



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

APLIKACE ROZHRANÍ MICROSOFT KINECT PRO PRÁCI S FOTOGRAFIEMI Z ELEKTRONOVÉHO MIKROSKOPU

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie

Autor práce: **Jiří Burýšek**
Vedoucí práce: Ing. Miloš Hernych



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří Buryšek**
Osobní číslo: **M12000113**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Aplikace rozhraní Microsoft Kinect pro práci s fotografiemi z elektronového mikroskopu**
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s hardwarem snímačů pohybu Microsoft Kinect a jeho softwarovou podporou - dostupnými API, SDK a vhodnými knihovnamí.
2. Navrhněte aplikaci, která vlastností tohoto rozhraní využije pro intuitivní vizualizaci a zpracování sad fotografií, pořizovaných elektronovým mikroskopem v různém zvětšení.
3. Tuto aplikaci zrealizujte a ve spolupráci s budoucími uživateli ověřte v reálném nasazení.

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **30–40 stran**

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

Seznam odborné literatury:

- [1] **JANA, Abhijit: Kinect for Windows SDK Programming Guide. Birmingham: Packt Pub, 2012. ISBN 18-496-9239-4.**
- [2] **BORENSTEIN, Greg.: Making things see 3D vision with Kinect, Processing, Arduino, and MakerBot. Sebastopol: O'Reilly Media, 2012. ISBN 978-144-9327-781.**
- [3] **MILES, Rob.: Learn Microsoft Kinect API. U.S.A.: Microsoft Press, 2012, xviii, 250 pages. ISBN 07-356-6396-3.**
- [4] **WEBB, Jarrett, ASHLEY, James: Beginning Kinect programming with the Microsoft Kinect SDK. New York: Apress, 2012. ISBN 14-302-4105-5.**
- [5] **KEAN, Sean, HALL, Jonathan and PERRY, Phoenix: Meet the Kinect: an introduction to programming natural user interfaces. New York: Apress, 2012, 220 p. ISBN 14-302-3888-7.**

Vedoucí bakalářské práce:

Ing. Miloš Hernych

Ústav mechatroniky a technické informatiky


Konzultant bakalářské práce:

doc. Ing. Dora Kroisová, Ph.D.

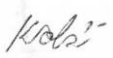
Katedra materiálu

Datum zadání bakalářské práce: **10. října 2014**

Termín odevzdání bakalářské práce: **15. května 2015**


prof. Ing. Václav Kopecký, CSc.
děkan




doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2014

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 15. 5. 2015

Podpis:



Abstrakt

Tato bakalářská práce zkoumá možnosti vývoje aplikací pro snímač pohybů těla Microsoft Kinect. Poskytuje ucelený přehled o historii a schopnostech zařízení včetně jeho zařazení do kontextu podobných periférií, ať už jako herního ovladače či jako nástroje s technickým využitím. Text se zabývá i možnostmi uplatněním pohybového ovládání ve skutečném nasazení. Další část je věnována možnostem vývoje aplikací pro Kinect, především je zaměřena na vývoj softwaru pro platformu Microsoft Windows. V této části jsou rozebrána aplikační rozhraní použitelná k programování pro dané zařízení, největší pozornost je pak věnována původnímu aplikačnímu rozhraní společnosti Microsoft. Práce se dále soustředí na návrh a realizaci skutečné aplikace pro určený snímač pohybu. Vytvářená aplikace slouží k vizualizaci a zpracování sérií fotografií pořízených elektronovým mikroskopem v různých zvětšeních. Cílem realizovaného softwaru je možnost manipulace s těmito fotografiemi pomocí intuitivní série pohybů. Práce eviduje celý průběh vývoje, zkoumá obecná řešení dávkového zpracování podobné série snímků, řeší její zpracování ať už pomocí samostatných fotografií, tak i jako série ve formátu videa. Za účelem implementace pohybového ovládání k manipulaci se snímky je zkoumáno, které pohyby jsou nejlepší z hlediska jednoduchosti a přívětivosti ovládání, zároveň však takové, které vyhovují specifickým potřebám zadaného způsobu manipulace. Závěrečná část popisuje konkrétní algoritmy, které slouží ke zlepšení celkové působivosti a vzhledu aplikace. Je zde uveden popis konečného řešení aplikace, včetně jejího vzhledu. Na základě této práce je možné získat obecnou průpravu a přehled, jakým způsobem funguje implementace funkce pohybového zařízení. Zároveň může sloužit jako motivace k dalšímu užití daného typu ovládání a její zjištění tak přispívají k rozvoji celé této oblasti.

Klíčová slova

Microsoft Kinect, ovládání pohybem, gesta, elektronový mikroskop, Kinect API

Abstract

This bachelor thesis analyses possibilities of developing an application used with body movement sensor Microsoft Kinect. It provides a review of the history of this device and its capabilities, including the background of similar peripherals, both as a game controller and a device of technical use. The study also deals with usage of such devices in a practical way. The following section analyses possibilities of development applications for Kinect, specifically in a combination with Microsoft Windows operating system. The mentioned part describes various application interfaces created for this particular device, focusing its attention on original application interface by Microsoft. The thesis moves on to describe a process of designing an actual application for the device. This application should provide a way to view and process a series of photographs captured with electron microscope in various scales. The goal of the developed software is to provide an interface used to control and transform these photographs with a set of intuitive gestures and moves. The theses reflects the whole process of development, analyses the solution of processing a vast series of images, either in a form of sole photographs or in a form of video file. To achieve the best experience in implementation of the movement control, various moves are discussed in order to seek the ideal balance between simplicity and specific needs of the given way of transformations and control. The final section describes particular set of algorithms used to improve the overall impression and layout of the application. On the basis of the results of this thesis, one can get an overall summary of the principles of movement control implementation. It may also became a source of motivation for other projects based on this particular way of control and its results provide some support for growth of this whole area of peripherals.

Keywords

Microsoft Kinect, movement control, gestures, electron microscope, Kinect API

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 12 |
| 2 | Historie pohybového ovládání | 13 |
| 2.1 | PrimeSense | 13 |
| 2.2 | Vývoj zařízení Microsoft Kinect | 15 |
| 2.3 | Technické využití pohybového ovládání | 16 |
| 3 | Microsoft Kinect | 18 |
| 3.1 | Varianty zařízení | 18 |
| 3.1.1 | Specifikace první generace | 18 |
| 3.1.2 | Rozdíly v modifikacích první generace | 19 |
| 3.1.3 | Specifikace druhé generace | 19 |
| 3.1.4 | Rozdíly v modifikacích druhé generace | 19 |
| 3.2 | Programování pro Kinect | 20 |
| 3.2.1 | Rozdíly v modifikacích druhé generace | 20 |
| 3.2.2 | Funkce API | 21 |
| 3.2.3 | Vrstva barev obrazu <i>ColorFrame</i> | 21 |
| 3.2.4 | Vrstva hloubky obrazu <i>DepthFrame</i> | 21 |
| 3.2.5 | Vrstva kostry osoby <i>SkeletonFrame</i> | 21 |
| 3.2.6 | Vrstva pro manipulaci s ovládacími prvky <i>Interaction-Stream</i> | 22 |
| 3.2.7 | Gesta SDK verze 1.7 a vyšší | 22 |
| 3.2.8 | Gesto zatnutí pěsti | 23 |
| 3.2.9 | Gesto zmáčknutí | 23 |
| 3.2.10 | Přepínání uživatelů podle jejich interakcí | 23 |
| 4 | Návrh aplikace | 24 |

