



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

KVANTIFIKACE PORUCH PROPRIOCEPČNÍHO APARÁTU POMOCÍ VIRTUÁLNÍ REALITY

QUANTIFICATION OF PROPRIOCEPTIVE DISORDERS USING VIRTUAL REALITY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Petr Šopák

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Adam Chromý, Ph.D.

BRNO 2024



Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Petr Šopák

ID: 221022

Ročník: 2

Akademický rok: 2023/24

NÁZEV TÉMATU:

Kvantifikace poruch propriocepčního aparátu pomocí virtuální reality

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit systém v prostředí virtuální reality, který bude sloužit pro analýzu a kvantifikaci při léčbě poruch propriocepčního aparátu v lidském těle. Předpokládá se nejen návrh a implementace aplikace, ale také ověření její přesnosti. Výsledek práce bude použit v reálné medicínské praxi a v navazujícím lékařském výzkumu v rámci CEITEC.

Zadání:

1. Seznamte se s helmou virtuální reality Oculus Quest 2. Provedte rešerši všech dostupných zdrojů pojednávajících o principu lokalizace této helmy a jejích ovladačů.
2. Navrhněte a proveďte experimenty, které poskytnou data pro ověření přesnosti lokalizace. Analyzujte a definujte podmínky, které je nutné dodržet, abychom se pohybovali ve zjištěném intervalu přesnosti.
3. Dle pokynů vedoucího vytvořte novou VR aplikaci v prostředí Unity, která bude sloužit při léčbě poruch propriocepčního aparátu. Aplikace by měla poskytovat nejméně veškeré funkce stávajícího řešení, kterým bude nová aplikace inspirována.
4. Ověřte funkčnost a použitelnost vytvořeného systému praktickým experimentem.

DOPORUČENÁ LITERATURA:

VEGA, José A. a Juan COBO. Proprioception. B.m.: BoD – Books on Demand, 2021. ISBN 978-1-83968-069-4.

Termín zadání: 5.2.2024

Termín odevzdání: 15.5.2024

Vedoucí práce: Ing. Adam Chromý, Ph.D.

doc. Ing. Petr Fiedler, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zabývá vývojem aplikace pro analýzu propriocepce využívající virtuální realitu a herní platformu Unity. Aplikace byla vyvinuta na základě předlohy ve formě instalace obdobné aplikace (bez možnosti přístupu k jejím zdrojovým kódům). Dále aplikace byla obohacena o nové funkce, jako je například černá scéna, a zlepšení stávajících funkcí, jako je přidání realistických modelů, vytvoření nového uživatelského rozhraní a kalibrace pro shodu virtuálních objektů se skutečnými. Byly navrženy a realizovány také experimenty k ověření přesnosti lokalizace zařízení Oculus Quest 2.

KLÍČOVÁ SLOVA

virtuální realita, Oculus Quest 2, Unity, propriocepce, imerze, přesnost lokalizace

ABSTRACT

This thesis explores the development of an application for proprioception analysis utilizing virtual reality and the Unity gaming platform. The application was developed based on a prototype similar in installation but without access to its source codes. Additionally, the application was enhanced with new features such as a black scene, and improvements to existing functionalities, including the addition of realistic models, the creation of a new user interface, and calibration to align virtual objects with their real-world counterparts. Experiments were also designed and conducted to verify the accuracy of device localization using the Oculus Quest 2.

KEYWORDS

virtual reality, Oculus Quest 2, Unity, proprioception, immersion, localization accuracy

ŠOPÁK, Petr. *Kvantifikace poruch propriocepčního aparátu pomocí virtuální reality*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2024. Vedoucí práce: Ing. Adam Chromý, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Bc. Petr Šopák
VUT ID autora: 221022
Typ práce: Diplomová práce
Akademický rok: 2023/24
Téma závěrečné práce: Kvantifikace poruch propriocepčního aparátu pomocí virtuální reality

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Na začátek bych chtěl vyjádřit upřímné poděkování své rodině za neustálou podporu, povzbuzení a důvěru. Speciální díky patří panu Ing. Adam Chromý, Ph.D., za jeho odborné vedení, trpělivost a cenné rady, jež byly nezbytné pro úspěšné dokončení této práce.

Obsah

Úvod	12
1 Virtuální realita	13
1.1 Oculus Quest	14
1.1.1 Architektura helmy	15
1.1.2 Generování map v reálném čase a sledování pozice	15
1.1.3 Sledování ovladačů	17
1.1.4 Sledování rukou	18
1.2 Simultánní lokalizace a mapování (SLAM)	19
1.2.1 Monokulární SLAM	21
1.2.2 Stereo SLAM	23
1.2.3 Monokulární-Stereo SLAM	24
1.2.4 Shrnutí	26
1.3 Virtuální realita v medicíně	27
1.3.1 Aplikace v rehabilitaci	28
1.3.2 Aplikace v psychologii	30
2 Návrh řešení	33
3 Ověření lokalizace Oculus Quest 2	35
3.1 Verifikace lokalizace Vicon Trackeru	35
3.1.1 Návrh experimentu	35
3.1.2 Předzpracování výsledků	38
3.1.3 Výsledky měření	41
3.2 Experiment ověření Oculus Quest 2	41
3.2.1 Návrh standardizovaného prostředí	42
3.2.2 Držáky lokalizačních markerů pro OQ2	44
3.2.3 Aplikace pro získávání dat při měření	44
3.2.4 Postup měření experimentu	47
3.2.5 Předzpracování změřených dat	48
4 Aplikace pro analýzu propriocepce ve VR	50
4.1 Propojení VR a Unity	50
4.2 Návrh scény	51
4.3 Realizace modelů rukou a jejich animací	52
4.3.1 Tvorba uživatelských rukou	52
4.3.2 Animace a pohyby rukou	53
4.4 Implementace hlavní funkce aplikace	54

4.4.1	Experiment 1 a 3 - Box	58
4.4.2	Experiment 2 - Hodnocení	59
4.4.3	Distrakční úlohy	62
4.5	Kalibrace systému	64
4.6	Další funkce	66
4.6.1	Validace pokusu experimentu 2	66
4.6.2	Blednutí modelu ruky	67
4.7	Hodnocení aplikace a její srovnání	69
Závěr		73
Literatura		74
A Obsah elektronické přílohy		82

Seznam obrázků

1.1	Zařazení pojmů do kontinua Virtuality-reality [4]	14
1.2	Architektura sledování helmy [8]	16
1.3	Průběh sledování ovladačů s LED diody [12]	18
1.4	Fáze procesu pro odhad pozice ruky a úhlů prstů [17]	19
1.5	Průběh zpracování SLAM [21]	20
1.6	Přehled algoritmu ORB-SLAM [23]	22
1.7	Přehled algoritmu S-PTAM [24]	23
1.8	Přehled masky segmentace a výsledná reprezentace klíčových bodů ze snímku [25]	24
1.9	ORB-SLAM2: a) Přehled algoritmu. b) přepracování vstupů [26]	25
1.10	Hlavní komponenty ORB-SLAM3 [27]	27
1.11	Významné přínosy v medicíně [28]	28
1.12	Vizuální přehled fází rehabilitačního systému: a) s zapnutou vizuální zpětnou vazbou; b) s vypnutou vizuální zpětnou vazbou [30]	29
1.13	Vizuální přehled jednotlivých her: a) cvičení - zvedání; b) cvičení - konzumace jablka; c) cvičení - otevírání a zavírání afektované ruky [32]	30
1.14	Vizuální přehled experimentů iluze vlastnictví falešné ruky: a) pacient s HMD; b) změna průhlednosti končetiny; c) změna velikosti ruky [34]	31
1.15	Virtuální prostředí experimentu s doktorem a úlohy na řešení [36]	32
1.16	Experiment pro změření RHI: a) průběh experimentu; b) očekávaná trajektorie pohybu ruky v VR pro řešení úloh [37]	32
3.1	Návrh rozmístění kamer Vicon Trackeru a definování středu T	36
3.2	Konstrukce scény: a) manipulátor b) rozložení kamer	36
3.3	Sledovaný objekt - deska s markery: a) připevněna deska na manipulátoru b) nákres desky připevněné na manipulátoru (světlemodrá)	37
3.4	Porovnání pozic pro měření prvního měření u první kalibrace Viconu - translační pohyb: Vlevo zobrazení pozic v 3D prostoru a vpravo závislost jednotlivých souřadnic na frame	39
3.5	Zarovnání modelů: (vlevo) před zarovnáním (vpravo) po zarovnání	40
3.6	Porovnání pozice objektu po kalibraci: (vlevo) zobrazení pozic v 3D prostoru (vpravo) závislost jednotlivých složek pozice na framu	41
3.7	Konstrukce plátna	42
3.8	Návrh plátna optimálního prostředí pro lokalizaci HMD	43
3.9	Návrh modelů: a) modely pro HMD; b) model pro pravý ovladač; c) realizace modelů	44
3.10	Stavový diagram aplikace pro experiment 2	45

3.11 Druhy tratí z pohledu uživatele: a) Přímka; b) Sférická spirála ; c) lemniskáta	46
3.12 Simulace pohybu objektu u třetí dráhy - lemniskáta. Dochází ke změně pozice i orientace	47
4.1 Pohled HMD na scénu a zvýrazněné ovladače za pomoci paprsků . . .	50
4.2 Výsledná scéna místnosti	51
4.3 Mužský (nahore) a ženský (dole) model ruky	52
4.4 Animation system pro stlačení předního tlačítka	53
4.5 Animator controller pro levou ruku	53
4.6 Hlavní smyčka programu	54
4.7 Vstupní uživatelské rozhraní	55
4.8 Prvotní pohled uživatele - informační tabule a modely rukou na stole	56
4.9 Kolidory modelu ruky: kolidor ukazováček a dlaň	56
4.10 Pozice neviditelných prahů ve scéně (bokorys)	57
4.11 Experiment 1 a 3 - box: a) scéna pro experiment; b) interakce uživatele se scénou	58
4.12 Stavový diagram Experimentu 1 a 3	59
4.13 Stavový diagram Experimentu 2	60
4.14 Algoritmus experimentu 2 z více pohledů: a) řešení distrakčních úloh; b) zvýraznění hodnocené ruky po dokončení distrakčních úloh; c) vypnutí zvýraznění ruky po překročení prahu a výběru odpovědi . . .	61
4.15 Průběh experimentu 2 z pohledu uživatele: a) distrakční úloha; b) zvýraznění hodnocené ruky; c) vypnutí zvýrazněné ruky při výběru odpovědi	62
4.16 Druhy distrakčních úloh: a) geometrická úloha; b) matematická úloha	62
4.17 Vizuální zobrazení správnosti úlohy: a) výběr správné odpovědi; b) výběr špatné odpovědi	63
4.18 Kalibrace pohledu HMD v aplikaci	64
4.19 Náskres výpočtu kalibrace: a) výpočet rotace; b) výpočet translace . .	65
4.20 Uživatelské rozhraní pro experiment 2 se zobrazeným upozorněním o invalidním pokusu: a) aktuální pokus je invalidní; b) předchozí pokus byl invalidní	67
4.21 Blednutí ruky v aplikaci	68
4.22 Návrh objektu pro vytvoření blednutí ruky	68
4.23 Srovnání aplikací: a) návrh scény; b) výsledek kalibrace; c) experiment 1 - box; d) experiment 2 - hodnocení polohy ruky. Původní aplikace (vlevo) a nová aplikace (vpravo).	70
4.24 Srovnání vstupního uživatelského rozhraní: a) původní aplikace; b) nová aplikace.	71

4.25 Srovnání distrakčních úloh: a) geometrická úloha; b) matematická úloha. Původní aplikace (vlevo) a nová aplikace (vpravo).	72
--	----

Úvod

Propriocepce hraje klíčovou roli v našem vnímání polohy a pohybu těla v prostoru. Poruchy propriocepce mohou mít značný dopad na každodenní činnosti jedince, jeho schopnost se orientovat a pohybovat, což může vést k závažným bezpečnostním rizikům a snížení kvality života. Tradiční metody diagnostiky těchto poruch jsou často závislé na subjektivních hodnoceních a mohou být omezené ve své přesnosti a objektivitě. V posledních letech však technologický pokrok, zejména v oblasti virtuální reality, nabízí nové možnosti pro přesnější měření a kvantifikaci propriocepčních funkcí. V rámci tohoto pokroku se tato diplomová práce zaměřuje na vývoj a implementaci aplikace v prostředí virtuální reality, která by byla schopna analyzovat a objektivně kvantifikovat poruchy propriocepčního aparátu pro potřeby medicínské praxe a výzkumu.

Cílem této diplomové práce je navrhnout a vyvinout novou aplikaci pro virtuální realitu, která bude používána v lékařském prostředí. Vychází z obdobné instalace aplikace, která neobsahovala možnost přístupu k jejím zdrojovým kódům, scén, textur a podobně, neboli byla vytvořena od úplného počátku. Tato aplikace bude implementována v herním prostředí Unity a speciálně upravena pro použití s headsetem Oculus Quest 2. Zvláštní důraz bude kladen na zahrnutí nových funkcí a splnění specifických požadavků od lékařských odborníků. Součástí práce bude rovněž série experimentů zaměřených na ověření schopnosti lokalizace tohoto zařízení, což je klíčové pro zajištění požadované úrovně přesnosti v medicínských aplikacích.

