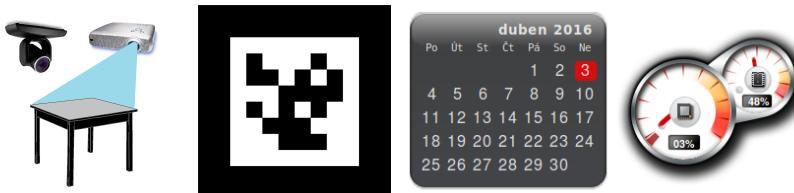


Rozšířené uživatelské rozhraní

Vladík Ondřej*



Abstrakt

Cílem této práce je navrhnout a implementovat Rozšířené uživatelské rozhraní, které bude využívat systému kamera-projektor a implementace knihovny funkcí pro poloautomatickou kalibraci kamery a celého systému. Rozhraní funguje tak, že na pracovní ploše (stůl, případně stěna nebo tabule) jsou rozmístěny různé značky. Kamera tyto značky snímá ve videosekvenci. Ze snímků se určí pozice značky v prostoru. Projektor bude promítat prvky uživatelského rozhraní na tuto pracovní plochu ke značkám. Promítané prvky v tomto případě představují widgety, tyto aplikace většinou reprezentující věci na pracovním stole nebo zobrazující různé informace o systému. Kalibrace kamery a celého systému je implementována samostatně, aby mohla být případně použita i pro jiné účely, než pro tuto práci. Celá je napsána v jazyce C++ a implementována formou pluginu do správce oken. Aktuálně autorem probíhá testování celého systému.

Klíčová slova: Kamera-projektor — Kalibrace kamery — Widget

Přiložené materiály: N/A

*xvladi02@fit.vutbr.cz, Fakulta informačních technologií v Brně

1. Úvod

Uživatelé se v této době stále více zajímají o alternativní ovládací prvky. Mezi takové prvky patří například senzor Leap Motiona rozhraní využívající kameru a projektor. Rozhraní využívající tyto prvky se mohou ovládat různými způsoby, pohyby těla uživatele, gesty rukou nebo pomocí různých značek. Účelem těchto rozhraní je rozšířit možnosti klasického počítače. Využití je velké. Rozhraní je možné využít pro prezentace, studování různých modelů nebo hraní her.

Tato práce se zabývá návrhem rozhraní, které využívá systému kamera-projektor. Rozhraní by sloužilo pro ovládání widgetů, které by se promítaly k papírovým značkám. Těmito značkami by uživatel libovolně pohyboval po pracovní ploše a widgety by se pohybovaly s nimi.

Pro jeho použití je nutné mít všechny komponenty tohoto rozhraní správně nakalibrované. Kalibrace se dá

rozdělit na dvě části:

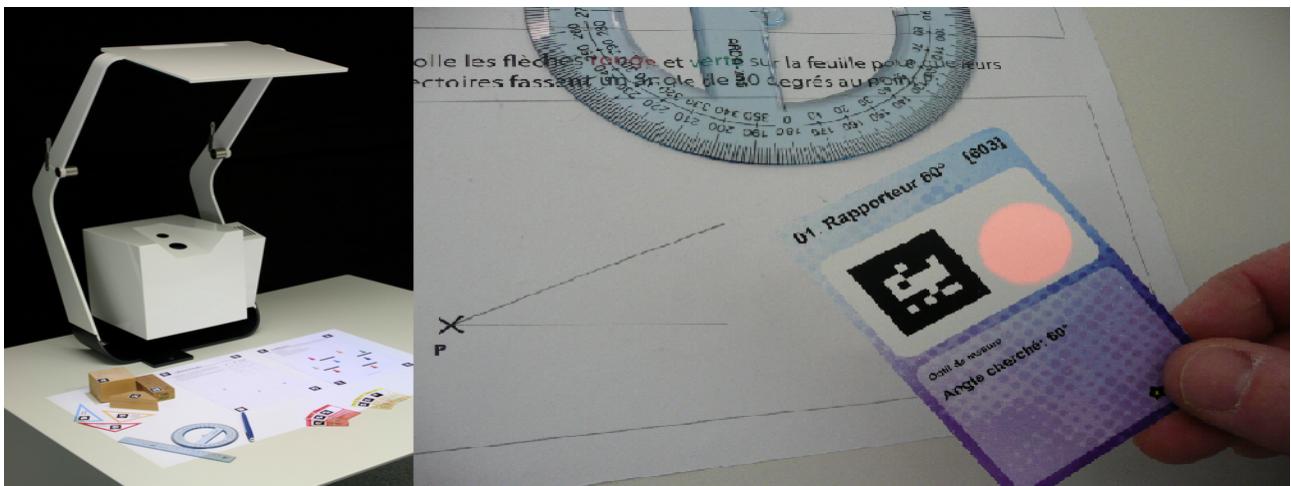
- Kalibrace kamery
- Kalibrace projektoru

