti TUI pro správu aplikačních oken přináší tato práce inovaci v podobě použití obyčejné kamery a tištěných výchozích značek, čímž se snižuje pořizovací cena celého rozhraní. Dále je zde nově použito napojení na správce oken Compiz, které umožňuje rychlou práci s „živými“ aplikačními okny. V další fázi projektu by bylo vhodné zaměřit se především na přidání nových možností interakce s různými dalšími předměty. Další možností by bylo sledovat lidské ruce v rámci projekční plochy a reagovat na jimi prováděná gesta.

Literatura

- [Ben08] Banerjee S., „*Projective geometry, camera models and calibration*“ Dept. Computer Science and Engineerin, IIT Delhi, 2008 [on-line]. Dostupný z WWW: <http://www.cse.iitd.ernet.in/~suban/vision/geometry.pdf> (leden 2010)
- [Bis92] D. Bishop, “*Marble Answering Machine*”, 1992 [online]. [cit. 7.12.2010] Dostupný z WWW: <http://design.cca.edu/graduate/uploads/pdf/marbleanswers.pdf> (prosinec 2010)
- [cam10] Allied Vision Technologies Canada : *GC1020 / GC1020C Technical Manual* . 2010 [online]. 1.11.2010 [cit. 15.12.2010]. Dostupný z WWW: http://www.alliedvisiontec.com/fileadmin/content/PDF/Products/Technical_Manual/GC_Series/700024A_-_GC1020_User_Manual.pdf (prosinec 2010)
- [Dur09] R. Duraiswami, “*Camera Calibration*“ , lecture for subject Scientific Computing on Graphics Processors, Department of computer science, University of Maryland , 2009 , [cit. 14.12.2010] Dokument dostupný na URL: <http://www.umiacs.umd.edu/~ramani/cmsc828d/lecture9.pdf> (prosinec 2010)
- [FIB95] Fitzmaurice, G.W., Ishii, H. & Buxton, W. Bricks “*Laying the Foundations for Graspable User Interfaces*”, in Proceedings of CHI'95 , 442-449. [online]. [cit. 8.12.2010]. Dostupný z WWW: <http://web.media.mit.edu/~anjchang/ti01/fitzmaurice-chi95-bricks.pdf> (prosinec 2010)
- [FHM09] G. Falcao, N. Hurtos, J. Massich, D. Fofi, “*Projector-Camera Calibration Toolbox*”, 2009. [on-line]. [cit. 15.12.2010] Dostupný z WWW: <http://code.google.com/p/procamcalib> (prosinec 2010)
- [FK03] Frahm J-M., Koch R., „*Camera calibration with known rotations*“. In IEEE International Conference on Computer Vision (ICCV), 2003. Dostupný z WWW: <http://portal.acm.org/citation.cfm?id=946736> (prosinec 2011)
- [Han10] Hanning T., “*High Precision Camera Calibration*“ , Vieweg+Teubner Verlag , Oct 2010, pp. 153-157
- [HVA+05] Holman, D., Vertegaal, R., Altosaar, M., Troje, N., and Johns, D. 2005. “*Paper windows: interaction techniques for digital paper*“ In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Portland, Oregon, USA, April 02 - 07, 2005). CHI '05. ACM, New York, NY, 591-599. [online] [cit. 9.12.2010]. Dostupný z WWW: http://portal.acm.org/ft_gateway.cfm?id=1055054&type=pdf&CFID=1426118&CFTOKEN=22266175 (prosinec 2010)
- [HZ04] R.Hartley and A. Zisserman, “*Multiple View Geometry in Computer Vision Second Edition*“ , Cambridge University Press, 2004, pp. 153-157 [cit. 14.12.2010]
- [ChISR09] Cheng, K., Itzsten, B., Sztajer, P., and Rittenbruch, M. (2009) “*A unified multi-touch & multi-pointer software architecture for supporting collocated work on the desktop*”, Technical Report ATP-2247, NICTA, Australian Technology Park, Sydney, Australia, September 2009. [online] [cit. 9.12.2010]. Dostupný z WWW: http://sydney.edu.au/engineering/it/~kcheng/2247_A_unified_2.pdf (prosinec 2010)