První kapitola se podrobně věnuje tématu virtuální reality, přičemž klíčovým bodem bude rozbor všech dostupných zdrojů týkajících se headsetu Oculus Quest 2 a jeho využití v medicínském sektoru. Budou zde prezentovány základní principy virtuální reality a její současný přínos k oblasti zdravotní péče. Druhá kapitola bude obsahovat podrobný návrh řešení, které bylo vyvinuto v reakci na konkrétní požadavky lékařů. Zde se zaměří na definování cílů projektu, specifikace funkcí, které aplikace musí obsahovat. Ve třetí kapitole se práce zaměří na metodologii návrhu a provádění experimentů, které mají za úkol ověřit lokalizační schopnosti Oculus Quest 2 v rámci navržené aplikace. Budou zde stanoveny podmínky, které je nutné splnit pro dosažení požadovaného intervalu přesnosti. Poslední kapitola je věnována implementaci a testování nové aplikaci, která bude sloužit k analýze a léčbě poruch propriocepce, a také přidání nových funkcí.

1 Virtuální realita

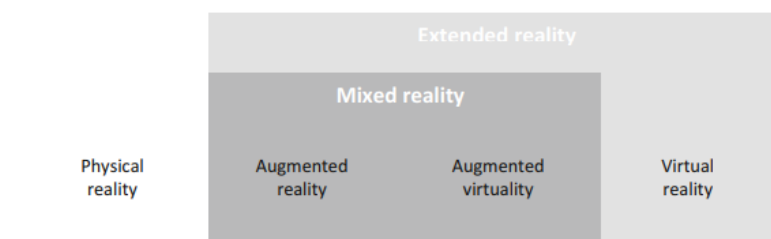
Virtuální realita (VR) simuluje prostředí, které pohltí uživatele do takové míry, že nepocituje odlišnost od reálného světa. Vědci charakterizují VR za pomoci tří vlastností: *(tele-)prezence*, *interaktivita* a *vnoření téže imerze* [1]. Prezence je chápána jako pocit fyzického bytí jinde, než kde se skutečně člověk nachází. Interaktivita ovlivňuje přítomnost a odkazuje na rozsah, v jakém mohou uživatelé manipulovat se svým virtuálním prostředím v reálném čase. Ponoření nemá jednotnou definici. Někteří vědci [2] charakterizují vnoření za pomoci technologických schopností, které jsou objektivně měřitelné jako *inkluzivita* (do jaké míry je vyloučena realita), *rozsáhlost* (rozsah adresových smyslových modalit), *okolí* (velikost zorného pole), *živost* (rozlišení či kvalita displeje) a *přizpůsobení* (rozsah, ve kterém je proprioceptivní zpětná vazba na pohyby těla v souladu s zobrazenou informací). Další vědci [3] charakterizují vnoření jako účast subjektivní složky, která může nabývat libovolnou formu, včetně *kognitivního vnoření* (uživatel pocituje během řešení složitého problému), *smyslově-motorického vnoření* (uživatel pocituje při získávání zpětné vazby o pohybu) a *prostorového vnoření* (uživatel pocituje při provádění rozsáhlých manévru).

Je zapotřebí si definovat několik pojmů VR. První pojmy jsou *virtuální prostředí* a *virtuální svět*. Obvykle se týkají softwaru, který běží na hardwaru, a který může nebo nemusí být založen na VR. Virtuální prostředí se skládá ze softwarových reprezentací skutečných objektů a procesů; rozhraní člověk-počítač pro zobrazování a interakci s těmito modely. Zatímco virtuální svět je specifickým typem virtuálního prostředí, které poskytuje sdílený simulovaný prostor mezi více uživateli. [4]

Dalšími pojmy jsou *smíšená realita*, *rozšířená realita*, *rozšířená virtualita*. Rozšířená realita (AR) odkazuje na kombinaci digitálních a fyzických informací v reálném čase prostřednictvím různých technologických zařízení. To znamená, že přidává virtuální informace do zobrazené skutečné reality. Na rozdíl rozšířená virtualita (AV) odkazuje na virtuální svět rozšířený o mapování snímků nebo videí z reálného světa. Smíšená realita (MR) je konkrétní podmnožina technologií související s virtuální realitou (VR), která zahrnuje sloučení skutečného a virtuálního světa. V souladu s tím VR obvykle vytváří zcela umělé virtuální prostředí, které nabízí plnou virtualizaci, zatímco AR a AV rozšiřují skutečnou svět o virtuální informace a naopak. Tedy MR je širší pojem, který kombinuje anebo spojuje virtuální a skutečné světy.

Poslední pojem je *rozšířená realita* (XR). Jedná se o zastřešující termín, který je definován jako všechna reálná a virtuální kombinovaná prostředí a interakce člověk-stroj generované počítačovou technologií a nositelnými zařízeními. Samotné zařazení pojmů mezi skutečnou a virtuální realitou lze vidět na obrázku 1.1. [4] [5]

Z hlediska imerze lze rozdělit technologie na tři druhy: *neimmerzivní VR*, *poloimmer-*



Obr. 1.1: Zařazení pojmů do kontinua Virtuality-reality [4]

živní VR a imerzivní VR. Neimerzivní VR jsou veškeré technologie, které poskytují počítačově generované prostředí. Umožňují uživateli zůstat v povědomí a mít kontrolu nad svým fyzickým prostředím. Jedná se zde o počítač nebo konzoli. Na rozdíl poloimerzivní VR poskytuje uživateli pocit bytí v jiné realitě, ale zároveň umožňuje spojení s fyzickým okolím. Technologie, poskytující tento typ imerze, jsou vytvořeny za pomoci 3D grafiky a je používána nejčastěji pro vzdělávací či školicí účely jako simulátory. Tato práce se bude zabývat imerzivních VR, které nabízí uživateli plně pohlcující virtuální realitu a komunikaci s ní. Jsou tvořeny z hlavní jednotky, tzv. *virtuálních brýlí* (HMD). Umožňují ponořit uživatele do virtuálního světa s vysokým rozlišením a širokým zorným polem, a blokovat pohled do fyzického okolí. Používají se v mnoha odvětvích, například v herním průmyslu, medicíně, vzdělávání a dalších. [5]

Technologie VR se dále rozděluje do dvou metod získávání informací z okolí: *inside-out* a *outside-in*. Outside-in (sledování zvenčí dovnitř) je metoda, kdy místnost obsahuje mnoho senzorů, které sledují zařízení uživatele, ve které jsou vestavěné LED diody. Na rozdíl metoda inside-out (sledování zevnitř ven) integruje senzory do samotné helmy, která sleduje signály v místnosti. Tuto metodu používá Oculus Quest či Oculus Rift, které budou dále uvažovány v rámci této diplomové práce. [6]

1.1 Oculus Quest

Oculus Quest se oproti konkurenčním VR zařízením vyznačuje autonomním provozem, pokročilým hardwarem, vysokým rozlišením obrazu a zejména cenovou dostupností. Jeho vysoký výkon v kombinaci s přívětivou cenou jej učinil nejpopulárnějším VR headsetem na trhu. Dále poskytuje uživatelům rozsáhlé knihovny a doplňky pro tvorbu vlastních her nebo aplikací, a to především v rámci herního engine Unity. [7]

Jedná se zcela o *all-in-one* (vše v jednom) a bezdrátový herní systém VR. Hlavní snahou je odemknout plný potenciál VR a AR zážitků. To znamená, že technologie potřebuje pracovat kdekoli, přizpůsobit se novému okolí, kde se vyskytuje a jak se uživatel v reálném prostředí pohybuje.

Hlavní výhody jsou odebrání přesné sledované polohy a jejich dostupnost v reálném čase, a samostatná helma, díky které se zvýší kompatibilita a energetická úspornost.

Technologie Oculus Quest se nazývá *Oculus Insight*. Je to první případ, který používá helmu s šesti stupni volnosti (6DoF) a sledování ovladačů. Využívá systémy počítačového vidění CV a *vizuálně inerciální simultánní lokalizaci a mapování* (SLAM). Zkráceně umožňuje vypočítat každou milisekundu přesnou polohu helmy a ovladačů v reálném čase, aby převedl pohyby uživatele do VR. Využívá SLAM ke sledování polohy helmy uživatele a sledování konstelace (constellation tracking) ke sledování pozic ovladačů. CV systémy spojují vícenásobné vstupy z helmy a ovladačů, aby zvýšily přesnost a dobu odezvy sledování polohy systému. [8]

1.1.1 Architektura helmy

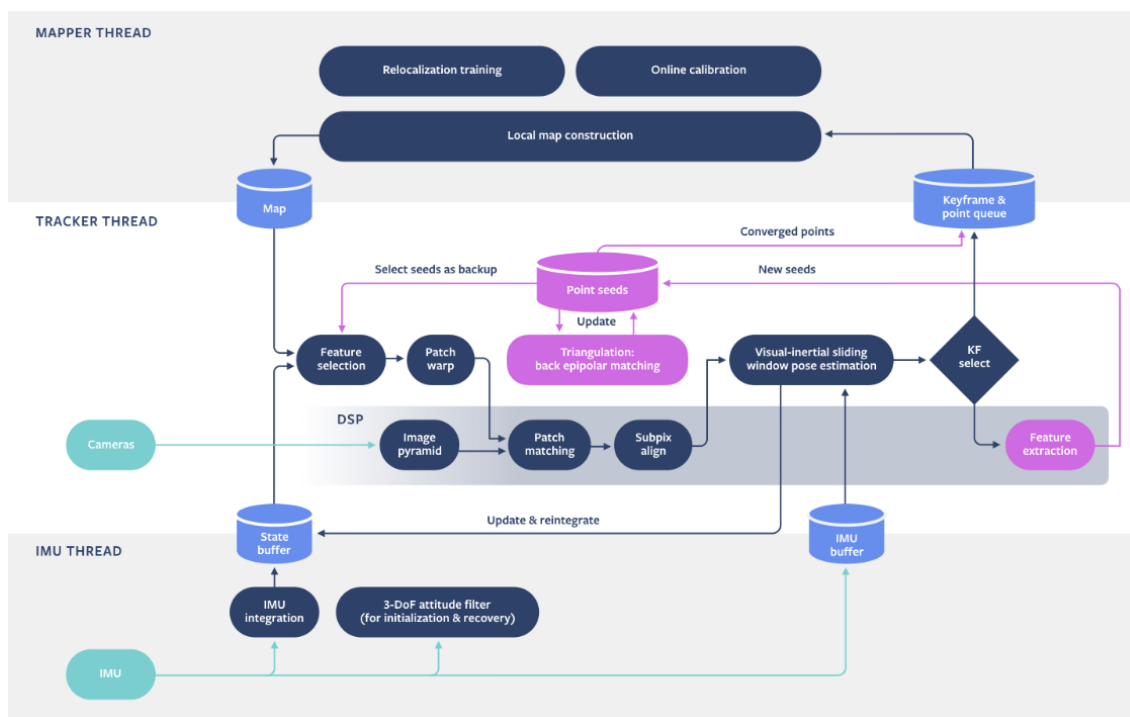
Architektura helmy pro zpracování sledování je zobrazena na obrázku 1.2. Architektura byla vytvořena tak, aby zefektivnila výpočetní výkon a energetickou úsporu. Společně s pokročilými algoritmy počítačového vidění (vizuálně-inerciální mapování, rozpoznávání prostoru a rekonstrukce geometrie) určí polohu objektů ve vztahu k jiným objektům v daném místě. Díky této sadě algoritmu umožňuje zařízení VR přesně určit svou polohu, identifikovat aspekty geometrie místností (například umístění polohy) a sledovat polohy helmy a ovladačů s ohledem na 3D mapu, kterou generuje a neustále aktualizuje Insight. Z obrázku 1.2 lze vidět, že Oculus Insight zpracovává několik vláken dat najednou v reálném čase. Vlákno mapovače upravuje mapu a odesílá aktualizované kopie do vlákna trackeru, který používá snímky kamery k odhadu pozic v snímcích poskytovaných mapovačem, zatímco vlákno *inerciální měřící jednotky* (IMU) využívá měření z IMU pro aktualizaci posledního stavu SLAMu. [8]

1.1.2 Generování map v reálném čase a sledování pozice

Oculus Insight pro generování map a sledování pozice získává tři druhy dat. *Data lineárního zrychlení a rychlosti otáčení* z IMU. V helmě jsou integrována tak, aby sledovala orientaci a polohu každého z nich s nízkou latencí. *Obrazová data* z kamer v helmě pomáhají vytvářet 3D mapu prostoru, která přesně určuje orientační body (landmarks), jako jsou rohy nebo vzory. Tyto orientační body jsou pozorovány opakovaně, což umožňuje kompenzovat posun (drift). Posuny jsou běžným problémem s IMU, kde se díky jeho integračním vlastnostem s rostoucím časem hromadí nesrovnalosti měření prostoru a následně způsobují nepřesnosti sledování polohy. [8]

Článek [9] se zaměřuje na minimalizaci driftu systému Oculus Insight za účelem zlepšení přesnosti sledování a pochopení jeho omezení. Během experimentu byly

Headset tracking compute architecture



Obr. 1.2: Architektura sledování helmy [8]

upravovány podmínky v prostředí a mezi uživateli. Výsledky experimentu ukázaly, že prostředí s vysokokontrastními vzory dosahuje lepšího výkonu sledování, ale bylo zjištěno i zlepšení výkonu u pravidelného mřížkového vzoru. Délka trvání experimentu nevykazuje významný vliv na efektivitu sledování, což umožňuje její zanedbání v dalších experimentech.