| | | |
|----------|--|-----------|
| 4.1 | Ovládání aplikace | 24 |
| 4.1.1 | Škálování obrazu | 25 |
| 4.1.2 | Rotace obrazu | 25 |
| 5 | Realizace aplikace | 26 |
| 5.1 | Výběr prostředků | 26 |
| 5.1.1 | Výběr hardwaru | 26 |
| 5.1.2 | Výběr API | 26 |
| 5.1.3 | Výběr programovacích prostředků | 27 |
| 5.2 | Zpracování pohybu | 27 |
| 5.2.1 | Ovládání aplikace | 28 |
| 5.2.2 | Inicializace | 28 |
| 5.2.3 | Zajištění interagující postavy | 29 |
| 5.2.4 | Implementace SDK 1.7 k vylepšení ovládání | 29 |
| 5.2.5 | Gesto zatnutí pěsti | 30 |
| 5.2.6 | Gesto zmáčknutí | 30 |
| 5.2.7 | Přepínání uživatelů podle jejich interakcí | 30 |
| 5.3 | Zpracování sérií fotografií | 30 |
| 5.3.1 | Nahrávání fotografií ze složky | 31 |
| 5.3.2 | Prolínání | 32 |
| 5.3.3 | Průměrování pohybů | 32 |
| 5.3.4 | Nahrazení jednotlivých fotografií videem | 33 |
| 5.4 | Vzhled aplikace | 33 |
| 5.4.1 | Systém odezvy | 35 |
| 5.4.2 | Nápověda | 35 |
| 5.4.3 | Obrysy uživatelů | 36 |
| 5.5 | Požadavky systému reálného času | 36 |
| 5.6 | Program pro výrobu sekvence | 37 |

Seznam obrázků

| | | |
|---|--|----|
| 1 | První generace Kinectu pro Xbox 360 | 16 |
| 2 | Umístění zpracovávaných bodů těla | 22 |
| 3 | Ukázka snímku sejmutého elektronovým mikroskopem | 34 |
| 4 | Pohled aplikace na volbu série fotografií | 34 |

1 Úvod

Cílem práce je vytvoření komplexního řešení využití hardwarového prostředku Kinect od firmy Microsoft k prezentaci fotografií vytvořených elektronovým mikroskopem. Za účelem vytvoření této aplikace je nezbytné nastudovat snímač pohybů Kinect ve svém kontextu. Bude provedena studie historie jeho vzniku a výběr typu přístroje z nabízených variant. Kromě hardwarové stránky produktu je potřeba vytvořit si přehled o softwarových možnostech. Ze získaných informací je pak stanoveno, která knihovna představuje tu nejlepší pro programování zadaného softwarového řešení.

Daná aplikace je postavena na technických možnostech snímání a softwarového zpracování pohybů člověka, respektive pohybů jeho kostry. Prezentace sledu fotografií musí probíhat intuitivním způsobem, aby uživatel výsledného softwarového produktu snadno pochopil, jak s uživatelským prostředím pracovat. Je tedy potřeba vyprojektovat samotné uživatelské prostředí a možnosti jeho ovládání. Dále je nezbytné navrhnout způsob, jakým bude program se sérií fotografií manipulovat a předkládat je koncovému uživateli.

Konečně počítá práce s implementací navržených řešení do finální aplikace. Tato projde testy se skutečnými uživateli k ověření vytyčených cílů přívětivosti ovládání a kladného dojmu z uživatelského prostředí.

2 Historie pohybového ovládání

System pohybového ovládání se může jevit technologicky natolik náročný, že technologie Kinectu představuje jedno z prvních takových řešení. Pokusy o takové zařízení se však datují až k 60. letům minulého století [1]. V počátcích vývoje tato zařízení implementovala prostředky podobné těm, které se v dnešní době běžně označují jako „motion capture“. Zásadní rozdíl pak spočíval v možnosti nějakým způsobem interagovat s počítačem, na kterém je software provozován.

Už tehdy bylo patrné, že možnosti práce se zařízením, které musí mít uživatel na sobě, jsou značně omezené a efektivita je velice nízká. Příští řešení už tedy pracovala s myšlenkou, kdy uživatel je snímán bez jakéhokoli zařízení umístěném přímo na těle. Mezi prvními takovými projekty bylo i Pantomation, původně zaměřené na uměleckou tvorbu. Toto zařízení by se už dalo nazvat přímým předchůdcem Kinectu a vytvářelo definici pohybového ovládače. Jednalo se o zařízení, které dovedlo v reálném čase snímat osobu a její momentální pozici matematicky vyjádřit. Ve spojení s příslušným softwarem bylo tedy možné prvky tohoto softwaru ovládat.

2.1 PrimeSense

Izraelská společnost PrimeSense vznikla roku 2005 s jasným cílem vytvořit zařízení s možností pohybového ovládání, které bude natolik levné, aby si jej mohl dovolit běžný uživatel počítače. Už rok po svém založení [2] zařadila firma do portfolia svůj hlavní produkt, který byl tou dobou stále ve vývoji. Tímto produktem se stal mikročip SoC¹, který umožňoval postavit

¹System on a chip je integrovaný obvod, který v sobě kombinuje všechny nezbytné součásti k chodu daného zařízení.

zařízení svou funkcí v zásadě totožné s Kinectem. Tedy takové zařízení, jež při přidání softwarového vybavení snímalo lidskou postavu a její pohyby. S odvoláním na uvedený článek serveru Geektime, byl tento čip nabídnut všem hlavním výrobcům konzolí té doby, tedy Sony, Nintendo i Microsoftu. Z vyjmenovaných firem přijal nabídku pouze Microsoft a vytvořil tak projekt kódového označení Natal.

Microsoft postavil svůj Kinect sice na čipu dodávaném PrimeSensem, ovšem softwarové řešení vytvořil původní. PrimeSense však ke své technologii vytvořilo vlastní software, čímž ze svého systému učinil produkt, který lze nabídnout komukoli. Je možné se tedy setkat s výrobky povahou takřka identickými s Kinectem, zřejmě neúspěšnější z nich je Asus WAMI Xtion [3]. Právě zázemí velikosti Microsoftu však učinilo z Kinectu zdaleka nejvyužívanější přístroj ve své kategorii.

Důvody, proč je právě PrimeSense první společností, které se podařilo nabídnout pohybové ovládání běžnému uživateli, jsou zřejmé. Jedná o užití jiného principu, z něž je zpracován obraz postavy, než dosud užívaly konkurenční společnosti. Dosud firmy v podobných produktech aplikovali systém zvaný „Time-of-flight“. Funkci systému lze zjednodušeně přirovnat k radaru, avšak na laserovém základě. Zařízení tedy počítá dobu, za kterou se emitované fotony vrátí zpět k zařízení. PrimeSense však postavil svou technologii na jiné bázi a totiž na porovnávání obrazu vyslaného a přijímaného. Tímto postupem je zajištěn dostatečný výkon za tak nízkou cenu, za kterou může být daný čip montován do běžně dostupných periférií.

