

Analýza osob pomocí pohybového senzoru Kinect s využitím Kinect for Windows SDK

The Analysis of Human Body Movement Using Kinect Sensors
and Kinect for Windows SDK

Antonín Horáček

Bakalářská práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Antonín HORÁČEK**

Osobní číslo: **A08013**

Studijní program: **B 3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

Téma práce: **Analýza osob pomocí pohybového senzoru Kinect s
využitím Kinect for Windows SDK**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Analyzujte možnosti senzoru Kinect.
3. Analyzujte možnosti Kinect for Windows SDK.
4. Vypracujte návrh aplikace pro identifikaci osob pomocí senzoru Kinect.
5. Demostrujte výsledky a formulujte závěr.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **BENEŠ, Bedřich; SOCHOR, Jiří; FELKEL, Petr; ŽÁRA, Jiří. Moderní počítačová grafika. Vyd 1. Brno: Computer Press, 2004, 609 s. ISBN 80-251-0454-0.**
2. **WEBB, Jarrett; ASHLEY, James. Beginning Kinect Programming with the Microsoft Kinect SDK. Apress. ISBN 14-302-4104-7.**
3. **KRAMER, Jeff; BURRUS, Nicolas; HERRERA, Daniel; ECHTLER, Florian; PARKER, Matt. Hacking the Kinect. Apress. ISBN 14-302-3867-4.**
4. **Microsoft Kinect SDK for Developers. MICROSOFT CORPORATION. Microsoft.com [online]. 2012 [cit. 2012-01-30]. Dostupné z: <http://www.microsoft.com/en-us/kinectforwindows/>**
5. **HADÁČEK, Jan. Přizpůsobení 3D modelu lidské postavy s použitím reálných dat. Plzeň, 2011. Bakalářská práce. Západočeská Univerzita v Plzni.**

Vedoucí bakalářské práce:

Ing. Erik Král

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

24. února 2012

Termín odevzdání bakalářské práce:

8. června 2012

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Příjmení a jméno: Obor:

P R O H L Á Š E N Í

Prohlašuji, že

- беру на ве́домі, же оdevzdáním diplomové/bakalářské práce souhlasím se zveřejněníм své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby ¹⁾;
- беру на ве́домі, же дипломová/бакалářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- был/а jsem seznáмен/а s tím, же на мой/и дипломову/бакалářскую работу се плně vztahuje закон ч. 121/2000 Sb. о праву авторском, о правах соувisejících с правом авторским а о změně некоторых законов (авторский закон) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3 ²⁾;
- беру на ве́домі, же podle § 60 ³⁾ odst. 1 авторского закона má UTB ve Zlíně право на uzavření лиценční сmlouvy о использовании школьного дila в rozsahu § 12 odst. 4 авторского закона;
- беру на ве́домі, же podle § 60 ³⁾ odst. 2 а 3 могу использовать свое дило – дипломову/бакалářскую работу или предоставить лицензию к ее использованию только с предхожим письменным согласием Univerzity Tomáše Bati ve Zlíně, которая является опра́внена в таком случае от меня требовать соразмерный взнос на оплату расходов, которые были Univerzitой Tomáše Bati ve Zlíně на создание дila выноложены (аž до их фактической суммы);
- беру на ве́домі, же pokud bylo к выполнению дипломové/bakalářské práce využito softwaru poskytnutého Univerzitой Tomáše Bati ve Zlíně или другими субъектами только к учебным а научным целям (только к некоммерческому использованию), нельзя результаты дипломové/bakalářské работы использовать к коммерческим целям;
- беру на ве́домі, же pokud je выходом дипломové/bakalářské работы как-либо программный продукт, считаются за часть работы также и исходные коды, напр. исход-

ry, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Ve Zlíně

.....

1) zákon č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, § 47 Zveřejňování závěrečných prací:

(1) Vysoká škola nevydělčně zveřejňuje disertační, diplomové, bakalářské a rigorózní práce, u kterých proběhla obhajoba, včetně posudků oponentů a výsledku obhajoby prostřednictvím databáze kvalifikačních prací, kterou spravuje. Způsob zveřejnění stanoví vnitřní předpis vysoké školy.

(2) Disertační, diplomové, bakalářské a rigorózní práce odevzdané uchazečem k obhajobě musí být též nejméně pět pracovních dnů před konáním obhajoby zveřejněny k nahlížení veřejnosti v místě určeném vnitřním předpisem vysoké školy nebo není-li tak určeno, v místě pracoviště vysoké školy, kde se má konat obhajoba práce. Každý si může ze zveřejněné práce pořizovat na své náklady výpisy, opisy nebo rozmnoženiny.

(3) Platí, že odevzdáním práce autor souhlasí se zveřejněním své práce podle tohoto zákona, bez ohledu na výsledek obhajoby.

2) zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, § 35 odst. 3:

(3) Do práva autorského také nezasahuje škola nebo školské či vzdělávací zařízení, užije-li nikoli za účelem přímého nebo nepřímého hospodářského nebo obchodního prospěchu k výuce nebo k vlastní potřebě dílo vytvořené žákem nebo studentem ke splnění školních nebo studijních povinností vyplývajících z jeho právního vztahu ke škole nebo školskému či vzdělávacího zařízení (školní dílo).

3) zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, § 60 Školní dílo:

(1) Škola nebo školské či vzdělávací zařízení mají za obvyklých podmínek právo na uzavření licenční smlouvy o užití školního díla (§ 35 odst. 3). Odpírá-li autor takového díla udělit svolení bez vážného důvodu, mohou se tyto osoby domáhat nahrazení chybějícího projevu jeho vůle u soudu. Ustanovení § 35 odst. 3 zůstává nedotčeno.

(2) Není-li sjednáno jinak, může autor školního díla své dílo užít či poskytnout jinému licenci, není-li to v rozporu s oprávněnými zájmy školy nebo školského či vzdělávacího zařízení.

(3) Škola nebo školské či vzdělávací zařízení jsou oprávněny požadovat, aby jim autor školního díla z výdělku jím dosaženého v souvislosti s užitím díla či poskytnutím licence podle odstavce 2 přiměřeně přispěl na úhradu nákladů, které na vytvoření díla vynaložily, a to podle okolností až do jejich skutečné výše; přitom se přihlédne k výši výdělku dosaženého školou nebo školským či vzdělávacím zařízením s užitím školního díla podle odstavce 1.

ABSTRAKT

Cílem mé bakalářské práce, je analýza pohybového senzoru Microsoft Kinect a jeho možností v souvislosti s identifikací osob. Obsahem práce je seznámení se senzorem, jeho hardwarovým vybavením a softwarovými prostředky, potřebnými pro připojení k osobnímu počítači a k vytváření samostatných aplikací. Důraz je kladen především na využití tohoto přístroje při analýze osob, schopnost senzoru, rozpoznat ve svém zorném poli přítomnost člověka, analyzovat jeho tělesnou stavbu a hlasový projev. Součástí práce je samostatná aplikace, na níž je možno demonstrovat nasazení senzoru Kinect v praxi.

Klíčová slova: senzor, analýza, aplikace, Kinect

ABSTRACT

The aim of my bachelor's thesis is analysis of the motion sensor Microsoft Kinect and its potential in the personal identification field. Contents of the thesis comprise of a general introduction on the sensor, its hardware equipment and extensive software tools, for connection to a personal computer and for creating the standalone applications with stress on their use. Emphasis is placed on device's potential of active service in the personal detection, sensor's efficiency in detecting a human presence in its field of vision and in analyzing a body construction and vocal expression of the target. The standalone application for demonstration of Kinect sensor practical capacities is included.

Keywords: sensor, analysis, applications, Kinect

Poděkování:

Chci poděkovat vedoucímu práce Ing. et Ing. Eriku Královi za cenné rady a připomínky, podnětné konzultace a především vstřícný přístup, což významně přispělo k úspěšnému dokončení mé bakalářské práce. Dále děkuji rodině a přátelům za součinnost při testování vytvořené aplikace a psychickou podporu.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
Podpis bakaláře

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST.....	10
1 ANALÝZA OSOB.....	11
1.1 IDENTIFIKACE OSOB	11
1.2 OBECNÉ METODY IDENTIFIKACE	12
1.3 VÝCHOZÍ METODY PRO SENZOR KINECT.....	13
1.3.1 Identifikace osob podle tělesných měř	13
1.3.2 Identifikace osob podle hlasu.....	14
1.3.3 Identifikace podle geometrie tváře.....	15
2 SENZOR KINECT.....	16
2.1 STRUČNÁ HISTORIE	16
2.2 ZÁKLADNÍ TECHNICKÉ PARAMETRY	17
2.3 POPIS JEDNOTLIVÝCH ČÁSTÍ	20
2.3.1 Motorizovaná základna	21
2.3.2 Barevná VGA kamera	21
2.3.3 Infračervený vysílač	22
2.3.4 Monochromatický snímač	23
2.3.5 Více směrový mikrofon.....	23
2.3.6 Elektronické desky	24
2.4 OBECNÉ PRINCIPY ČINNOSTI.....	26
2.4.1 Určování vzdálenosti objektu.....	26
2.4.2 Zpracování obrazu.....	26
2.4.3 Rozpoznání lidského těla	27
2.4.4 Rozpoznání obličeje	27
2.4.5 Rozpoznání hlasu	29
3 VÝVOJOVÁ PROSTŘEDÍ.....	30
3.1 CL NUI PLATFORM.....	30
3.2 OPENNI.....	31
3.3 KINECT FOR WINDOWS SDK.....	32
II PRAKTICKÁ ČÁST	33
4 NÁVRH SAMOSTATNÉ APLIKACE	34
4.1 POŽADAVKY NA FUNKCE APLIKACE	34
4.2 PROSTŘEDKY NUTNÉ PRO TVORBU APLIKACE	34
4.2.1 Hardwarové prostředky	35
4.2.2 Softwarové prostředky	35
4.3 ROZVRŽENÍ APLIKACE	35
4.4 VYTVOŘENÍ NOVÉHO PROJEKTU	36
4.5 UŽIVATELSKÉ ROZHRANÍ	37
4.6 ZÁLOŽKA SENZOR.....	38
4.6.1 Inicializace senzoru Kinect	38
4.6.2 Nastavení výchozích parametrů	38
4.6.3 Vykreslení RGB obrazu	40

4.6.4	Vykreslení kosterního systému	40
4.6.5	Výpočet délky jednotlivých kostí.....	44
4.7	ZÁLOŽKA GRAFY	45
4.8	DATABÁZE OSOB	46
4.9	PRÁCE S APLIKACÍ.....	48
4.9.1	Obsluha aplikace	48
4.9.2	Hlasové ovládání	50
5	OVĚŘENÍ PŘESNOSTI SENZORU	51
5.1	VÝSLEDKY MĚŘENÍ S VÝPOČTEM PRŮMĚRNÉ HODNOTY	51
5.2	VÝSLEDKY MĚŘENÍ S VYČÍSLENÍM NEJČASTĚJŠÍ HODNOTY	53
	ZÁVĚR	55
	SEZNAM POUŽITÉ LITERATURY.....	56
	SEZNAM OBRÁZKŮ	59
	SEZNAM TABULEK.....	60

ÚVOD

V dnešní době jde pokrok rychle dopředu. Věci, se kterými jsme se dříve setkávali jen v podobě vědecké fikce, se stávají součástí běžného života. Například vynález telefonu přinesl lidem možnost vzájemné komunikace téměř odkudkoliv a kamkoliv. Lidé mají rádi vše, co je praktické, pohodlné a trendy, a proto byl vynalezen telefon mobilní, který byl oproštěn od nutnosti kabelového připojení a postupně obohacen o další a další prvky, uspokojující poptávku. Podobně je to ve všech odvětvích, ať už se jedná o informatiku, strojírenství či jiné obory, zabývající se rozvojem všemožných technologií. Lidé vyvíjejí stále nová zařízení, plnící různé účely nebo spojující několik zařízení v jedno, kombinující jejich vlastnosti.

Tato práce se zabývá jedním z takovýchto zařízení, jedná se o senzor Microsoft Kinect pro Xbox 360 (dále jen Kinect), využívaný především jako pohybový ovladač v zábavním průmyslu. Všechny ostatní, doposud využívané ovladače, se vyznačovali nutností přímého styku s uživatelem, jako například klasická počítačová myš, klávesnice, gamepad apod., které byly navíc omezeny nutností vzájemného fyzického propojení s ovládaným médiem. Jisté vylepšení přinesly bezdrátové technologie, kdy však stále zůstala nutnost přímého styku s ovladačem. Senzor Kinect je sice fyzicky propojen s ovládaným médiem, styk mezi uživatelem a ovladačem je zde však bezdotykový a k ovládní daného média (počítač, televize, konzole Xbox 360), postačí uživateli pouze jeho přirozený, pohybový a hlasový projev. Senzor analyzuje fyzické vlastnosti člověka, dokáže sledovat jeho pohyb, shromažďovat data, udávající jeho fyzickou stavbu, tvar obličeje a dokonce rozpoznávat jeho hlasové pokyny. Nabízí tak širokou škálu využití v mnoha oborech. Tématem této práce je jeho využití v souvislosti s identifikací osob a to na základě analýzy tělesné stavby a hlasového projevu. Součástí práce je i samostatná aplikace, jejímž účelem je pouze demonstrovat, s jakou přesností jsou data získávána a jakým způsobem lze senzoru využít pro účely identifikace osob.

I. TEORETICKÁ ČÁST

1 ANALÝZA OSOB

Výraz analýza pochází z řeckého slova „*ana-lyó*“, což znamená rozvazovat a rozebírat. V tomto případě se jedná o analýzu neboli rozbor osob a ten můžeme provádět jak z fyziologického tak z psychologického hlediska. Tato práce pojednává o analýze osob s použitím pohybového senzoru Microsoft Kinect for Xbox 360 (dále jen Kinect) [1]. Bude nás tedy zajímat hledisko fyziologické. O tom, co vše tento přístroj dokáže, kromě toho, že slouží jako pohybový ovladač pro herní konzoli, se lze dočíst v dalších kapitolách. Prozatím postačí, že umí ve svém zorném poli rozpoznat přítomnost člověka, zaznamenat jeho hlas, tělesnou stavbu a sledovat jeho pohyb. Využijeme-li nějaké databáze s uloženými vzorky a příslušnou aplikaci, lze jej použít také pro identifikaci osob. A do jaké míry je identifikace osoby přesná a spolehlivá, o tom se přesvědčíme v dalších částech této práce.

1.1 Identifikace osob

Všeobecný význam slova identifikace je porovnání nezaměnitelných charakteristik objektu s následným určením nebo vyloučením shodnosti. V případě této bakalářské práce jde o rozpoznání objektu jako člověka a následné určení či vyloučení konkrétního jedince.

Je mnoho důvodů, proč potřebujeme zjistit totožnost osoby. Může se jednat o osobu podezřelou z trestného činu, osobu, která není schopna vlastní identifikace (například po nehodě) nebo může jít o systém zabezpečení objektu, kdy je povolen vstup pouze osobám, které jsou identifikovány jako oprávněné. Ale důvodů je mnohem více a ne všechny kladou takový důraz na přesnost identifikace daného jedince. Senzor Kinect je využíván prozatím hlavně v zábavním průmyslu a co se identifikace týče, jde především o rozpoznání přítomnosti člověka, sledování jeho pohybu a ztotožnění s jeho virtuálním profilem. Nepředpokládá se zde přílišná snaha o podvod, manipulace s profily cizích uživatelů apod.

V podstatě máme dva druhy identifikace. Záleží na tom, zda je osoba fyzicky přítomna nebo máme pouze nějakou informaci, jako fotografii, otisk prstu, pachovou stopu apod. Totožnost pak prověřujeme na základě získaných informací, které porovnáváme v dostupných databázích. Nejznámější obor, který se identifikací osob (a nejen jich) zabývá, se nazývá kriminalistika.

Procesy užívané při identifikaci osob můžeme rozdělit na tři základní [2]:

Verifikace – proces, kdy úkolem biometrického systému je potvrdit či vyvrátit, zda se jedná o určitého jedince, srovnáním sejmutého vzorku s již dříve zapsaným (tzv. šablonou neboli template). Jedná se o tzv. princip one-to-one.

Identifikace – proces, kdy úkolem systému je zjistit, zda lze daného jedince ztotožnit na základě sejmutého vzorku, srovnaného se všemi vzorky nacházejícími se v určité databázi. Tento princip označujeme jako one-to-many.

Autentizace – neboli rozpoznávání, je proces, na jehož konci uživatel získá určitý status, např. oprávněný/neoprávněný atd.