Během pohybu Oculus Insight detekuje pixely na snímcích s vysokým kontrastem, jako jsou rohy místnosti či okna. Tyto oblasti jsou sledovány a spojovány v průběhu času, od jednoho snímku k dalšímu. Díky dostatečně dlouhé základní linii pozorování (baseline of observation), Oculus je schopný triangulovat 3D polohu každého bodu v okolí. To tvoří základ 3D mapy prostředí systému. Toto řeší SLAM, neboli automatické rozpoznávání rysů.

V článku [10] byl proveden experiment pro zjištění přesnosti sledování pozice a orientaci u Oculus Quest 2 a byl porovnán s SteamVR Tracking. Ve srovnání se SteamVR Tracking Oculus Quest 2 vykazuje vyšší přesnost, kde parametr porovnání byla výška sledovaného objektu. Statistika středních chyb výšky sledovaného objektu u Oculus Quest 2 jsou $\mu = -0,001 \text{ m}$ a $\sigma = 0,004 \text{ m}$.

V poslední řadě jsou použity infračervené LED diody v ovladačích, které jsou detekovány za pomoci čtyř kamer umístěné na HMD. Umožňuje systému svázat

posun polohy ovladače. [8] Pro Oculus Quest 2 byl realizován i druhý způsob pro detekci rukou - *XR marker-less motion tracking* neboli znám pod jménem jako *hand-tracking system*. [17]

Oculus také snižuje limitace imerzivity. Využívá dlouho testovaný *kinematický predikční algoritmus*, který byl vytvořen pro předpovídání přesného pohybu v řádech milisekund. Tímto algoritmem se zvyšuje přesnost mezi skutečnou pozicí a pozicí ve virtuální realitě. Dále je využit pro generování plynulejšího pohybu pro uživatele. [8] V Článku [11] se prezentuje metoda založená na *reinforcement learning*, která zpracovává signály z HMD (a dvou ovladačů) a simuluje pravděpodobné a fyzikálně platné pohyby celého těla.

1.1.3 Sledování ovladačů

Sledování ovladačů je realizován za pomoci *LED matching* technologií. Jsou použity algoritmy, které běží na každém získaném snímku a spadají do dvou hlavních tříd. První je statistická metoda *brute matching*, která intenzivně hledá řešení. Jsou používány, pokud nejsou k dispozici žádné předchozí informace o poloze a orientaci ovladačů. Když máme pozici ovladače z předchozího snímku, je použit *proximity matching*, což vyhledává v velmi menším okně snímku.

Prvotně je použita metoda, u které jsou hledány LED diody postupně v jedné kameře po druhé. Je zapotřebí nasnímat jednou kamerou vždy dvě diody generující hypotézu a dvě pro ověření. To znamená, že je zapotřebí alespoň detekovat čtyři LED diody pro správnou shodu. Kromě toho to zvyšuje pravděpodobnost protichůdných výsledků párování napříč více kamerami. Aby se tomu zabránilo, byl realizován proces, který využívá kalibračních dat stereokamery při vyhodnocování korespondence LED diod v různých kamerách. To umožňuje spolehnout se na vzájemný vztah každé kamery a vyřešit nejistoty.

Tato metoda bohužel selhává, když jsou ovladače blízko okraje zorného pole, příliš daleko nebo příliš blízko kamer anebo když dochází ke okluzi. Díky tomu, že se výsledky vyhodnocují celkově a ne pouze z jednotlivých kamer, vylučuje možnost nekonzistentních výsledků shody. Toto pomáhá rozšířit sledovatelnost ovladačů a snížit výskyt 3DOF v důsledku ztráty v blízkosti okrajů zorného pole, což umožňuje plynulejší sledování.

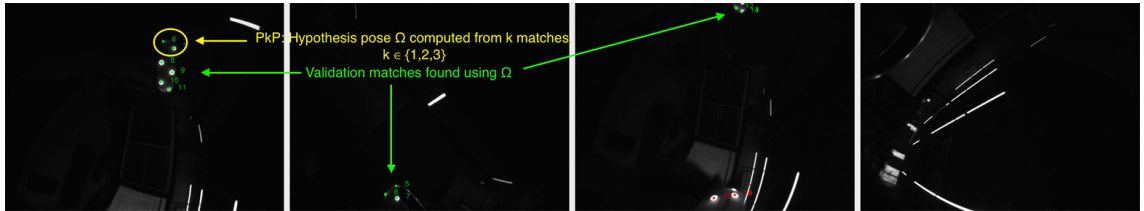
Aby bylo možné zjistit pozici ovladače, když kamera vidí méně než čtyři LED diody, byl navržen *Perspective-1-Point pose solver* (P1P). [12]

Perspective-n-Point (PnP) je základní problém v oblasti počítačového vidění, který spočívá v určení polohy a orientace kamery pomocí sady n 3D bodů ve světě a jejich odpovídajících 2D projekcí na snímku kamery. [13]

P1P používá předpokládanou pozici pro přímé ověření shody namísto ověřování pomocí statistických předpovědí nebo předpovědí nejbližšího souseda. Snižuje minimální požadavek na shodu na 2 shody (zajišťuje správné omezení překladu a měřítko pro optimalizaci stereo pozice). V případě méně než 4 LED diod používá pouze optimalizaci stereo pozice, aby se zabránilo nedostatečné orientaci.

Dále je použit *P2P pose solver*, který využívá dvě shody a předchozí orientaci pozice k řešení pozice komponenty pózy. Sníží minimální požadavek na shodu na tři (dvě shody generující hypotézy a jednu ověřující shodu). [12]

Obrázek 1.3 zobrazuje čtyři snímky, na nichž je zachyceno detekování LED diod (ověřovacích a generativních). Na základě počtu identifikovaných diod je použit solver k určení konečné pozice ovladače. Snímky demonstrují robustnost algoritmu, který efektivně ignoruje jiné potenciální markery v prostředí.



Obr. 1.3: Průběh sledování ovladačů s LED diody [12]

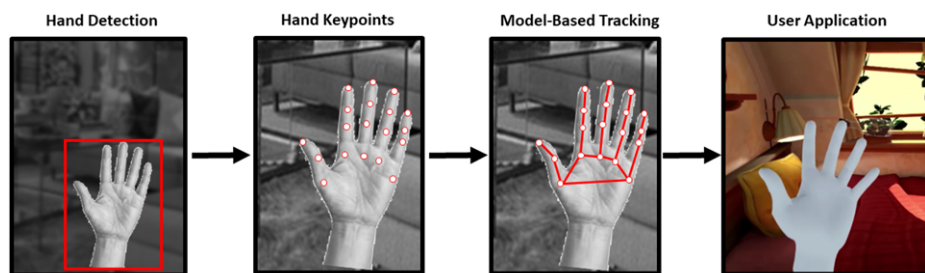
V článcích [14] [15] byly provedeny experimenty pro zjištění přesnosti pozice a orientaci u ovladačů ve fyzickém a virtuálním světě. Bylo zjištěno, že se jedná o velice robustní systém, který má chyby sledování orientace a polohy menší než $0,85^\circ$ a $0,7\text{ cm}$.

1.1.4 Sledování rukou

Oculus Quest 2 nabízí optický systém [16], který sleduje ruce uživatele bez použití značek. Tento systém umožňuje uživateli manipulovat s virtuálním prostředím pouze holýma rukama a bez nutnosti používat ovladače. Tento systém funguje na zařízeních bez potřeby PC a používá vícestupňový proces pro odhad pozice ruky a úhlů prstů v reálném čase.

Na obrázku 1.4 jsou zobrazeny fáze procesu. První fáze *detekce ruky* (Hand Detection) zahrnuje správnou identifikaci rukou od okolních objektů a pozadí. Následně jsou identifikovány a označeny *klíčové body* (Hand Keypoints) na ruce a prstech, které slouží jako vstup pro *model založený na inverzní kinematice ruky* ve další fázi (Model-Based Tracking). Tento model používá klíčové body k odhadu globální pozice ruky a úhlů kloubů prstů. Ve finální fázi jsou výstupy tohoto modelu pak

zpracovány do uživatelské aplikace jako data o pozici a úhlech kloubů, například pro zobrazení odpovídající virtuální ruky ve virtuálním prostředí. [17]



Obr. 1.4: Fáze procesu pro odhad pozice ruky a úhlů prstů [17]

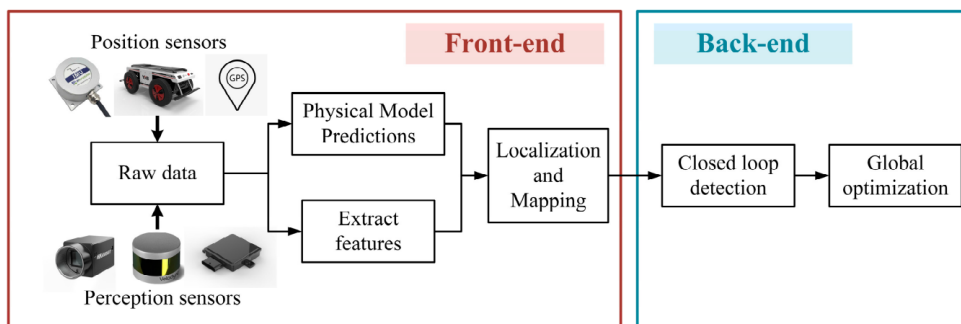
1.2 Simultánní lokalizace a mapování (SLAM)

Oculus Quest 2 neuvádí, jaký přesně algoritmus SLAM využívá. Pro navržení odpovídajícího experimentálního prostředí, které bude maximalizovat efektivitu a přesnost svoje lokalizace, je nezbytné se seznámit s dostupnými algoritmy SLAM, které následně umožní definovat adekvátní podmínky pro experiment.

SLAM je jedna ze základních výzev robotiky. Tato metoda se používá, například u autonomního vozidla, pro zrekonstruování mapy neznámého okolí. Kromě toho vozidlo identifikuje vlastní polohu a orientaci opakovanou identifikací prostorových charakteristik během pohybu. Díky tomuto je často používána pro zjišťování informací pro následné plánování vlastní trasy či vyhýbání se překážek. [18] [19] [20]

Na obrázku 1.5 lze vidět moderní SLAM systém, který se rozděluje na dvě velké části *front-end* a *back-end*. Front-end je zodpovědná za odhad aktuální pozice snímku v reálném čase a ukládání odpovídajících mapových informací. Back-end část je zodpovědná za rozsáhlou optimalizaci pózy a scény. *Detekce uzavřené smyčky* (closed loop detection) je jedním z klíčových problémů SLAM, který pomáhá robotu identifikovat navštívené scény a spouštět korekci posunu v globálním měřítku. Rozsáhlá *globální optimalizace* (Large-scale global optimization) je také hlavním rozdílem mezi SLAM a moderní odometrií. Tyto dvě metody mají mnoho podobností v odhadu pozice. Většina moderních technologií s více senzory působí na front-endu, aby dosáhla vysoké přesnosti a nízkého driftu, filtrování zdrojů více dat. [21]

Lokalizační technologie se liší v závislosti na prostředí a požadavků (účinnost, přesnost, rychlost a spolehlivost). Toto může být dosažitelné s *globálním polohovým systémem (GPS)*, *inerciálním měřicí jednotkou (IMU)*, či jinými prostředky. Bohužel každý z nich může selhat z různých důvodů, například nepřesných přenosů, či intenzivního ohrožení životního prostředí u GPS anebo IMU mají častou kumulativní



Obr. 1.5: Průběh zpracování SLAM [21]

poruchu. S rychlým rozvojem SLAM systémy vybavené kamerou, IMU, LiDARem a dalšími senzory, který se objevil v posledních letech, jsou tyto problémy postupně a důsledně překonávány. [20]

Nejčastěji se SLAM rozděluje dle použitých senzorů, dle nich i jejich celý front-end, na dvě skupiny: *Laser-SLAM* a *Visual-SLAM*. Jedná se o *jedno-senzorové SLAM systémy*. Tyto systémy jsou nespolehlivé a plné nejistot. Nejsou schopny se vypořádat se s více komplexním prostředím, jako jsou vysokorychlostní scény, malé prostory či současně otevřené a velké scény. [20] [21]

Existují i *multi-senzorové systémy* nebo též fúze, které většinou využívají kombinaci Laser-SLAM a Visual-SLAM nebo jiné kombinace senzorů. Toto lze kategorizovat jako *volně sprážené, resp. těsně propojené režimy*. Volně propojený systém zpracovává naměřená data každého senzoru samostatně a spojuje je ve filtr, který marginalizuje data aktuálního snímku, aby se dosáhlo nejnovějších výsledků odhadu stavu. Pevně propojený systém společně optimalizuje naměřená data všech senzorů a kombinuje pozorovací charakteristiky a fyzikální modely každého senzoru pro získání robustnějšího odhadu pozice. Jedná se zde například o *LIDAR-IMU systém* anebo pro tři použité senzory *Visual-LIDAR-IMU systém*. [21]

Z krátkého popisu funkčnosti Oculus Insight je patrné, že se bude jednat zejména o **Visual SLAM**. Nadále se text bude zabývat zejména tímto typem. *Visual SLAM* je sada SLAM technik, které používají pouze snímky získané z kamer či jiných obrazových snímačů. Na rozdíl od senzorů používaných v tradičních SLAM, jako GPS nebo LiDAR jsou kamery cenově dostupnější a jsou schopny shromáždit více informací o prostředí, jako je barva, textura a vzhled. Navíc moderní fotoaparáty jsou kompaktní, nízkonákladové a mají nízkou spotřebu energie. Lze tento systém nalézt v mnohu různých aplikacích jako například ovládání humanoidních robotů, bezpilotních vzdušných a pozemních prostředků anebo endoskopie.