Jednotlivé části kalibrace jsou popsány v dalších sekcích článku.(3)(4)

Při tvorbě tohoto rozhraní bylo použito více knihoven, především knihovna počítačového vidění OpenCV [1], opengl, knihovna pro detekci značek ARToolkit [2]. Pro správu widgetů je použit Manager Screenlets.

2. Podobné práce

Rozhraní, které využívají kamery a projektor je mnoho. Většina ovšem nepoužívá tento typ značek a jsou ovládány pomocí gest rukou nebo pomocí objektů, které obsahují různá elektronická zařízení, pro jejich detekci.



Obrázek 1. Projekt Rozšířený papír. Vlevo celé zařízení s pracovní plochou. Vpravo ukázka kontroly zadaného úkolu.

Proto je zde ukázáno existující řešení, které je této práci nejblíže.

Rozšířený papír

[3] Rozhraní využívá systému kamera-projektor a je určeno především do školního prostředí. Kamera snímá značky na pracovní ploše, podle druhu značky a její pozice mezi ostatními značkami se promítají různé geometrické obrazce a hodnoty. Kamera i projektor jsou umístěny do jedno zařízení a snímání a promítání je umožněno přes zrcadlo. Díky tomu je celé zařízení velice dobře přenositelné, například mezi učebnami školy.

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

3. Kalibrace kamery

Pomocí kalibrace jsou získány interní a externí parametry této kamery. Tyto parametry jsou nutné pro výpočet prostorové pozice objektu, o známé velikosti, vzhledem ke kameře. „[4] Kalibrace probíhá tak, že, kamera provádí mapování mezi 3D světovým metrickým systémem a 2D souřadným systémem obrazu v pixelech. Aby bylo možné provést inverzní proces, tedy spočítat 3D metrické souřadnice z 2D pixelů, je tedy nutné znát interní a externí parametry této kamery.“ Kalibrace kamery byla celá samostatně implementována a je plně funkční.

Interní parametry

Následující rovnice ukazuje matici interních parametrů: [4]

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

Znázorňuje tyto parametry:

- α_x, α_y - Ohnisková vzdálenost vyjádřená v pixelzech.
- u_0, v_0 - Reprezentují polohu hlavního snímkového bodu.
- γ - Koeficient radiálního a tangenciálního zkreslení.

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

Externí parametry

Těmito parametry se vyjadřuje transformace mezi souřadným systémem kamery a světovým souřadným systémem, vůči němuž je kamera kalibrována. Vztah pro vyjádření pozice kamery (2), kde C označuje pozici kamery. R vyjadřuje rotaci a T označuje pozici počátku souřadnic reálného systému vyjádřené v souřadnicích souřadného systému kamery.[5]

$$C = -R^{-1}T \quad (2)$$

4. Kalibrace projektoru

Tato kalibrace se provádí, protože se souřadnice objektu v obou rovinách liší. Aby je bylo možné převádět

je nutné znát polohu kamery i projektoru. Díky tomu, že seví, že plocha kam se bude obraz promítat je zcela rovná, není nutné znát prostorové umístění vzhledem ke kameře a je možné provést kalibraci projektoru pomocí homografie.

Kalibrace projektoru není stále plně funkční.