Jak bylo řečeno, Microsoft v Kinectu pracoval s vlastním softwarem. Zároveň princip fungování technologie PrimeSensu nezůstal tajemstvím, Microsoft tedy neměl důvod kontrakt s touto společností prodlužovat. Práce na

Kinect už nyní probíhají pouze v Microsoftu. Nedlouho po konci kontraktu byl PrimeSense roku 2013 odkoupen společností Apple.

2.2 Vývoj zařízení Microsoft Kinect

Ovládání herních zařízení bylo v minulosti vždy řešeno pomocí periferních ovladačů, které byly osazeny mechanickými členy typů tlačítek či kloubů. Ať už byly spojeny s herním zařízením pomocí kabelu či bezdrátově, signály z takových zařízení byly dodávány ve formě vhodné ke snadnému zpracování.

Zásadní změnou v tomto zažitém postupu bylo zavedení speciálního herního ovladače Wii firmy Nintendo. Tato firma v roce 2006 představila speciální herní ovladač s akcelerometrem, který pomocí bezdrátových signálů zasílal herní konzoli údaje o pohybu ovladače. Úspěch tohoto řešení byl mimořádný a ve své době se jednalo o nejprodávanější konzoli. Do roku 2012 se jen v Evropě prodalo 30 milionů kusů tohoto herního zařízení. Bylo jen otázkou času, kdy se i ostatní výrobci herních konzolí pokusí představit svoje zařízení určené k pohybovému ovládání. Největším konkurentem konzole Wii se z hlediska objemu prodeje stala konzole Xbox 360 firmy Microsoft.

V roce 2009 požádal Microsoft o udělení patentové ochrany pro ovladač s názvem Xbox Loop [4]. Jednalo se o elektromyografický přenosný ovladač zaznamenávající svalové biosignály, které by dokázal převést do podoby povelů k ovládání her a přístrojů.

V témže roce na veletrhu E3 představil Microsoft projekt Natal. Jednalo se o snímací systém, který již byl schopen sledovat pohyby celého těla. Později byl tento systém uveden na trh pod názvem Kinect a připojil se k existující herní konzoli Xbox 360.

Vývoj Kinectu stál desítky milionů dolarů a jeho uvedení na trh pro zařízení Xbox 360 podpořila reklama v hodnotě 500 milionů dolarů. [5] Pohybový



Obrázek 1: První generace Kinectu pro Xbox 360,

Zdroj: <https://msdn.microsoft.com/en-us/library/hh438998.aspx>

ovladač Kinect měl napodobit úspěch zařízení Nintendo Wii. Microsoft chápal reklamní kampaň jako dlouhodobou investici. Návratnost nákladů měly zajistit prodané hry. To byl také důvod, aby již v den uvedení zařízení na trh bylo k dispozici mimo přibalené hry dalších deset titulů sportovních her.

2.3 Technické využití pohybového ovládání

Technologie Kinectu byla vyvinuta přednostně pro potřeby herního průmyslu. Ukazuje se však, že možnosti pro její užití jsou daleko širší. V podstatě lze uvažovat o jejím užití při všech situacích, kdy pro ovládání technického zařízení není z jakéhokoliv důvodu možný anebo vhodný přímý kontakt s ovládacími prvky. Velmi dobrým příkladem z poslední doby je použité řešení, které zvolila televize Nova v souvislosti se spuštěním nového zpravodajského studia v březnu 2014.

Při předpovědi počasí jsou grafika a mapy řízena gesty moderátorů za použití pohybového senzoru Kinect. Touto aplikací se podařilo dosáhnout skutečnosti, že moderátor není k divákům otočen zády při ovládání toku informací, map a obrazů, jako tomu bylo doposud při ovládání plazmové obrazovky. Tuto měl moderátor v minulosti za sebou. Nyní vše ovládá bez-

dotykově, čelem k divákovi. Takové řešení použité TV Nova je unikátní v celosvětovém měřítku a ve zpravodajství bylo použito vůbec poprvé.

V liberecké iQLANDi, která přibližuje svět vědy a techniky všem věkovým skupinám připravila Technická univerzita v Liberci expozici týkající se výroby a použití nanovláken, fluorescenčních a teplocitlivých materiálů a především expozici, kde bude možné pouhým pohybem těla pronikat do hmoty, prohlížet si její strukturu až na úrovni zobrazení elektronového mikroskopu. Právě v této expozici najde uplatnění softwarové řešení ovládané pohybem těla.

3 Microsoft Kinect

3.1 Varianty zařízení

Snímač pohybu Kinect firmy Microsoft dospěl v době tvorby práce do druhé generace. V obou generacích byl produkt na trh uvolněn ve dvou modifikacích:

1. Kinect pro herní konzoli Xbox (360/One v závislosti na generaci)
2. Kinect pro Windows

3.1.1 Specifikace první generace

V těle zařízení lze najít dvě kamery a emitor infračerveného záření. Jedna z kamer obsahuje snímač typu CMOS o rozlišení VGA, tedy 640×480 pixelů. Barevná hloubka této kamery je 32 bitů formátu RGB, snímat lze frekvencí 30 snímků za vteřinu. Druhá kamera snímá monochromatický, 16 bitový obraz infračerveného záření o rozlišení 320×240 pixelů při kmitočtu taktéž 30 snímků za vteřinu. Obě kamery pak disponují systémem automatického ostření.

Kinect dále disponuje čtyřmi mikrofony, které mají v konečném důsledku zajistit systém zpracování mluveného slova. Zároveň jsou v takové sestavě, aby dokázaly určit zdroj zvuku v prostoru a eliminovat okolní šum. Mikrofony jsou rozmístěny v rozložení tři na pravé straně na jeden na levé straně. Disponují shodnými parametry rozlišení 16 bitů o taktu 16 kHz.

Tělo zařízení stojí na podstavci vybaveném motoriky, které umožňují naklánění i natáčení přístroje. Elektrická spotřeba Kinectu se pohybuje na úrovni 12 W. Z tohoto důvodu má zařízení vlastní aktivní chlazení. Co se

týká využití systémových prostředků, při zapojení Kinectu ve verzi pro Xbox 360 k této konzoli, zařízení zabere méně než 10% celkového výkonu.

Přístroj je schopen zaregistrovat nejvýše šest uživatelů, z toho dva dokáže monitorovat za účelem ovládní. U takového uživatele je registrováno 20 tělesných bodů, ideální pozice před senzorem se pak pohybuje od 1,2 do 3,5 metrů.[6]

3.1.2 Rozdíly v modifikacích první generace

Protože Microsoft neuvádí specifikace zařízení Kinect, lze tato zařízení porovnat pouze z naměřených hodnot. Jediný parametr, kterým Kinect pro Xbox 360 nedisponuje, je mód pro snímání blízké osoby. Senzor je tedy schopen zachytit pohyby člověka už několik desítek centimetrů před sebou.