V závislosti na typu kamery existují tři základní typy SLAM: *monokulární, stereo* a *RGB-D*. Stereo SLAM je multi-kamerový SLAM, který umožňuje získat určitý

stupeň rozlišení trajektorie a zároveň poskytuje výhodu být více univerzální oproti třeba RGB-D, který je více citlivý při používání ve venkovním prostředí. Běžně používané vizuální kamery jsou *monokulární RGB kamery*, *stereo kamery*, *všesměrové kamery* nebo *RGB-D kamery*. Všesměrová kamera zachytává snímky současně ve 360° a RGB-D kamera snímá další informaci (hloubku pixelu) k barevnému snímku.

Jak bylo řečeno dříve SLAM je reprezentován třemi hlavními sekcemi a to lokalizace, mapování a hledání trasy. Vizuální SLAM systémy pro lokalizaci polohy uživatele a orientaci v prostoru triangulují polohu nastavených bodů prostřednictvím po sobě jdoucích snímků kamery a současně tyto informace používají k odhadu polohy kamery. Neboli je snaha o mapování okolí ve vztahu k poloze uživatele za účelem usnadnění navigace. Pokud je dostatečný počet bodů, lze rychle porozumět jak orientaci senzoru, tak struktuře fyzického prostředí a lze toto dosáhnout pomocí jediné 3D kamery oproti jiným forem SLAM. Cílem lokalizace je minimalizovat chybu reprojekce, neboli rozdíl mezi projektovanými a skutečnými body, za pomocí algoritmických úprav *úprava svazků* (bundle adjustment - BA).

Mapování je prováděno za pomoci kamer k získávání dat o okolního prostředí. Následně za pomoci kombinace počítačového vidění a algoritmů odometrie pro mapování prostředí umožňuje robotům navigovat se samostatně a zlepšit lokalizaci. Ve většině případů je přidána inerciální měřicí jednotka (IMU) pro měření pohybu jejího těla. Není vhodné se jenom spoléhat na odometrii anebo na vlastní výpočty polohy, protože dochází k akumulaci chyb způsobena šumem a výsledná poloha se stává nejistou. Zde je nezbytné použít *uzavření smyčky* (Loop closer). Je vyhledávána lokalita, která byla již dříve mapována - *detekce smyčky* (loop closure detection). Poté je možné opravit a upravit vygenerovanou mapu pro tyto akumulované šумы. Díky těmto všem nasbíraným informacím lze vygenerovat celková mapa.

Pokud systém lokalizuje vlastní pozici a má namapované okolí kolem sebe, tak naplánování cesty k uživatelem vybraném místě již nepředstavuje problém. [22]

1.2.1 Monokulární SLAM

Tento typ se za účelem mapování, lokalizaci a hledání cesty výhradně spoléhá na sekvenci *monokulárních snímků* zachycených pohybující se kamerou. Obrazová sekvence je obvykle soubor obrazů, kterou jsou si navzájem podobné. Zde patří algoritmy jako *MonoSLAM*, *PTAM*, *ORB-SLAM* a další. Tyto algoritmy postrádají schopnost dobrého výkonu a hlavně ve velkých, přeplněných vnitřních prostředích. [22] Zde je zásadní popsat zejména metodu ORB-SLAM, protože z ní pak vychází další metody.

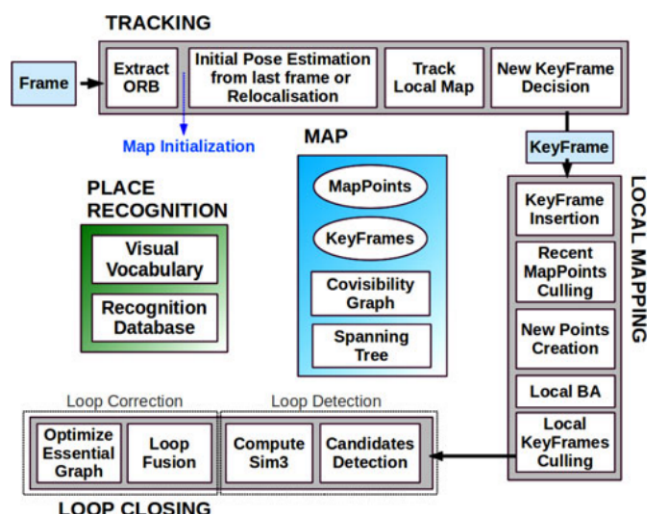
ORB-SLAM

Systém, který je založený na extrahování rysů ze snímků, schopný pracovat v malých ale i ve velkých vnitřních nebo vnějších prostorech v reálném čase. Celý algoritmus lze vidět na obrázku 1.6.

Jak je z pojmenování systému zřejmé, základem jsou *ORB rysy* (Oriented FAST and Rotated BRIEF), které jsou použity pro efektivní a robustní detekci rysů, následného porovnávání těchto rysů a manipulaci s rotací snímku. ORB rysy kombinují dvě komponenty: *FAST detekce klíčových bodů* a *BRIEF generování deskriptorů*. FAST je algoritmus detekování míst, kde dochází k významným změnám intenzit. BRIEF je binární deskriptor, který představuje místní vzory intenzit obrazu kolem klíčových bodů. Využívá binární testy k porovnání intenzit sousedních pixelů, které se následně využívají pro generování binárních řetězců, které zakóduje informace o místní struktuře.

Sledování a lokalizace je prováděná identifikací ORB rysů v každém novém snímku a porovnává je s rysy předchozího snímku. Tento sledovací proces pomáhá odhadnout pohyb kamery vzhledem k předchozí pozici. K generování mapy a vytvoření uzavřené smyčky se nepoužívají všechny snímky, ale jsou z nich vybrány znova klíčové snímky, ve kterých dochází velkým změnám. Samotné generování mapy se provádí za pomoci triangulace 3D pozice rysů z klíčových snímků na základě jejich odpovídajících 2D obrazových souřadnic z více snímků. V poslední řadě dochází k detekci uzavřené smyčky, která vždy porovnává rysy aktuálního snímku s uloženým prvotním klíčovým snímkem.

Je zapotřebí inicializovat algoritmus počátečním odhadem pozice kamery a počátkem pro generování mapy. [23]



Obr. 1.6: Přehled algoritmu ORB-SLAM [23]

1.2.2 Stereo SLAM

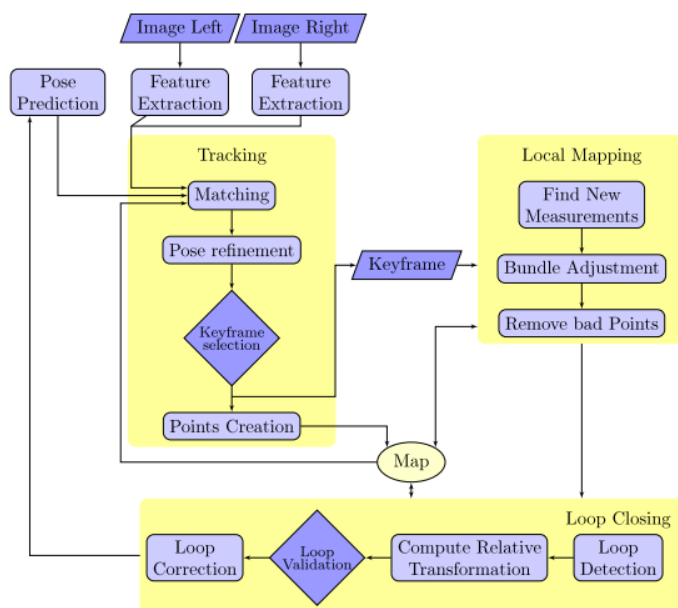
Stereo SLAM spoléhá na funkční body k odhadu trajektorie kamery a vytvoření mapy prostředí. Hlavní body jsou obvykle body ze všech rohů okolí. Výkon je ovlivněn v prostředí s nízkou texturou, kde je velmi obtížné najít dostatečný počet spolehlivých bodů. [22]

S-PTAM

S-PTAM systém využívá stereo kamery, díky nim je systém schopen srovnat stejné klíčové body. Když robot projíždí prostředím, je následně možné sledovat tyto klíčové body v následujícím novém získaném snímku. Též umožňuje zlepšit odhad hloubky od objektu a sledování pozice robota.

Systém je založen na extrahování rysů z klíčových snímků, kde extraktor rysů je kombinace detekčního postupu klíčových bodů a výpočtu jedinečného podpisu, tzv. *deskriptoru*, pro každý takový detekovaný bod. S-PTAM nabízí mnoho metod k výběru. Nejčastěji používané detektory jsou ORB, GFTT či FAST a deskriptory jako ORB, SURF či BRIEF.

Algoritmus je znázorněn na obrázku 1.7. Z obrázku je patrné, že tento systém je založený na dvou paralelně pracujících bločcích: *sledování kamery* a *optimalizace mapy*. V bločku sledování kamery dochází k srovnávání rysů a extrahování nových bodů, a odhadování svojí pozici pro každý nový snímek. V bloku mapování iterativně zpřesňuje blízké klíčové body, které tvoří mapu. S každým novým snímkem je snaha k uzavření smyčky. [24]

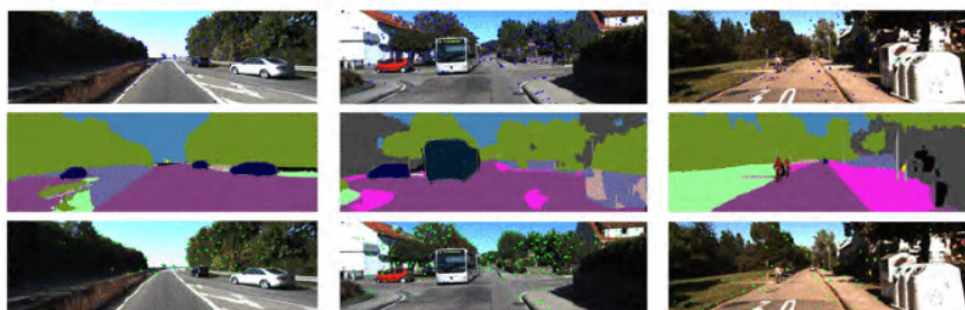


Obr. 1.7: Přehled algoritmu S-PTAM [24]

DOC-SLAM

Tento algoritmus využívá *panoptickou segmentační síť* pro zachytávání statických informací o pozadí a odlišnostech každého pohybujícího se objektu. U většiny SLAM může pohybující se objekt poškodit kvalitu odhadu pozice kamery a dokonce vést k selhání systému. Díky zavedení dříve zmíněné sítě se zvyšuje robustnost systému v prostředích s pohyblivými instancemi.

DOC-SLAM je tvořen z pěti modulů: panoptická segmentace, extrakce bodů, sledování, kontrola konzistence pohybu, doplněk klíčových bodů a back-end. Pro segmentaci je využita konvoluční neuronová síť *UPNet*. Díky ní jsou získávány segmentované masky ze vstupních snímků. Dále vstupní snímky jsou současně přivedeny do modulu extrakce klíčových bodů. Body jsou extrahovány za pomoci algoritmu FAST. V modulu kontroly konzistence pohybu jsou na základě masky extrahovány body absolutních statických vlastností a jsou brány jako klíčové body, které jsou použity pro řešení počáteční transformace. Za pomoci této počáteční transformace a instance masky, v bloku pro odstranění dynamických bodů, jsou odstraněny prvky skutečně pohybujících se objektů, ale současně jsou zachovány statické klíčové body. Pokud počet klíčových bodů je nedostačující, jsou vygenerovány další klíčové body za pomoci masky scény s pozadím a statickými objekty. Všechny klíčové body jsou nakonec přivedeny pro optimalizaci pozice kamery do back-end modulu. Na obrázku 1.8 lze vidět masky po segmentaci a výsledné klíčové body. [25]



Obr. 1.8: Přehled masky segmentace a výsledná reprezentace klíčových bodů ze snímku [25]

1.2.3 Monokulární-Stereo SLAM

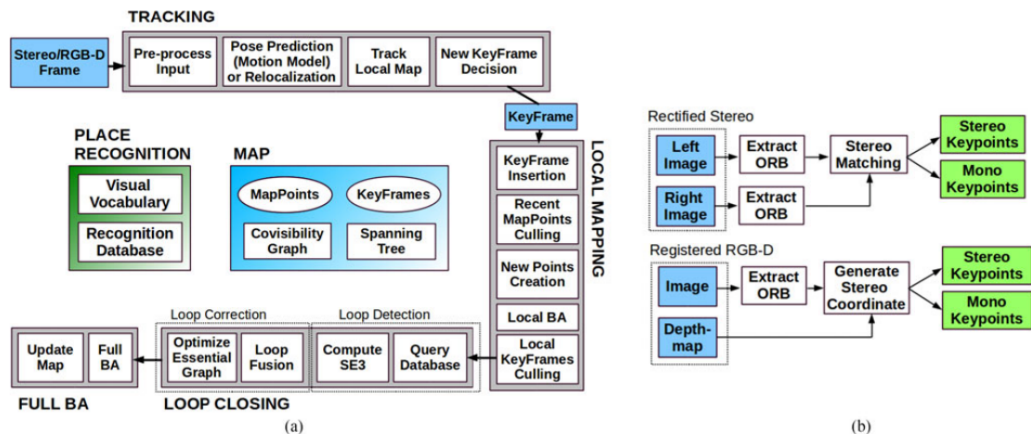
Tento typ je kombinace dříve popsáných skupin, které používají sekvenci snímků anebo pouze rysové body pro mapování, lokalizaci a hledání trasy. Většina algoritmů používá úpravu svazků a některé *lokální úpravu svazků* (LBA). LBA je metoda odhadu geometrie sekvencí snímků pořízených kalibrovanou kamerou. Díky ní se

sníží výpočetní složitost a umožňuje zpracování v reálném čase se srovnatelnou přesností jako se standardní úpravou svazků. Zde patří například *ORB-SLAM2* či *ORB-SLAM3*. [22]

ORB-SLAM2

ORB-SLAM2 je open-source systém a je určen pro stereo a RGB-D kamery. Vychází z dříve zmíněného monokulárního ORB-SLAM a pracuje přímo na standardním CPU od malých zařízení až třeba po autonomní auta. Tedy umí pracovat i s monokulárním vstupem. Systém je znova založený na extrakci ORB rysů. Celý algoritmus lze vidět na obrázku 1.9 za a) a předpracování vstupních snímku za b). Algoritmus má tři hlavní paralelní vlákna: 1) sledování pro lokalizaci kamery s místní mapou a minimalizací reprojekční chyby použitím *BA pouze pro pohyb* (motion-only BA); 2) lokální mapování pro správu mapy a její optimalizaci - provádí za pomoci *lokální BA*; a 3) uzavření smyčky pro detekci velkých smyček a korekce nahromaděného driftu provedením optimalizace polohového grafu. Díky těmto vláknům se spouští čtvrté vlákno, které provede *plnou BA* po optimalizaci pozice grafu, aby se vypočítala optimální struktura a řešení pohybu.