1.2 Obecné metody identifikace

Metod pro identifikaci osob existuje skutečně mnoho, dle mechanismu bychom je mohli rozdělit na tři základní a to identifikaci heslem, předmětem či biometrickým prvkem. Cílem této práce však není, popisovat je všechny, budeme se zabývat především těmi, které jsou založené na rozpoznávání jedinečných biologických charakteristik živé osoby, tedy metody biometrické. Právě ty vycházejí z přesvědčení, že některé biologické charakteristiky (morfologické, fyziologické) jsou pro každého živého člověka jedinečné a neměnitelné.

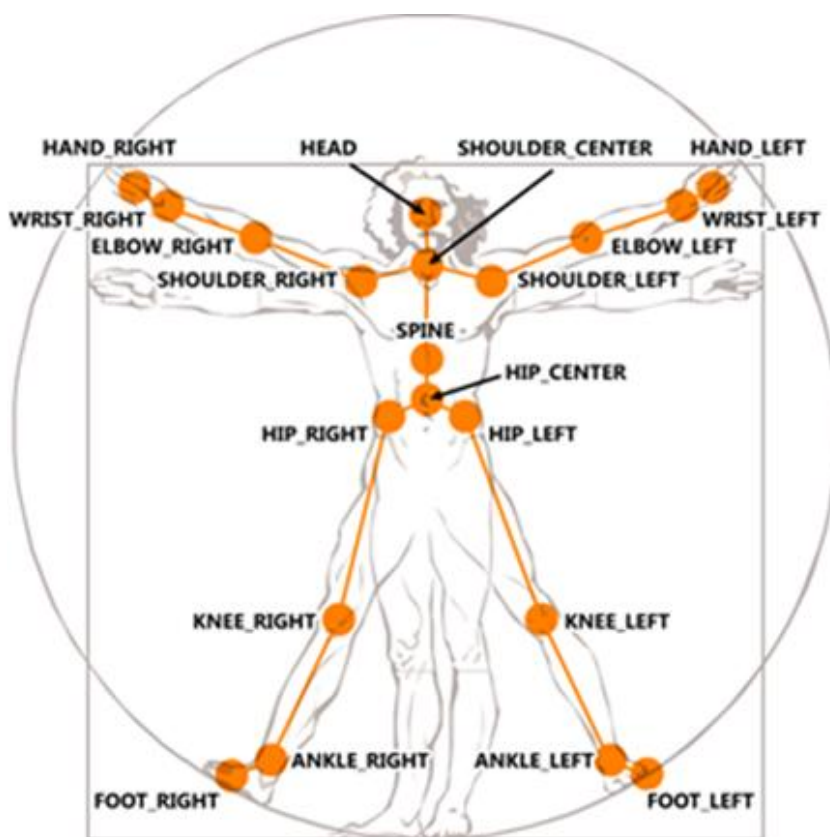
Vzhledem k využití senzoru Kinect, jsem se zaměřil na metody, u kterých identifikace probíhá za použití samostatného automatizovaného zařízení, případně zařízení připojeného k osobnímu počítači, které přímo získává a vyhodnocuje, zda lze identifikovat danou osobu či nikoliv. S rozvojem počítačových technologií již na konci 60. let, začalo i biometrické rozpoznávání člověka podléhat automatizaci. Z různých výzkumů bylo potvrzeno, že lidské tělo má mnoho specifik, která jsou neměnná a jedinečná. Na základě těchto skutečností byly vyvinuty technologie, které dokážou tyto části zaznamenávat, srovnávat a vyhodnocovat. Mezi nejznámější patří například identifikace osob podle oční duhovky či sítnice, otisků prstů, akustické charakteristiky hlasu, pachu, tvaru ušního boltce, dynamiky chůze či podpisu atd. Tyto jednotlivé technologie jsou dobře popsány např. ve studijním textu VŠB TU Ostrava - Biometrické metody identifikace osob v bezpečnostní praxi [2].

1.3 Výchozí metody pro senzor Kinect

Tématem této práce je identifikace osob pomocí senzoru Kinect, představíme si tedy biometrické metody, se kterými je tento senzor schopen pracovat.

1.3.1 Identifikace osob podle tělesných měr

V roce 1879 vznikl tzv. systém identifikace osob podle tělesných měr, jehož zakladatelem byl úředník pařížské policie Alphonse Bertillon (23. 4. 1853 - 13. 2. 1914), který se zasloužil i o jeho zavedení do praxe [3]. Bertillonův antropometrický systém (tzv. bertillonáž) spočíval v měření jedenácti základních rozměrů, které byly evidovány v kartotéce pro následnou identifikaci. Jednalo se o výšku ve stoje a vsedě, délku hlavy, šířku hlavy, délku pravého ucha, šířku pravého ucha, délku levého chodidla, délku prostředníku levé ruky, délku prsteníku levé ruky a délku předloktí levé ruky. Tento systém byl však použitelný až od určitého věku člověka, kdy se již zmíněné tělesné proporce nemění. Na místech, kde je kladen vysoký důraz na přesnost a spolehlivost identifikace osoby, se dnes preferují jiné přesnější metody, jako daktyloskopické snímání otisků prstů apod., ale i tyto metody mají svá úskalí, každá se pohybuje v jiné cenové relaci, podléhají určitým nárokům na manipulaci, údržbu atd. Ale budeme-li se držet identifikace osob podle tělesných měr, jednou z možností je právě využití senzoru Kinect, který dokáže zaznamenat délku hlavních částí kosterního systému člověka, respektive velikost spojnic jeho jednotlivých bodů. Tyto body se nazývají „jointy“ (z angl. slova joint = kloub, spoj) a senzor jich u jedné osoby dokáže zaznamenat až 20 (Obr. 1) [4].



Obrázek 1: Uspořádání kosterního systému [4]

1.3.2 Identifikace osob podle hlasu

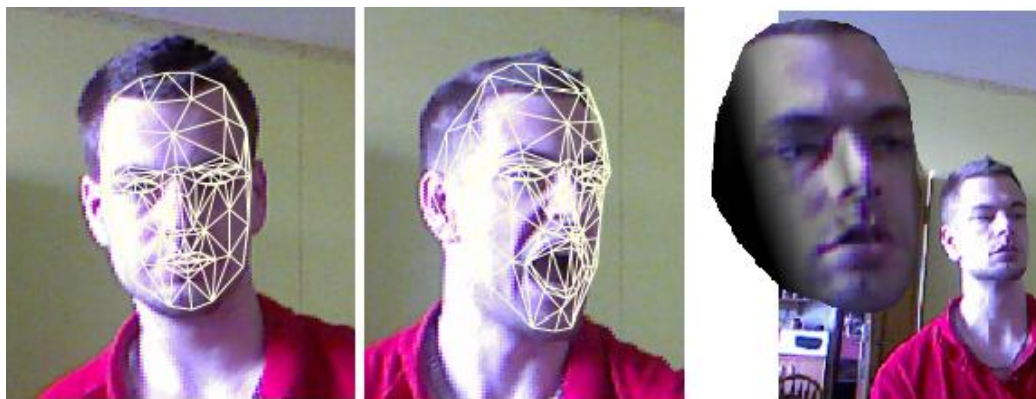
Další metodou identifikace osob je hlasová biometrie. Metoda byla vyvinuta již roku 1970, tehdy však technologie nebyly na takové úrovni jako dnes, kdy je například ovládání počítače za použití hlasových příkazů dostupné každému. Jako příklad může posloužit program JetVoice [5]. Ten je dostupný zdarma a s použitím klasického mikrofону umožní uživateli obsluhu základních činností na počítači pomocí hlasových příkazů.

Identifikace podle hlasu je možná jen tehdy, pokud již máme vytvořenou nějakou databázi, ve které je zaznamenán vzorek hlasu, tzv. hlasový otisk. Pro určení, zda se získaný vzorek hlasu, shoduje se vzorkem uloženým v databázi, slouží tzv. hlasová verifikace. Často je zaměňována s rozpoznáváním hlasu. Jsou to však dvě odlišné technologie, jež spojuje pouze hlas. Rozpoznávání řeči funguje na principu převodu zvukového záznamu na text s cílem zjistit význam slov, zatímco hlasová verifikace pracuje s porovnáváním hlasových otisků a cílem je zjistit shodu s otiskem, uloženým v databázi, na významu slov nezáleží. Senzor Kinect teoreticky zvládne obojí, záleží pouze na navržené aplikaci, jak se získanými daty pracuje. Hlavní výhodou verifikace pomocí digitálních otisků hlasu je nízká

cena, poměrně vysoká spolehlivost a naprostá neinvazivnost technologie. Pro vyšší zabezpečení je možné využití klíčových vět, kdy je pro správnou identifikaci, kromě hlasu samotného, nutná znalost použitých výrazů.

1.3.3 Identifikace podle geometrie tváře

Problematika, zabývající se identifikací osob podle tváří, je velmi obsáhlá. Verifikace tváře probíhá srovnáváním obrazu sejmutého kamerou s obrazem, uloženým v databázi. K jednoznačné identifikaci slouží převážně tvar obličeje, poloha opticky významných míst na tváři, jako jsou oči, nos, ústa a obočí. Obrazová data mohou být uložena jako matice jasových úrovní, častěji jsou však diskriminována nějakou funkcí, která snižuje jejich redundanci. Neuchovává se tedy přesná poloha daných míst, ale jen vzdálenost očí, vzdálenost rtů od nosu, úhel mezi špičkou nosu a jedním okem, atd. 21. 5. 2012 vyšla nová verze Microsoft Kinect SDK 1.5, která již umožňuje práci s geometrií tváře [6]. Vzhledem k datu vydání jsem již nestihl tuto funkci využít do mé aplikace, proto ji zde pouze zmiňuji. Na základě získaných dat z hloubkového senzoru se určí důležité body, které propojí do síťového trojrozměrného modelu, včetně RGB obrazu, který navíc dokáže reagovat na pohyb snímané osoby v reálném čase. (Obr. 2).



Obrázek 2: Příklad snímání geometrie tváře

Nové technologie obecně, umožňují rozpoznání obličeje s dříve nevídanou přesností. Dnešní tzv. „3-D senzory“ dokážou zachytit informace o tvaru obličeje. Tyto informace mohou být pak použity k identifikaci specifických vlastností na povrchu tváře, jako obrysu očních důlků, nosu či brady. Výhodou je zde fakt, že na schopnost rozpoznání nemá vliv kvalita osvětlení, jako u jiných technologií a navíc je možné snímání z různých úhlů, včetně profilu.

2 SENZOR KINECT

Senzor Kinect slouží jako vstupní zařízení pro bezdotykové ovládání herní konzole Xbox 360 a nově nabízí využití u osobních počítačů, což je také tématem mé bakalářské práce. Dle společnosti Microsoft dokáže senzor rozpoznat ve svém zorném poli člověka, sledovat jeho pohyb, hlasový projev a následně pracovat se získanými daty [7]. Vzhledem k jeho možnostem je využíván primárně v zábavním průmyslu, jako pohybový ovladač pro různé aplikace, především u výše uvedené konzole Xbox 360. Získaná data o uživateli je však možno uchovávat a následně využít k identifikaci jednotlivých uživatelů. Do jaké míry je přístroj schopen spolehlivě určit identitu daného uživatele se pokusím ověřit v praktické části této práce.

2.1 Stručná historie

Technologie, ze které senzor vychází, byla vynalezena v izraelské společnosti Prime Sense již v roce 2005, jako metoda a systém pro rekonstrukci objektu [8]. Samotný senzor Kinect byl poprvé oznámen 1.6.2009 na výstavě Electronic Entertainment Expo, také nazývané E3, tehdy ještě pod krycím názvem "Projekt Natal". Později v roce 2010 byl oznámen oficiální název, který vznikl spojením dvou anglických slov, vyjadřujících podle Microsoftu nejlépe podstatu systému, slova "kinetic" a "connect", tedy Kinect. Senzor byl představen, jako periferní zařízení k herní konzoli Microsoft Xbox 360, jehož prostřednictvím lze bezdotykově ovládat samotnou konzoli a především videohry, pro tento senzor speciálně vytvořené. Všechny vyspělejší doposud používané ovladače musel uživatel fyzicky držet v ruce nebo se jich jinak dotýkat, což přinášelo jistá omezení. Kinect je umístěn ideálně pod nebo nad zobrazovacím zařízením a pro interakci s ním uživateli stačí použití vlastního těla a hlasu. Senzor Kinect lze však využít i jinde, než u herní konzole Xbox 360. Prostřednictvím USB konektoru lze zařízení připojit k osobnímu počítači. 16. 6. 2011 společnost Microsoft vydala nekomerční Kinect Software Development Kit pro systém Windows 7, který široké veřejnosti umožnil, vyzkoušet si tvorbu všemožných aplikací a to prostřednictvím známých programovacích jazyků jako C++ či C#. Současná verze je k dispozici na oficiálních stránkách společnosti Microsoft.

2.2 Základní technické parametry

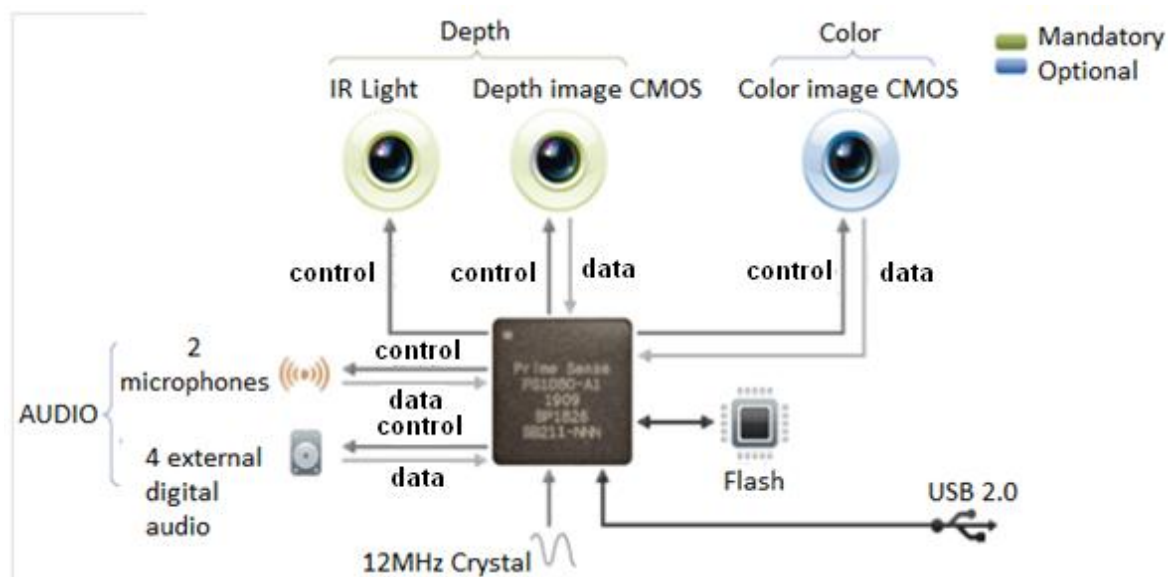
Senzor Kinect má 4 hlavní části, které jej charakterizují (Obr. 3) [9]:

- Barevná VGA kamera
- Infračervený vysílač
- Monochromatický CMOS snímač
- Více směrový mikrofón



Obrázek 3: Základní prvky senzoru Kinect [9]

Na následujícím obrázku (Obr. 4) je znázorněno schéma uspořádání hlavních funkčních prvků [10]. Popisuje vnitřní čip PS1080 od společnosti PrimeSense, který zpracovává signál z CMOS (complimentary metal-oxide semiconductor) snímače a signál z vysílače IR záření. Vyhodnocuje změny v časovém posunu IR záření a vytváří trojrozměrný snímek prostoru. Jedním z výstupů tohoto čipu je 11 bitová informace (2048 odstínů šedi) o hloubce (vzdálenosti) každého z 307 200 (640 x 480) bodů v prostoru. Vnitřní čip se stará o zpracování informace o hloubkovém rozložení prostoru a na základě hloubkové mapy detekuje případné uživatele.



Obrázek 4: Schéma uspořádání hlavních funkčních prvků senzoru [10]

Senzor Kinect má při použití se softwarem konzole Xbox 360 praktický limit v rozmezí 1,2 - 3,5 m [11]. Oblast, v níž se musí uživatel pohybovat, je přibližně 6 m². Senzor však dokáže udržovat snímání uživatele přes rozšířený rozsah cca 0.7 - 6 m, ale aplikace jsou navrženy tak, aby uživatele upozornili v případě, že vybočí mimo praktický limit. Docílí se tak přesnějšího ovládání. Senzor má zorné pole 57 ° horizontálně a 43 ° vertikálně, zatímco motorizovaná základna je schopna naklápět senzor až na 27 ° směrem nahoru nebo dolů. Horizontální pole při pozorovací vzdálenosti 0.8 m, je tedy přibližně 87 cm a vertikální pole přibližně 63 cm, což vede k rozlišení více než 1,3 mm na 1 pixel. Mikrofon má čtyři mikrofonní kapsle, a pracuje s každým kanálem 16 - bitovým zpracováním zvuku na vzorkovací frekvenci 16 kHz.