- [Ipa10] "iPad official site", Dostupný z WWW: <http://www.apple.com/ipad/> (květen 2011)
- [IU97] Ishii, H., and Ullmer, B. (1997). "Tangible Bits: Towards seamless interfaces between people, bits and atoms", Proceedings of CHI'97 Conference, ACM, 234-241. [online]. [cit. 9.12.2010]. Dostupný z WWW: www.sigchi.org/chi97/proceedings/paper/hi.htm (prosinec 2010)
- [JWF+09] Johnson T., Welch G., Fuchs H., la Force R., Towles H. 2009. „A Distributed Cooperative Framework for Continuous Multi-Projector Pose Estimation“. In *Proceedings of the 2009 IEEE Virtual Reality Conference (VR '09)*. IEEE Computer Society, Washington, DC, USA, 35-42. DOI=10.1109/VR.2009.4810996 Dostupný z WWW: <http://dx.doi.org/10.1109/VR.2009.4810996> (duben 2011)
- [Kat00] Kato, H., "ARToolKit" [on-line]. 2000 [cit. 16.12.2010] Dostupný z WWW: <http://www.hitl.washington.edu/artoolkit/> (prosinec 2010)
- [KNF09] Koike H., Nishikawa W., Fukuchi K., "Transparent 2-D markers on an LCD tabletop system", in Proc. CHI, 2009, pp.163-172. [on-line]. 2010 Dostupný z WWW: <http://portal.acm.org/citation.cfm?id=1518728> (květen 2011)
- [Kra06] E. Král, "Analýza obrazové informace kamerového systému", diplomová práce, Zlín, FAI UTB ve Zlíně, 2006 [on-line] cit. 14.12.2010] Dostupný z WWW: http://dspace.knihovna.utb.cz/bitstream/handle/10563/1370/kr%C3%A1l_2006_dp.pdf?sequence=1 (prosinec 2010)
- [Lee06] W. Lee "Technical report: Real Camera vs. Virtual Camera", 2006 Dostupný z WWW: <http://old.uvr.gist.ac.kr/wlee/web/techReports/ar/Camera%20Models.html> (březen 2011)
- [LW05] W. Lee, W. Woo, "TARBoard: Tangible Augmented Reality System for Table-top Game Environment" 2nd International Workshop on Pervasive Gaming Applications (PerGames2005), 2005 Dostupný z WWW: http://old.uvr.gist.ac.kr/wlee/web/papers/2005/PerGames_TARBoard_wlee.pdf (únor 2011)
- [Ocv09] OpenCV 2.0 C Reference, "Camera Calibration and 3D Reconstruction" [on-line], 2009, Dostupný z WWW: http://opencv.willowgarage.com/documentation/camera_calibration_and_3d_reconstruction.html (prosinec 2010)
- [RF06] Remondino, F., Fraser, C., 2006. "Digital camera calibration methods: considerations and comparisons." ISPRS Commission V Symposium 'Image Engineering and Vision Metrology' pp. 266-272 [cit. 15.12.2010] Dostupný z WWW: http://www.photogrammetry.ethz.ch/general/persons/fabio/Remondino_Fraser_ISPRS_V_2006.pdf (prosinec 2010)
- [RFK98] Rauterberg, M., M. Fjeld, H. Krueger, M. Bichsel, U. Leonhardt and M. Meier, "BUILD-IT: A Planning Tool for Construction and Design". In Summary of Conference on Human Factors in Computing Systems (CHI '98). New York: ACM [online]. [cit. 8.12.2010]. Dostupný z WWW: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.23.4135&rep=rep1&type=pdf> (prosinec 2010)

- [Ros01] Rosco Screens ,*"The Basics of Choosing a Screen"* [on-line]. 2010
[cit. 13.12.2010] Dokument dostupný na URL:
http://www.rosco.com/us/technotes/screens/technote_tbes.cfm (prosinec 2010)
- [RP04] A. Raij and M. Pollefeys. „*Auto-calibration of multi-projector display walls*“. Proc. of ICPR, 2004. Dostupný z WWW: www.cs.unc.edu/~marc/pubs/RaijICPR2004.pdf (únor 2011)
- [Sch07] Schmalstieg, D. , „*Studierstube Tracker*“ [on-line]. 207
[cit. 16.12.2010] Dostupný z WWW:
http://studierstube.icg.tu-graz.ac.at/handheld_ar/stbtracker.php (prosinec 2010)
- [SST09] Spindler, M., Stellmach, S., & Dachsel, R. (2009), „*PaperLens: advanced magic lens interaction above the tabletop*“ Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (p. 69-76). ACM.
[online]. [cit. 8.12.2010]. Dostupný z WWW:
<http://www.wisg.cs.uni-magdeburg.de/uise/Forschung/Publikationen/2009-ITS-PaperLens.pdf> (prosinec 2010)
- [Ste06] Štefanová, E. , „*Kalibrace neměřických digitálních kamer*“, Přírodovědecká fakulta, UK v Praze , [cit. 15.12.2010] Dostupný z WWW:
http://www.vugtk.cz/odis/sborniky/jine/geos06/paper/53_stefanova_e/paper/53_stefanova_e.pdf (prosinec 2010)
- [STM99] Streitz, N.A., Tandler, P., Muller-Tomfelde, C., Konomi, S. „*i-LAND: An Interactive Landscape for Creativity and Innovation*“, Proc. CHI 1999 , 120-127. [online].
[cit. 7.12.2010]. Dostupný z WWW:
<http://homepage1.nifty.com/konomi/shinichi/chi99Reprint.pdf> (prosinec 2010)
- [Svo09] Svoboda, T., „*Kalibrace kamery ze (známých) rotací*“ ,CVUT FEL, Centrum strojového vnímání , 2009 , [online]. [cit. 13.12.2010]. Dostupný z WWW:
http://cmp.felk.cvut.cz/cmp/courses/Y33ROV/Y33ROV_ZS20082009/Lectures/Geometry/calibrotation-print.pdf (prosinec 2010)
- [SWGZ07] Song,P.,Winkler,S., Gilani, S.O., Zhou, Z.Y., „*Vision-Based Projected Tabletop Interface for Finger Interactions*“,CVHCI07(49-58) [on-line]. 2007
[cit. 15.12.2010] Dostupný z WWW:
<http://www.springerlink.com/content/m41u5617p3816772/fulltext.pdf> (prosinec 2010)
- [SWS08] Schmalstieg, D., Waldner, M., Steinberger, M. , „*Deskothèque*“ , ICGV TU Graz , 2008 Dostupný z WWW:
<http://studierstube.icg.tu-graz.ac.at/deskothèque/#projects> (prosinec 2010)
- [Wal06] Manuela Waldner, „*Tangible Tiles: Ein Tangible User Interface zur Verwaltung und Manipulation von digitalen Daten*“ Master's Thesis August 2006 Dostupný z WWW:
http://manuela-waldner.net/download/Waldner_Diplomarbeit.pdf (květen 2011)
- [Wel93] Wellner, P., „*Interacting with Paper on the Digital Desk*“, ACM Comm , 36(7), 1993, 86-96. [online]. [cit. 7.12.2010]. Dostupný z WWW:
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=4360CAE333CA4759261C65B03B40589F?doi=10.1.1.53.7526&rep=rep1&type=pdf> (prosinec 2010)