Pokud se z dlouhodobého hlediska výrazně nemění podmínky prostředí, lze použít *režim lokalizace* k umožnění *lehké dlouhodobé lokalizaci* (lightweight long-term localization). Pokud je to potřeba, v tomto módu jsou vlákna lokálního mapování a uzavření smyčky deaktivovány a kamera je nepřetržitě sledována za pomoci relokalizace. Tento mód využívá shody *vizuální odometrie* a shody k bodům mapy. Shoda vizuální odometrie je shoda mezi ORB rysem v aktuálním snímku a 3D body vytvořeny v předchozím snímku ze stereo/hloubkové informace. Díky tomuto lokalizaci je robustní v nenamapované oblasti, ale drift se může postupně hromadit. Shody bodů na mapě zajišťují lokalizaci bez driftu. [26]



Obr. 1.9: ORB-SLAM2: a) Přehled algoritmu. b) předpracování vstupů [26]

ORB-SLAM3

Je založený na ORB-SLAM2 a ORB-SLAM-VI, kde ORB-SLAM-VI je *Vizuálně-inerciální SLAM*, což je kombinace vizuální a inerciálního senzoru poskytující odolnost vůči špatné textuře, rozmazání pohybu a okluzím. V případě monokulárního systému, činí měřítko pozorovatelné. Používá vizuálně-inerciální BA založené na preintegraci IMU.

ORB-SLAM3 je multimapový a multisekcový systém schopný pracovat v čistě vizuálních anebo vizuálně-inerciálních režimech s monokulárními, stereo nebo RGB-D senzory využívajícími modely kamer s *pin-hole* a *fish-eye*. Na obrázku 1.10 jsou vykresleny všechny komponenty tohoto systému. Jedná se o ORB-SLAM2 s nově přidanými komponentami.

Sledovací vlákno zpracovává informace senzoru a vypočítává pozici aktuálního snímku vzhledem k aktivní mapě v reálném čase, čímž se minimalizuje chyba při reprojekci odpovídajících mapových prvků. Také na něm závisí, který aktuální snímek se stane klíčovým snímkem. Rychlost těla a odchylky IMU se odhadují za pomoci inerciálních reziduí při optimalizaci. Při ztrátě sledování se vlákno pokusí přemístit aktuální snímek ve všech mapách Atlasu.

Lokální mapovací vlákno přidává klíčové snímky a body do aktivní mapy, a současně odstraňuje nadbytečné snímky. Postupně zpřesňuje mapu za pomoci vizuálního nebo vizuálně-inerciálního BA, které pracuje v lokálním okně klíčových snímků blízko aktuálního snímku. Pokud se jedná o inerciální případ, zde parametry IMU jsou inicializovány a zpřesňovány mapovacím vláknem.

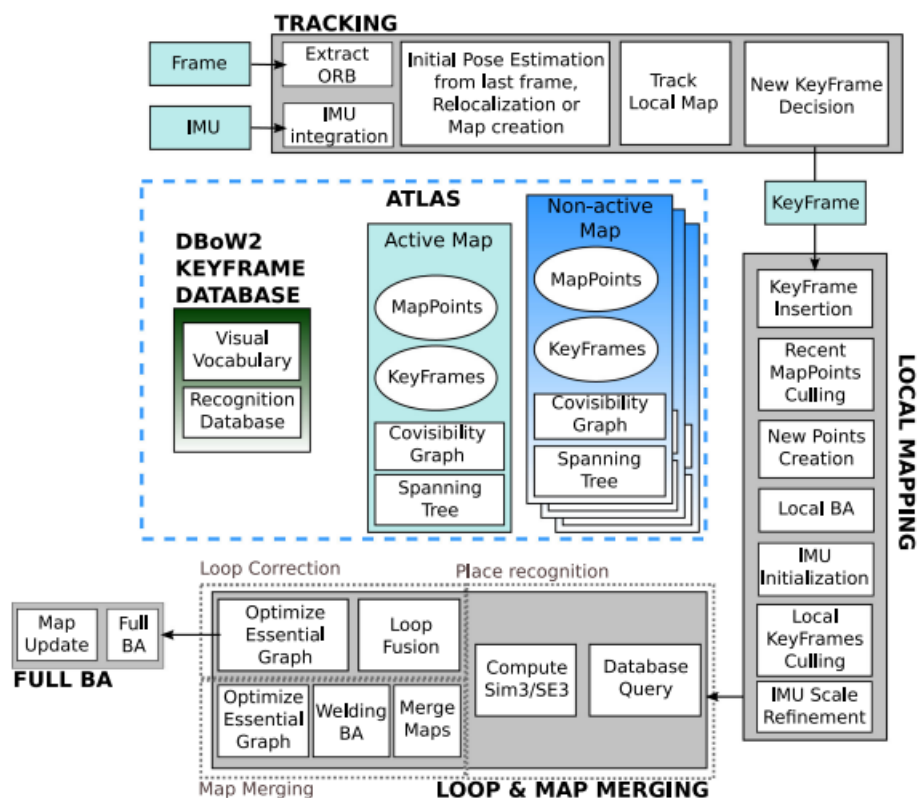
Vlákno slučující smyčku a mapu detekuje společné oblasti mezi aktivní mapou a celým Atlasem. Pokud společná oblast patří do aktivní mapy, provede se korekce smyčky. Ale pokud patří do jiné mapy, obě mapy se sloučí do jediné, která se stává aktivní mapou. Po opravě smyčky se spustí plný BA, aby se mapa dále zpřesnila.

Atlas je multimapová reprezentace složená ze sady oddělených map. Rozděluje mapy na *aktivní mapy*, kde sledovací vlákno lokalizuje příchozí snímky a je průběžně optimalizováno, a na *neaktivní mapy*, to jsou všechny ostatní mapy uložené v Atlasu. Tedy je vytvářena databáze klíčových snímků. [27]

1.2.4 Shrnutí

Ideální prostředí pro efektivní využití SLAM by mělo splňovat několik klíčových kritérií, aby maximalizovalo přesnost a výkon těchto systémů. Strukturované prostředí s jasně rozpoznatelnými, vysoce kontrastními a stabilními orientačními body zlepšuje schopnost SLAM algoritmů sledovat pohyb a vytvářet přesné mapy. Statické prostředí, kde se objekty nemění ani nepohybují, je rovněž klíčové, jelikož dynamické změny mohou vést k chybám v mapování a lokalizaci.

Pro algoritmy závislé na vizuálních datech je důležité dobré a konzistentní osvětlení, zatímco prostředí s minimálním opakováním textur a vzorů pomáhá v rozlišování mezi jednotlivými orientačními body. Rozmanitost v prostředí podporuje rozlišovací schopnosti SLAM systémů, a omezené, dobře definované prostředí pomáhá udržet přesnost v rozsáhlých nebo otevřených oblastech.

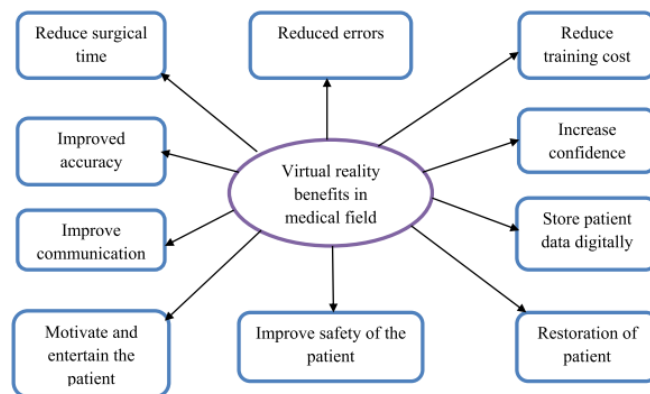


Obr. 1.10: Hlavní komponenty ORB-SLAM3 [27]

1.3 Virtuální realita v medicíně

Virtuální realita se používá v mnoha odvětvích jako v herním průmyslu, zdravotnictví, armádě a podobně. Tato kapitola se zejména zaměří na využití virtuální reality v medicíně. Hlavně na takové aplikace, které přinesou poznatky užitečné pro návrh a vytvoření aplikace v rámci této práce.

V dnešní době je snaha využít VR pro zvýšení účinnosti léčby pacientů, ale také pro školení nového zdravotnického personálu novým dovednostem v bezpečném prostředí. Třeba simulace praxe na operačním sále, anatomie či při kontrole infekcí. Obrázek 1.11 ukazuje současné významné přínosy VR v oblasti medicíny. [28]



Obr. 1.11: Významné přínosy v medicíně [28]

VR v medicíně přináší mnoho výhod. Zvýšení vzdělanosti v lékařském prostředí a jejího zrychlení bez strachu a případné újmy na zdraví, například simulace postupu provádění zákroku anebo příprava před složitou operací. Také napomáhá snižování nákladů na péči o pacienty. Navyšuje počet metod pro zrychlení rekonvalescence pacientů jako například rehabilitace po nehodě. Také použití VR napomáhá mentálnímu zdraví a onoho vybudování lékařů, například sebevědomí při provádění operací anebo diagnostiky onemocnění či postupu léčby.

Hlavní překážkou pro komerční využití v této oblasti je vysoká cena, která však s průběhem let postupně klesá. Vyžaduje rozsáhlou softwarovou a hardwarovou podporu a u některých pacientů může docházet k fyzickým nebo psychologickým vedlejším účinkům. Hlavně je omezena zejména pro simulaci a demonstraci, nelze s ní zatím použít pro samotnou identifikaci příznaků nových onemocnění.

Jak bylo popsáno dříve, VR má potenciál velmi přínosného uplatnění v mnoha odvětvích, zejména v rehabilitaci, vzdělávání, psychologické terapii a další. V dalších podkapitolách se budeme věnovat jednotlivým odvětvím a jejich využitím. [28]

1.3.1 Aplikace v rehabilitaci

Zdravotnický systém neslouží pouze k záchraně životů ohrožených pacientů. Též se snaží dodat péči, která pomáhá zajistit, aby lidé byli schopni se začlenit do svého kruhu nejbližších, zaměstnání, ale také vzdělání. Je snaha, aby jednotlivci nebyli izolováni od společnosti a prosazování vlastního systému, který by jim zmenšoval naději na plnohodnotný život po prodělání nějaké nečekané situace s následky. Tento lékařský obor se jmenuje *rehabilitace*.

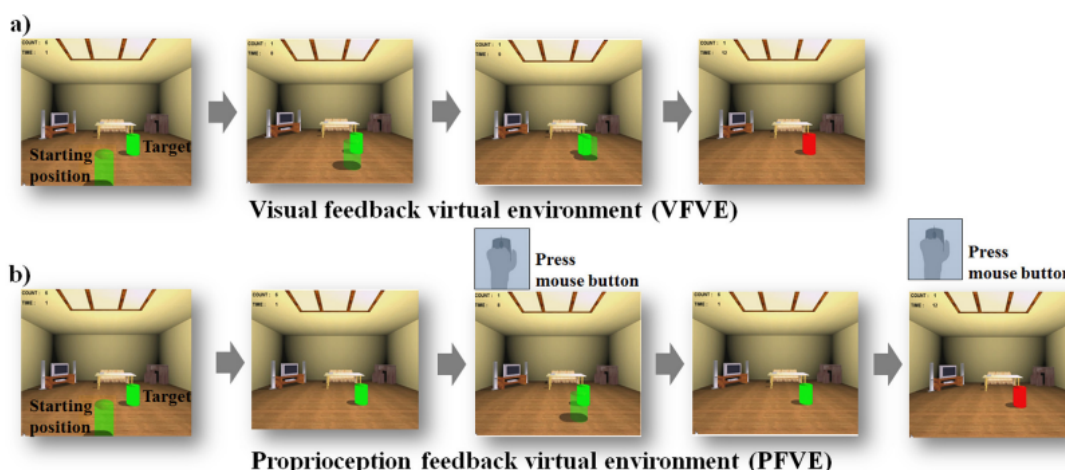
Uplatňuje se ve velkém spektru stavů a nemocí pacienta. Rehabilitace vykazuje pozitivní výsledky v celém spektru. Nejčastěji používané rehabilitace jsou například neurologické stavy (infarkt, motor-neurologické onemocnění), respirační stavy a také

muskuloskeletální poruchy (zlomeniny, subakutní anebo chronické bolesti části těla). [29]

Článek [30] se zabývá vyvinutí rehabilitačního systému pro pacienty po mozkové mrtvici. Využívají propriocepční zpětnou vazbu při rehabilitaci horních končetin za pomoci blokování vizuální zpětné vazby.