Výhody:

- Snadná realizace – senzor je přenositelný, poměrně levný a ve spojení například s běžným notebookem, jej lze využít téměř kdekoli, včetně domácností.
- Snadná obsluha – lze vytvářet samostatné, automatizované aplikace, které mohou být ovládány, kromě klasické myši a klávesnice, také pohyby lidského těla či hlasem - využitelnost v zábavním průmyslu.
- Víceúčelovost – senzor disponuje množstvím funkcí, umožňuje záznam obrazových a zvukových dat, dokáže rozpoznat a sledovat pohyb člověka ve svém zorném

poli a v závislosti na dané aplikaci, reagovat na jednoduché pohyby či hlasové příkazy. Jednotlivé funkce lze využívat samostatně nebo je různě kombinovat.

Nevýhody:

- Přesnost měření – při použití pro snímání metrických dat, například za účelem identifikace osob, vykazuje malou přesnost, snímané hodnoty hodně kolísají a jeho užití je tedy nevhodné v místech, kde je kladen důraz na vysoké zabezpečení.
- Rušivé vlivy – nutnost vytvoření správných podmínek snímaného prostředí, v závislosti na požadované funkci (zajištění výskytu omezeného množství osob v zorném poli senzoru, odstranění nežádoucích objektů, které by mohl senzor chybně vyhodnotit jako člověka, nevhodné osvětlení či přílišný hluk).

Častým problémem, který se v běžných domácnostech z hlediska konzole Xbox 360 vyskytuje, je nedostatek prostoru. Senzor dokáže ve svém zorném poli rozpoznat až 6 osob a 2 z nich určí jako aktivní. Při snímání dvou uživatelů se však zvyšuje vzdálenost, kterou musí uživatelé dodržet, aby je senzor spolehlivě snímal. Pro tento účel byl vytvořen jednoduchý optický nástavec od společnosti Nyko (Obr. 5), který nevyžaduje žádné napájení a při tom snižuje vyžadovanou vzdálenost od přístroje Kinect o 40 % [12]. Zde si však někteří uživatelé stěžovali na větší šum v získaném obraze a občasné chyby v ovládání, vzniklé umístěním lesklých předmětů v zorném poli senzoru.



Obrázek 5: Optický nástavec od společnosti Nyko [12]

Senzor je ke konzoli Xbox 360 nebo k PC připojen prostřednictvím klasického USB konektoru. Vzhledem k tomu, že vyžaduje vyšší příkon, než může poskytnout USB, je zvlášť napájen dodatečným adaptérem (Obr. 6).



Obrázek 6: Napájecí adaptér

2.3 Popis jednotlivých částí

Nyní se podrobněji seznámíme s jednotlivými částmi senzoru Kinect, k jejichž popisu jsem využil serveru ifixit.com, kde byl senzor kompletně rozebrán a jednotlivé části vyobrazeny [13].



Obrázek 7: Rozebraný senzor Kinect [13]

2.3.1 Motorizovaná základna

Otočná základna zajišťuje naklápění senzoru, tak aby uživatele udržel ve svém zorném poli. Pohyb zajišťuje malý elektromotorek (Obr. 8), podobný, jako můžeme nalézt například u DVD mechanik. Převody, zajišťující naklápění nahoru a dolů, tvoří několik plastových ozubených koleček. Před použitím je senzor potřeba naklopit pod takovým úhlem, aby snímal pokud možno celou postavu uživatele, což závisí na umístění senzoru. U konzole Xbox 360 je software naprogramován tak, aby senzor při každém spuštění nejprve nasnímal celý prostor, tedy od nejnižší po nejvyšší polohu a ze získaných dat určí, kterou plochu bude brát, jako podlahu. Pokud je senzor umístěn například na široké polici, či stolku, měl by se nacházet na jeho okraji směrem k uživateli. Mohl by totiž vyhodnotit polici, jako podlahu a vyžadoval by zbytečně větší vzdálenost uživatele.



Obrázek 8: Motorizovaná základna a elektromotorek [13]

2.3.2 Barevná VGA kamera

Společnost Microsoft ji nazývá "RGB kamerou" (Obr. 9). Odkazuje se na tři barevné složky (Red, Green, Blue), které detekuje a zobrazuje v klasickém VGA rozlišení 640 x 480, jako běžné web kamery a to při frekvenci 30 Hz. Přesné označení tohoto zařízení je VNA38209015. Uživatelé jsou pak rozpoznáváni v obraze detekcí různobarevných objektů. Pomocí barevné rozličnosti tak může být uživatel oddělen od okolí, podobně, jako se to dělá ve filmovém průmyslu, kde je herec snímán před modrým či zeleným pozadím a zbytek je dokreslen v počítači.



Obrázek 9: Barevná VGA
kamera [13]

2.3.3 Infračervený vysílač

Tento vysílač s označením OG12 / 0956 / D306 / JG05A (Obr. 10) vysílá emitované infračervené záření, které se odráží od jednotlivých objektů v prostoru. Toto záření je následně zachytáváno níže popsáním CMOS snímačem a z vyhodnocených dat dokáže senzor určit vzdálenost a umístění každého objektu v prostoru. Vzhledem k tomu, že má senzor vlastní zdroj infračerveného záření, není závislý na světelných podmínkách a funguje i v úplné tmě. Společně s CMOS snímačem umožňují „vidět“ místnost trojrozměrně nebo-li 3D, proto se jim také říká 3D hloubkové senzory.



Obrázek 10: Infračervený
vysílač [13]

2.3.4 Monochromatický snímač

Třetí částí kamerového systému je monochromatický přijímač nebo také CMOS snímač (320×240) s označením Microsoft X853750001 / VCA379C7130 (Obr. 11). Stará se o snímání odražených paprsků. Pakliže se paprsek v prostoru neodrazí, je místo vyhodnoceno, jako místo s nulovou vzdáleností. K tomu může dojít například, je-li objekt ve větší vzdálenosti.



Obrázek 11: Monochromatický
CMOS snímač [13]

2.3.5 Více směrový mikrofón

Jedná se o pole čtyř mikrofónů, které dokážou oddělit hlasy lidí od hluku v místnosti, tedy od okolního šumu. To umožní být i několik metrů od mikrofónu za stálé hlasové kontroly senzoru Kinect. Audio řešení umožní pro Kinect otestovat okolní hluk, hlasitost reproduktorů, kalibraci mikrofónu a schopnost systému Kinect slyšet uživatele v okolním prostředí. Pracuje s 16 - bitovým zpracováním zvuku na vzorkovací frekvence 16 kHz.

2.3.6 Elektronické desky

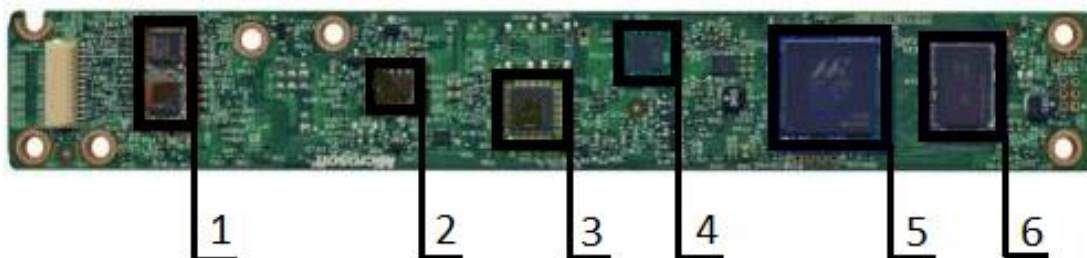
Uvnitř senzoru jsou 3 malé elektronické desky a ventilátor, zajišťující cirkulaci vzduchu (Obr. 12).



Obrázek 12: Elektronické desky a ventilátor [13]

První deska obsahuje (Obr. 13):

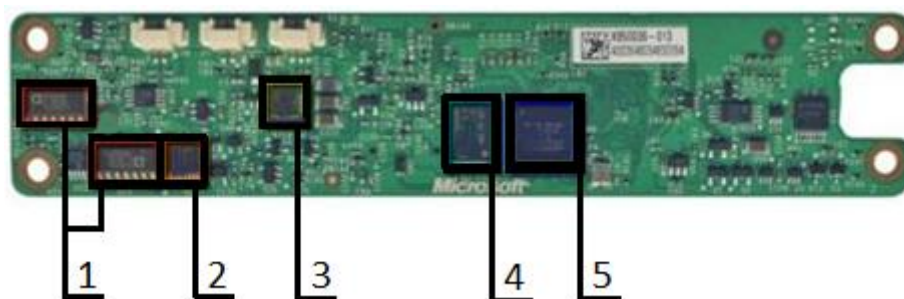
- 1.) Wolfson Microelectronics WM8737G - stereo ADC s mikrofonním předzesilovačem, navržené pro přenosná zařízení, hlasové záznamníky apod.
- 2.) Fairchild Semiconductor FDS8984 - N-Channel PowerTrench MOSFET slouží pro zlepšení celkové výkonnosti DC / DC měniče pomocí synchronního nebo klasického přepínání PWM regulátorů.
- 3.) NEC uPD720114 - USB 2.0 řadič
- 4.) H1026567 XBOX1001 X851716-005 GEPP – paměťový čip SPI flash
- 5.) Marvell AP102 – SoC (System-on-Chip) s řadičem pro kameru
- 6.) Hynix H5PS5162FF - 512 megabit DDR2 SDRAM



Obrázek 13: Popis první elektronické desky [13]

Na další desce můžeme vidět několik důležitých mikrosoučástek (Obr. 14):

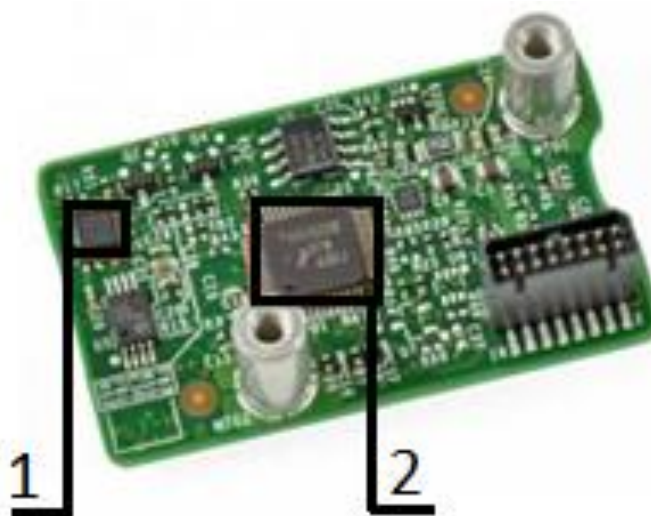
- 1.) Analog Devices AD8694 - výstupní operační zesilovač
- 2.) TI ADS7830I - 8bitový, 8kanálový A/D převodník s I2C rozhraním
- 3.) Allegro Microsystems A3906 - řadič krokového motorku
- 4.) ST Microelectronics M29W800DB - 8 Mbit ($1\text{Mb} \times 8$ nebo $512\text{kb} \times 16$) NV Flash Memory
- 5.) PrimeSense PS1080-A2 - SoC procesor, zpracovávající snímáný obraz



Obrázek 14: Popis druhé elektronické desky [13]

Na nejmenší třetí desce najdeme především (Obr. 15):

- 1.) Kionix MEMS KXSD9 akcelerometr
- 2.) TI TAS1020B USB audio řadič



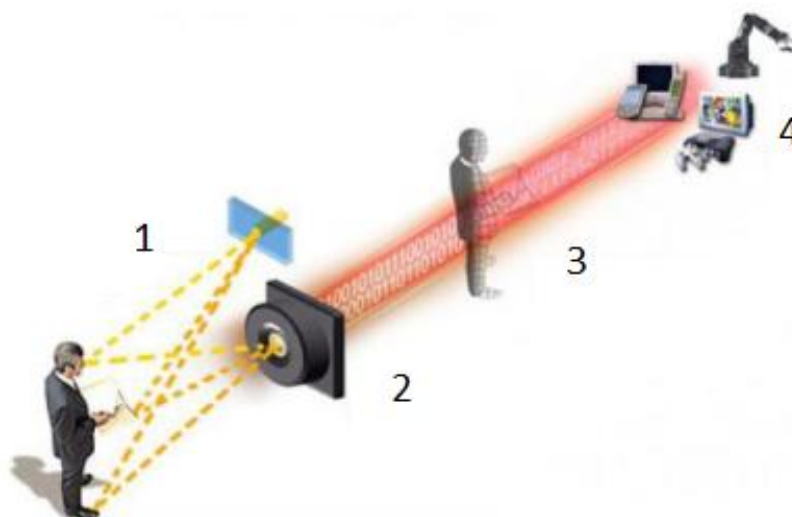
Obrázek 15: Popis třetí elektronické desky [13]

2.4 Obecné principy činnosti

V předchozích kapitolách jsou popsány jednotlivé části senzoru. Dále je zjednodušenou formou popsán obecný princip, jak senzor pracuje.

2.4.1 Určování vzdálenosti objektu

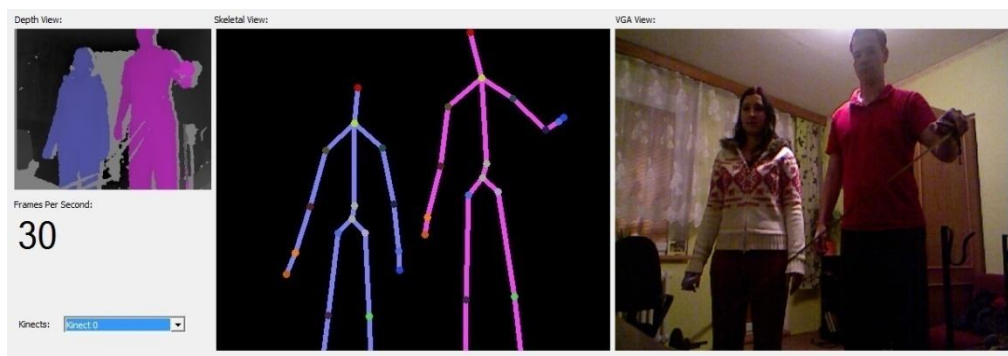
Princip určování vzdálenosti objektů je vidět na následujícím obrázku (Obr. 16) [14]. Generované infračervené záření se šíří okolím (1), CMOS snímač zachytává odražené paprsky (2) a takto zachycená data jsou zpracována vnitřním čipem PS1080 a předána k dalšímu zpracování (3). Vzniká trojrozměrná mapa snímaného prostoru, která je následně zobrazena na displeji připojeného zařízení (4).



Obrázek 16: Princip určování vzdálenosti objektů [14]

2.4.2 Zpracování obrazu

Obrazová data RGB kamery jsou získávána ve VGA rozlišení, každý pixel je reprezentován třemi bajty, které jsou nositelem informace o jednotlivých barevných složkách. Hloubková mapa je reprezentovaná maticí o stejném rozlišení (640 x 480), jednotlivé pixely nesou informaci vzdálenosti objektu od senzoru. Vzdálenost je reprezentována jedenácti bitovým číslem. Hloubkovou mapu můžeme vidět v levé části následujícího obrázku (Obr. 17), i s rozpoznáním obrysů postav, uprostřed je pak snímán kosterní model a vpravo je záznam z klasické VGA kamery. Tato malá aplikace nazvaná "Skeletal viewer" je přímo součástí balíku Microsoft Kinect 1.0 SDK [15].



Obrázek 17: Ukázka aplikace Skeletal viewer

2.4.3 Rozpoznání lidského těla

Rozpoznání lidského těla zajišťuje tzv. Skeletal Tracking System [4]. Pokud je detekována lidská postava, stojící před senzorem, je možné z Kinectu získat data, která definují tuto postavu pomocí skeletálního modelu. Tento model zjednodušuje lidské tělo na 20 tzv. jointů (z angl. slova joint = kloub, spoj). Hloubková kamera posílá informaci o souřadnicích každého jointu v kartézském souřadném systému s počátkem u senzoru kamery. Kinect Skeletal Tracking System je software, který používá anatomická data a data získaná z hloubkových senzorů pro nalezení jointů daného uživatele. Umožňuje sledovat zároveň 2 uživatele a u každého až 20 jointů. Popis jednotlivých jointů, tak jak je snímá senzor, můžeme vidět v kapitole 1.3.1 (Obr. 1).

