

# Západočeská univerzita v Plzni

Semestrální práce předmětu ZDO

Rozšířená realita v medicíně



**Vypracoval:**  
MICHAL KUBÁT  
4. června 2014

# 1 Zadání

Tato práce navazuje na zadání z předmětu Projekt 5, kde bylo cílem vytvořit prostředí pro rozšíření reality v medicíně pomocí Kinectu a digitálního projektoru. Těžištěm bylo poskládání různých technologií do funkčního celku a vytvoření kalibrace pro nastavení vzájemné polohy snímače Kinect s projektorem. Na tělo stojícího či pohybujícího se člověka je pak promítán referenční bod resp. obraz.

Hlavním cílem této semestrální práce je vytvořit automatickou kalibraci, která by nahradila předchozí "ruční", značně nestabilní řešení.

## 2 Řešení

### 2.1 Použitý hardware

- **Kinect** je vstupní zařízení pro snímání pohybu od firmy Microsoft pro Xbox 360 a PC. Kinect je schopen číst souřadnice bodů jednotlivých částí lidského těla (např: hlava, krk, ramena, ruce, adt.)
- **Digitální projektor**
- **Lokální PC** - běží na něm Kinect Server, pro posílání dat.
- **PC** - běží na něm program pro kalibraci a promítání bodu.

### 2.2 Použitý software

- **Web Socket** je komunikační technika založená naplně duplexní komunikaci po spojení v protokolu TCP/IP. Programátorské rozhraní (API) pro webové sockety je standardizováno sdružením W3C. Umožňuje lepší interaktivitu mezi prohlížečem a serverem, hodí se pro komunikaci v reálném čase.
- **Python** je dynamický intepretovaný programovací jazyk. Je to hybridní jazyk, umožňuje při psaní programů používat nejen objektově paradigma, ale i procedurální a v omezené míře i funkcionální, podle toho co komu vyhovuje nebo se pro danou úlohu hodí nejlépe. Python má díky tomu vynikající vyjadřovací schopnosti. Kód programu je ve srovnání s jinými jazyky krátký a dobře čitelný.
- **Pygame** je multiplikativní sada modulů jazyka Python určená k tvorbě počítačových her. Obsahuje knihovny pro práci s grafikou, zvukem a vstupními zařízeními.
- **GitHub** je samostatně použitelný systém pro správu repozitářů a umožňuje efektivní práci s velkými projekty. Jedná se o mocnou podporu pro nelineární vývoj projektu. Podporuje rychlé vytváření a rychlé slučování. Git poskytuje každému vývojáři lokální kopii celé historie vývoje. Změny se kopírují z jednoho úložiště do jiného. Úložiště může

být zveřejněno prostřednictvím protokolů HTTP, FTP nebo protokolu Git realizovaného buď přes obyčejné sockety nebo přes ssh.

- **OpenCV** je multiplatformní knihovna pro manipulaci s obrazem. Je zaměřena především na počítačové vidění a zpracování obrazu v reálném čase. Původně jí vyvíjela společnost Intel.
- **wsclient** je webová aplikace, která umožňuje zobrazovat aktuální data z Kinectu. Při odeslání zprávy *skeleton* zobrazí jednotlivé souřadnice částí lidského těla, které Kinect snímá. Pokud je odeslána zpráva *rgb*, dostaneme obrázek z kamery i jeho maticový popis. Daný obrázek je pak možno jednoduše stáhnout. Rovněž lze zobrazit hloubkovou mapu.

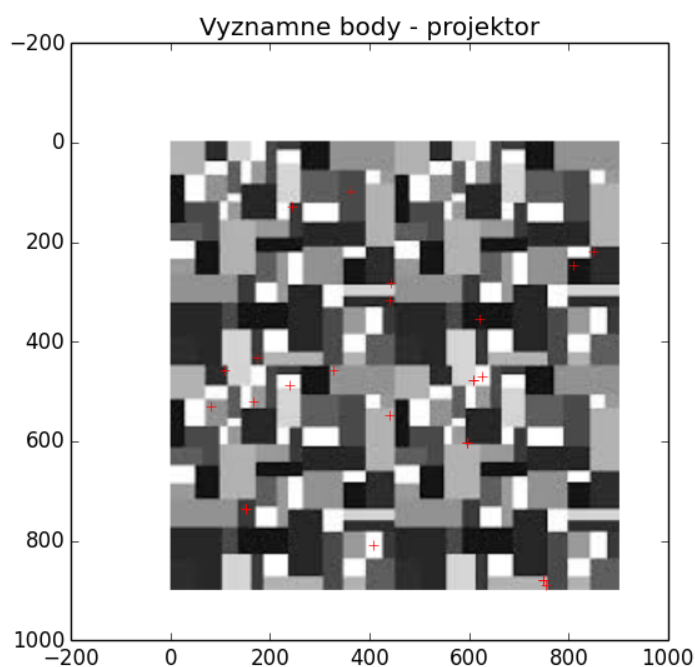
## 2.3 Vytvořené skripty

### 2.3.1 Fakeserver

Tento skript má pouze jediný úkol, a to spouštět se a běžet současně s hlavním programem *sledovani* a generovat mu falešné body, které by jinak získával z Kinectu. Fakeserver je použit pro spuštění *sledovani* "offline", tedy při absenci Kinectu.