Propriocepce patří do *somatosenzorického systému* a je definována jako smyslové vjemy polohy, pohybu nebo rovnováhy. Tedy se jedná o vnímání či uvědomování si polohy částí těla vůči sobě v prostoru. K tvorbě dochází v důsledku reakcí recepto-rových systémů v tkáních lidského těla, nejčastěji ve svalech, ale také ve šlachách, kloubech i v kůži. Tento systém je rozhodující například v zajišťování koordinace pohybů anebo rovnováhy v klidu či pohybu. Z tohoto důvodu propriocepce zahrnuje dvě složky. První, která vzniká ve statické poloze - *statická propriocepce* a druhá se objevuje během pohybu - *dynamická propriocepce*. [31]

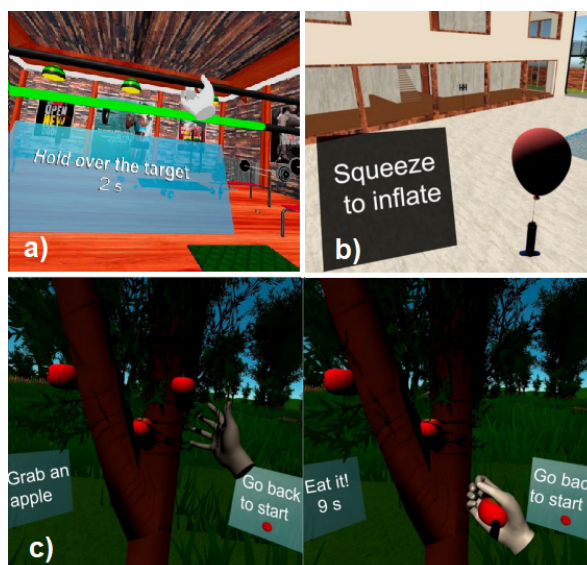
Virtuální prostředí navržené v článku [30] zapíná/vypíná vizuální zpětnou vazbu během tréninku horních končetin v simulovaném prostředí s úlohou běžného dne. Aplikace lze vidět na obrázku 1.12. Úloha se snaží, aby pacient přemístil objekt na konkrétní místo ve dvou fázích (zapnutá a vypnutá vizuální zpětná vazba). V zapnuté fázi pacient vidí semi-transparentní objekt a konkrétní místo, kde objekt musí být přemístěn. Tento objekt vidí po celou dobu. Na rozdíl ve vypnuté fázi objekt lze vidět jen v počátečním bodu a následně je jeho vzor v prostoru odebrán. Po celou dobu byla měřena vzdálenost mezi pozicí objektu a počátečního bodu a v době ukončení vzdálenost mezi pozicí objektu a koncového bodu.



Obr. 1.12: Vizuální přehled fází rehabilitačního systému: **a)** s zapnutou vizuální zpětnou vazbou; **b)** s vypnutou vizuální zpětnou vazbou [30]

Využití imerzivní virtuální reality v motorické rehabilitaci (horních končetin) popisuje článek [32]. Byla navržena aplikace tvořená ze tří her. Pacienti měli za

pomocí těchto her porovnat dva typy interakce s virtuálním prostředím - s ovladači a bez ovladači. Jednotlivé typy her lze vidět na obrázku 1.13. Všechny hry spočívaly, aby byly využity jiné části ruky.

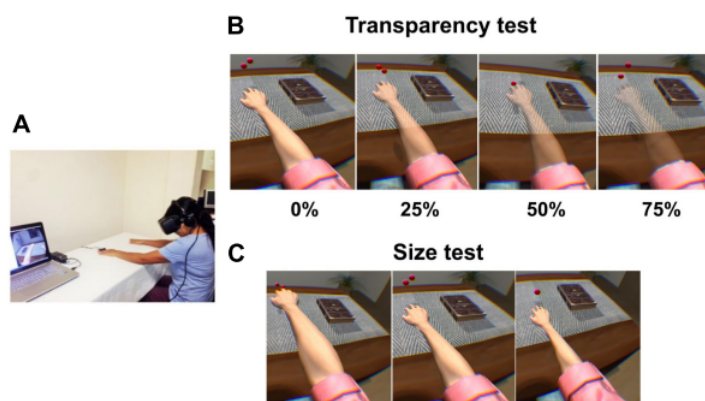


Obr. 1.13: Vizuální přehled jednotlivých her: **a)** cvičení - zvedání; **b)** cvičení - konzumace jablka; **c)** cvičení - otevírání a zavírání afektované ruky [32]

Dalším příkladem využití imerzivní virtuální reality je vyvolání iluze vlastnictví falešné končetiny. Tato iluze je intenzivně zkoumána a může mít klinické důsledky pro stavy pacienta, které zahrnují změněný obraz těla, jako je chronická bolest. [33] Tyto zkoumané možnosti jsou využívány v *neurorehabilitaci* v oblasti bolesti. Článek [34] využívá VR pro snížení bolesti u pacientů s chronickou bolestí paže prostřednictvím změny obrazu virtuálního těla. Pacient byl podroben dvěma experimentům: experiment změny průhlednosti končetiny a experiment změnou velikosti končetiny. Přehled experimentu lze vidět na 1.14. Z experimentů bylo zjištěno, že se správnou strategií (pro každou bolest se strategie liší) dochází až k polovičnímu snížení bolesti na dočasnou dobu.

1.3.2 Aplikace v psychologii

VR hraje důležitou roli v diagnostice a také v léčbě duševních poruch. Nabízí hlavní tři důležité výhody, které jiné technologie nevykazují a jsou výhodné pro psychologii. První výhodou lze nalézt v imerzi, která je předurčena pro zásahy v oblasti *expozice*. V *klasické expoziční terapii* mohou být pacienti konfrontováni se svou úzkostí a naučit se ji zmírnit. Lze vytvořit i situace, které by v reálném světě nešly realizovat jako balancování na laně mezi dvěma mrakodrapy. Druhá výhoda už byla zmíněna a to simulace reality s relevantními symptomy. Toto je velice vhodné pro vývoj



Obr. 1.14: Vizuální přehled experimentů iluze vlastnictví falešné ruky: **a)** pacient s HMD; **b)** změna průhlednosti končetiny; **c)** změna velikosti ruky [34]

nových testovacích prostředí, které jsou ekologicky platné a vysoce standardizované. Zatřetí VR působivě demonstruje, jak poddajný je často vlastní *mysl pro sebe* a *pocit reality*, zvláště pokud člověk trpí duševní chorobou. *Psychoedukce založená na zkušenostech* by mohla být přínosná u duševních poruch, u kterých je vlastní model sebe nebo model reality abnormálně změněn, a kde konvenční zásah selhává.

Virtuální realita našla uplatnění v klinické psychologii pro léčení mnoho duševních poruch jako deprese, úzkosti, ale také demence, poruchy příjmu potravy, schizofrenie a další. [35]

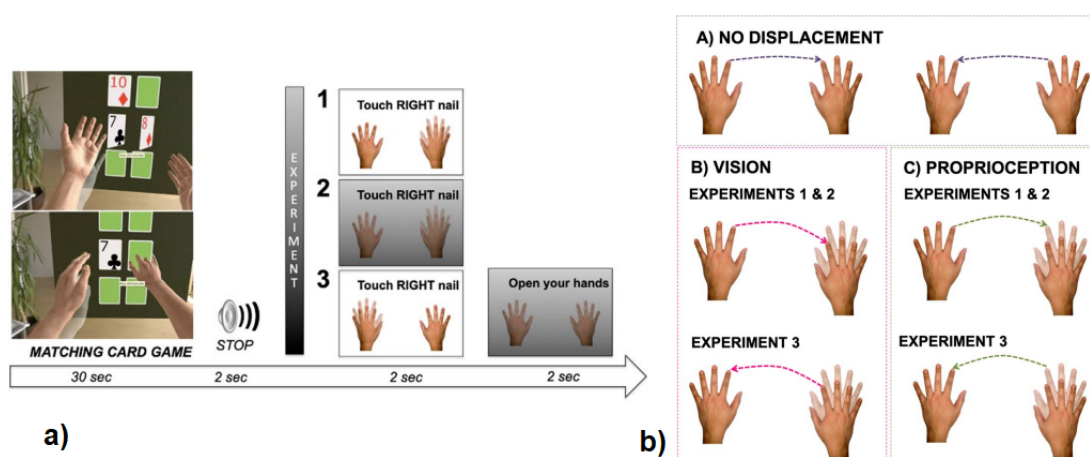
Pro *podporu klinického rozhodování* při hodnocení duševních poruch byla využita VR v článku [36]. Byla vytvořena simulace zaměřena na podmínky schizofrenie. Experiment se snaží z abnormálního chování (pohyby očí) rozhodnout o mentální poruše u pacienta. Pacient musel řešit specifické úlohy, kdy v průběhu je spuštěn audio záznam vytvořený z popisu od lidí trpících schizofrenií. Prostředí simulace je na obrázku 1.15 s úlohami na tabuli. V průběhu řešení byly získávány data pozicí a pohybů očí. Z klinických poznatků a získaných dat lze rozhodnout o jakou poruchu se jedná. Například člověk s depresí ztrácí zájem o plnění úloh, s schizofrenií mají změněné vnímání sebe sama, zvláštní chování a může pociťovat negativní či kognitivní příznaky.

V článku [37] bylo zaměřeno, jak posunutá umělá ruka v VR ovlivňuje akční výkon skutečné ruky. Tomuto jevu se říká *Rubber hand illusion* (RHI), která se používá k prozkoumání flexibility tělesného schématu - jak je systém citlivý na vizuální iluzi. Účastníci vykazují systematické zkreslení při dotázání o odhad pozice ruky, že zdánlivá poloha jejich ruky je posunuta směrem ke umělé ruce. Tomuto jevu se říká *proprioceptivní drift*, neboli je to nepřímá míra vlastnictví umělé ruky.

Byla vytvořena série experimentů na bázi hraní karetní hry. Experiment lze vi-



Obr. 1.15: Virtuální prostředí experimentu s doktorem a úlohy na řešení [36]



Obr. 1.16: Experiment pro změření RHI: **a)** průběh experimentu; **b)** očekávaná trajektorie pohybu ruky v VR pro řešení úloh [37]

děť na 1.16 za *a)*. Byly testovány měnící se prostorové a časové parametry, když účastník provádí pohyb. V experimentech se zapínal či vypínal vzhled umělé ruky pro odebrání zrakové zpětné vazby. Ve všech experimentech se zjišťovala vzdálenost driftu při vlivu zrakové zpětné vazby a proprioceptivní zpětné vazby. Na obrázku 1.16 za *b)* jsou způsoby, jak vypadá odhadovaná trajektorie pohybu ruky při jednotlivých experimentech. Ze získaných dat je naznačováno, že vizuální obraz umělé ruky prezentované ve VR ovlivňují schéma a akční výkon skutečné ruky.

S jinou strategií se zaměřuje článek [38], který se snaží o potlačení RHI. Vytváří matematický model, který umí predikovat aktivní komponentu RHI, tj. tendence vygenerovat síly směrem virtuální ruce. Vychází z principu *free energy principle*.

2 Návrh řešení

Cílem této práce je vyvinout novou aplikaci pro virtuální realitu v prostředí vývojové platformy Unity, která bude sloužit k léčbě poruch propiocepčního aparátu. Nová aplikace bude vytvořena na základě původní aplikace, která bohužel neumožňovala přístup ke svým zdrojovým souborům. Z tohoto důvodu bude nová aplikace vyvinuta zcela od začátku, přičemž původní aplikace poslouží jako vzor. Navíc bude nová aplikace rozšířena o nové funkce na základě specifických požadavků lékařů. Součástí práce bude také vytvoření experimentů pro ověření lokalizační schopnosti headsetu Oculus Quest 2 a jeho ovladačů, s cílem definovat standardizované podmínky pro zachování přesnosti. Data shromážděná z těchto experimentů budou analyzována pomocí statistických metod, aby poskytla relevantní informace a poznatky. Tyto výsledky následně poslouží pro další analýzy a aplikace, jak v případě nově vyvíjené aplikace, tak i v dalších odvětvích.

V první řadě se práce zaměří na vytvoření experimentů pro ověření lokalizace. Tyto experimenty budou rozděleny do dvou fází. V první fázi bude ověřena přesnost sledování objektu systémem Vicon Tracker za specificky definovanými podmínkami, aby bylo možné tuto měření opakovat. Následně bude Vicon Tracker použit k experimentu pro ověření lokalizace Oculus Quest 2. Pro tento účel bude opět vytvořeno standardizované prostředí, které umožní maximalizovat přesnost lokalizace. Pro sledování jednotlivých komponent Oculus Quest 2 bude nutné vytvořit modely s nasazenými sledovacími markery. V poslední fázi bude vyvinuta aplikace, která zaznamenává pozici a orientaci HMD i ovladačů a vytvoří virtuální prostředí ukazující uživateli instrukce pro provádění měření. Takto podrobné a vícefázové ověření přesnosti je klíčové pro zjištění co nejpřesnější míry chyby lokalizace headsetu Oculus Quest 2. Tento přístup umožňuje identifikovat a rozdělit jednotlivé dílčí chyby. Získané poznatky nebudou užitečné pouze pro tento konkrétní případ, ale mají potenciál přinést nové informace i do dalších oblastí medicíny, jelikož dosud neexistuje podobná studie, která by se zabývala ověřením přesnosti systémů VICON a Oculus.