- [wik01] Wikipedie: Otevřená encyklopedie: “*Tangible User Interface*” [on-line]. 2010 [cit. 8.12.2010] Dostupný z WWW: http://en.wikipedia.org/wiki/Tangible_User_Interface (prosinec 2010)
- [wik02] Wikipedie: Otevřená encyklopedie: “*Pinhole camera*” [on-line]. 2010 [cit. 14.12.2010] Dostupný z WWW: http://en.wikipedia.org/wiki/Pinhole_camera (prosinec 2010)
- [wik03] Wikipedie: Otevřená encyklopedie: “*Distortion (optics)*” [on-line]. 2010 [cit. 14.12.2010] Dostupný z WWW: http://en.wikipedia.org/wiki/Distortion_%28optics%29 (prosinec 2010)
- [wik04] Wikipedie: Otevřená encyklopedie: “*Homography*” [on-line]. 2010 [cit. 15.12.2010] Dostupný z WWW: <http://en.wikipedia.org/wiki/Homography> (prosinec 2010)
- [wik05] Wikipedie: Otevřená encyklopedie: “*ARToolKit*” [on-line]. 2010 [cit. 16.12.2010] Dostupný z WWW: <http://en.wikipedia.org/wiki/ARToolKit> (prosinec 2010)
- [wik06] Wikipedie: Otevřená encyklopedie: “*Compiz*” [on-line]. 2010 Dostupný z WWW: <http://cs.wikipedia.org/wiki/Compiz> (prosinec 2010)
- [wik07] Wikipedie: Otevřená encyklopedie: “*OpenCV*” [on-line]. 2010 Dostupný z WWW: <http://en.wikipedia.org/wiki/OpenCV> (prosinec 2010)
- [wik08] Wikipedie: Otevřená encyklopedie: “*OpenGL*” [on-line]. 2010 Dostupný z WWW: <http://en.wikipedia.org/wiki/OpenGL> (prosinec 2010)
- [wik09] Wikipedie: Otevřená encyklopedie: “*Reprojection error*” [on-line]. 2011 Dostupný z WWW: http://en.wikipedia.org/wiki/Reprojection_error (květen 2011)
- [wik10] Wikipedie: Otevřená encyklopedie: “*Augmented reality*” [on-line]. 2011 Dostupný z WWW: http://cs.wikipedia.org/wiki/Augmented_reality (květen 2011)

Seznam příloh

Příloha 1. Manuál - Kalibrační program

Příloha 2. Manuál - Řídící program pro TUI

Příloha 3. DVD

Příloha 1. Manuál - Kalibrační program

Program je závislý na frameworku Deskotheque, který je kompatibilní pouze s OS Linux. Takže nebylo možné vyzkoušet funkčnost pod jinými OS. Program byl úspěšně zkompileován a odzkoušen pod OS Ubuntu 10.04 LTS - Lucid Lynx. Nezbytnou součástí pro správný běh je připojená a správně nakonfigurovaná kamera a projektor. Při testování byla použita kamera Prosilica GC1020[cam10]. V podadresáři `lib/PvSDK-1.20/bin-pc/x64/` respektive `lib/PvSDK-1.20/bin-pc/x86/` se nachází zkompileovaný program `SampleViewer` pro otestování správného připojení a funkčnosti kamery typu Prosilica. Program byl také úspěšně odzkoušen s integrovanou webovou kamerou připojenou přes usb. Program by měl dále fungovat s libovolnou kamerou, ke které umí přistupovat knihovna OpenCV.