3.1.3 Specifikace druhé generace

Barevná kamera nyní pracuje s rozlišením 1080p, zatímco monochromatická s rozlišením 512×424 pixelů. Největší změny ale doznala technologie zpracování. Kinect druhé verze využívá systému „Time-of-flight“, kterému se detailněji věnuje část 2.1. Je paradoxní, že až příchod prvního Kinectu postaveného na jiném principu, pomohlo této technologii dospět do takové fáze, kdy mohla původní technologii nahradit. Dále byl navýšen počet osob, jejichž pohyby lze monitorovat, konkrétně na šest.[7]

3.1.4 Rozdíly v modifikacích druhé generace

Stejně jako v první generaci, i v druhé je Kinect pro Windows lépe uzpůsoben pro snímání osob v kratší vzdálenosti. V tomto případě se však již zřejmě jedná o firmwarovou úpravu. Microsoft ukončil prodej Kinectu pro Windows

v dubnu 2015 [8]. Kinect už je tedy pro všechny platformy stejný, pro jeho užití v kombinaci s Windows je požadováno speciálního adaptéru.

3.2 Programování pro Kinect

Přirozeným krokem, kde začít studovat základy práce s aplikačním rozhraním Kinectu byla k tomu určená webová stránka distributora. Předtím bylo ovšem nutné prověřit, eventualitu, zda existují jiné knihovny, případně jaké jsou možnosti programování pro Kinect při jejich užití. Tyto knihovny mohli posloužit účelům práce lépe.

Při zahájení projektu existovaly *otevřené* knihovny [10]. Tyto knihovny postavila komunita na bázi dostupných ovladačů [11] dodavatele technologie Kinectu, tedy firmy PrimeSense, viz část 2.1. Kinect API² umožňuje vyvářet program v nativním kódu C++, nebo jako projekt WPF v jazycích C#, Visual Basic a XAML.

3.2.1 Rozdíly v modifikacích druhé generace

Z učebních materiálů byly použity především tyto dva zdroje: Prvním z nich je shromaždiště vývojářských informací Microsoftu, Microsoft Development Network (MSDN), konkrétně pak samozřejmě části [12] přímo pojednávající o programování Kinectu pro Windows, Druhým zdrojem se pak staly samotné ukázkové aplikace obsažené ve vývojářském balíku Developer Toolkit, ke stažení na webových stránkách Kinectu pro Windows [9]. Většina znalostí k tvorbě aplikace byla tedy nabyta procesem reverzního inženýrství.

²Application Programming Interface, tedy aplikační rozhraní je sbírka knihovny, které lze využít při programování.

3.2.2 Funkce API

Fungováním senzoru Kinect dostává programátor posléze k dispozici celkem čtyři unikátní vrstvy zpracovaného obrazu. Jedná se o tyto vrstvy:

- **3.2.3 Vrstva barev obrazu *ColorFrame***

Vrstva *ColorFrame* činí totéž, co klasická kamera, tedy poskytuje obrazový záznam dění probíhajícího před zařízením.

- **3.2.4 Vrstva hloubky obrazu *DepthFrame***

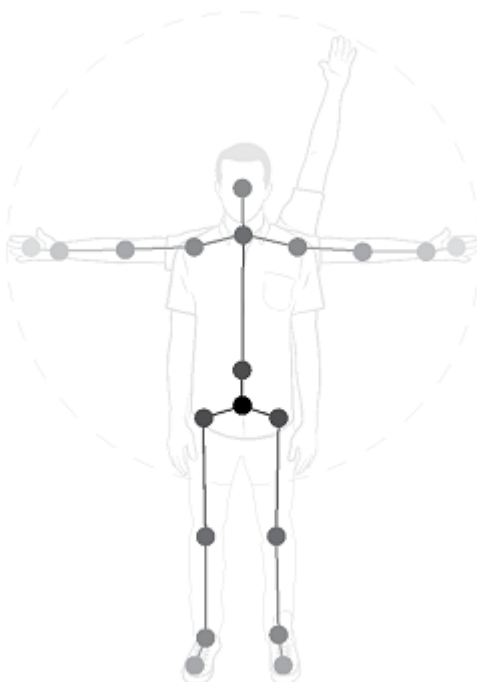
Díky této vrstvě lze získat informace o struktuře a rozmístění pohybů v prostoru před senzorem. Toho je užito pro oddělení snímaných osob od prostředí. Zároveň se nabízí široké, dosud málo využívané možnosti sledování dalších objektů, mimo lidí.

- **3.2.5 Vrstva kostry osoby *SkeletonFrame***

Tato vrstva představuje zřejmě tu nejdůležitější ze všech funkcí Kinectu. Poskytuje již zpracovaná data bez přímé potřeby další úpravy. Díky této vrstvě získáme data o umístění, pohybu a náklonu každé části těla a to ve vektorových souřadnicích trojrozměrného prostoru.

- **3.2.6** Vrstva pro manipulaci s ovládacími prvky *InteractionStream*

InteractionStream stojí na trochu jiné úrovni než předcházející tři vrstvy. Byla přidána v SDK³ verze 1.7 a je založena na stávajících možnostech. Její hlavní výhodou je přidání gest rukou.



Obrázek 2: Umístění zpracovávaných bodů těla,

Zdroj: <https://msdn.microsoft.com/en-us/library/hh973073.aspx>

3.2.7 Gesta SDK verze 1.7 a vyšší

Vývojářský balík SDK verze 1.7 přinesl novinky, z nichž mnohé nejsou součástí API, a jejich užití funguje přes krkolonné připojení poskytovaného

³Software development kit je balík nástrojů určený pro vývojáře k vývoji softwaru.

kódu. Jednou z nejpodstatnějších změn je nová vrstva obrazu „Interaction-Stream“, která se určitým způsobem snaží kompenzovat problematiku neustálého přísunu dat. Obsahuje sadu událostí, které jsou spojeny s manuálními gesty a přibližuje Kinect blíže do pozice běžné periférie.

3.2.8 Gesto zatnutí pěsti

Zařízení je nyní schopno rozpoznat, když uživatel zatne ruku v pěst. Tato funkce má zatím značné rezervy a pohybuje se na prahu použitelnosti, neboť míra chybovosti je vysoká.

3.2.9 Gesto zmáčknutí

Dosud aplikace pro Kinect řešily úmysl označení prvku prostým podržením ruky nad vybraným prvkem. Gesto zmáčknutí vychází z představy fyzického zmáčknutí, při kterém uživatel nejprve položku zvolí pozičně a poté ruku natáhne ještě blíže k senzoru. To je pro uživatele daleko přirozenější, zároveň je však potřeba uživatele předem informovat, jak by měl postupovat.

3.2.10 Přepínání uživatelů podle jejich interakcí

Díky implementaci nové vývojářské sady je možné elegantněji vyřešit přepínání uživatelů. Zmíněná vrstva „InteractionStream“ dokáže zachytit úmysl uživatele s aplikací manipulovat pomocí jejich pohybů.