Homografie

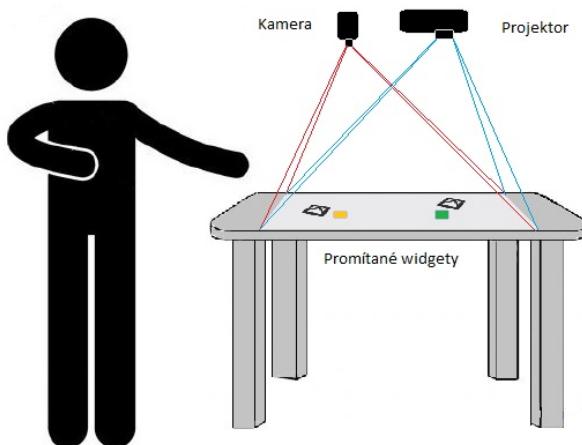
Homografie se znázorňuje maticí H. Tato matice vyjadřuje projektivní transformaci bodů mezi rovinami. Díky tomu je možné vypočítat souřadnice nějakého bodu v obraze projektoru ze souřadnic téhož bodu v obraze kamery. Výpočet probíhá s množinou bodů, u kterých jsou zadány souřadnice z jedné i druhé roviny.

5. Návrh rozhraní

Nevhodnější rozvržení celého systému je ve vodorovné poloze. Kamera a projektor jsou pevně umístěny nad stolem. Na kterém leží značky, ke kterým se projektem promítají widgety.

Před prvním spuštěním se provede kalibrace celého systému, která se už opakovat nemusí, pokud se nebude hýbat s projektem nebo kamerou. Potom si uživatel zvolí jaké widgety mohou být promítány. Jakou značku bude každý widget představovat. Poté se aplikace spustí.

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



Obrázek 2. Ukázka rozhraní. Kamera snímá pracovní plochu, na které jsou rozmištěny značky. Projektor promítá widgety na tuto plochu ke značkám.

Hardwareová sestava

Hlavní části systému:

- Kamera -Snímkuje pracovní plochu.

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

Značky

Značky neboli také markery je mnoho typů. [4] Artoolkit knihovna [2] používá značky typu tzv. template marker. Jsou to černobílé čtverce se specifickými rozměry a určitým obrazcem uvnitř.

Tyto značky jsou používány kvůli jednoduché detekci i při zhoršené kvalitě obrazu.

Tyto značky mají několik pravidel pro případné vytváření vlastních:

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



Obrázek 3. Vzorová značka knihovny ARToolkit.

Widgety

Widgety jsou GUI aplikace reprezentující věci, které se většinou nachází na psacím stole, jako jsou kalendář, hodiny, pravítko. Kromě nich jsou to i aplikace zobrazující stav a informace o systému nebo také například počasí nebo právě přehrávanou hudbu. Jsou zobrazovány v pozadí na ploše a je možné s nimi libovolně hýbat podle potřeb uživatele. Jejich velkou výhodou je mít tyto věci neustále po ruce a ušetření místa na pracovním stole.[6]

Programů pro jejich správu je více. Mezi nejpoužívanější na OS Linux patří Screenlets[7]. Je to jednoduchý nástroj pro jejich zobrazení a práci s nimi, umožňuje jednotlivé widgety vypínat, zapínat nebo také například nastavit, aby se spouštěli po přihlášení počítače.



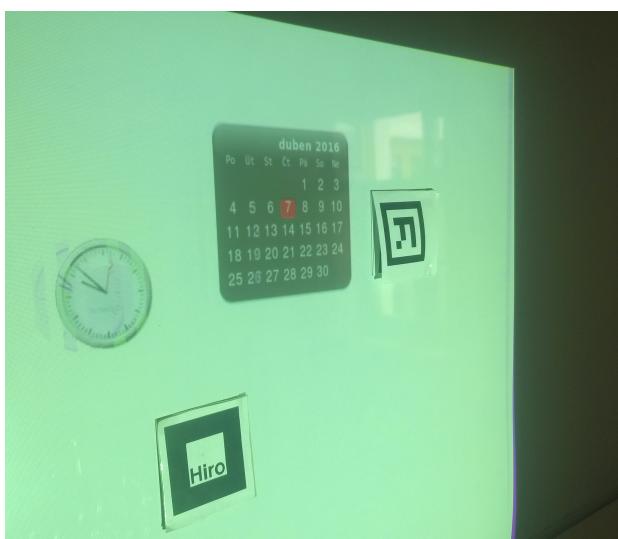
Obrázek 4. Ukázka několika widgetů.

6. Testování a vyhodnocení

Testování probíhá několika formami. Jedním z prvních je zjištění, jak velký vliv má počet snímků při kalibraci kamery a jaký je jejich nejvhodnější počet. Aby byla kamera nejlépe nakalibrována.