Kompilace – Linux

Pro úspěšné zkompileování programu je třeba nainstalovat následující knihovny a nástroje:

- program CMAKE 2.8 – Ubuntu balíčky:
 - `cmake`
- knihovna OpenGL – Ubuntu balíčky:
 - `freeglut3-dev`
- knihovna IL – Ubuntu balíčky:
 - `libdevil-dev`
- knihovna OpenCV – Ubuntu balíčky:
 - `libcv-dev`
 - `libhighgui-dev`
 - `libcvaux-dev`
- knihovna X Input / Xlib – Ubuntu balíčky:
 - `libxi-dev`

Pro kompilaci je připraven CMAKE skript. V kořenovém adresáři programu stačí zavolat příkaz „`cmake . && make`“. Spustitelný program je následně vygenerován v podadresáři s názvem `bin`.

Spouštění programu

Program je nutné spouštět s parametry následovně:

```
./cal <in_xml> <out_xml>
```

Popis parametrů:

- <in_xml> Název souboru, který obsahuje parametry pro konfiguraci programu.
- <out_xml> Název souboru pro ukládání nebo načítání výsledků kalibrace.

Příklad spuštění

V adresáři /bin zadejte následující příkaz:

```
./cal data/input/dctl-taget-05x07-50x50mm-3x3.xml data/output/cameraname_date.xml
```

Popis obsahu vstupních souborů

Soubor : <in_xml>

Tagy:

- Calibration - hlavní tag, obaluje všechny ostatní tagy
- Camera - obsahuje parametry pro nastavení kamery
 - Parametry:
 - id - identifikační číslo kamery pro výpis výsledků
 - name - název kamery pro výpis výsledků
 - width - rozlišení kamery v pixelech – šířka
 - height - rozlišení kamery v pixelech – výška
- Prosilica - nastavení kamery Prosilica, nastavuje se pouze při použití kamery tohoto typu, jinak se nastavuje následující tag CV
 - Parametry:
 - uid - unikátní identifikační číslo kamery typu Prosilica
- CV - nastavení kamery, která je podporována knihovnou OpenCV
 - Parametry:
 - index - identifikační číslo kamery
- Target - definice objektu, který má být detekován
 - Parametry:
 - id - identifikační číslo objektu

- type - typ objektu, možné hodnoty jsou CHECKER – šachovnice, CODED_MARKER – cíl poskládaný z kódových značek, které využívá knihovna Studierstube
- Params - parametry detekovaného objektu, liší se podle druhu objektu
 - Parametry – typ CHECKER:
 - rows - počet řádků vnitřních rohů šachovnice
 - cols - počet sloupců vnitřních rohů šachovnice
 - boarder - šířka okraje okolo šachovnice
 - width - šířka jednoho políčka šachovnice
 - heigth - výška jednoho políčka šachovnice
 - Parametry – typ CODED_MARKER:
 - rows - počet řádků vykreslovaných/detekovaných značek
 - cols - počet sloupců vykreslovaných/detekovaných značek
 - size - délka hrany značky v pixelech
 - origin_X – souřadnice x levého horního rohu první značky v rámci okna
 - origin_Y – souřadnice y levého horního rohu první značky v rámci okna
 - initial_marker_filename - název (včetně cesty) souboru obsahující obrázek první značky, musí obsahovat číslo, názvy ostatních souborů jsou generovány jako sekvence souborů se stejným názvem, ale jiným pořadovým číslem
 - treshold - hodnota prahu pro detektor kódových značek
 - boarder – šířka okraje okolo jednotlivých značek

Příklad souboru:

```
<?xml version="1.0" ?>
<Calibration>
  <Camera id="0" name="Prosilica 41074" width="1024" height="768">
    <Prosilica uid="41074" />
    nebo
    <CV index="0" />
  </Camera>
  <Target id="0" type="CHECKER">
    <Params rows="5" cols="7" border="0" width="50.0"
      height="50.0" />
  </Target>
  <Target id="1" type="CODED_MARKER">
    <Params rows="3" cols="3" size="100" origin_X="100"
      origin_Y="40"
```

```

        initial_marker_filename="./data/stb/bch/BchThin_0050.png"
        threshold="100" boarder="0.125"/>
    </Target>

</Calibration>

```

Tento soubor je možné nalézt v bin/data/input/dctl-taget-05x07-50x50mm-3x3.xml .

Soubor : <out_xml>

Tagy:

- Calibration - hlavní tag, obaluje všechny ostatní tagy
- Camera - obsahuje informace o kameře a obaluje tagy s parametry, které se vztahují ke kalibraci kamery
 - Parametry:
 - id - identifikační číslo kamery
 - name - název kamery
 - width - rozlišení kamery v pixelech – šířka
 - height - rozlišení kamery v pixelech – výška
 - type - typ použité kamery – hodnoty Prosilica nebo CV
 - uid - unikátní identifikační číslo kamery typu Prosilica
- Projector - obsahuje tagy s parametry, které se vztahují ke kalibraci projektoru
- Intrinsics - interní parametry kalibrovaného zařízení
 - Parametry:
 - fx - ohnisková vzdálenost násobená počtem pixelů na délkovou jednotku na ose x
 - fy - ohnisková vzdálenost násobená počtem pixelů na délkovou jednotku na ose y
 - cx - offset základního bodu na ose x
 - cy - offset základního bodu na ose y
- Distortion - koeficienty geometrického zkreslení
 - Parametry:
 - k1 - koeficient radiálního zkreslení
 - k2 - koeficient radiálního zkreslení
 - p1 - koeficient tangenciálního zkreslení
 - p2 - koeficient tangenciálního zkreslení
- Extrinsics - externí parametry kalibrovaného zařízení
 - Parametry:

- r_1, r_2, r_3 - vektor otočení mezi souřadnicovým systémem kalibrovaného zařízení a systémem světových souřadnic
- t_1, t_2, t_3 - vektor posunutí mezi souřadnicovým systémem kalibrovaného zařízení a systémem světových souřadnic

Příklad souboru:

```
<?xml version="1.0" ?>
  <Calibration>
    <Camera id="0" name="Prosilica 41074" width="1024"
      height="768" type="Prosilica" uid="41074">
      <Intrinsics fx="1035.955957" fy="1029.661561"
        cx="515.266390" cy="360.011990"/>
      <Distortion k1="-0.263910" k2="0.116981" p1="-0.001378"
        p2="-0.000279"/>
      <Extrinsics r1="-0.129828" r2="0.079489" r3="0.018113"
        t1="-25.709472" t2="-159.924239" t3="2387.796855"/>
    </Camera>
    <Projector>
      <Intrinsics fx="2397.765918" fy="2375.098523"
        cx="727.262427" cy="689.805311"/>
      <Distortion k1="0.071725" k2="-0.082701" p1="0.006148"
        p2="0.015688"/>
      <Extrinsics r1="-0.101477" r2="0.012587" r3="0.012318"
        t1="-142.231723" t2="-407.794347" t3="2571.733205"/>
    </Projector>
  </Calibration>
```

Tento soubor je možné nalézt v bin/data/output/dctl-taget-05x07-50x50mm-3x3-result.xml . Soubor je možné načíst do programu ,a tím přeskočit kalibraci kamery. Případně, pokud jsou vyplněny interní parametry(včetně geometrických zkreslení), lze získat pouze externí parametry obou zařízení vzhledem k detekované šachovnici. Pro samostatnou kalibraci kamery nemusí být zadány parametry pro cílový objekt typu CODED_MARKER a soubor s názvem <out_xml> nemusí existovat nebo může být prázdný.

Ovládání programu

Program je ovládán pomocí klávesnice. Význam jednotlivých kláves:

- p - P - přechod mezi režimem kalibrace kamery a projektoru
- c - C - výpočet interních parametrů zařízení z detekovaných hodnot
- t - T - přechod do/z testovacího režimu
- s - S - uložení všech vypočtených parametrů
- l - L - načtení výsledků ze souboru <out_xml> , pouze v režimu kalibrace projektoru
- d - D - přechod do/z režimu detekce, pouze v režimu kalibrace projektoru
- f - F - přepnutí okna z/do režimu celé obrazovky

Při kalibraci projektoru nelze libovolně přecházet mezi testovacím a detekčním režimem. Předchozí používaný režim je vždy nutné vypnout, aby bylo možné přejít do nového.

Běh programu

Po spuštění programu je automaticky načteno nastavení ze souboru <in_xml> . Pokud dojde ke správné inicializaci kamery, zobrazí se dvě okna. V prvním je obraz z kamery a v druhém se vykreslují kódové značky.

Kalibrace kamery

Program je po spuštění v režimu pro kalibraci kamery, přičemž je ihned zahájena detekce šachovnice. Vezměte vytištěnou šachovnici a pohybujte s ní před kamerou. Počet úspěšných rozpoznání cíle(= počet platných snímků) se zobrazuje na obrazovce. Doporučuje se získat 80-100 platných snímků. Po dosažení požadovaného množství snímků zapněte výpočet interních parametrů kamery. Výsledek lze otestovat v testovacím režimu, kde se zobrazuje obraz kamery bez geometrického zkreslení. Pokud jsou reálné přímky také v obraze zobrazeny jako přímky, je výsledek kalibrace v pořádku. Přičemž platí pravidlo, že čím více snímků je pořízeno během detekce, tím je výsledek kalibrace lepší. Pro uložení výsledků stiskněte písmeno S.