4 Návrh aplikace

Pro účely aplikace byly pořízeny modelové fotografie různých objektů pomocí elektronového mikroskopu. Základní požadavek aplikace vyžaduje možnost prohlížet si tyto fotografie v postupné sérii, k vytvoření pocitu kontinuálního zvětšování a zmenšování objektu. Tento hlavní prvek aplikace je třeba zasadit do uživatelského prostředí, které obsahuje celkový pohled na možné objekty, které si lze prohlížet. Po volbě objektu musí následovat pohled na samotnou sérii fotografií. Zároveň je třeba poskytnout uživateli zpětnou vazbu a nápovědu k celé aplikaci.

Velký důraz je kladen na snadné a pochopitelné pohyby těla, jejichž pomocí lze například realizovat průchod od bližšího bodu série až k celkovému pohledu na fotografii. S tímto úkolem souvisí hledání optimální cesty ke zpracování a použití fotografie pro její optimální interakci s užitým softwarovým řešením.

První návrh aplikace tedy zobrazuje přírodní prostředí, ve kterém jsou umístěny ilustrace živočichů a rostlin k vyvolání celistvého dojmu. Piktoqramy umístěné na okraji obrazovky by měly sloužit jako nápověda, jakými pohyby lze aplikaci ovládat. K vyvolání patřičné série je třeba najet rukou nad daný objekt a tím jej zvolit. Příští obrazovka zobrazuje jednu fotografii ze série. Pomocí pohybů rukou se lze pohybovat v sérii oběma směry a zároveň je možné snímkem otáčet.

4.1 Ovládání aplikace

V současnosti není běžný uživatel zvyklý na ovládání jinými perifériemi, než je myš a klávesnice. Z hlediska designu aplikace představovala největší výzvu právě nutnost naučit uživatele pracovat s Kinectem v nejkratším možném

čase při minimálním množství chyb. Bylo tedy potřeba najít pokud možno ideální spouštěcí mechanismus k vyvolání interakce, který by suploval například kliknutí myší. V zásadě lze odhadnout, které pohyby si člověk většinou spojí s různými proveditelnými akcemi.

4.1.1 Škálování obrazu

Základní požadavek ve smyslu ovládní aplikace tkví v uživatelské možnosti fotografii libovolně zvětšovat a zmenšovat. Právě zde zjevně funguje intuice, kdy uživatel upravuje velikost pomocí zvětšování a zmenšování vzdálenosti mezi rukama.

4.1.2 Rotace obrazu

V návaznosti na manipulace s velikostí přirozeně následuje otáčení fotografie. I zde lze spoléhat na uživatelskou spontaneitu. Pro dosažení tohoto efektu uživatel zřejmě bude pohybovat rukama v opačném směru ve vertikální rovině.

5 Realizace aplikace

5.1 Výběr prostředků

5.1.1 Výběr hardwaru

V počátcích tvorby této práce byla na trhu dostupná pouze první generace snímače pohybu Kinect. Rozhodovací proces tedy pracoval pouze s ní. Na výběr tedy byly pouze dvě varianty, tedy Kinect pro Windows a Kinect pro Xbox 360⁴.

Logické důvody pro volbu řešení zadání práce pomocí Kinectu pro Windows jsou především přímý přístup k celému průběhu ladění programu bez nutnosti testování na konzoli a teoreticky neomezenému výpočetnímu výkonu počítače, ke kterému je program připojen. Naproti tomu ke kvalitnímu ladění aplikací pro konzoli Xbox 360 je nezbytné mít tento produkt ve vývojářské typizaci, přičemž její výkon je pevně daný.

Cena je ale pádným argumentem pro volbu Kinectu pro Xbox 360. Komplet Xboxu 360 s Kinectem měl při spuštění prodeje produktu jen o málo vyšší peněžní hodnotu než samotný Kinect pro Windows.

Vzhledem k tomu, že Kinect pro Xbox 360 byl v době realizace práce snadněji dostupný, zdálo se řešení užití Kinectu pro Xbox 360 logickým vyústěním rozhodovacího procesu, jaký hardware užít pro vypracování zadání bakalářské práce.

5.1.2 Výběr API

Zde bylo nutné rozhodnout, zda využít oficiální API Microsoftu, či se spolehnout na otevřené řešení. Kvalita otevřených prostředků nebyla bohužel v

⁴První generace umožňuje užít Kinect pro Xbox 360 ve spolupráci s Windows pouze pro účely testování aplikací.

době hledání správného řešení na dostatečné úrovni, aby mohla nabídnout takovou eleganci tvorby, jako oficiální knihovny, a jejich smysl spočíval spíše na účelnosti otevřeného kódu, než na lepším řešení. Naproti tomu původní API firmy Microsoft nabízelo nesrovnatelně lepší a především snazší získávání dat ze zařízení. S přihlédnutím k tomuto faktu byla konečná aplikace postavena na oficiálních uzavřených knihovnách firmy Microsoft.

5.1.3 Výběr programovacích prostředků

Požadovaná aplikace je založená na práci s grafikou. Kód projektů WPF je postaven na rozhraní Microsoft DirectX, čímž je zajištěna plná podpora grafické akcelerace výsledného projektu. Kromě toho nabízí WPF skvělou grafickou podporu tvorby uživatelského rozhraní v grafickém režimu, a to díky prostředí XAML a jeho *Živému náhledu*. S přihlédnutím k těmto důvodům se užití nativního C++ jevílo jako vynakládání zbytečných prostředků k dosažení stejných výsledků, jako při užití WPF. Z popisovaných důvodů bylo rozhodnuto postavit aplikaci jako projekt WPF.

5.2 Zpracování pohybu

Programování pro Kinect je specifické neustálým přísunem dat. Programy tvořené pro užití s běžnými perifériemi využívají události k iniciaci patřičné odpovědi. Tento postup nelze za použití Kinectu tradičním způsobem aplikovat. Metoda, která data zpracovává, tedy delegát události, se neustále cyklicky opakuje. Tento proces však v celkovém odvětví informačních technologií není zdaleka ojedinělý, například počítače typu PLC vykazují obdobné chování. Lze tedy využít i postupů určených pro tato zařízení. Zjednodušeně řečeno je nezbytné uplatnit vyšší samostatnou kontrolu větvení průběhu algoritmu pomocí přidaného množství přepínačů.

5.2.1 Ovládání aplikace

V první verzi byla jako vrstva zajišťující pohyb zvolena „SkeletonFrame“. Za pomoci dat takto získaných bylo naprogramováno ovládání základního pohledu aplikace. Modelový obraz bylo tedy možné zvětšovat, zmenšovat a rotovat přesně v intencích návrhu.

5.2.2 Inicializace

Každé interakci musí nezbytně předcházet proces, při němž uživatel dá najevo svůj úmysl s programem pracovat. Kinect snímá postavy před sebou bez toho, aby jakkoli rozlišoval tento úmysl a informace předává dál. V tomto bodě bylo načrtnuto několik plánů a některé byly zpracovány i do fáze implementace, aby se nejvhodnější přístup ukázal pomocí testování.