2.4.4 Rozpoznání obličeje

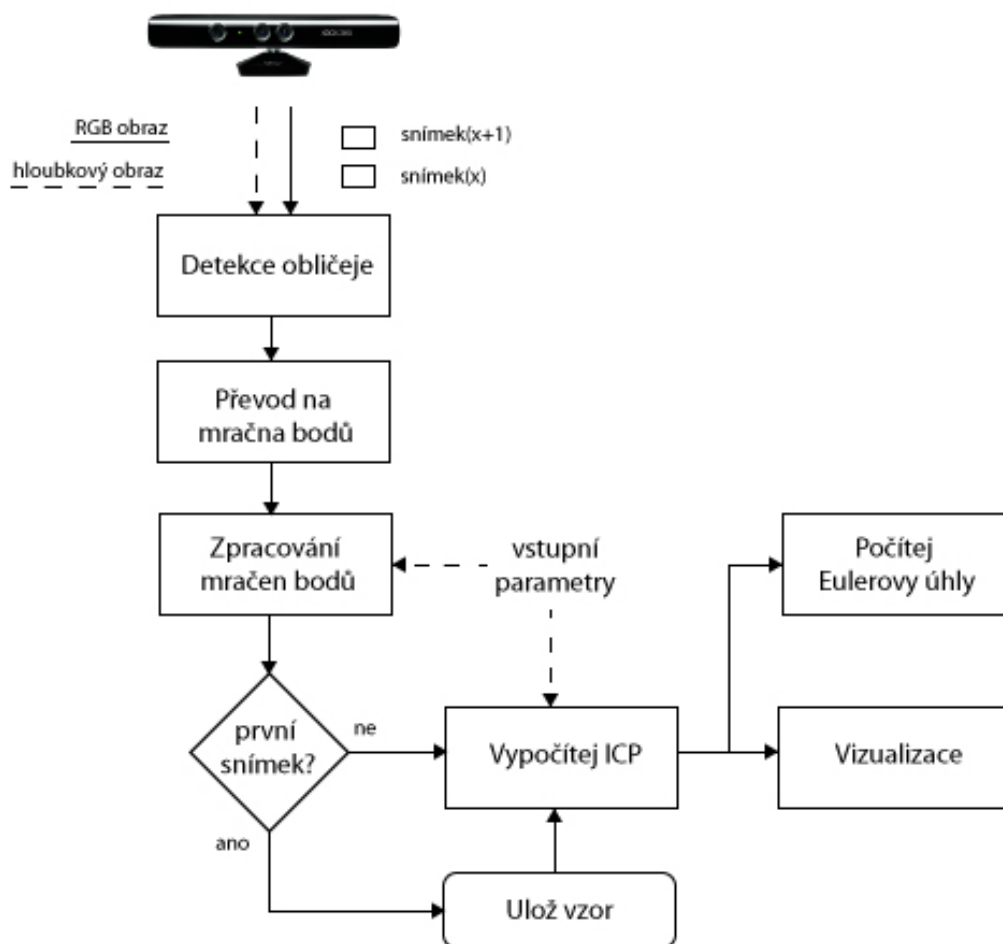
K rozpoznávání lidského obličeje je využito prvku, známého jako akvizice neboli přírůstek prostorové informace v podobě mračen bodů (Obr. 18) [16].



Obrázek 18: Příklad záznamu
mračna bodů obličeje

Do prostoru je vysílačem IR emitována pevná konfigurace bodů, které jsou zachycovány CMOS snímačem a promítají se do scény. Na základě vyměřování mezi konfigurací bodů sejmutých IR snímačem a konfigurací bodů vyslaných IR vysílačem, je vypočtena potřebná prostorová informace. Tuto metodu vyvinula společnost PrimeSense a nazvala ji „Light Coding“ [17]. O zpracování n-rozměrných mračen bodů a 3D geometrie se stará rozsáhlá knihovna PCL (Point Cloud Library). Tato knihovna je multiplatformní a byla úspěšně využita u operačních systémů, jako Linux, MacOS, Windows a Android. Pro představu, jak vypadá systém pro detekci a sledování polohy hlavy se můžeme podívat na následující obrázek (Obr. 19) [18]. Pro práci se senzorem uvolnila společnost PrimeSense OpenNI ovladače, které umožnili data ze senzoru Kinect získávat ve dvou formátech:

- 320x240 16-bit hloubková mapa - 30 snímků/sekundu
- 640x480 32-bit RGB obraz - 30 snímků/sekundu



Obrázek 19: Schéma systému pro detekci a sledování polohy hlavy [18]

Nejprve je na RGB obraze provedena detekce obličeje a detekovaná část obrazu je převedena na mračno bodů. Dochází k převodu do datové struktury PCD (Point Cloud Data), se kterou se dále pracuje v rámci knihovny PCL. Další jsou filtrovány nadbytečné části mračna a dochází k podvzorkování z důvodu snížení výpočetní náročnosti dalších operací. Snímky jsou zpracovány jeden po druhém a první vstupní snímek je uložen jako referenční pro pozdější implementaci registrace mezi snímky. Pro registraci je použit algoritmus ICP (Iterative Closest Point). Výstupem algoritmu je transformační matice, ze které se dále počítají Eulerovy úhly. Poslední blok algoritmu je vizualizace výsledků.

2.4.5 Rozpoznání hlasu

Obecný princip rozpoznávání hlasu je již popsán v kapitole 1.3.2. Hlavním prvkem, potřebným pro identifikaci, je hlasový otisk. Jedná se o datovou matici, obsahující charakteristiku hlasu dané osoby, která není delší než 20 KB. Vlastní proces pak probíhá prostřednictvím telefonu či mikrofону. Osobě, která podléhá identifikaci, je položena otázka nebo série otázek a systém porovná její odpovědi s uloženým hlasovým vzorkem. Tímto způsobem by teoreticky mohlo dojít ke zneužití záznamu hovoru jinou osobou. Proto musí být systém ošetřen proti tzv. „replay attacku“. Člověk nedokáže danou frázi zopakovat vždy úplně přesně, digitální záznam je vždy jiný, a pokud je použit záznam, stroj to pozná.

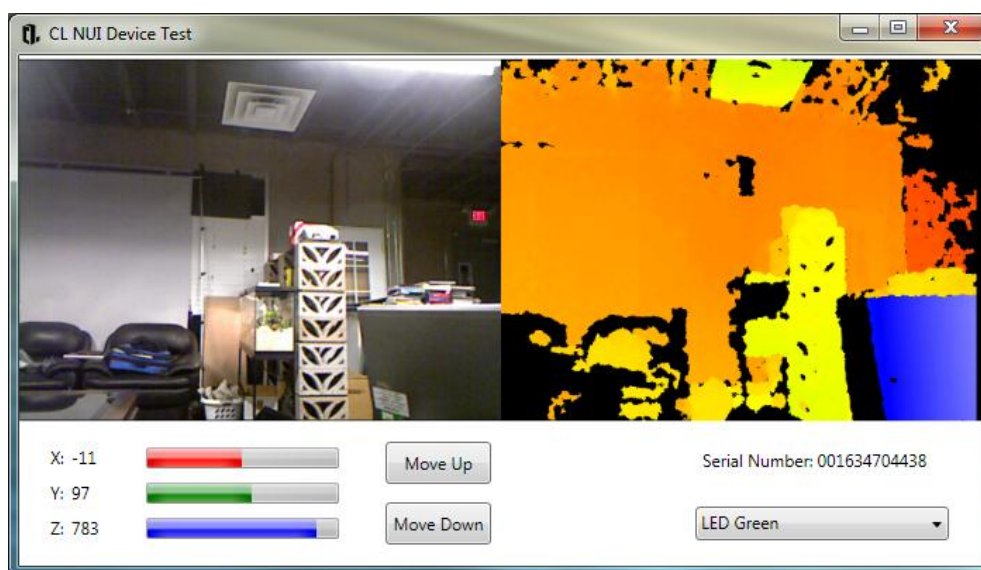
Součástí balíku Kinect SDK je i několik miniaplikací pro práci se zvukem. Ty dokážou zaznamenat zvuk ve formátu wav (waveform audio file format), který byl vytvořen společností IBM a Microsoft pro ukládání zvuku na osobních počítačích. Nejvyužívanější funkcí z hlediska záznamu zvuku, je u senzoru Kinect rozpoznávání řeči, což je zmíněno již v kapitole 1.3.2. Pomocí speciálního softwaru dokáže počítač nebo herní konzole, rozpoznat v lidské řeči konkrétní slova nebo fráze a v závislosti na dané aplikaci jim přidělit určitý význam či funkci. Daná slova nebo celé fráze však musí být předem definována, například můžeme použít běžně užívané anglické výrazy přehrávačů hudby či videa (play, stop, fullscreen, apod.). Práce s hlasovým rozpoznáváním je včetně zdrojových kódů dobře popsána v publikaci Beginning Kinect Programing with MS SDK [19].

3 VÝVOJOVÁ PROSTŘEDÍ

Senzor Kinect samotný by u osobních počítačů nenašel využití, nebýt potřebných ovladačů a vývojového prostředí, ve kterém lze vytvářet různorodé aplikace. Společnost Microsoft se nejprve bránila snaze využít senzor jinde, než u herní konzole Kinect, ale programátoři si i tak našli cestu, jak jej využít u osobních počítačů a vytvořili vlastní "hacky" (CLNUI, OpenKinect [20][21]). Na internetu se tak objevilo mnoho ukázek toho, co lze s tímto senzorem na osobním počítači dělat. Také společnost PrimeSense, která za vývojem technologie Kinect stojí, přispěla tím, že vydala open-source ovladače pro RGB-D senzor, jež je hlavním prvkem zařízení Kinect, tedy PrimeSense's Development Kit. Microsoft nakonec uznala, že nemá smysl, bránit svému produktu v dalším rozvoji a sám vydal oficiální vývojové sady, kterých jsem využil i já, Microsoft Windows SDK, Platform SDK a .NET Framework SDK, obsahující hlavičkové soubory, knihovny, vzory, dokumentaci a potřebné vývojové nástroje.

3.1 CL NUI Platform

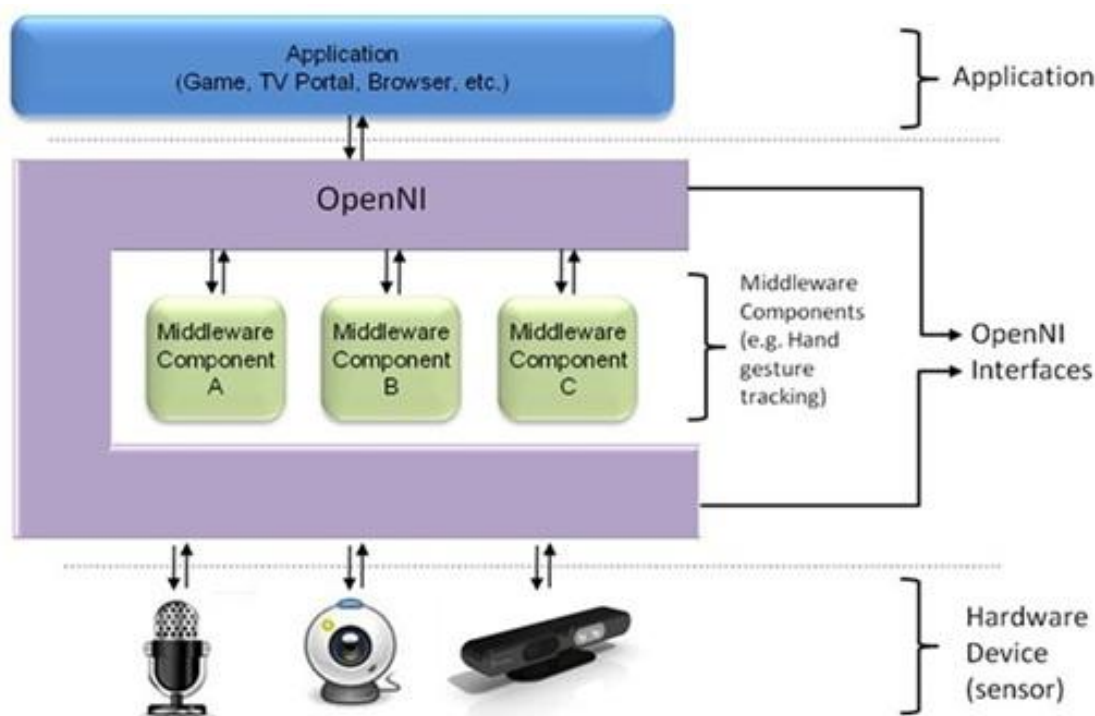
První hack senzoru Kinect se podařil 06.11.2010, členu skupiny NUI Group, známému jako Alex P. To zaznamenalo velký ohlas u široké veřejnosti a tato skupina se rozhodla ve vývoji pokračovat. Vytvořili stabilní platformu pro využití jednotlivých částí senzoru, jako jsou RGB kamera, hloubkové senzory, naklápěcí základna a sada mikrofونů pro práci se zvukem. Na následujícím obrázku (Obr. 19) je vidět ukázková aplikace, vytvořená v tomto prostředí [20].



Obrázek 20: Ukázková aplikace z prostředí CL NUI Platform [20]

3.2 OpenNI

OpenNI (Open Natural Interaction) je vícejazyčný, multiplatformní rámec, definující API (Application Programming Interface) pro psaní aplikací využívajících přirozenou interakci, tedy prostředí pro práci se zařízeními, která dokážou "vidět" a "slyšet" člověka v přirozeném prostředí [21]. Poskytuje softwarové komponenty, analyzující zvuková a vizuální data (tzv. middleware). Lze jej tedy využít také pro práci se senzorem Kinect. Více je však podporován velmi podobný senzor, za kterým stojí (podobně jako u Kinectu) společnost PrimeSense a který ve spolupráci se společností Asus nyní nabízí pod názvem X-tion Pro. Ten je určen přímo pro osobní počítače, postrádá však klasickou RGB kameru. V prosinci 2010 vydala společnost PrimeSense své vlastní open-source ovladače tzv. middleware NITE, z čehož také čerpali vývojáři aplikací pro senzor Kinect. Práce s OpenNI, včetně návodů, pro instalaci do systémů, jako Mac OS X, OS Windows a OS Linux, je poměrně podrobně popsána v publikaci Making Things See [22]. Na následujícím obrázku (Obr. 20) je zobrazena třívrstvá koncepce OpenNI.



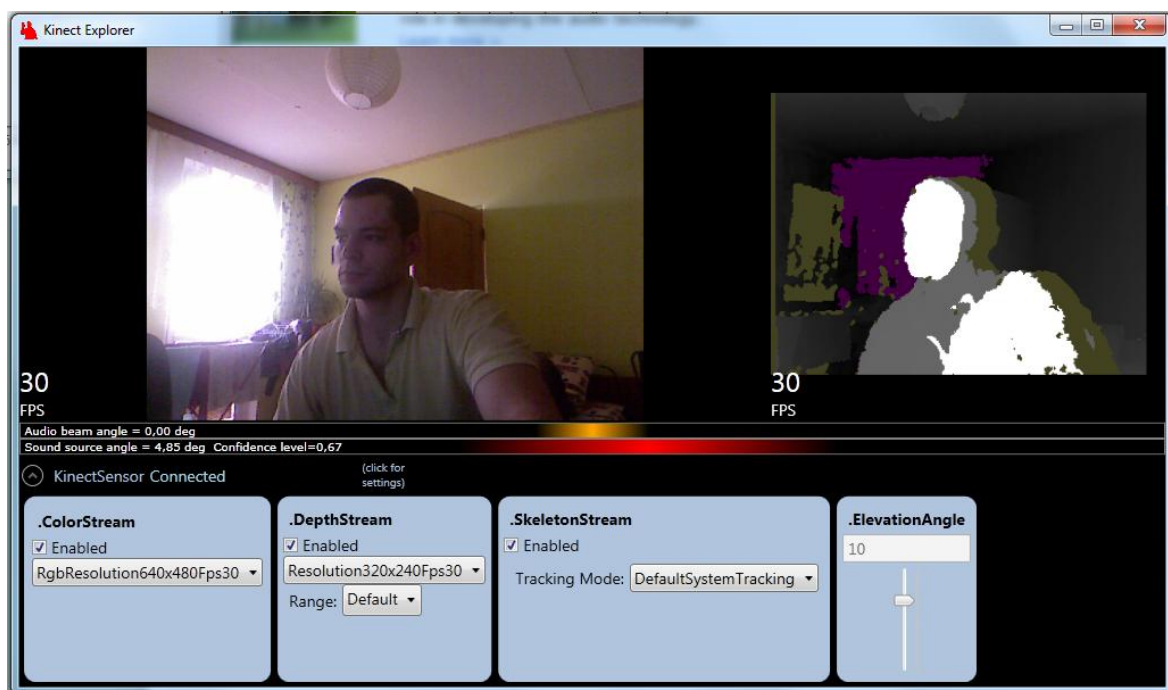
Obrázek 21: Třívrstvá koncepce OpenNI [21]

Horní vrstva představuje software, využívající daného zařízení a na základě získaných dat vytváří interakci s uživatelem. Střední vrstva poskytuje komunikační rozhraní, pracující se zařízeními a komponenty, které analyzují data ze snímače. Spodní vrstva pak znázorňuje samotná zařízení, zachycující vizuální a zvukové prvky scény.