Kalibrace projektoru

Vytištěnou šachovnici připevněte na bílou desku. Deska musí být dostatečně velká , aby se na ní vešel promítaný vzor společně s šachovnicí a navzájem se nepřekrývali. Spustěte program a přejděte do režimu kalibrace projektoru. Okno, kde se zobrazují ID markery, přesuňte pro zobrazení na plátno a nastavte do režimu celé obrazovky. Pro správnou funkčnost musí být toto okno po celou dobu kalibrace aktivní. Umístěte kameru tak, aby zabírala celý obraz promítaný projektorem. Případně načtěte parametry z souboru <out_xml> pomocí písmena L. Spustěte detekci. Deskou pohybujte před projektorem tak, aby na ní byly vždy zobrazeny všechny promítané značky. Připevněná šachovnice

musí být vždy osvětlena celá projektořem. Platný snímek je zaregistrován, když je zároveň rozpoznána šachovnice a všechny zobrazované značky. Přitom jsou na místa detekovaných rohových bodů šachovnice promítnuty křížky. Doporučuje se získat 80-100 platných snímků. Po dosažení požadovaného množství snímků zapněte výpočet interních parametrů projektořu. Pro otestování výsledku a výpočet externích parametrů přejděte do testovacího režimu.

V testovacím režimu umístěte šachovnici do prostoru plátna. Doporučuje se umístit šachovnici co nejbliže k promítanému vzoru. Na základě prvního snímku, kde jsou rozpoznány značky společně s šachovnicí, jsou vypočteny externí parametry obou zařízení. Pro určení co nejpřesnějších externích parametrů je důležité, aby všechny rozpoznávané vzory byly v jedné rovině. Po výpočtu externích parametrů se přestane promítat vzor a začne rozpoznávání/sledování samostatné šachovnice. Pohybujte šachovnicí před projektořem. V případě, že je šachovnice rozpoznána, dojde k promítnutí křížků na její vnitřní rohy. Úspěšnost detekce lze vyhodnotit jako odchylku mezi promítanými křížky a skutečnými vnitřními rohy šachovnice. Výsledek celého procesu uložte pomocí písmene S.

Doporučení pro získání co nejpřesnější kalibrace projektořu:

- Čočka projektořu musí být posunuta na střed.
- Je vhodné použít co největší šachovnici. Např. vytisknout na papír o velikosti A3.
- Použít standardní expozici kamery. Pokud je obraz příliš světlý musí se upravit clona.
- Kamera musí být velmi dobře nakalibrována.

Příloha 2. Manuál - Řídící program pro TUI

Program je závislý na frameworku Deskothèque a správci oken Compiz, který je kompatibilní pouze s OS Linux. Takže nebylo možné vyzkoušet funkčnost pod jinými OS. Program byl úspěšně zkompilován a odzkoušen pod OS Ubuntu 10.04 LTS - Lucid Lynx a Compiz verze 0.8.4. Nezbytnou součástí pro správný běh je připojená a správně nakonfigurovaná kamera a projektor. Při testování byla použita kamera Prosilica GC1020[cam10]. V podadresáři lib/PvSDK-1.20/bin-pc/x64/ respektive lib/PvSDK-1.20/bin-pc/x86/ se nachází zkompilovaný program SampleViewer pro otestování správného připojení a funkčnosti kamery typu Prosilica. Program byl také úspěšně odzkoušen s integrovanou webovou kamerou připojenou přes usb. Program by měl dále fungovat s libovolnou kamerou, ke které umí přistupovat knihovna OpenCV.

Instalace – Linux

1. Kořenový adresář dktTangible překopírovat do složky \$HOME/src
2. Upravit nastavení proměnných prostředí pro načítání knihoven pro Compiz plugin podle následujícího postupu:
 - 2.1. Překopírovat soubor Tangible.env do adresáře \$HOME/src.
 - 2.2. Upravit bashrc skript, který je spouštěn při spuštění unixového shellu (např. konsole, gnome-terminal, bash) následovně:
 - a) Otevřít soubor bashrc např. pomocí příkazu „sudo gedit ~/.bashrc“.
 - b) Na konec souboru bashrc přidat následující kód kde řetězec "\$HOME/src" musí odkazovat na umístění souboru Tangible.env (případně je nutné upravit podle aktuálního umístění):

```
export base="$HOME/src"
## Execute specific Tangible project ENV file
echo "--- Executing $base/Tangible.env"
if [ -f $base/Tangible.env ]; then
    cd $base
    . Tangible.env
    cd - > /dev/null
fi
echo "--- $base/Tangible.env Finished."
```

- c) Uložit soubor bashrc.

Kompilace – Linux

Pro úspěšné zkompileování programu je třeba nainstalovat stejné knihovny jako v případě kalibračního programu. Navíc je třeba nainstalovat následující knihovny a nástroje:

- Compiz – Ubuntu balíčky:
 - compiz
 - compiz-plugins
 - compiz-core
 - compiz-dbg
 - compiz-gnome
 - compiz-fusion-bcop
 - compiz-fusion-plugins-main
 - compiz-dev
 - compizconfig-settings-manager
- Xlib – Ubuntu balíčky:
 - libxtst-dev
 - libxtst6
- Logování – Ubuntu balíčky:
 - liblog4c-dev
 - liblog4c3

Pro kompilaci je připraven CMAKE skript. V kořenovém adresáři programu stačí zavolat příkazy „cmake . && make“, čímž se vytvoří sdílená knihovna pro zpracování obrazu a výpočet transformací s názvem libDktTang.so a spustitelný testovací program s názvem tang v podadresáři s /bin.