První opatření implementované za tímto účelem spočívalo v možnosti interakce pro ruce zvednuté nad úroveň ramen. V empirických testech tento systém fungoval dobře. Problémy se nicméně objevily při úmyslu rotace obrazu, který je počítán z rozdílu pozice rukou ve vertikální rovině. V určitém bodě otáčení se uživatel nutně dostal do pozice, při které jedna z rukou opusťla úroveň nad výškou ramen. To ale nebyl jediný problém: Pro modelové uživatele byla tato pozice velice málo intuitivní, chtělo by se říci, že byla až obtěžující. Uživatelé často ruce v průběhu interakce snižovali, následkem čehož aplikace přestávala reagovat, aniž by si uvědomili důvod výpadku. Od tohoto způsobu bylo v konečné aplikaci upuštěno.

Další zvažovanou možností bylo podmínění ovládání pro člověka stojícího na určitém bodě. Po delší úvaze však bylo shledáno, že tento způsob sice řeší problém identifikace aktivního uživatele, nikoli jeho úmysl s fotografií, potažmo aplikací jako takovou manipulovat. Tato možnost se tedy nedostala ani do fáze implementace.

Poslední zvažovaná varianta v prvním plánu si vypůjčila myšlenku z implementace Kinectu v prostředí Xboxu. Zde probíhá inicializace pomocí zamávání před senzorem. Tento způsob ovšem ukazuje svůj problém už v prostředí Xboxu. Kinect má totiž stále rezervy v rozeznávání částí těla v zákrytu. Pokud tedy uživatel mává na snímač s rukou umístěnou před tělem, velice často se objeví problém tuto sérii pohybů kontinuálně zaznamenat a tento způsob ověření tak selhává. Problém lze eliminovat při provádění pohybu paží mimo tělo. Pro snadné a intuitivní ovládání aplikace je však tento způsob při tak vysoké možnosti selhání, jak bylo později patrné z implementace, nepoužitelný.

Do okamžiku implementace SDK vyšší verze pracovala aplikace s první popsanou formou inicializace, tedy zvednutí rukou nad úroveň ramen.

5.2.3 Zajištění interagující postavy

Užitý Kinectu rozlišuje až šest osob najednou. Pro účely této aplikace je sledování více než jedné osoby nejen redundantní, ale i nežádoucí. V implicitní konfiguraci dává Kinect eventuální možnost ovládat aplikaci všem osobám, které před snímačem stojí. Toto bylo třeba omezit. V prvním prototypu tak Kinect zachytil osobu, která jediná mohla aplikací ovládat a možnost ovládání nebyla předána do té doby, než dotyčná osoba opustila zorný úhel snímače. Zde tedy vznikl velký prostor pro vylepšení postupu.

5.2.4 Implementace SDK 1.7 k vylepšení ovládání

Po důkladném testování implementovaných reakcí a studování gest v SDK byla postavena aplikace s konečnými možnostmi ovládání. Dosud užívaná vrstva „SkeletonFrame“ byla nahrazena novou vrstvou „InteractionStream“. Dále byla implementována probíraná gesta rukou.

5.2.5 Gesto zatnutí pěsti

Toto gesto představuje pro aplikaci zásadní průlom z hlediska řešení inicializace. Dosud užívaná inicializace pomocí zvednutí rukou nad úroveň ramen byla odstraněna a plně nahrazena podporou gesta stisknutí ruky v pěst. Uživatel tedy může s úmyslem aplikaci ovládat pohybovat rukama kdekoli v zorném poli senzoru, avšak až po stisknutí ruky získá schopnost manipulace. Konkrétně po stisknutí jedné ruky je aktivována rotace fotografií a po stisknutí obou rukou je aktivováno přibližování či oddalování pohledu.

Aplikace v základním pohledu zobrazuje plochu s rozesetými tlačítky. Pro rolování této plochy je opět užito gesta stisknutí ruky.

5.2.6 Gesto zmáčknutí

V konečné aplikaci bylo nezbytné vyřešit volení přechodových tlačítek. Zde bylo s úspěchem využito gesta zmáčknutí tlačítka.

5.2.7 Přepínání uživatelů podle jejich interakcí

Díky implementaci nové vývojářské sady je možné elegantněji vyřešit přepínání uživatelů. Zmíněná vrstva „InteractionStream“ dokáže zachytit úmysl uživatele s aplikací manipulovat analýzou jejich pohybů. Pomocí tohoto je v konečné aplikaci uživateli dovolen přístup k interakci, pokud s aplikací v dané chvíli zrovna nepracuje někdo jiný. Zároveň je do určité míry umožněn přístup k interakci dvěma uživatelům najednou.

5.3 Zpracování sérií fotografií

Po vyřešení problematiky ovládání aplikace, bylo třeba propojit patřičné metody se skutečnou manipulací s obrazem. Rotace obrazu neobnáší nic jiného

než aplikaci známého postupu transformace obrazu na základě programovacích prostředků WPF. Avšak další, zdánlivě triviální úkol přechodu mezi fotografiemi nakonec představoval velkou výzvu. Toto bylo způsobeno především důrazem na estetickou stránku této části a přinášelo s sebou nesnáze, kdy je třeba nejdříve navržený algoritmus implementovat a teprve poté lze posoudit, zda takový algoritmus působí na uživatele přívětivým dojmem. Naprogramováno tedy bylo postupně několik možností, jak fotografie zpracovat, i algoritmy, jak učinit jejich procházení ještě esteticky příjemnějším zážitkem.

5.3.1 Nahrávání fotografií ze složky

Prvotní řešení průchodu série fotografií bylo postaveno na algoritmu, který při své inicializaci načte do paměti veškeré fotografie z dané složky (z důvodu minimalizace času nutného pro jejich změnu) a postupně měnil zdroj obrázku, který je zatím přítomen jen jako prostor pro zobrazení aktuálního snímku. Libovolná manipulace s takovým obrázkem byla proto triviální, neboť se pouze upravovaly jeho parametry.

Obrázek tedy umožnil vlastní zvětšení, respektive zmenšení, přičemž při dosažení hraničních hodnot byl snímek zaměněn za další v řadě. V tomto přístupu se již objevil první problém a totiž v prioritním zpracování určitých úkonů v projektu WPF na vrub jiných. Při rychlých pohybech a tedy i rychlé potřebě zaměnit několik fotografií za sebou byl nejdříve obsluhován tento požadavek, zatímco úkon změny velikosti obrázku byl upozaděn.

O výměnu aktuální fotografie se starala třída, která rozpočítala počet nahraných fotografií ze složky do rozpětí paží uživatele. Pakliže tedy uživatel dosáhl jedné z hraničních hodnot, které už byly exaktně rozpočítané, fotografie se zaměnila za příští snímek v pořadí.

Postup samotné záměny pracoval bezchybně, ale výsledný dojem působil rušivě: Obrázků nebyl dostatečný počet pro vyvolání pocitu plynulé animace a každý přechod byl vnímán jako sledování nového snímku bez návaznosti na předcházející. Na výsledném dojmu se ještě podepsal fakt, že série fotografií nezachycovala zvětšování objektu směrem k jeho středu. Jedním z prvních kroků, jak tento postup vylepšit, bylo nasnímání nových sérií fotografií. Metodika snímání se změnila a zajišťovala plynulý průchod, při kterém byl obraz předcházejícího snímku vždy umístěn ve středu současného.