Pro rozhraní je velice důležité změřit, jak velká je odezva mezi pohnutím značky a promítaným widgetem. Tuto odezvu je dobré změřit při různé zátěži celého systému. Při pohybu jen jednou značkou, mnoha značkami, hodně rychlými pohyby se značkou.

Proto aby kamera detekovala značky, je vhodné určit velikost, jakých mohou widgety být. Pro detekci je také potřeba ověřit vliv barvy pozadí, které promítá projektor spolu se značkami.



Obrázek 5. Ukázka jednoho z prvních spuštění celé aplikace. Promítací plocha je tabule na zdi.

Výsledky

Tyto výsledky jsou pouze z prvních pokusů testování, které byly uskutečněny, bez kalibrace projektoru.

Při testování běhu celé aplikaci byla určena vhodná barva pozadí plochy. Bílá barva, kromě ohrazení plochy, zaručuje pro kameru lepší detekci značek. Zároveň není tak potřebné další osvětlení pro tuto plochu nebo celou místnost.

Odezva mezi pohybem značky a následným pohybem widgetů se pohybuje mezi 0,1 až 0,4s. Největší zpoždění se uskuteční po pohybu se značkou mimo pracovní plochu pro detekci a jejím opětovném navrácení. Mezi příčiny patří počet aktuálně promítaných widgetů, vytížení procesoru, zpoždění při načítání snímků.

Velikost značek je hlavně závislá na rozlišení kamery. Při testování byla použita kamera s rozlišením 640x480 pixelů a značky o velikosti 4x4cm. Pracovní plocha byla umístěna do 1 metru. Pro tuto vzdálenost byla velikost značek dostačující. Pro případné větší vzdálenosti je doporučena kamera s větším rozlišením nebo značky větší velikosti.

Testování kalibrace kamery a projektoru není stále dokončené, důvodem je nedokončená kalibrace projektoru.

7. Závěr

Cílem této práce bylo navrhnout a implementovat uživatelské rozhraní, které by využívalo systému kamera-projektor a kalibraci celého tohoto systému. Ve (2) kapitole je ukázáno existující řešení podobného typu rozhraní. V kapitole (3) a (4) byla popsána kalibrace kamery i projektoru. Další kapitola (5) se věnuje návrhu rozhraní a popisu jednotlivých částí. Kapitola (6) se věnuje testování, které stále probíhá, jejich formám a současným výsledkům.

Rozhraní bylo implementováno. Podle prvních testů je pohyb widgetů skoro plynulý, samozřejmě s určitým zpožděním a občasným chvěním. Detekce značek probíhá v pořádku, při správném osvětlení, pokud není bílé pozadí.

Kalibrace kamery je také implementována, ale ještě není dokončeno testování. Kalibrace projektoru není stále dokončena.

Tato práce uživateli rozšiřuje pracovní plochu počítače. Je to další příklad využití tohoto typu rozhraní. Usnadní uživateli práci na počítači a rozšíří možnosti jeho ovládání.

Poděkování

Rád bych chtěl poděkovat panu Ing. Jiří Zahrádkovi za odborné vedení a cenné rady při tvorbě této práce.

Literatura

- [1] G. Bradski. *Dr. Dobb's Journal of Software Tools*.
<http://opencv.org/>.
- [2] Dr. Hirokazu Kato, March 1999. <https://www.hitl.washington.edu/artoolkit>.
- [3] Pierre Dillenbourg Quentin Bonnard, Frédéric Kaplan, 2012. <http://craft.epfl.ch/lang/en/PaperTangibleInterface>.
- [4] Ing. Jiří Zahrádka. *Rozšířené uživatelské rozhraní*. Brno, 2011.
- [5] „Camera resectioning“ Wikipedie: Otevřená encyklopédie, 2016. https://en.wikipedia.org/wiki/Camera_resectioning.
- [6] „Widget (GUI)“ Wikipedie: Otevřená encyklopédie, 2014. [https://en.wikipedia.org/wiki/Widget_\(GUI\)](https://en.wikipedia.org/wiki/Widget_(GUI)).
- [7] Screenlets, 2013. <http://www.noobslab.com/2013/09/desktop-screenlets-for-ubuntulinux.html>.