Pro kompilaci pluginu pro Compiz, který je v podadresáři Plugin je nutné přejít do adresáře /Plugin/build a zavolat příkaz „cmake .. && make install“. Tím se plugin nainstaluje pro použití s Compizem. Pro správnou kompilaci pluginu je nutné mít vytvořenou uvedenou knihovnu libDktTang.so.

Spouštění testovacího programu

Program je nutné spouštět s parametry následovně:

```
./tang <calibration_xml> <tangibles_xml>
```

Popis parametrů:

- `<calibration_xml>` Název souboru, který obsahuje kalibrační parametry kamery a projektoru. XML formát obsahu souboru je shodný s výstupním souborem kalibračního programu popsáným v příloze číslo 1.
- `<tangibles_xml>` Název souboru obsahujícího xml definici použitých objektů a značek.

Příklad spuštění

V adresáři /bin zadejte následující příkaz:

```
./tang data/input/1-exampleCV.xml data/input/tangibles1.xml
```

Popis obsahu vstupních souborů

Soubor : `<tangibles_xml>`

Definované délkové hodnoty se uvádějí jednotkách daných nastavením při kalibraci.

Tagy:

- Tangibles - hlavní tag, obaluje všechny ostatní tagy
- Tangible - definuje objekt sledovaný v rámci rozhraní, obsahuje definice značek použitých pro tento objekt
 - Parametry:
 - `type` - typ objektu, může nabývat hodnot:
 - `window` - deska pro manipulaci s okny, nutná definice parametrů `width` a `height`, `uid`
 - `drop` - značka jejíž detekce signalizuje odpojení okna od desky, nutná definice parametru `link_uid`
 - `pen` - dotykové pero, nutná definice parametru `button`
 - `width` - šířka promítaného okna
 - `height` - výška promítaného okna
 - `uid` - unikátní identifikační číslo objektu pro párování
 - `link_uid` - číslo odkazující na objekt na tímto `uid` pro tvoření vazeb mezi objekty
 - `button` - typ simulovaného tlačítka myši, může nabývat hodnot:
 - `L` - levé tlačítko myši
 - `R` - pravé tlačítko myši
- Target - tag obalující definici jednotlivých značek
- Marker - definice konkrétní použité značky

- Parametry:
 - type - typ značky, může nabývat hodnot:
 - BCH - ID marker
 - FRAME - Frame marker
 - id - identifikační číslo značky
 - size - délka hrany použité čtvercové značky
 - rotation - úhel pootočení značky ve stupních, osa otáčení je umístěna uprostřed značky a je na ní kolmá – volitelný parametr
- Offset - definice posuvu značky vzhledem k výchozímu bodu jejího souřadného systému – volitelný tag
 - Parametry:
 - x - posuv na ose x
 - y - posuv na ose y
 - z - posuv na ose z
- UpDirection - tag pro definici prostorového natočení značky vzhledem k jejímu výchozímu bodu, tag konkrétně definuje vektor kolmý na vrchní stranu značky
 - Parametry:
 - x - hodnota složky vektoru pro osu x
 - y - hodnota složky vektoru pro osu y
 - z - hodnota složky vektoru pro osu z
- RightDirection - tag pro definici prostorového natočení značky vzhledem k jejímu výchozímu bodu, tag konkrétně definuje vektor kolmý na pravou stranu značky (při pohledu shora)
 - Parametry:
 - x - hodnota složky vektoru pro osu x
 - y - hodnota složky vektoru pro osu y
 - z - hodnota složky vektoru pro osu z

Příklad souboru:

<Tangibles>

```

<Tangible type="window" width="235.0" height="235.0" uid="7">
  <Target>
    <Marker type="FRAME" id="7" size="281.0"/>
  </Target>
</Tangible>

```

```

<Tangible type="drop" link_uid="7">
  <Target>
    <Marker type="BCH" id="113" size="135.0"/>
  </Target>
</Tangible>
<Tangible type="pen" button="L" >
  <Target>
    <Marker type="BCH" id="41" size="125.0">
      <Offset x="0" y="305" z="0"/>
    </Marker>
  </Target>
</Tangible>
<Tangible type="pen" button="R">
  <Target>
    <Marker type="BCH" id="42" size="91.0">
      <Offset x="0" y="313" z="0"/>
    </Marker>
  </Target>
</Tangible>
<Tangible type="window" width="68" height="68" uid="2">
  <Target>
    <Marker type="BCH" id="21" size="80.0">
      <Offset x="0" y="0" z="50"/>
      <RightDirection x="1" y="0" z="0"/>
      <UpDirection x="0" y="1" z="0"/>
    </Marker>
    <Marker type="BCH" id="22" size="80.0">
      <Offset x="0" y="-50" z="0"/>
      <RightDirection x="1" y="0" z="0"/>
      <UpDirection x="0" y="0" z="1"/>
    </Marker>
    <Marker type="BCH" id="23" size="80.0">
      <Offset x="50" y="0" z="0"/>
      <RightDirection x="0" y="1" z="0"/>
      <UpDirection x="0" y="0" z="1"/>
    </Marker>
  </Target>
</Tangible>

```

```

<Tangible type="drop" link_uid="2">
  <Target>
    <Marker type="BCH" id="111" size="135.0"/>
  </Target>
</Tangible>
</Tangibles>

```

Ukázkový soubor je možné nalézt v bin/data/input/tangibles1.xml .