Druhá část se zaměří na vytvoření nové aplikace od základů, s ohledem na požadavky lékařů. Bude zahrnuta funkce komunikace mezi herním prostředím Unity a virtuální realitou Oculus Quest 2, což umožní kompatibilitu i s dalšími verzemi Quest nebo jinými zařízeními. Aplikace bude také schopna získávat veškerá data z HMD a ovladačů, včetně pozice, rotace a rychlosti.

Po správném nastavení komunikace bude vytvořena scéna, která bude obsahovat stejné prvky jako původní aplikace. Hlavní scéna bude obsahovat dvě kamery – jednu na HMD a druhou na připojeném počítači, která bude sledovat trajektorii kamery HMD a zobrazovat pohled z HMD přímo na monitor počítače. Navrhnuta bude také místnost pro uživatele, s důrazem na vysokou úroveň imerze, včetně textur, osvětlení

a modelů objektů. Dále bude provedena kalibrace VR systému pro zajištění shody pozic objektů ve virtuálním a skutečném prostředí.

Ve vytvořené scéně proběhne interaktivní úkol, který pomůže uživateli orientovat se při manipulaci a řešení tří experimentů – Experiment 1, 2 a 3, které budou založeny na stejném principu jako původní aplikace, ale upraveny dle nových požadavků. Experiment 2 navíc umožní simulaci černého prostředí a přidání nových interakcí. Mezi experimenty budou vloženy distrakční úlohy (matematická a geometrická), které se budou střídat během řešení hlavních experimentů.

Budou vytvořeny dvě uživatelské rozhraní (UI), které budou viditelné pouze na monitoru počítače, nikoli v HMD. První UI se objeví na začátku aplikace a bude rozděleno na dvě části: první část bude určena pro vyplnění dotazníku o uživateli a druhá část pro nastavení parametrů experimentů a spuštění kalibrace. Informace z dotazníku a nastavené parametry budou uloženy do textového souboru. Druhé UI se zobrazí během provádění experimentu 2 a bude sloužit k poskytování informací o průběhu experimentu.

Jsou také definovány další nové požadavky, jako je vytvoření funkce pro odstranění aktuálního nebo předchozího pokusu v experimentu 2, která bude aktivována pomocí kláves. Po aktivaci se v druhém UI zobrazí informace o vykonané akci. Dalším požadavkem je nahrazení základního modelu ruky za realističtější model, včetně modelu ovladače VR, s animacemi interakce s tlačítky a joystickem ovladače, jako je dotyk, stisknutí nebo pohyb.

3 Ověření lokalizace Oculus Quest 2

Pro kvantifikaci poruch propriocepčního aparátu s využitím virtuální reality je nutné nejprve stanovit podmínky, které zaručí přesné měření. Je důležité určit, jak přesně Oculus Quest 2 detekuje polohu a orientaci headsetu a ovladačů.

Pro měření pozice Oculus Quest 2 byl použit Vicon Motion Tracker [39], který s vysokou přesností sleduje markery umístěné na monitorovaném objektu. Ačkoliv je Vicon známý svou přesností, je třeba tuto přesnost verifikovat prostřednictvím dalšího experimentu s použitím šestiosého EPSON C3 manipulátoru [40]. Tato kapitola se tedy zaměřuje na dva experimenty ověřující přesnost lokalizace Oculus Quest 2 a Vicon Trackeru.

3.1 Verifikace lokalizace Vicon Trackeru

Pro ověření přesnosti Vicon trackeru byl využit EPSON manipulátor. Manipulátor EPSON je řádově přesnější než VICON tracker a z něho změřená data můžou sloužit jako referenční. Následující podkapitoly popisují návrh experimentu, definici standardních podmínek, předzpracování naměřených hodnot a prezentaci výsledků.

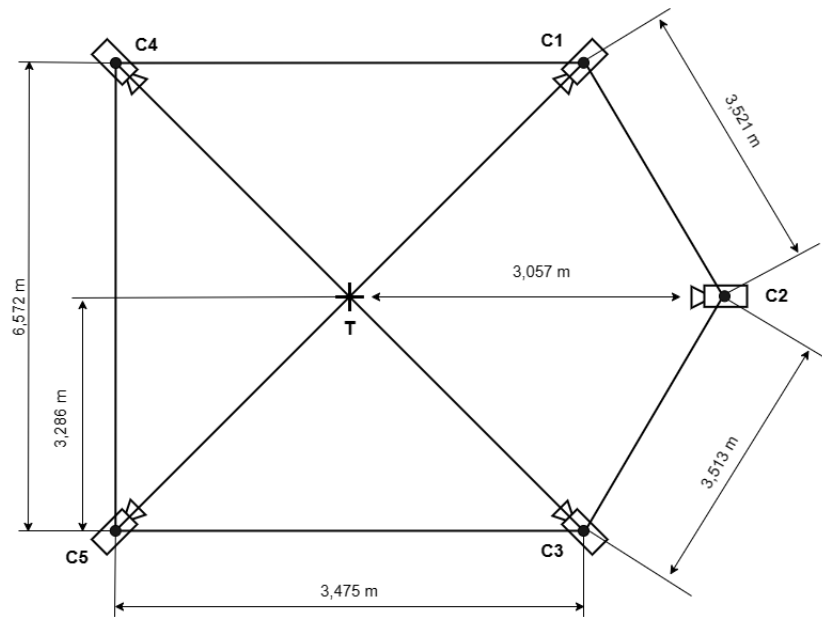
3.1.1 Návrh experimentu

Na rameno manipulátoru byla připevněna deska se čtyřmi markery, která je během experimentu sledována. Manipulátor provádí dva druhy pohybu - translační a rotační pohyb, přičemž každý typ pohybu je opakován čtyřikrát v alternující posloupnosti. Trajektorie byla vygenerována za pomoci MATLAB ve formátu příkazů pro manipulátor. Tento skript byl vytvořen vedoucím práce z důvodu bezpečnosti trajektorie při pohybu manipulátoru.

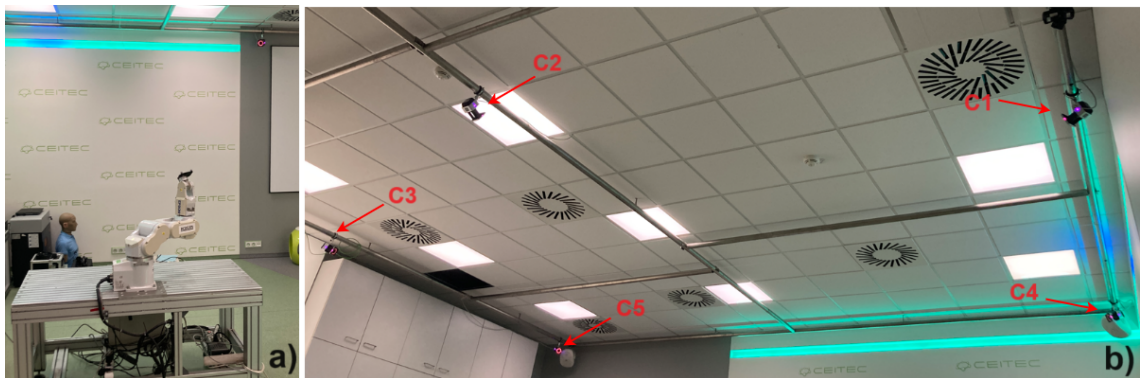
Před zahájením experimentu je nutné definovat prostředí, včetně umístění kamer Viconu a polohy manipulátoru. Rozmístění kamer je ilustrováno na obrázku 3.1. Bod T označuje počátek souřadnicové osy manipulátoru, výška středu je $h = 2,873\text{ m}$. Zrekonstruovaná scéna je zobrazena na obrázku 3.2.

Nastavení Vicon trackeru

Software Vicon Tracker (verze 3.10), používaný pro konfiguraci a měření, vyžaduje před každým měřením nastavení v pěti krocích. Nejdříve byla vytvořena maska k eliminaci lesklých objektů, které by mohly vést k nesprávné lokalizaci. Poté byly všechny kamery kalibrovány pomocí *wand* pohybováním v prostoru zorného pole kamer. Po kalibraci kamer byl *wand* umístěn do středu T k určení *originu*.

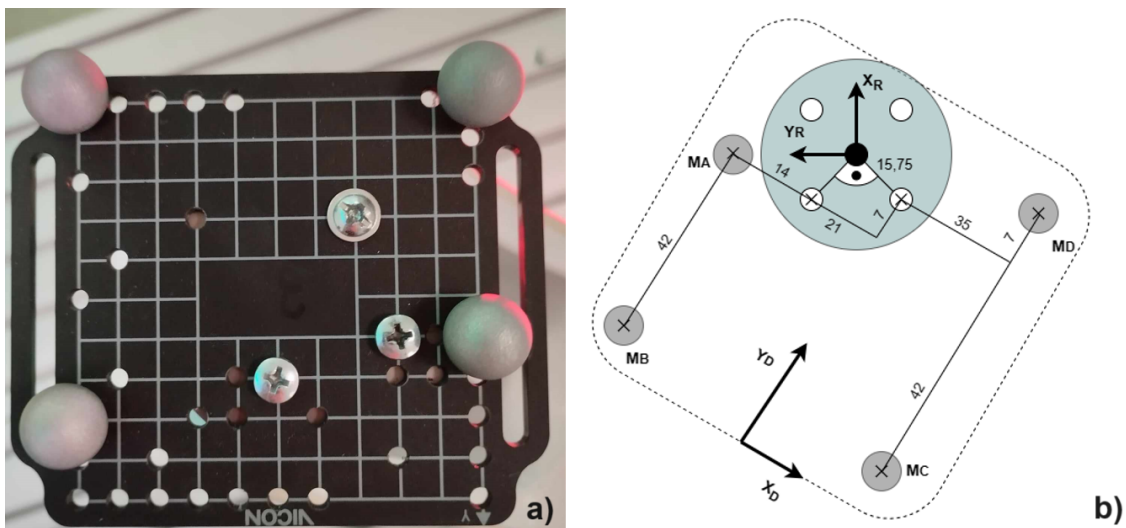


Obr. 3.1: Návrh rozmístění kamer Vicon Trackeru a definování středu T



Obr. 3.2: Konstrukce scény: a) manipulátor b) rozložení kamer

Objekt je definován zakliknutí markerů, které jsou v dané chvíli viditelné. Po zadání markerů do systému dochází k vytvoření objektu, jehož pozice a orientace jsou určeny v těžišti tohoto objektu. Tyto údaje jsou výchozí a vypočteny pomocí software Vicon, avšak neodpovídají přesně pozici a orientaci, které zobrazuje manipulátor. Pozice a orientace jsou vypočítány podle obrázku 3.3. Na tomto obrázku v části b) je znázorněn problém, kde je manipulátor vyznačen světle modrou barvou a markery šedě M_A, M_B, M_C, M_D . Diagram je doplněn o veškeré dostupné údaje o rozměrech. Informace o velikosti destičky a o rastru byly získány z dokumentace Viconu. Pozice závitů na manipulátoru byly získány z dokumentace manipulátoru EPSON. Ještě je potřeba definovat výšku $h = 97,5 \text{ mm}$, což představuje součet poloměru kuličky a výšky destičky.



Obr. 3.3: Sledovaný objekt - deska s markery: **a)** připevněna deska na manipulátoru **b)** náčrtek desky připevněné na manipulátoru (světlemodrá)

Pomocí geometrických a goniometrických výpočtů jsou určeny pozice, které jsou aplikovány na jednotlivé markery. Tímto postupem je možné získat přesnou celkovou pozici a orientaci objektu. Nejprve se předpokládá, že souřadnicové osy budou otočeny o -90 stupňů, což zjednoduší přehlednost výpočtů. Dále je nutné vypočítat posuny a na ose X_D desky a b v ose Y_D .

$$a = 15,75 \cdot \sin \left(45 + \tan \left(\frac{7}{21} \right) \right) = -14,08 \text{ mm} \quad (3.1)$$

$$b = 15,75 \cdot \cos \left(45 + \tan \left(\frac{7}{21} \right) \right) = 7,04 \text{ mm} \quad (3.2)$$

Pomocí těchto posunů a rastru destičky je možné vypočítat pozice markerů pro danou souřadnicovou osu:

$$\begin{bmatrix} M_A \\ M_B \\ M_C \\ M_D \end{bmatrix} = \begin{bmatrix} x_A & y_A \\ x_B & y_B \\ x_C & y_C \\ x_D & y_D \end{bmatrix} = \begin{bmatrix} a & b + 14 \\ a - 42 & b + 14 \\ a - 42 & b - 56 \\ a - 14 & b - 56 \end{bmatrix} = \begin{bmatrix} -14,08 & 21,04 \\ -56,08 & 21,04 \\ -56,08 & -48,96 \\ -0,08 & -48,96 \end{bmatrix} \quad (3.3)$$

Všechny hodnoty jsou uvedeny v milimetrech. Dále je třeba tyto pozice otočit kolem osy z o úhel θ , který je vypočítán podle následujícího vzorce:

$$\theta = -\tan \left(\frac{7}{21} \right) - 90 = -109,47^\circ \quad (3.4)$$

Rotace je provedena pomocí rotační matice pro všechny pozice markerů, jak je uvedeno v 3.5. Výsledné hodnoty jsou následně zavedeny do softwaru Vicon.

$$\begin{bmatrix} M'_A \\ M'_B \\ M'_C \\ M'_D \end{bmatrix} = \begin{bmatrix} M_A \\ M_B \\ M_C \\ M_D \end{bmatrix} \cdot \mathbf{R}(\theta)^T = \begin{bmatrix} -6,71 & 24,41 \\ -46,55 & 37,69 \\ -68,68 & -28,72 \\ -15,55 & -46,42 \end{bmatrix} \quad (3.5)$$

Měření experimentu

Manipulátor provádí pohyby dle trajektorie generované skriptem, přičemž na každé definované pozici se zastaví na 0,5 s. Tento proces se opakuje u všech předem stanovených pozic.