3.3 Kinect for Windows SDK

16. června 2011 Microsoft uvolnila Kinect Software Development Kit pro systém Windows 7, jako nekomerční beta verzi určenou pro vývojáře. Kinect SDK umožnil psát aplikace v C++, C#, Visual Basic a Visual Studio. Celkem byly vydány dvě beta verze a 2. února 2012 vyšla finální verze SDK 1.0, která obsahuje řadu vylepšení, jako je podpora až čtyř senzorů Kinect připojených k jednomu počítači nebo vylepšené rozpoznávání lidského těla a jeho pohybu. Kinect SDK, zahrnující ovladače kompatibilní s OS MS Windows 7, obsahuje tyto hlavní funkce:

- přístup k základní úrovni datových toků z hloubkového snímáče, barevné RGB kamery a z víceúrovňového mikrofону.
- schopnost sledovat kosterní obraz jedné nebo dvou osob, jež se pohybují v zorném poli senzoru Kinect.
- pokročilé zvukové možnosti, akustické potlačení šumu a ozvěny v prostředí, schopnost rozpoznat aktuální zdroj zvuku a jeho integraci s rozpoznáváním řeči v systému Windows API.



Obrázek 22: Aplikace Kinect Explorer z balíku Kinect for Windows SDK 1.0

II. PRAKTICKÁ ČÁST

4 NÁVRH SAMOSTATNÉ APLIKACE

Schopnosti senzoru Kinect, v souvislosti s analýzou osob, jsem se rozhodl demonstrovat na samostatné aplikaci, kterou jsem vytvořil. Tato aplikace není navržena pro praktické využití, slouží pouze, jako ukázka práce se senzorem a představuje možnosti jeho využití. Jedná se o identifikaci osob, které jsou uloženy v jednoduché databázi a cílem je rozpoznat, zda analyzovaná osoba je v databázi již obsažena či ne. Nazval jsem ji jednoduše CheckPerson.

4.1 Požadavky na funkce aplikace

Možnosti senzoru Kinect jsou popsány v teoretické části, z nich jsem si vytýčil jednotlivé body, které by daná aplikace měla umět:

- Zobrazení zorného pole senzoru pomocí RGB obrazu
- Rozpoznání člověka v zorném poli senzoru a jeho sledování
- Vykreslení kosterního systému člověka přímo v RGB obraze
- Záznam polohy jednotlivých jointů snímaného objektu (člověka)
- Ukládání získaných dat včetně nacionálů analyzované osoby
- Porovnání získaného vzorku s již uloženými – identifikace
- Zobrazení výsledku (ztotožněný či neznámý objekt)
- Vizualizace snímaných hodnot - automatické vytváření grafů
- Rozpoznávání hlasu pro účely ovládání aplikace a identifikace objektu

4.2 Prostředky nutné pro tvorbu aplikace

Pro návrh požadované aplikace je potřeba zvolit správné prostředky. Během procesu získávání dat je použito velké množství snímků pro každý z 20 jointů, což klade určité nároky na hardware počítače. Je proto potřeba mít dostatečně výkonný počítač, jak je uvedeno níže v kapitole 4.2.1. Dále jsou zde také jistá omezení ve volbě vývojových nástrojů. Díky mnohým nadšencům se možnosti postupně rozšiřují, v době tvorby této práce jich však mnoho není. Pro práci se senzorem a daným softwarem, jsou dle oficiálních údajů společnosti Microsoft dána určitá softwarová a hardwarová kritéria, která je potřeba dodržet [15].

4.2.1 Hardwarové prostředky

Pro práci se senzorem Kinect, připojeného k osobnímu počítači, jsou doporučeny následující parametry osobního počítače:

- 32-bitový (x86) nebo 64-bitový (x64) procesor, nejlépe dvou jádrový o frekvenci 2.66 GHz a výkonnější, USB 2.0 pro připojení senzoru, alespoň 2GB RAM.
- senzor Kinect od společnosti Microsoft se speciálním adaptérem, pro dostatečné napájení (viz. předchozí popis senzoru).

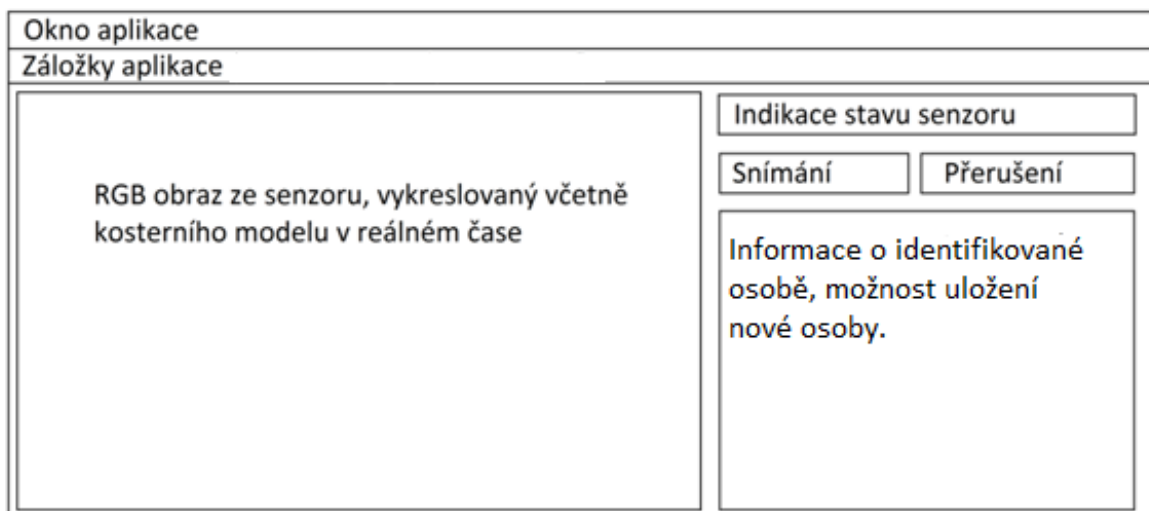
4.2.2 Softwarové prostředky

Pro návrh samotné aplikace je potřeba operačního systému MS Windows 7 a s tím související vývojové nástroje, tedy Microsoft® Visual Studio® 2010 Express nebo jeho další edice, Microsoft Kinect SDK (verzi 1.0 a vyšší), dále .NET Framework 4.0 a rozhraní Microsoft DirectX® 9. Většina aplikací vzniká za pomoci programovacích jazyků C#, pro který jsem se také rozhodl, případně C++, ale možností je více, záleží jen na schopnostech programátora.

4.3 Rozvržení aplikace

Nejprve je nutno, rozvrhnout si, jak bude výsledná aplikace vypadat. Aplikace má být navržena pro operační systém Windows 7 a spouštěna bude v samostatném okně. To bude rozděleno tak, že více prostoru bude věnováno vizualizaci a méně ovládacím prvkům. Okno bude mít předem stanovenou velikost (šířku a výšku). Je třeba dbát na to, že uživatelé mají monitory s různě vysokým rozlišením. Není tedy vhodné, navrhnout jej příliš velké, aby se vešlo na pracovní plochu běžného počítače. Pracovní rozlišení mého monitoru (notebooku), je 1366 x 768 (počet sloupců/počet řádků zobrazených pixelů), což již dnes považuji za standard, například u stolních PC bývá mnohem vyšší. Velikost okna jsem tedy zvolil 1050 x 680, což je dostačující. Horní část bude sloužit, jako lišta záložek, pro přepínání mezi jednotlivými částmi aplikace. Součástí první a zároveň hlavní záložky, bude zobrazení zorného pole senzoru v podobě RGB obrazu, ve kterém se bude současně vykreslovat kosterní model snímané osoby, prostřednictvím vzájemně propojených jointů. V dolní části měl být původně, po nasnímání potřebných dat, vykreslen graf. S tímto grafem však bylo okno aplikace příliš velké a nevešlo by se celé na mou obrazovku. Proto jsem pro graf zvolil samostatné okno, které se otevře po nasnímání všech potřebných dat. Zobrazuje rozsah naměřených hodnot pro jednotlivé jointy a jejich spojnice nebo-li kosti.

Pravá část bude sloužit jako ovládací, bude obsahovat informace o stavu senzoru, spuštění a přerušení snímání, zobrazení ztotožněných osob, či uložení nových a hlasové ověřování. V dalších záložce, kterou jsem pojmenoval Grafy, by mělo být zobrazení nasnímaných hodnot vybraných osob. Grafický návrh je zobrazen na následujícím obrázku (Obr. 23).



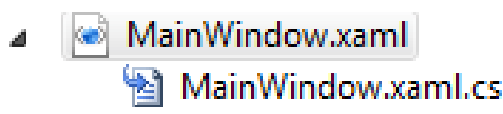
Obrázek 23: Grafický návrh, rozvržení aplikace

4.4 Vytvoření nového projektu

Aplikaci jsem vytvořil pomocí programovacího jazyka C#, který je součástí Microsoft Visual Studio 2010 a to jako tzv. solution. Ten v sobě umožňuje vytvářet a spravovat mnoho projektů, v podstatě samostatných aplikací, které mohou vzájemně spolupracovat, jeden projekt může například vytvářet dynamickou knihovnu, další zase program, jež tuto knihovnu využívá a další může sloužit třeba jako instalátor do Windows. Solution i projekt mohou mít stejný název, pojmenoval jsem ho tedy „CheckPerson“, což by se dalo přeložit jako prověření osoby. Visual studio umožňuje pomocí průvodce vytvářet již přednastavené solutiony, ty však obsahují mnoho, pro mne nepotřebného zdrojového kódu. Proto jsem využil volby „Empty projekt“, který obsahuje jen prázdný adresář References. Ten bude obsahovat referenční datové typy, ty neuchovávají na rozdíl od typů hodnotových pouze určitou hodnotu, ale především odkaz na místo v paměti, kde je uložena daná instance.

4.5 Uživatelské rozhraní

Jako uživatelské rozhraní aplikace poslouží okno, vytvořené prostřednictvím tzv. značkovacího jazyka XAML (eXtensible Application Markup Language). Pro tvorbu vzhledu je to ideální, rychlé a přehledné řešení, jak jednoduše nadefinovat vlastnosti, umístění a spojitosti jednotlivých prvků aplikace. Výhodou je i oddělení uživatelského prostředí od programové části. Okno se tak skládá ze dvou souborů (.xaml a .cs). V prvním souboru je definován vzhled okna (tlačítka, textová pole, barvy atd.) a ve druhém programová část (funkce tlačítek, spuštění animací atd.) (Obr. 24).



Obrázek 24: Zobrazení souboru XAML
v Solution Exploreru jazyka C#

Náhled na okno se záložkami je vidět na následujícím obrázku (Obr. 25). Okno je vytvořeno jednoduše jako objekt <Window>, pomocí parametru Title jsem jej pojmenoval „MainWindow“ a parametry Height (výška), Width (šírka) definují velikost v pixelech. Dále obsahuje typicky dva atributy typu xmlns, které slouží, jako identifikátory pro jmenný prostor (tzv. namespace). Na adresách těchto atributů jsou uloženy různé popisy, specifikace či schémata pro typizaci vzhledu. Záložky Senzor a Grafy jsou tvořeny prostřednictvím kontroly TabControl (umožňují přehledné přepínání mezi částmi aplikace v jednom okně). Zde je ukázka zdrojového kódu.

```
<Window x:Class="CheckPerson.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:KinectControls="clr-namespace:CheckPerson"
  Title="MainWindow" Height="880" Width="1050">

  <Grid>
    <TabControl>
      <TabItem Header="Senzor"><KinectControls:Sensor /></TabItem>
      <TabItem Header="Grafy"></TabItem>
    </TabControl>
  </Grid>
</Window>
```



Obrázek 25: Tvorba uživatelského rozhraní

4.6 Záložka Senzor

První ze záložek slouží jako hlavní část celé aplikace, pojmenoval jsem ji „Senzor“. U programovacích jazyků jsem zvyklý na anglický jazyk, a proto je také celý kód (mimo názvů v uživatelském prostředí) v tomto jazyce. Vytvořil jsem tedy soubory `Sensor.xaml` (definice vzhledu) a `Sensor.xaml.cs` (programová část). Zde je zdrojový kód poměrně obsáhlý, uvedu tedy pouze jeho důležité části.

4.6.1 Inicializace senzoru Kinect

Po spuštění aplikace je nejprve potřeba zjistit, zda je senzor Kinect připojen a připraven k použití. Samozřejmostí je instalace potřebných ovladačů, jež jsou součástí Kinect SDK. Základní příkazy a zdrojové kódy, pro práci se senzorem jsou výborně popsány ve dvou anglických publikacích (*Beginning Kinect Programming With the Microsoft Kinect Sdk* [19] a *Making Things See* [22]), z nichž jsem také čerpal.

Nejdříve tedy zjistíme, zda je senzor připojen a připraven k použití a pokud ano, předáme mu informaci, že má začít vykreslovat obraz z RGB kamery. Stav senzoru je neustále monitorován pomocí timeru (`timerCheckSensor`) a to každých 250 ms.

```
void timerCheckSensor_Elapsed(object sender, System.Timers.ElapsedEventArgs e)
{
    if(_kinectSensor != null && _kinectSensor.IsRunning)
    {
        if(!_worker.IsBusy)
        {
            _worker.RunWorkerAsync();
        }
        return;
    }
}
```

Aplikace zvolí první nalezený senzor (může jich být připojeno i více).

```
_kinectSensor = KinectSensor.KinectSensors.Where(item => item.Status == KinectSta-
tus.Connected).FirstOrDefault();
if(_kinectSensor != null)
```

4.6.2 Nastavení výchozích parametrů

Při spouštění senzoru je potřeba nastavit výchozí parametry pro filtrování tzv. surových dat. Filtrování je běžně využívaný postup při zpracování signálu. Filtry jsou obvykle používány k odstranění nebo zmírnění nežádoucích částí signálu. Tyto parametry jsou pro aplikaci poměrně důležité, proto uvádím jejich význam:

- **Smoothing** (vyhlazování) – určuje míru vyhlazení, výchozí nastavení je 0,5. Při nastavení 0 jsou vrácena všechna hrubá data a při zvyšování úrovně vyhlazování se zvyšuje latence (uloží jen každý n-tý vzorek). Hodnoty musí být v rozsahu [0, 1.0]

- Correction (korekce) – určuje míru korekce. Výchozí nastavení je 0,5. Čím nižší hodnota, tím se data zpracovávají pomaleji, ale jsou přesnější, vyšší hodnota zajistí rychlejší opravy dat, ale s menší přesností. Hodnoty musí být v rozsahu [0, 1.0].
- Prediction (predikce neboli odhad) – určuje počet předpokládaných snímků, výchozí nastavení je opět 0,5 a hodnoty musí být v rozsahu [0, 1.0].
- JitterRadius (poloměr chvění) – určuje redukci chvění. Jitter značí určité rozkolísání (nestabilitu). Výchozí nastavení je 0,05. Všechna odchýlení mimo rádius se upínají právě k jeho hranici.
- MaxDeviationRadius (poloměr maximální odchylky) – určuje maximální rádius, ve kterém se hodnoty zpracovaných dat mohou lišit od hrubých dat. Uvádí se v metrech, výchozí hodnota je 0,04. Filtrované hodnoty, které daný rádius přesáhnou, se upínají k tomuto poloměru.