Ovládání testovacího programu

Program je ovládán pomocí klávesnice. Význam jednotlivých kláves:

- f - F - přepnutí okna z/do režimu celé obrazovky

Běh testovacího programu

Po spuštění programu je automaticky načteno nastavení ze souborů daných parametry. Následně se zobrazí dvě okna, kde v prvním okně je obraz z kamery a v druhém pouze bílé pozadí. Přesuňte druhé okno na pracovní plochu, která je zobrazována projektorem a stiskněte tlačítko F pro zvětšení/roztážení na celou obrazovku. Poté vezměte desky se značkami a vložte je do scény. Na desky by se měli začít promítat obrázky. Pokud se nic nepromítá bude pravděpodobně chyba v definičním XML souboru, např. je špatně definováno id použitých značek, případně mohou být chybné kalibrační parametry.

Spouštění pluginu pro Compiz

Plugin je doporučeno spouštět v oddělené session X serveru. Novou X session je možné spustit např. příkazem „sudo startx -- :1“. V nové session je vhodné spustit Compiz v debug módu pro zobrazení výpisů informujících o běhu pluginu. Restartování do debug módu lze provést příkazem „compiz --replace --debug“.

Plugin je lze nastavit a aktivovat v Manažeru nastavení compizu (spustitelném např. příkazem „ccsm“), kde je k nalezení po správné instalaci pod označením Tangible. Před aktivací pluginu je nutné nastavit konfigurační soubory obsahující kalibrační parametry kamery a projektoru a definici použitých objektů a značek viz. <calibration_xml> <tangibles_xml> v sekci Spouštění testovacího programu. Správnost souborů je vhodné ověřit jejich použitím s testovacím programem.

Nastavení pluginu

V Manažeru nastavení Compizu je možné upravovat nastavení vytvořeného pluginu po kliknutí na ikonu s názvem Tangible.

Nastavení v sekci General

- Update Delay - Zpoždění mezi aktualizací/překreslováním obrazovky.
- Output screen - Compiz screen, na které se mají provádět transformace oken. Compiz screen pod sebou zahrnuje tzv. Devices, které odpovídají konkrétním zobrazovacím zařízením jako monitor, projektor. Tento parametr bude ve většině případů nastaven na screen 0.
- Output device - Číslo konkrétního zařízení na jehož zobrazované ploše má docházet k transformacím aplikačních oken. Např. pokud je projektor nastaven jako druhé zařízení (monitor je jako první) bude tento parametr nastaven na Device 1.
- Calibration parameters - Název a cesta k souboru, který obsahuje kalibrační parametry kamery a projektoru. XML formát obsahu souboru je shodný s výstupním souborem kalibračního programu popsaným v příloze číslo 1.
- Tangible objects definition - Název a cesta k souboru obsahujícím xml definici použitých objektů a značek.

Nastavení v sekci Windows

- Windows to be used - Nastavení typů aplikačních oken, která se mají napojovat na desky označené značkami. Může nabývat hodnot např. Dialog, Normal, Menu, Notification.
- Rendering scale - Nastavení změny měřítka transformovaných oken. Pokud je parametr nastaven na 0 měřítko se nebude měnit.
- Pose filter threshold - Nastavení prahu pro filtrování souřadnic značek podrobnosti viz. kapitola 6.2 sekce Sledování značek. Nastavením velmi vysoké hodnoty se nevyfiltrují žádné souřadnice – filtr se vypne.
- Pose change threshold - Nastavení prahu pro filtrování souřadnic značek. Nové souřadnice značky jsou použity jen pokud se liší minimálně o hodnotu zvoleného prahu od souřadnic, které byly použity naposledy. Nastavením hodnoty na 0 se povolí použití každých nově vypočtených souřadnic. Odstraňuje chvění obrazu v rozsahu 1-2 pixelů způsobené použitím téměř totožných souřadnic.
- Invisible period - Transformované aplikační okno je zneviditelněno pokud není po tuto dobu rozpoznána deska, ke které je okno napojeno.
- Touch distance threshold - Nastavení prahu tolerance při detekci průniku dotykového pera a projekční roviny nebo desky.

Běh pluginu

Po správném nastavení a spuštění pluginu by mělo ihned fungovat „nabírání“ oken na desky opatřené značkami. Všechna nastavení je možné měnit i při aktivovaném/spuštěném pluginu. Pokud systém nefunguje je třeba překontrolovat správnost všech popsaných nastavení.

Příloha 3. DVD

Obsah DVD:

- Zdrojové soubory a spustitelná verze vytvořených programů.
- Plakát prezentující tuto práci.
- Programová dokumentace.
- Text této práce.
- Video prezentující interakci s vytvořeným TUI.
- Sledované značky pro tisk.