5.3.2 Prolínání

Jako řešení nízké návaznosti bylo zvoleno přidání prostupné prolínačky mezi dva obrázky při jejich záměně. Technické řešení spočívalo v překopírování obrázku do jeho identické kopie, včetně jeho parametrů. Zatímco se zdrojová fotografie původního obrázku změnila, jeho kopie se postupně stávala plně průhlednou, čímž bylo dosaženo kýženého efektu.

5.3.3 Průměrování pohybů

K dalšímu vylepšení výsledného vizuálního dojmu přispělo i průměrování hodnot, které z Kinectu přicházejí. Senzor ještě není na tak vysoké úrovni, aby plynule zaznamenal rychlé pohyby, proto program vždy zpracovává poslední daný počet hodnot určených k manipulaci s fotografií a obraz upravuje až na základě jejich průměru. Co se týká užitého algoritmu, pracuje tento prvek na bázi klasické FIFO⁵ fronty. Uvedení tohoto kroku vizuální optimalizace vedlo k zásadnímu vylepšení působivosti prezentace.

⁵First In, First Out je metoda, která označuje prioritu zpracování. Čím dříve je prvek do fronty vložen, tím dříve je obslužen.

5.3.4 Nahrazení jednotlivých fotografií videem

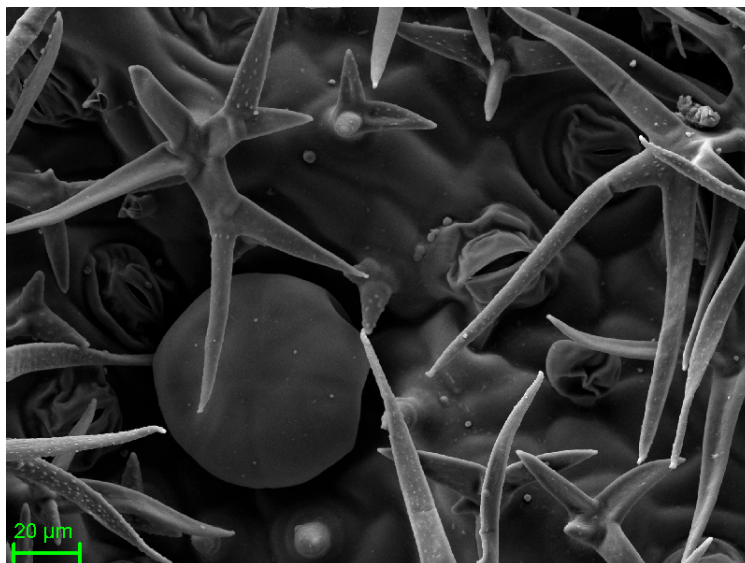
Přes veškeré snahy optimalizovat kód tak, aby dal výsledný produkt pocit plynulé animované sekvence, byl výsledný dojem stále rušivý. Bylo tedy upuštěno od přímého nahrávání jednotlivých obrázků ze složky i od animace prolínání.

Nové řešení vyžadovalo zpracování fotografií jako prezentace formou videa. Pro testovací účely bylo užito různých externích programů k tvorbě takových videí s proměnlivými výsledky. Pro příklad je možné uvést, že renderování podobného videa z 50 fotografií dokázalo generovat asi 150 snímků bez užití jakýchkoli přechodů a podobných prostředků. Z tohoto bylo patrné, že toto řešení vydá na dostatečný počet snímků pro plynulou animaci bez dalších pomocných algoritmů.

Nyní bylo potřeba přepracovat aplikaci, aby těžila jednotlivé snímky z dodaného videa. Kvůli výkonnostním nárokům aplikace nahraje video a z něj vytvoří jednotlivé snímky, místo aby video zanechávala celistvé a uživatel jej v principu jen přehrával. Za účelem práce s videem bylo užito frameworku AForge.NET [13], jehož knihovna pro práci s videem přesně splňuje požadavky této aplikace. Hlavní přínos popisovaného přístupu s použitím videa ovšem tkví právě mimo kód samotný. Při zpracování běžnými nástroji pro tvorbu videa a prezentací z fotografií dojde k nezbytné degradaci, ale i jistému prolnutí fotografií a zmíněnému zvýšení celkového počtu snímků.

5.4 Vzhled aplikace

Finální vzhled aplikace kopíruje stanovené zadání bez výraznějších změn. V průběhu testování doznal design jen několika modifikací pro lepší ergonomii.



Obrázek 3: Ukázka snímku sejmutého elektronovým mikroskopem

Při zadávání aplikace nebyla naplánovaná odezva uživatele na akce, které přímo nesouvisí s požadovaným výstupem. Hned úvodní plocha tedy obsahuje velkoformátovou fotografii přírodního prostředí, ve kterém jsou umístěny fotografie objektů fauny a flóry, jež slouží jako tlačítka. Podkladová fotografie i s umístěnými objekty byla v konečné aplikaci upravena pro podporu posouvání, čímž přidává na celkové interaktivitě produktu.

5.4.1 Systém odezvy

Od počátku vývoje bylo z testování patrné, že bude v budoucnu třeba integrovat systém kurzoru či podobného vizuálního prvku. V tomto směru byla opět velkým přínosem zmíněná vývojářská sada verze 1.7, která pomáhá vývojářům s integrací kurzoru, reagujícímu na „hlavní ruku“, tedy ruku, která byla pozdvihnutá k interakci jako první. Zobrazování kurzoru pro druhou ruku ale podporováno není. Z právního a technického hlediska však byla modifikace kódu vývojářského balíku problematická, proto bylo rozhodnuto postupovat



Obrázek 4: Pohled aplikace na volbu série fotografií

jinak. V tomto smyslu bylo nezbytné vytvořit vlastní systém vykreslování. Aplikace tedy využívá dvou nezávislých systémů vykreslování kurzoru, přičemž bylo důležité, aby tento rozdíl nebyl patrný.

Díky možnostem vrstvy „InteractionStream“ byla naprogramována stejná funkčnost pro oba kurzory, z hlediska kódu se ovšem rozcházejí, neboť mnohé události této vrstvy upravují pouze parametry kurzoru ruky určené jako primární. Události pro obě ruce tedy bylo třeba řešit separátně.

Kurzory zároveň fungují interaktivně a mění svůj vzhled na základě použitých gest. Například tedy gesto zatnutí ruky v pěst změni kurzor do patřičné vizuální podoby.

Po výběru objektu na hlavní obrazovce je vyvolána obrazovka prohlížeče fotografií. Zde prezentované snímky pořízené elektronovým mikroskopem pochopitelně odpovídají vybranému objektu. Neopomenutelnou součástí této obrazovky je pak samozřejmě tlačítko, jímž se lze vrátit na předcházející přehled objektů.

5.4.2 Náповěda

Z projednávaného tématu o intuici pohybového ovládání lze snadno vyčíst, že bylo potřeba představit i systém nápovědy, který by uživatele informoval, jaké pohyby může použít. Každá obrazovka tak obsahuje sérii piktogramů, jež vyjadřují pohyby aplikovatelné na té které obrazovce. Piktogramy záměrně nezobrazují reakci systémů na daná gesta, což má za cíl podnítit v uživateli zvědavost a chuť experimentovat.