Na základě několika měření s různým nastavením jsem zvolil tyto parametry:

```
var parameters = new TransformSmoothParameters
{
    Smoothing = 0.1f, Correction = 0.1f, Prediction = 0.1f,
    JitterRadius = 0.03f, MaxDeviationRadius = 0.04f
};
```

Dále se spustí funkce, která umožní snímat data kosterního systému ze senzoru:

```
_kinectSensor.SkeletonStream.Enable(parameters);
_kinectSensor.ColorStream.Enable(ColorImageFormat.RgbResolution640x480Fps30);
_kinectSensor.Start();
_kinectSensor.AllFramesReady += kinectSensor_AllFramesReady; PrepareColorImage();
_isReady = true; }
else{ _isReady = false;}
```

O stavu senzoru bude aplikace informovat uživatele v pravé horní části této záložky (Připraven / Není připraven).

```
lblState.Dispatcher.BeginInvoke(new Action(() => lblState.Content = _isReady ?
"Připraven" : "Není připraven"));
```


4.6.3 Vykreslení RGB obrazu

Vykreslování obrazových dat, přijatých z určitého zdroje se nazývá tzv. streaming (z anglického stream = proud). Jedná se o kontinuální přenos audiovizuálního materiálu mezi zdrojem a cíleným zařízením, který je přímo zobrazován uživateli. Vykreslování barevného obrazu z RGB kamery senzoru, je možno v podobě kódu rozdělit na tři části. Nejprve je streaming zapnut, poté začne aplikace zpracovávat data ze senzoru Kinect a následně je vykresluje na obrazovku monitoru, při čemž 2. a 3. krok je vykonáván neustále (dle nastavení aplikace) [23]. Poté, co aplikace ověří, zda je senzor připraven, se spustí vykreslování obrazu a o to se postará metoda `PrepareColorImage`:

```
private void PrepareColorImage()
{ if(!ColorImage.Dispatcher.CheckAccess())
  { ColorImage.Dispatcher.Invoke(new SimpleDelegate(PrepareColorImage),
    null); return; }

  ColorImageStream colorStream = _kinectSensor.ColorStream;

  _colorImageBitmap = new WriteableBitmap(colorStream.FrameWidth, colorStream.FrameHeight, 96, 96, PixelFormats.Bgr32, null);

  _colorImageBitmapRect = new Int32Rect(0, 0, colorStream.FrameWidth, colorStream.FrameHeight);

  _colorImageStride = colorStream.FrameWidth*colorStream.FrameBytesPerPixel;

  _colorImagePixelData = new byte[colorStream.FramePixelDataLength];

  ColorImage.Source = _colorImageBitmap;
}
```

4.6.4 Vykreslení kosterního systému

Jakmile je senzor připraven, provádí kromě snímání a reprodukce obrazových dat, také neustálé vyhodnocování dat ze senzoru hloubkového. Vysíláním a přijímáním odražených infračervených paprsků, jsou získávána prostorová data, na základě nichž je možno rozpoznat přítomnost osoby. Osoba je trojrozměrný objekt, jemuž jsou přiděleny jointy (přirovnáme-li tento model ke kostře člověka, jedná se v podstatě o klouby). Jointy jsou (nemusí být) spojeny, pro lepší vizualizaci, pomocí spojitých čar a pro nás představují kosti kosterního systému. O sledování a vykreslování jednotlivých jointů se stará následující část kódu:

```
void kinectSensor_AllFramesReady(object sender, AllFramesReadyEventArgs e)
{ bool hasData = false;
  using (SkeletonFrame skeletonFrame = e.OpenSkeletonFrame())
  { if(skeletonFrame != null)
    { hasData = true;
      _skeletonData = new Skeleton[skeletonFrame.SkeletonArrayLength];
    }
  }
}
```

```

        skeletonFrame.CopySkeletonDataTo(_skeletonData); }}
    if (hasData)
    { if ( _skeletonData != null && _skeletonData.Any(item =>
        item.TrackingState == SkeletonTrackingState.Tracked))
        { foreach (var skeleton in _skeletonData.Where(item =>
            item.TrackingState == SkeletonTrackingState.Tracked))
            { DrawBones(skeleton);
              CheckPersonInCamera();}}}}

```

K vykreslení spojnic určitých jointů do tzv. plátna (Canvas), tak aby byla vytvořena iluze kostí člověka, je použito kontroly Line (z angl. čára), jak ukazuje následující část kódu. Zvolíme první a druhý joint pro danou kost (v závislosti na typu kosti):

```

var joint1 = skeleton.Joints.Where(item => item.JointType == bone.JointType1 &&
item.TrackingState == JointTrackingState.Tracked).First();

var joint2 = skeleton.Joints.Where(item => item.JointType == bone.JointType2 &&
item.TrackingState == JointTrackingState.Tracked).First();

var line = new Line();line.StrokeThickness = 3;line.Stroke = Brushes.Red;

```

Přepočítají se surová data pozic jointů (z metrů) na pozice obrazových dat (v pixelech):

```

line.X1 = _kinectSensor.MapSkeletonPointToColor(joint1.Position, ColorImageFor-
mat.RgbResolution640x480Fps30).X;
line.Y1 = _kinectSensor.MapSkeletonPointToColor(joint1.Position, ColorImageFor-
mat.RgbResolution640x480Fps30).Y;
line.X2 = _kinectSensor.MapSkeletonPointToColor(joint2.Position, ColorImageFor-
mat.RgbResolution640x480Fps30).X;
line.Y2 = _kinectSensor.MapSkeletonPointToColor(joint2.Position, ColorImageFor-
mat.RgbResolution640x480Fps30).Y;

```

Následně je volána metoda pro vykreslení kostí do RGB obrazu:

```

SkeletonView.Children.Add(line);

```

Prostřednictvím metody, nazvané CheckPersonIsReady, je definováno, kdy má začít snímání dat (osoba musí udělat „lidský“ kříž rozpažením rukou) a následné porovnání sejmutých dat s těmi, které jsou uloženy v databázi. Rozpažení osoby je ošetřeno podmínkami, kdy příslušné jointy v oblasti rukou, dosahují požadovaných hodnot na osách x, y, z. Například joint, nacházející se v oblasti zápěstí levé ruky, musí být přibližně ve stejné vzdálenosti na ose z, jako joint pravé ruky a oba přibližně ve stejné výšce na ose y. Následující část kódu zajišťuje proces, probíhající po sejmutí dat. Jsou-li sejmuta data, následuje ověření, zda je osoba obsažena v databázi. Aplikace je ovládána také hlasem. V určitých fázích práce s aplikací je hlasové rozpoznávání nežádoucí a to například zde, po nashromáždění dat, kdy se zobrazí panel pro uložení nové osoby. Při testování docházelo k tomu,

že při vyplňování údajů osoby, kdy v okolí probíhal běžný hovor, aplikace zachytila některé ze slov, jako jeden z anglických výrazů, určených k jejímu ovládní. Proto jsem vytvořil proměnnou `_block`, která v požadovaných fázích práce s aplikací, zajišťuje vypnutí hlasového rozpoznávání.

```
if (_isDataGathered && !_block)
{
    _block = true;
}
```

Pro porovnávání kostí jednotlivých osob, jsem původně použil průměrné hodnoty délek každé kosti (průměr z 50 – 100 vzorků). Tento způsob se však příliš neosvědčil, jelikož hodnoty hodně kolísaly. Na kvalitu sejmutých dat mají výrazný vliv i okolní podmínky, jako osvětlení, vzdálenost od senzoru či další objekty ve snímaném prostoru. Z určitého počtu vzorků se však vždy nejčastěji opakovali hodnoty, udávající skutečnou vzdálenost dvou sousedních jointů (představujících kost). Ostatní hodnoty kolísaly velmi nepředvídatelně, což mnohdy zapříčinilo, že výsledná průměrná hodnota délky kosti se hodně lišila od předchozího měření (někdy i o 0.06 m, což je nežádoucí).

Proto jsem zvolil metodu, kdy délka měřené kosti odpovídá nejčastěji se opakující hodnotě v daném počtu vzorků. Při snímání je vždy délka kosti daného vzorku vypočtena pomocí matematického vzorce. Vychází se ze šesti hodnot, udávajících pozici dvou jointů (v ose x, y, z, pro každý joint). Jeden vzorek tedy odpovídá jednomu údaji o délce dané kosti. Tyto vzorky dosahují různých, ale velmi blízkých hodnot (v rozmezí několika cm). Vzorky jsou sbírány tak dlouho, dokud kterákoliv z hodnot nedosáhne daného počtu opakování (zvolil jsem 60). Tato hodnota je pak vyhodnocena jako výchozí údaj o délce kosti a následně porovnávána s databází, jak je vidět na následující části zdrojového kódu:

```
SetDiameterForMostFrequentSamples();
var skeletonData = new SkeletonData(_bones);
```

Porovnáme získaná data s databází:

```
_persons = PersonDL.CheckSkeletonData(skeletonData);
PersonView.Visibility = Visibility.Visible;
```

Pokud osoba není nalezena, zobrazí se příslušná hláška, s možností uložení dané osoby jako nové, tedy její zavedení do databáze.

```
if (_persons == null)
{
    personFind.Content = "Osoba nenalezena";
    txtFirstName.Visibility = Visibility.Visible;
}
```

```

txtLastName.Visibility = Visibility.Visible;
txtSecretWord.Visibility = Visibility.Visible;
btnSavePerson.Visibility = Visibility.Visible;
lblSecretWord.Visibility = Visibility.Visible;
lblFirstName.Visibility = Visibility.Hidden;
lblLastName.Visibility = Visibility.Hidden;
lblFirstName.Content = string.Empty;
lblLastName.Content = string.Empty;
btnSaveAsNew.Visibility = Visibility.Hidden;
}
Else { btnSaveAsNew.Visibility = Visibility.Visible;

```

Pokud je nalezena, zobrazí se její údaje:

```

if (_persons.Length == 1)
{
    SetPerson(_persons[0]);
}

```

Aplikace porovnává nasnímaná data s databází v určité toleranci (2cm na kost), může tedy dojít k tomu, že dané osobě odpovídá více osob, uložených v databázi a proto je zde pro každou osobu uloženo také „tajné slovo“, které v tomto případě snímaná osoba vysloví a aplikace ověří, zda vyřčený výraz (může to být více slov za sebou) odpovídá někomu, který je přiřazen osobám uloženým v databázi.

```

Else
{
    SinglePerson.Visibility = Visibility.Hidden;
    saySecretWord.Visibility = Visibility.Visible;
    btnSaveAsNew.Visibility = Visibility.Visible;
    personFind.Content = "Nalezeno více osob";
}

```

Pro hlasové rozpoznávání se využívají slovníky nebo spíše mluvnice (z angl. grammars), které obsahují konkrétní výrazy anglického jazyka, například výrazů pro ovládání aplikace nebo tajných slov, sloužících pro konečnou identifikaci osoby. V aplikaci jsou použity dva slovníky a to ApplicationGrammar a PersonGrammar, které obsahují potřebná slova. Metoda nazvaná _speechRecognitionEngine jich pak využívá pro rozpoznávání hlasových pokynů. Vyslovený výraz je porovnáván s oběma slovníky. Navážu-li na předchozí část kódu, kdy nalezneme více osob, odpovídajících snímané osobě, vyslovíme „tajné slovo“ a aplikace by měla toto slovo najít ve slovníku PersonGrammar, kam se také v tomto případě obrací (viz. následující část kódu):

```

var choices = new Choices(_persons.Select(item => item.SecretWord).ToArray());
var grammarBuilder = new GrammarBuilder();
grammarBuilder.Culture = _recognizerInfo.Culture;
grammarBuilder.Append(choices);
_personGrammar = new Grammar(grammarBuilder);
_personGrammar.Name = "PersonGrammar";
_speechRecognitionEngine.LoadGrammar(_personGrammar);}

```

4.6.5 Výpočet délky jednotlivých kostí

Senzor zaznamenává pozice jednotlivých jointů a tato data předává osobnímu počítači, kde jsou pomocí této aplikace zpracovávána. Abychom získaly délku každé z devatenácti spojníc neboli kostí, je potřeba určitého vzorce. Vzhledem k tomu, že každý joint je zaznamenán ve 3 osách, máme tak potřebné informace o jeho pozici v trojrozměrném prostoru. Použijeme-li následujícího matematického vzorce, převedeného pomocí příkazů programovacího jazyka přímo do kódu aplikace, získáme údaj o vzdálenosti mezi dvěma sousedícími jointy. Kdy x_1, y_1, z_1 zastupují pozici 1. jointu a x_2, y_2, z_2 pozici druhého.

Matematický vzorec: $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$

Tento vzorec aplikujeme pomocí objektu Math, určeného pro matematické funkce do kódu aplikace a proměnnou pro délky kostí nazveme třeba Length (z angl. délka):

```
public double Length
{get
{return Math.Round(Math.Sqrt(Math.Pow((BoneVectorJ1.X - BoneVectorJ2.X), 2) + Math.Pow((BoneVectorJ1.Y - BoneVectorJ2.Y), 2) + Math.Pow((BoneVectorJ1.Z - BoneVectorJ2.Z), 2)), 2); }}
```

Takto získáme délku kosti jednoho vzorku. My však potřebujeme získat výslednou délku, kterou použijeme při ověřování osoby. Tu získáme vyhodnocením délek jednotlivých vzorků. Původním záměrem bylo nastavit počet vzorků například na 50 a výslednou délku získat jejich zprůměrováním. To se však neosvědčilo, což je vysvětleno v kapitole 4.10.1 na provedeném měření. Zvolil jsem tedy jinou metodu. Výsledná délka bude odpovídat nejčastěji naměřené hodnotě. Proces snímání bude tedy nastaven tak, že pro každý vzorek bude probíhat přepočítání na délku kosti tak dlouho, dokud se u každé kosti nenashromadí stejných hodnot. První hodnota, která se bude při vzorkování opakovat n-krát (bude tedy nejčastější), bude použita jako výsledná délka kosti.

Hodnota n tedy představuje podmínku, která ukončí snímání dané kosti, jakmile se některá z délek jednotlivých vzorků opakuje n-krát. Počet opakujících se hodnot jsem nastavil na $n = 60$ a pro usnadnění (v kódu se vyskytuje hodnota n vícekrát), jsem si vytvořil proměnnou `_mostFrequentSampleCount`, kterou jsem nastavil na 60:

```
private int _mostFrequentSampleCount = 60;
```

Při snímání jsou pak sesbírány vypočtené délky kostí pro každý vzorek.

```
var groupedByMostFrequent = from bs in _boneSamples.Where(item => item.BoneType == bone.BoneType) group bs by bs.Length into grouped select new { Length = grouped.Key, Count = grouped.Count() };
```

A jakmile je u dané kosti zjištěno 60 stejných hodnot, tato nejčastější hodnota je označena jako výsledná a kontrolují se další kosti:

```
if (groupedByMostFrequent.Count() > 0 && groupedByMostFrequent.Max(item => item.Count) == _mostFrequentSampleCount)
{
    continue;
}
```

Po získání výsledných délek všech kostí nastává porovnání s databází.

4.7 Záložka Grafy

Poté, co jsou data aktuálně snímané osoby uložena do paměti, necháme aplikaci vykreslit jednoduchý graf (angl. chart) a to jak pro jointy, tak pro jejich spojnice neboli kosti. Grafy poslouží, jako vizualizace přesnosti senzoru, s jakou měří jednotlivé vzdálenosti každého jointu v osách x, y, z. V záložce Grafy je možno nahlížet na všechny osoby, uložené v databázi. Pro každou osobu jsou zde vyobrazeny krajní hodnoty délek jednotlivých kostí, včetně střední hodnoty. Senzor dokáže sledovat a zaznamenávat u jedné osoby až 20 jointů, pro účely aplikace jsem však zvolil pouze 16 jointů. Vynechal jsem jointy v oblasti dlaní a chodidel. Vzdálenosti od těchto jointů k sousedním, jsou velmi malé a především kolísavé. Pro identifikace jsou vhodné delší a ustálenější kosti. Pro vyobrazení těchto grafů je zde kontrola nazvaná ActualPersonCharts:

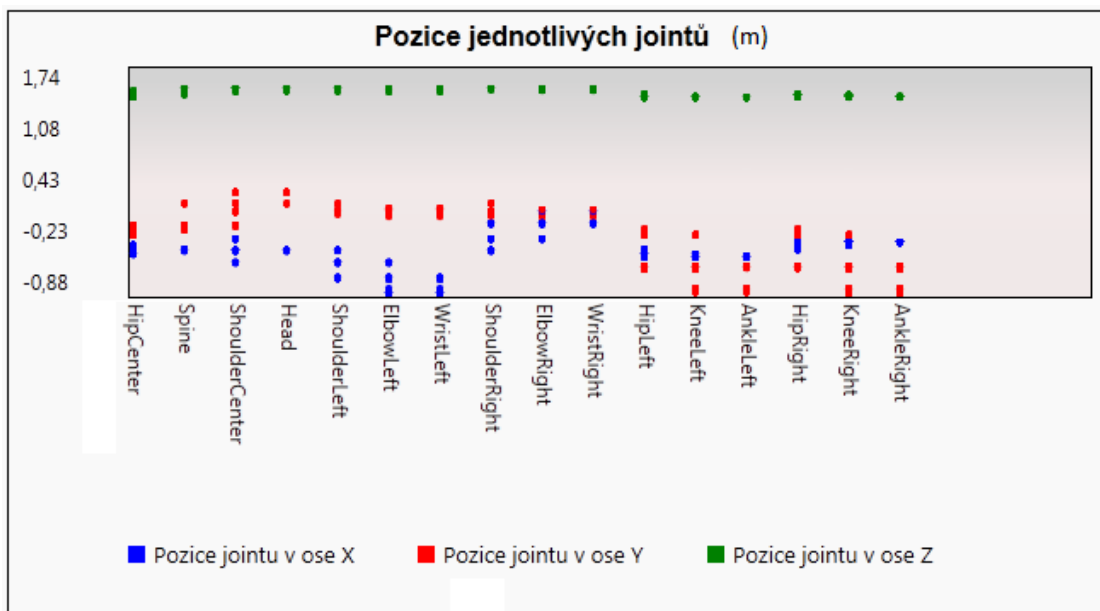
```
_personCharts = new ActualPersonCharts();
_personCharts.BoneSamples = _boneSamples;
_personCharts.Show();
```

Při zavolání metody Show se otevře kontrola ActualPersonCharts, která je definována jako okno (window). Zde je událost WindowLoaded, ve které se zavolá metoda DrawCharts, určená pro vykreslení grafů:

```
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    DrawCharts();
}
```

Graf pak může vypadat následovně (Obr. 26), kdy jednotlivé barvy znázorňují naměřené hodnoty os x, y a z pro jednotlivé jointy. V levé části jsou vyčísleny orientační hodnoty (v metrech) a to v rozsahu od minima po maximum. Z grafu je tedy patrné, že snímaná osoba (respektive některý její joint), se nacházela v maximální vzdálenosti 1,74m.