### 2.3.2 Automatic calibration

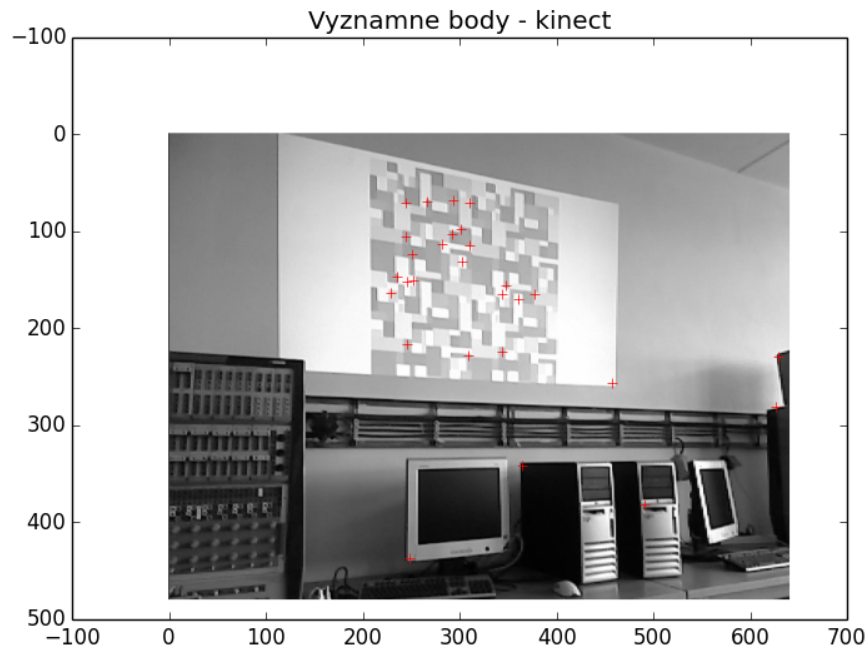
Hlavním skriptem vytvořeným v rámci této semestrální práce je *automatic calibration*. Vstupem tohoto programu jsou dva obrázky. Jedním je obraz promítaný projektorem na zeď, který je vhodně zvolený pro snadné nalezení významných bodů. Na obrázku 1 můžeme vidět jeden takovýto obraz s nalezenými významnými body.



Obrázek 1: Obraz promítaný projektorem s vyznačenými nalezenými významnými body

Druhým vstupem je obraz zachycený pomocí RGB kamery Kinectu. Pro jeho získání byl použit výše uvedený *wsclient*. Výsledný obraz opět s vyznačenými významnými body je na obrázku 2. Vstupy jsou nastavovány ve skriptu *config*.

Výstupem je transformační matice, vypočtená pomocí funkce *RANSAC*. Tato matice je potom použita ve skriptu *sledovani* pro kalibraci souřadnic snímaných bodů získaných Kinectem.



Obrázek 2: Obrázek získaný z Kinectu s vyznačenými významnými body

### 2.3.3 Sledovani

Vstupy jsou opět nastavovány ve skriptu *config* a jsou popsány v následující části. Tento skript tedy čte souřadnice požadovaných bodů podle volby buď z Kinectu nebo prostřednictvím *fakeserveru*. Souřadnice těchto bodů jsou poté pomocí funkce *getImageCoordinatesFromSkeletonCoordinates* přepočítány z toho důvodu, že Kinect používá různé rozlišení pro snímání souřadnic částí lidského těla než pro RGB kameru, pomocí níž byl pořízen obraz pro automatickou kalibraci. Dále je provedena kalibrace snímaných bodů pomocí vypočtené kalibrační (transformační) matice. Tyto body jsou pak podle nastavení ve skriptu *config* vykresleny buď jako prosté body obrázek nebo gif.

### 2.3.4 Config

Skript *Config* slouží pouze k nastavení jednotlivých konstant, vstupních obrázků a požadovaných módů programu. V části *automatic calibration* volíme již zmíněné vstupní obrázky a minimální počet vzájemně si odpovídajících nalezených významných bodů. V části *sledovani* zadáváme adresu kinect serveru, periodu čtení dat z Kinectu, požadované rozměry okna pro vykreslování a zobrazované obrázky. Dalším prvkem je volba složky s jednotlivými obrazy, které chceme promítat a volba adresáře se složkami sekvencí obrazů (promítání gifů). Při volbě *MODE* = "demo" nebudou souřadnice požadovaných bodů snímány Kinectem, ale generovány pomocí skriptu *fakeserver*. V tomto režimu rovněž nebude použita kalibrace

bodů. Dále volíme možnost otáčení a změny velikosti promítaného obrázku a vykreslení bodů snímaných jako torso, krk nebo hlava. Na oblast těla je pak možno zvolit promítat buď jednotlivé obrazy přepínané pomocí mezerníku nebo sekvenci obrazů, kde pomocí mezerníku přepínáme adresáře se sadami obrazů.

## 2.4 Postup spuštění aplikace

Nejdříve je zapotřebí promítnout na zeď vhodný kalibrační obraz a následně získat pomocí aplikace *wsclient* RGB obrázek z Kinectu. Tyto dva obrazy budou nadefinovány ve skriptu *config* a poslouží jako vstup pro *automatic calibration*. Po spuštění této aplikace dostaneme kalibrační matici, která je automaticky uložena a následně použita při spuštění skriptu *sledovani*.

## 3 Závěr

V rámci této práce byla vytvořena celkem přesná kalibrace projektoru s kinectem, která ovšem závisí na pozici, zejména vzdálenosti kinectu od stěny, na kterou je promítán vzorový obraz.