5.4.3 Obrysy uživatelů

Už v prvních hrách pro Kinect na konzoli Xbox 360 se objevoval prvek, který se později psal nepsaným pravidlem, kterak uživatele informovat, že je senzorem snímán. Tímto prvkem bylo zobrazení siluety uživatele, potažmo uživatelů. „InteractionStream“ tento prvek přinesl oficiálně a proto byl s výhodou užít i v této aplikaci. Prvek navíc graficky zvýrazňuje uživatele, který právě aplikaci ovládá, čímž řeší odezvu přepínání uživatelů z jejich pohledu.

5.5 Požadavky systému reálného času

V důsledku optimalizací animací a celkové náročnosti aplikace vyvstaly otázky latence ovládání. Přesná oficiální data odezvy Kinectu bohužel neexistují, vznikly ovšem neoficiální studie, které měří výkon senzoru. Podle studie publikované Livingstonem [14] je průměrná doba prodlevy 106 milisekund. Tato doba se dále mění, podle počtu snímaných osob. Kupříkladu při zátěži tří osob, se průměrná doba prodlevy více než zdvojnásobila, konkrétně na 234 milisekund. S odvoláním na tato data lze konstatovat, že uživatel tuto prodlevu pocítí velice zřetelně už před implementací vlastní logiky aplikace a senzor tedy nedává příliš prostoru pro další zdržení ze strany aplikace. Tento

fakt je problematický především kvůli implementovanému algoritmu průměrování. V základním nastavení je kapacita fronty nastavena na 4 hodnoty. Prezentace se tak pohybuje na hraně, kdy lze subjektivně ještě pokládat systém za funkční v reálném čase. Bohužel tento fakt není možné nějakým způsobem z pozice programátora kompenzovat.

5.6 Program pro výrobu sekvence

Jako konečné řešení vstupních obrazových dat do aplikace byly vybrány sekvence ve formě videa. Tvorba takového videa dosud spočívala na řešení externích programů, což nebylo z hlediska praktického nasazení dobré řešení. Mimo hlavní aplikaci byla tedy vytvořena další, určená především pro správce celého systému.

Aplikace v nejjednodušším nastavení generuje sérii fotografií do souboru videa, které je přímo použitelné v hlavním programu. Situace je komplikovanější, pokud jsou k dispozici jen malé sady fotografií, při jejichž užití by ani převedení do videa neznamenal dostatečné navýšení počtu snímků, aby vydalo na plynulý dojem animace. Pro takové případy aplikace implementuje algoritmus, kdy každou jednu fotografii zvětšuje do chvíle, než na její místo může dosadit fotografii následující.

Pro zajištění hladkého přechodu je užit algoritmus prolínání popsany v jedné z předcházejících částí. Tento algoritmus však pracuje pouze se statickými obrazy. Bylo jej tedy třeba upravit, aby prolínání obrazů probíhalo, zatímco se obě fotografie zvětšují, každá ovšem s jiným koeficientem zvětšení.

6 Závěr

Vytvořením aplikace pro určený hardware, která funkčně vyhovuje požadovanému úkolu, se podařilo splnit zadání práce. V průběhu práce na aplikaci se ukázalo, do jak vysoké míry je aplikace za užití podobného způsobu ovládání závislá na výsledném dojmu na straně uživatele. Mnohé z původně uvažovaných algoritmů, jejichž užití při ovládání běžnými periferiemi by bylo bezproblémové, bylo třeba nahradit jinými, které lépe vyhovují zadanému způsobu ovládání. V této skutečnosti také spočívá sekundární přínos aplikace, která může podobným projektům v budoucnosti usnadnit hledání nejlepšího řešení.

Zároveň se ukázalo, že uvedení přelomové periferie nelze považovat i za revoluci pro koncového uživatele. Nejdříve je třeba vyzkoušet možnosti, kterými lze takové zařízení využít, než se najde jeho skutečné uplatnění. V tomto směru práce splňuje účel. Microsoft i další výrobci pracují na nových verzích pohybových zařízení a s nimi je potřeba vyvíjet i stále lepší a efektivnější software.

Seznam použité literatury

- [1] *Prosthetic Knowledge Picks: Kinect Genealogy - A Brief History of Gestural Interfaces* [online]. [cit. 2015-03-10]. Dostupné z:
<http://rhizome.org/editorial/2012/oct/23/prosthetic-knowledge-picks-kinect-genealogy-brief->
- [2] *PrimeSense, the day after Kinect* [online]. [cit. 2015-03-05]. Dostupné z:
<http://www.geektime.com/2013/05/27/primesense-the-day-after-kinect>
- [3] *WAVI Xtion, Intuitive living room experience* [online]. [cit. 2014-09-05]. Dostupné z:
http://event.asus.com/wavi/product/WAVI_Xtion.aspx
- [4] *Microsoft secures patent for wearable controller* [online]. [cit. 2015-01-16]. Dostupné z:
<http://www.eurogamer.net/articles/2012-05-02-microsoft-secures-patent-for-wearable-controller>
- [5] *Microsoft dá půl miliardy dolarů na marketing pohybového ovládání Kinect* [online]. [cit. 2014-09-06]. Dostupné z:
<http://tech.ihned.cz/c1-47298300-microsoft-da-pul-miliardy-dolaru-na-marketing-pohyboveho-ovladani-kinect>
- [6] *Microsoft Kinect: nová éra, tělo jako ovladač* [online]. [cit. 2015-04-25]. Dostupné z:
<http://www.zive.cz/clanky/microsoft-kinect-nova-era-telo-jako-ovladac/sc-3-a-154556>

- [7] *Microsoft's new Kinect is official: larger field of view, HD camera, wake with voice* [online]. [cit. 2015-02-20]. Dostupné z:
<http://www.engadget.com/2013/05/21/microsofts-new-kinect-is-official>
- [8] *Microsoft no longer making Kinect for Windows sensors* [online]. [cit. 2015-04-26]. Dostupné z:
<http://www.zdnet.com/article/microsoft-no-longer-making-kinect-for-windows-sensors>
- [9] *Developing with Kinect* [online]. [cit. 2015-03-10]. Dostupné z:
<http://www.microsoft.com/en-us/kinectforwindows/develop>
- [10] *OpenKinect* [online]. [cit. 2015-03-11]. Dostupné z:
<http://openkinect.org>
- [11] *PrimeSense releases open source drivers, middleware that work with Kinect* [online]. [cit. 2014-12-27]. Dostupné z:
<http://www.joystiq.com/2010/12/10/primesense-releases-open-source-drivers-middleware-for-kinect>
- [12] *Kinect for Windows SDK* [online]. [cit. 2015-04-30]. Dostupné z:
<http://msdn.microsoft.com/en-us/library/hh855347.aspx>
- [13] *AForge.NET Framework* [online]. [cit. 2015-01-03]. Dostupné z:
<http://www.aforgenet.com/framework>
- [14] LIVINGSTON, M. A., J. SEBASTIAN, Z. AI a J. DECKER. 2012. *2012 IEEE Virtual Reality*. Piscataway: IEEE. ISBN 978-146-7312-479.