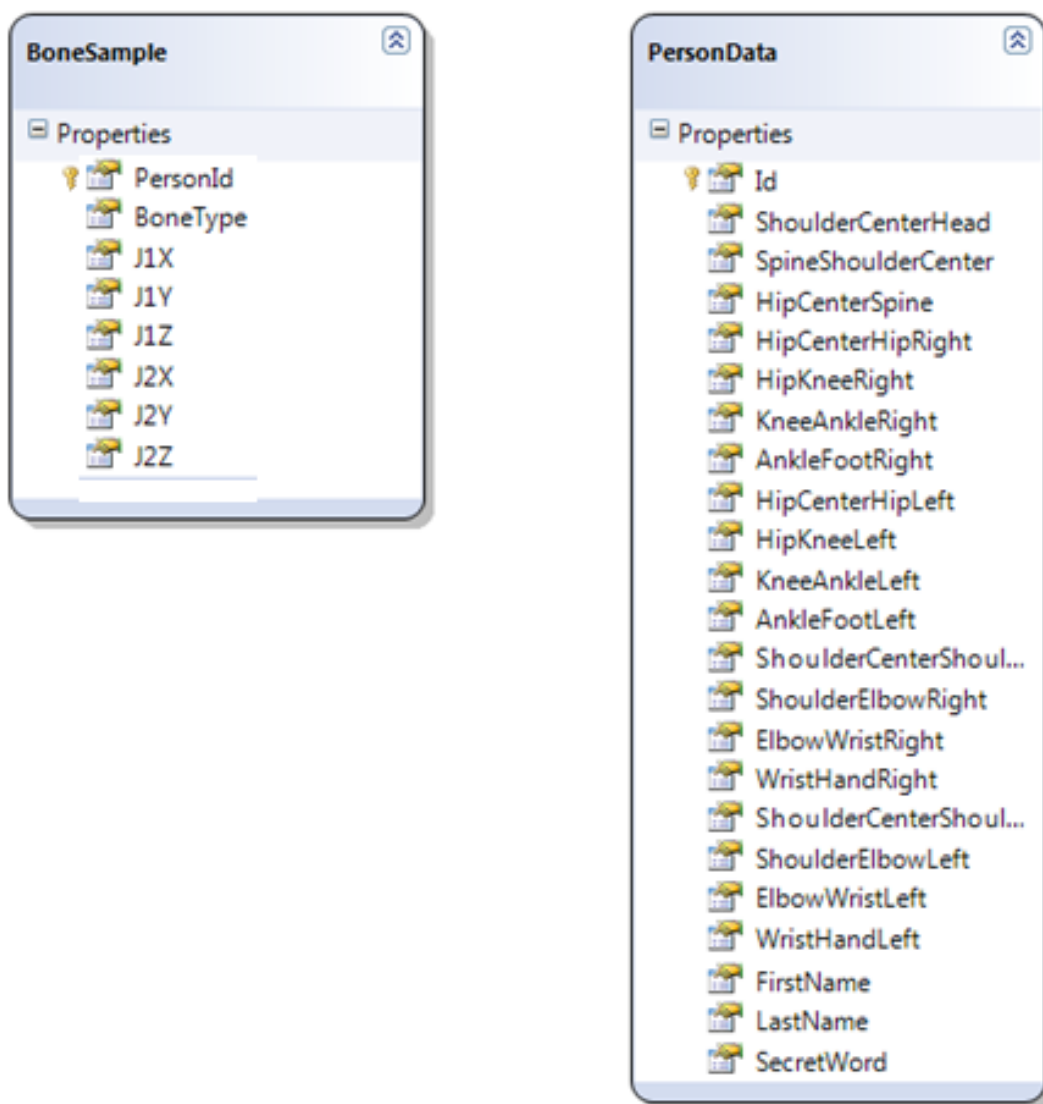
Hodnoty os X a Y (mohou být i záporné) u této osoby hodně kolísají, protože se příliš pohybovala nebo zde byly špatné světelné podmínky.



Obrázek 26: Graf naměřených hodnot pro jednotlivé jointy

4.8 Databáze osob

Pro ukládání a evidenci osob, včetně potřebných dat, je potřeba vytvořit databázi. Tato aplikace slouží jako ukázková a nebude se připojovat na žádný server, databáze tedy poběží na lokálním počítači. Abych se vyhnul použití klasického databázového systému MS SQL, zvolil jsem jednodušší verzi, pro méně náročné aplikace, tedy SQL Server Compact edition, která je zdarma a umožní vytvoření jednoduché databáze, v podobě datového souboru .sdf (standard data file) [24]. Výhodou této edice je fakt, že neběží jako služba, ale je představována pouze tímto databázovým souborem. S takovouto databází pak můžeme snadno pracovat pomocí Database Exploreru, který je součástí Visual Studia. Databázi tvoří dvě tabulky, nazvané BoneSample a PersonData (Obr. 27).



Obrázek 27: Uspořádání tabulek v databázi

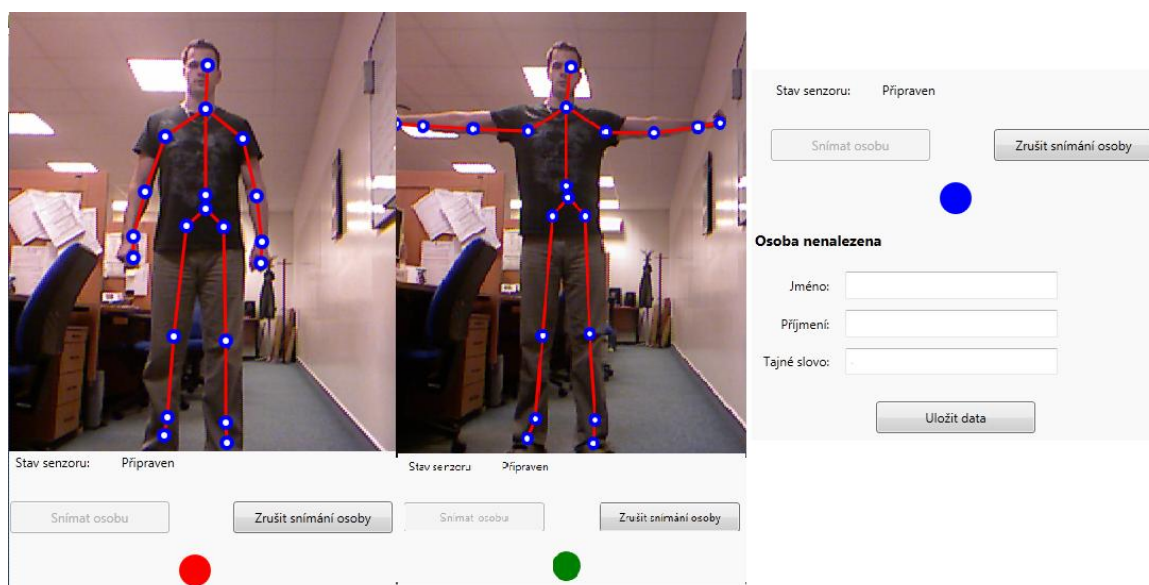
Tabulka PersonData slouží pro uložení osob a jejich vlastností. Obsahuje primární klíč, typicky Id (jedinečný identifikační údaj), jméno a příjmení osoby, tajné slovo (pro případ konečné identifikace, pokud aplikace najde více osob s téměř totožnými délkami kostí) a všechny kosti kosterního systému, kam se ukládají jejich délky, vypočtené z poloh jednotlivých jointů. Tabulka BoneSample obsahuje uložené vzorky surových dat, tedy polohu jednotlivých jointů ve všech osách. Pro potřeby výpočtů, se pro každou kost ukládají pozice dvou sousedních jointů (J1X, J1Y, J1Z, J2X, J2Y, J2Z), dále všechny typy kostí (BoneType) a PersonId pro přiřazení konkrétní osobě.

4.9 Práce s aplikací

4.9.1 Obsluha aplikace

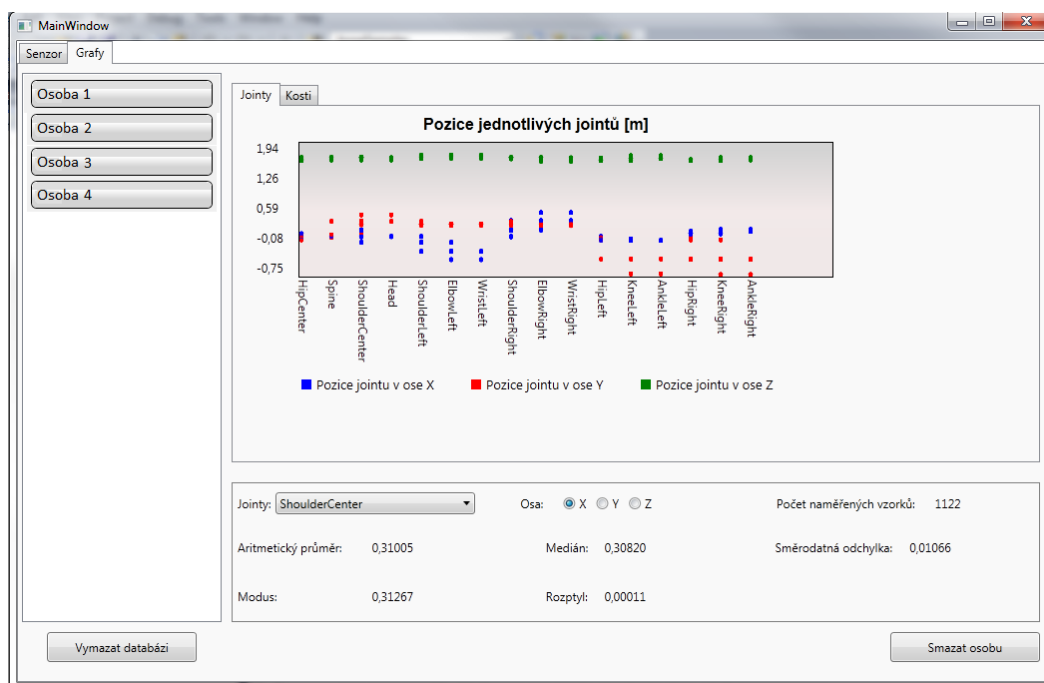
Aplikaci spustíme klasickým souborem typu .exe a to v podobě okna (MainWindow). Záložka Sensor je nastavena, jako hlavní. V pravém horním rohu je zobrazen stav senzoru (připojen/nepřipojen). S malým zpožděním (několika sekund) se po spuštění zároveň začne vykreslovat RGB obraz. Nyní budeme chtít ověřit, zda osoba, která se nachází v zorném poli senzoru, odpovídá některé z osob, uložených v databázi. Sběr dat, potřebných k porovnání s databází, bude spuštěn pomocí tlačítka „Snímat osobu“.

Na následujícím obrázku (Obr. 28) jsou vidět jednotlivé fáze snímání. Nejprve počkáme, až senzor rozpozná ve svém zorném poli osobu a ustálí se naklápěcí základna, která automaticky sleduje jeden konkrétní joint, jež se nachází zhruba uprostřed osoby. Osoba se tedy může libovolně přibližovat a oddalovat od senzoru a je neustále sledována (rozsah pohybu je samozřejmě omezen). Indikátor snímání (barevný puntík) je zatím červený. Sběr dat započne rozpažením osoby aneb vytvořením „lidského kříže“. Po celý čas snímání je indikátor zelený, a jakmile je splněna podmínka pro ukončení snímání (je nasbíráán dostatečný počet vzorků pro každou kost), indikátor změní barvu na modrou. Nyní buďto rozpozná osobu porovnáním s databází nebo nabídne uložení nové osoby.



Obrázek 28: Práce s aplikací

Po ukončení snímání se zároveň v nově vytvořeném okně vykreslí grafy jointů a kostí. Tyto jsou dostupné k prohlížení prostřednictvím záložky Grafy a to kdykoliv v pozdější době, podmínkou je samozřejmě jejich uložení, jakožto nové osoby (Obr. 29). V levé části jsou prostřednictvím ListBoxu (seznamu položek), seřazeny všechny osoby, uložené v databázi. V pravé části se pak vykresluje graf, stejně jako při snímání aktuální osoby a zde si můžeme zpětně prohlížet, s jakými naměřenými hodnotami, byla osoba uložena.



Obrázek 29: Záložka Grafy – prohlížení uložených osob

Ve spodní části jsou tlačítka pro vymazání vybrané osoby nebo vymazání celé databáze a pod grafem jsou zobrazeny některé důležité hodnoty pro jednotlivé kosti, které blíže popisují.

Aritmetický průměr – vyjadřuje průměrnou hodnotu pozice daného jointu pro osy x, y, z nebo průměrnou délku dané kosti.

Modus – je hodnota s největší relativní četností výskytu neboli hodnota, která se opakuje nejčastěji.

Medián – je střední hodnota, jež dělí řadu podle velikosti seřazených výsledků na dvě stejně početné poloviny a vyjadřuje střed této řady (při sudém počtu vzorků jednu ze dvou hodnot nejbližších středu).

Rozptyl – vyjadřuje variabilitu rozdělení souboru náhodných hodnot kolem její střední hodnoty (také označována jako střední kvadratická odchylka).

Počet naměřených vzorků – počet nasnímaných vzorků, je stejný pro všechny kosti, jelikož se snímá tak dlouho, dokud není u každé kosti zjištěno 60 stejných hodnot.

Směrodatná odchylka – jedná se o kvadratický průměr odchylek hodnot od jejich aritmetického průměru, vypovídá o tom, jak moc se od sebe nejčastější hodnoty liší.

4.9.2 Hlasové ovládání

Kromě klasické myši a klávesnice, lze aplikaci ovládat také hlasem. Rozpoznávání hlasu bylo optimalizováno pro několik světových jazyků. Pro nás je nejvhodnějším jazyk anglický. Využijeme tedy libovolných výrazů tohoto jazyka, kterým přidělíme určité funkce. Vyslovení takového výrazu (může jím být slovo, ale i několik po sobě následujících slov) vyvolá přerušení, na jehož základě proběhne předem definovaná událost. Například vysloví-li osoba výraz „Start“, započne snímání potřebných dat. Přerušit jej můžeme výrazem „Cancel“. Výrazů je v aplikaci užito málo, můžeme si je tedy uvést:

- Start – spuštění snímání dat
- Cancel – přerušení snímání dat
- Angle Up – naklonění naklápečí základny o určitý úhel nahoru.
- Angle Down – naklonění naklápečí základny o určitý úhel dolů.
- Sensor – přepnutí na záložku Senzor
- Charts – přepnutí na záložku Grafy
- Close Charts – zavření okna s grafem pro aktuálně snímanou osobu
- Focus Application – přesune hlavní okno aplikace na popředí
- Focus Charts – přesune okno s grafem aktuálně snímané osoby do popředí

5 OVĚŘENÍ PŘESNOSTI SENZORU

Pro ověření, s jakou přesností senzor dokáže měřit pozice jednotlivých jointů, jsem provedl několik měření za určitých podmínek (světlo, oblečení, vzdálenost od senzoru) a to nejprve s použitím původní podmínky (délka kosti, jako průměr hodnot snímaných vzorků) tak s konečnou podmínkou (nejčastější hodnota). Aplikace tedy pomocí vzorce vypočítá pro každý ze vzorků délku dané kosti. Tak získáme pro každou kost určitý počet vzorků, ze kterých se na základě podmínky stanoví výsledná délka dané kosti, která se poté porovnává s databází. Hodnoty jsou zaokrouhleny na dvě desetinná místa.

5.1 Výsledky měření s výpočtem průměrné hodnoty

Při prvním měření bylo pro určení výsledné délky nasnímáno pro každou kost 50 vzorků (pozice 2 sousedních jointů ve všech osách). Pro každý vzorek byla vypočtena délka kosti pomocí příslušného vzorce, který je uveden v kapitole 4.6.5. Dostaneme tedy 50 hodnot, z nichž každá odpovídá délce kosti v 1 okamžitém snímku senzoru. Konečnou délku dané kosti získáme, spočítáme-li průměrnou hodnotu z nasbíraných vzorků. V následující tabulce (Tab. 1) jsou zachyceny výsledky deseti měření pro 6 snímaných kostí. Osoba vždy rozpaží ruce a vysloví příkaz (start) pro snímání. Snímají se tedy kosti od krku k rameni, k lokti, až po dlaň (tyto jsem zvolil jako nejvhodnější). Hodnoty jsou uvedeny v metrech.

Měření	SCSR	SER	EWR	SCSL	SEL	EWL
1	0,21	0,22	0,24	0,21	0,22	0,23
2	0,21	0,24	0,23	0,21	0,22	0,23
3	0,24	0,27	0,24	0,22	0,28	0,24
4	0,24	0,26	0,22	0,21	0,27	0,22
5	0,23	0,25	0,23	0,22	0,26	0,23
6	0,24	0,27	0,23	0,22	0,26	0,23
7	0,24	0,27	0,23	0,22	0,27	0,26
8	0,25	0,26	0,23	0,22	0,27	0,25
9	0,24	0,26	0,23	0,21	0,25	0,23
10	0,24	0,27	0,24	0,22	0,25	0,23

Tabulka 1: Výsledky měření – metoda s výpočtem průměru

Kosti jsou v tabulce zapsány zkráceně jako ShoulderCenterShoulderRight (SCSR), ShoulderElbowRight (SER), ElbowWristRight (EWR), ShoulderCenterShoulderLeft (SCSL), ShoulderElbowLeft (SEL), ElbowWristLeft (EWL).

Z tabulky je vidět, že přesnost měření je pro účely identifikace velmi špatná. Hodnoty dosti kolísají. Například délka kosti, označené zkratkou SEL (od ramene k lokti), byla v jednom případě naměřena s hodnotou 0,22 m a v jiném případě s hodnotou 0,28 m, což je velmi nepřesné. V praxi by pak docházelo k tomu, že kdybychom například uložili právě hodnotu 0,28 m, osoba by při toleranci 0,02 m byla rozpoznána jen v 6 z 10 případů. Na následujícím obrázku (Obr. 30) je v případě levého chodidla vidět, že senzor má občas potíže správně určit pozici některých jointů a v různých okamžicích tak můžou být nasbírané vzorky dosti odlišné od ostatních. Stejně tak se může stát, že senzor na malý okamžik vyhodnotí některý z objektů v jeho zorném poli (například židli) jako osobu a takto získaný vzorek je samozřejmě nepoužitelný. V aplikaci by šlo tyto nepřesnosti ošetřit například odebráním určitého procenta odlišných naměřených hodnot. Vzhledem k tomu, že výskyt těchto nežádoucích odchylek je ve větším počtu vzorků malý, zvolil jsem jinou metodu, pro získání výsledné délky (viz. následující kapitola).



Obrázek 30: Chybné určení pozice jointu

5.2 Výsledky měření s vyčíslením nejčastější hodnoty

Tato metoda je popsána v kapitole 4.6.5. Vzhledem k výsledkům následujících tabulek (Tab. 2 a Tab. 3), jsem ji zvolil jako výchozí. Výsledek každého měření, je stále stejný, což je přesně to, co pro identifikaci potřebujeme. Všechna měření jsem uložil jako samostatné osoby, aplikace tak našla shodu ve všech těchto osobách a vyzvala snímanou osobu k vyslovení „tajného slova“, na jehož základě byla identifikována daná osoba. Snímány byly 2 různé osoby, různého pohlaví a věku. Tolerance snímání byla nastavena na 0,01 m, zde jí však nebylo potřeba. Na výsledcích měření lze vidět, že nejčastější hodnota skutečně převažuje nad ostatními, jelikož se ani v jednom z 10 měření nezměnila.

Objekt 1

Měření	SCSL	SEL	EWL	SCSR	SER	EWR
1	0,18	0,26	0,22	0,17	0,24	0,21
2	0,18	0,26	0,22	0,17	0,24	0,21
3	0,18	0,26	0,22	0,17	0,24	0,21
4	0,18	0,26	0,22	0,17	0,24	0,21
5	0,18	0,26	0,22	0,17	0,24	0,21
6	0,18	0,26	0,22	0,17	0,24	0,21
7	0,18	0,26	0,22	0,17	0,24	0,21
8	0,18	0,26	0,22	0,17	0,24	0,21
9	0,18	0,26	0,22	0,17	0,24	0,21
10	0,18	0,26	0,22	0,17	0,24	0,21

Tabulka 2: Výsledky měření – metoda nejčastější hodnoty (Objekt 1)

Objekt 2

Měření	SCSL	SEL	EWL	SCSR	SER	EWR
1	0,19	0,25	0,24	0,2	0,28	0,23
2	0,19	0,25	0,24	0,2	0,28	0,23
3	0,19	0,25	0,24	0,2	0,28	0,23
4	0,19	0,25	0,24	0,2	0,28	0,23
5	0,19	0,25	0,24	0,2	0,28	0,23
6	0,19	0,25	0,24	0,2	0,28	0,23
7	0,19	0,25	0,24	0,2	0,28	0,23
8	0,19	0,25	0,24	0,2	0,28	0,23
9	0,19	0,25	0,24	0,2	0,28	0,23
10	0,19	0,25	0,24	0,2	0,28	0,23

Tabulka 3: Výsledky měření – metoda nejčastější hodnoty (Objekt 2)

Měření jsem prováděl v běžné obývací místnosti s nábytkem a to jak za světla, tak za úplné tmy, hloubkový senzor tedy plní svou funkci. Přestože jsou výsledky lepší než v předchozím případě, stále je vidět, že kosti levé paže se liší od pravé paže, v jednom případě až o 0,03 m. Proto zvolena tolerance (0,01 - 0,02 m). Snímání pozic jointů není příliš přesné a senzor tedy není vhodný do míst, kde je kladen důraz na zabezpečení. Tato funkce je vhodná spíše v zábavním průmyslu, například pro rozpoznání členů rodiny apod.

ZÁVĚR

Senzor Kinect je výrobcem označován jako pohybový ovladač, což přesně vystihuje jeho hlavní dominantu. Sledování pohybu lidského těla je velmi kvalitní, což jsem si při analýze senzoru ověřil. Z výsledného měření jsem však zjistil, že pro využití při identifikaci osob je sice senzor použitelný, ale je třeba počítat s menší přesností. Identifikace byla provedena srovnáváním délky kostí. Jednotlivé vzorky se však dosti lišily, protože výsledky měření nejsou za všech podmínek stejné a snímání narušuje více okolností, jako odlesky světla na snímací optice či objekty v zorném poli senzoru, chybně vyhodnocené jako člověk. Při rozpoznávání osob je potřeba nastavit určitou toleranci, s jakou jsou získaná data vyhodnocována. Bez této tolerance by mohlo docházet k nutnosti opakovaného snímání osoby, což je nevyhovující. S vyšší tolerancí zase může dojít k tomu, že snímaná osoba sice bude rozpoznána, ale aplikace jí přiřadí identitu osoby, jejíž vzorky jsou v toleranci (jsou tedy podobné, ne však stejné). Nastavíme-li vhodné podmínky pro snímání osob, můžeme dosáhnout uspokojivých výsledků, nikdy však nebude identifikace osob zdaleka tak přesná, jako například snímání otisků prstů nebo povrchu oční duhovky. Senzor Kinect hodnotím, jako zajímavé zařízení, nabízející mnoho funkcí za dobrou cenu, je však potřeba znát jeho možnosti a tomu přizpůsobit nároky na něj kladené.

Cílem této bakalářské práce bylo analyzovat senzor Kinect a s použitím příslušného softwaru, vytvořit samostatnou aplikaci, na které by bylo možno demonstrovat jeho schopnosti při identifikaci osob. To se povedlo a výsledná aplikace je schopna s určitou tolerancí a za vhodných podmínek identifikovat osoby, jejichž nasnímaná data jsou uložena v databázi aplikace. Přesnost měření však není taková, v jakou jsem zpočátku doufal, ale dostačující pro nenáročné účely, jako nabízí například zábavní průmysl a zde také nachází své uplatnění.

SEZNAM POUŽITÉ LITERATURY

- [1] Kinect pro Xbox 360 - Xbox.com. MICROSOFT. [online]. 2012 [cit. 2012-06-05]. Dostupné z: <http://www.xbox.com/cs-CZ/Kinect?xr=shellnav>
- [2] ŠČUREK, Radomír. Biometrické metody identifikace osob v bezpečnostní praxi: Studijní text. [online]. 2008 [cit. 2012-06-05]. Dostupné z: http://www.fbi.vsb.cz/miranda2/export/sites-root/fbi/040/cs/sys/resource/PDF/biometricke_metody.pdf
- [3] Alphonse Bertillon - Wikipedie. *Wikipedie: Otevřená encyklopedie* [online]. 8. 2. 2012 [cit. 2012-06-05]. Dostupné z: http://cs.wikipedia.org/wiki/Alphonse_Bertillon
- [4] MERCAN, Murat. Kinect Part 5: Kinect Skeleton Tracking. *Murat Mercan: Ideas a Sketches* [online]. 8.11.2011 [cit. 2012-06-05]. Dostupné z: <http://www.mrtmrcn.com/en/post/2011/11/08/Kinect-Part-5-Kinect-Skeleton-Tracking.aspx>
- [5] JetVoice - hlasové ovládání počítače. [online]. 31.12.2011 [cit. 2012-06-05]. Dostupné z: <http://www.jrr.cz/jetvoice.php>
- [6] MSDN Blogs: Kinect for Windows Blog. MSDN [online]. 2012 [cit. 2012-06-05]. Dostupné z: <http://blogs.msdn.com/b/kinectforwindows/archive/2012/05/21/kinect-for-windows-runtime-and-sdk-version-1-5-released.aspx>
- [7] Začínáme se senzorem Kinect - Xbox.com. MICROSOFT. [online]. 2012 [cit. 2012-06-05]. Dostupné z: <http://www.xbox.com/cs-CZ/Kinect/GetStarted>
- [8] Patentscope: Method and system for object reconstruction. WIPO [online]. 19.4.2007 [cit. 2012-06-05]. Dostupné z: <http://www.wipo.int/patentscope/search/en/WO2007043036>
- [9] HADÁČEK, Jan. Přizpůsobení 3D modelu lidské postavy s použitím reálných dat. Plzeň, 2011. bakalářská práce (Bc.). ZÁPADOČESKÁ UNIVERZITA V PLZNI. Fakulta aplikovaných věd
- [10] Project Natal: The Mystery of Project Natal Revealed. GIZMODO [online]. [cit. 2012-06-05]. Dostupné z: <http://gizmodo.com/5506395/the-mystery-of-project-natal-revealed>

- [11] BORENSTEIN, Greg, Andrew ODEWAHN a Brian JEPSON. *Making things see: 3D vision with Kinect, Processing, Arduino, and MakerBot*. 1st ed. [Make: Books], 2012c2012, 416 s. ISBN 14-493-0707-8.
- [12] Nyko Xbox 360 Products: Zoom. NYKO [online]. 2011 [cit. 2012-06-05]. Dostupné z: <http://nyko.com/products/product-detail/?name=Zoom>
- [13] Microsoft Kinect Teardown. IFIXIT [online]. 2012 [cit. 2012-06-05]. Dostupné z: <http://www.ifixit.com/Teardown/Microsoft-Kinect-Teardown/4066/1>
- [14] How Motion Detection Works in Xbox Kinect. GADGED LAB [online]. 3.11.2010 [cit. 2012-06-05]. Dostupné z: <http://www.wired.com/gadgetlab/2010/11/tonights-release-xbox-kinect-how-does-it-work/>
- [15] Kinect for Windows: Download Kinect for Windows SDK. MICROSOFT. [online]. 2012 [cit. 2012-06-05]. Dostupné z: <http://www.microsoft.com/en-us/kinectforwindows/develop/overview.aspx>
- [16] MICHL, Vladimír. Cad Fórum - Wiki: Mračno bodů. [online]. 19.8.2010 [cit. 2012-06-05]. Dostupné z: <http://www.cadwiki.cz/Print.aspx?Page=Mracno-bodu>
- [17] PrimeSense Natural Interaction. PRIMESENSE [online]. 2011 [cit. 2012-06-05]. Dostupné z: <http://www.primesense.com/en/technology>
- [18] Detekce a sledování polohy hlavy ve video sekvencích s využitím zařízení Microsoft Kinect. ELECTROREVUE [online]. 3.1.2012 [cit. 2012-06-05]. Dostupné z: <http://www.elektrorevue.cz/cz/clanky/zpracovani-signalu/0/detekce-a-sledovani-polohy-hlavy-ve-video-sekvencich-s-vyuzitim-zarizeni-microsoft-kinect/>
- [19] WEBB, Jarrett a James ASHLEY. *Beginning Kinect Programming With the Microsoft Kinect Sdk*. 23.2.2012. Apress. ISBN 14-302-4104-7.
- [20] CL NUI Platform. CODELABORATORIES. [online]. [cit. 2012-06-05]. Dostupné z: <http://codelaboratories.com/kb/nui>
- [21] Introducing OpenNI. OPENNI. [online]. 2011 [cit. 2012-06-05]. Dostupné z: <http://www.openni.org>
- [22] BORENSTEIN, Greg, Andrew ODEWAHN a Brian JEPSON. *Making things see: 3D vision with Kinect, Processing, Arduino, and MakerBot*. 1st ed. [Make: Books], 2012c2012, 416 s. ISBN 14-493-0707-8.

- [23] KRAMER, Jeff, Nicolas BURRUS, Daniel HERRERA C., Florian ECHTLER a Matt PARKER. *Hacking the Kinect*. Apress, 2012, 268 s. ISBN 978-1-4302-3867-6.
- [24] Microsoft SQL Server Compact 4.0. MICROSOFT. [online]. 2012
[cit. 2012-06-05]. Dostupné z: <http://www.microsoft.com/en-us/download/details.aspx?id=17876>

SEZNAM OBRÁZKŮ

Obrázek 1: Uspořádání kosterního systému [4].....	14
Obrázek 2: Příklad snímání geometrie tváře.....	15
Obrázek 3: Základní prvky senzoru Kinect [9]	17
Obrázek 4: Schéma uspořádání hlavních funkčních prvků senzoru [10]	18
Obrázek 5: Optický nástavec od společnosti Nyko [12].....	19
Obrázek 6: Napájecí adaptér.....	20
Obrázek 7: Rozebraný senzor Kinect [13].....	20
Obrázek 8: Motorizovaná základna a elektromotor [13]	21
Obrázek 9: Barevná VGA.....	22
Obrázek 10: Infračervený	22
Obrázek 11: Monochromatický	23
Obrázek 12: Elektronické desky a ventilátor [13]	24
Obrázek 13: Popis první elektronické desky [13].....	24
Obrázek 14: Popis druhé elektronické desky [13].....	25
Obrázek 16: Princip určování vzdálenosti objektů [14]	26
Obrázek 15: Popis třetí elektronické desky [13].....	25
Obrázek 17: Ukázka aplikace Skeletal viewer	27
Obrázek 18: Příklad záznamu mračna bodů obličeje.....	27
Obrázek 19: Schéma systému pro detekci a sledování polohy hlavy [18]	28
Obrázek 20: Ukázková aplikace z prostředí CL NUI Platform [20]	30
Obrázek 21: Třívrstvá koncepce OpenNI [21]	31
Obrázek 22: Aplikace Kinect Explorer z balíku Kinect for Windows SDK 1.0	32
Obrázek 23: Grafický návrh, rozvržení aplikace	36
Obrázek 24: Zobrazení souboru XAML.....	37
Obrázek 25: Tvorba uživatelského rozhraní.....	37
Obrázek 26: Graf naměřených hodnot pro jednotlivé jointy	46
Obrázek 27: Uspořádání tabulek v databázi	47
Obrázek 28: Práce s aplikací.....	48
Obrázek 29: Záložka Grafy – prohlížení uložených osob	49
Obrázek 30: Chybné určení pozice jointu.....	52

SEZNAM TABULEK

Tabulka 1: Výsledky měření – metoda s vypočtením průměru	51
Tabulka 2: Výsledky měření – metoda nejčastější hodnoty (Objekt 1).....	53
Tabulka 3: Výsledky měření – metoda nejčastější hodnoty (Objekt 2).....	53