Měření bylo provedeno dvakrát při zopakování kalibrace Viconu, aby byla vyhodnocena také chyba kalibrace systému.

Data o pozici a orientaci se získávají z manipulátoru a pomocí systému Vicon. Vicon standardně exportuje data rotace v *helikální reprezentaci*, avšak pro další zpracování bude využíváno *kvaternionů*. Protože nelze simultánně spustit sledování systémem Vicon a pohyb manipulátoru, a je třeba sjednotit data z obou zdrojů, je nezbytné data předem zpracovat.

3.1.2 Předzpracování výsledků

Data získaná z Viconu a manipulátoru je nutné před vyhodnocením předzpracovat. K tomuto účelu byl použit software MATLAB. Předzpracování zahrnuje synchronizaci dat a kalibraci.

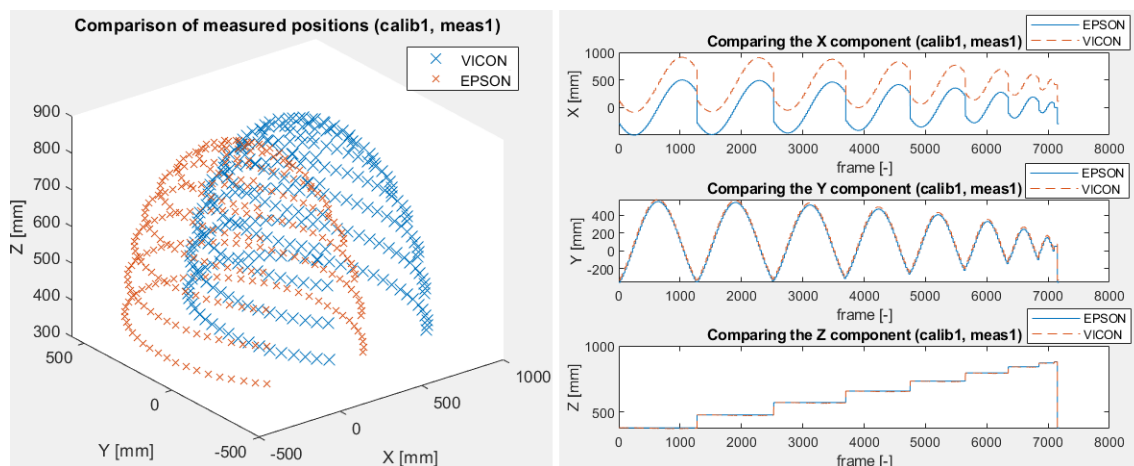
Synchronizace dat

Díky specifickému způsobu generování dat, kdy manipulátor vždy před vykonáním nového příkazu čekal 0.5 s, byla implementace synchronizace zjednodušena. Manuálně byl určen bod, kde začíná sledování trajektorie, tedy počátek intervalu čekání manipulátoru v první póze. Od tohoto bodu byl proveden posun o polovinu čekacího intervalu, aby se získala poloha uprostřed čekací doby. Následně byla z dat Viconu odstraněna data z části intervalu odpovídající čtvrtině vzdálenosti čekací doby od středu tohoto intervalu. Z této střední poloviny se pak posunulo o dobu pohybu manipulátoru (zahrnutou v exportovaném souboru) a 0,25 s, aby se dosáhlo následující pozice trajektorie. Tento proces byl opakován iterativně až do konce dílčího měření.

Celé měření probíhalo nepřetržitě a jeden dataset obsahuje osm jednotlivých měření při stejné kalibraci Viconu, což celkově trvalo přibližně 40 minut. To vedlo k mírnému zpoždění dat v dalších měřeních. V tomto případě to nebylo příliš problematické, jelikož byly sledovány pouze pozice, ve kterých se manipulátor nehýbal. Avšak bylo zjištěno, že obecný iterativní algoritmus by pro tento případ nebyl

vhodný. Pro zjednodušení byla proto vždy manuálně vyčtena počáteční data sledování jednotlivých dílčích měření a od těchto bodů byl spuštěn algoritmus.

Na obrázku, který je označen jako 3.4, je zobrazeno porovnání změřených pozic z prvního dílčího měření datasetu první kalibrace.

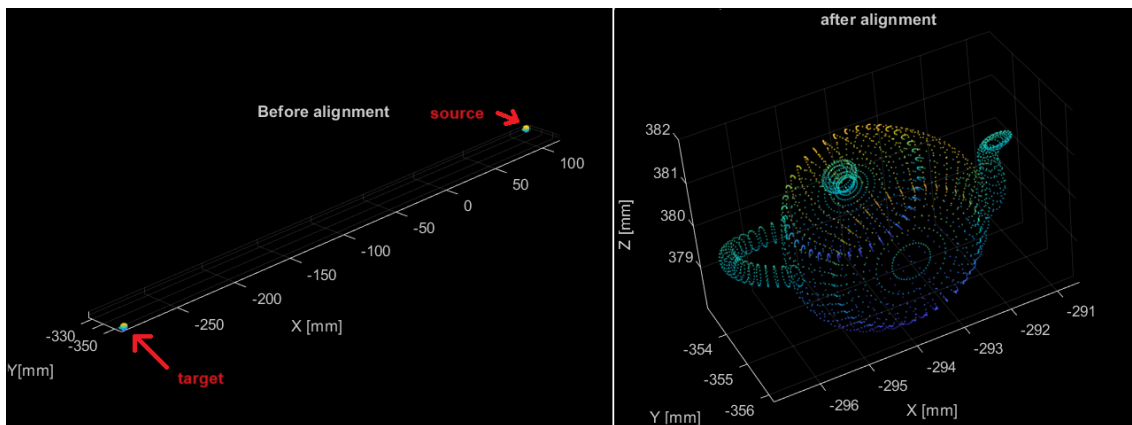


Obr. 3.4: Porovnání pozic pro měření prvního měření u první kalibrace Viconu - translační pohyb: Vlevo zobrazení pozic v 3D prostoru a vpravo závislost jednotlivých souřadnic na frame

Kalibrace dat

Na obrázku 3.4 je patrné, že zejména v ose X jsou hodnoty vzájemně posunuty a možná i rotovány. Tento jev je způsoben posunutím a rotací souřadných systémů vůči sobě. Aby bylo možné souřadné systémy sjednotit, je nutné vypočítat transformační matici. Tato matice umožní transformovat všechny body z jednoho souřadného systému do druhého. K výpočtu této matice byly použity první pozice a orientace změřené pomocí Vicon a EPSON. Po výpočtu transformační matice byla tato aplikována na všechna ostatní data. První pózy objektu, které byly použity pro výpočet, byly následně odebrány z datasetu.

Pro výpočet transformační matice byl použit *ICP algoritmus Point-to-Point*. Základní princip tohoto zarovnávacího algoritmu byl popsán v bakalářské práci [41]. Jako vstupní data pro tento algoritmus byly použity první pozice a orientace z EPSONu (target) a průměrné hodnoty prvních změřených pozic a orientací objektu z Viconu (source). Pro správné zarovnání za pomoci ICP algoritmu je nutné mít stejné modely. V tomto případě byl použit model *teapot* [42], na který byla aplikována translace a rotace, aby odpovídal změřené pozici a orientaci. Na obrázku 3.5 je možné vidět modely *teapot* před a po aplikaci zarovnání.



Obr. 3.5: Zarovnání modelů: (vlevo) před zarovnáním (vpravo) po zarovnání

Transformační matice získaná z ICP algoritmu je následně aplikována na všechna data změřená pomocí Viconu. Nová pozice je vypočítána pomocí maticového násobení podle vztahu 3.6. V tomto vztahu jsou transformační matice $\mathbf{T}$ a pozice $\mathbf{p}_{vicon}$ reprezentovány v homogenních souřadnicích, což umožňuje provádět translace a rotace v jediném kroku. Výsledkem maticového násobení jsou nové souřadnice, které přesně odpovídají cílovému souřadnému systému.

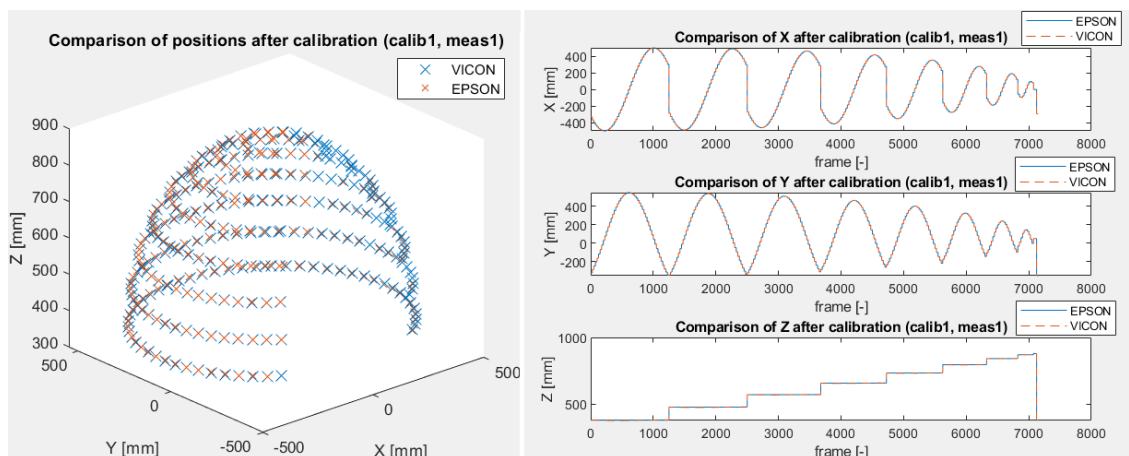
$$\mathbf{p}'_{vicon} = \mathbf{T} \cdot \mathbf{p}_{vicon} \quad (3.6)$$

Nová orientace objektu je vypočítána pomocí vztahu 3.7, kde na původní orientaci $\mathbf{R}_{vicon}$ je aplikována rotační matice $\mathbf{R}_T$ získaná z transformační matice. Tento proces zahrnuje maticové násobení rotačních matic, což umožňuje převést orientaci objektu z jednoho souřadného systému do druhého.

$$\mathbf{R}'_{vicon} = \mathbf{R}_T \cdot \mathbf{R}_{vicon} \quad (3.7)$$

Na obrázku 3.6 je zobrazeno porovnání pozic po aplikaci vypočtené transformační matice. Na grafech je patrné, že došlo ke korektnímu zarovnání. Toto ukazuje, že transformační matice správně převedla data Viconu do cílového souřadného systému, což potvrzuje správnost výpočtu a účinnost použitého algoritmu pro synchronizaci a kalibraci dat.

Posledním krokem bylo převedení rotací do Eulerových úhlů ve formátu ZYX a export datových sad do CSV souboru pro následnou analýzu.



Obr. 3.6: Porovnání pozice objektu po kalibraci: **(vlevo)** zobrazení pozic v 3D prostoru **(vpravo)** závislost jednotlivých složek pozice na framu

3.1.3 Výsledky měření

Analýza změřených dat nebyla přímou součástí tohoto zadání diplomové práce, ale byla provedena Mgr. Hynkem Cíglerem, Ph.D. z Fakulty sociálních studií Masarykovy univerzity v rámci širší spolupráce v CEITEC.

Z naměřených dat vyplývá, že systematická chyba v pozicích jednotlivých os byla minimální. Při úloze s rotačním pohybem byla pozice nižší o 0,59 mm v ose X, vyšší o 0,62 mm v ose Y a vyšší o 1,789 mm v ose Z. U úlohy s translačním pohybem byla chyba v ose X 1,048 mm, v ose Y -1,216 mm a v ose Z -1,469 mm.

Systematická chyba orientace objektu byla rovněž velmi nízká. U rotačního pohybu byla chyba $0,226^\circ$ v ose X, $0,201^\circ$ v ose Y a $0,566^\circ$ v ose Z. U translačního pohybu byla chyba v ose X $-0,019^\circ$, v ose Y $0,028^\circ$ a v ose Z $-0,128^\circ$.

Analýza dat pro pozice ukázala průměrnou hodnotu 1,509, přičemž variabilita měření byla vyjádřena standardní odchylkou 0,844. Přesnost odhadu průměru, určená standardní chybou, dosáhla 0,003. Co se týče dat rotací, průměrná hodnota byla 0,430, se směrodatnou odchylkou 0,381 a standardní chybou 0,001.

Z výsledků je zřejmé, že Vicon Tracker vykazuje velmi nízkou chybu při sledování lokalizačních markerů po celou dobu měření, aniž by došlo ke zhoršení přesnosti v průběhu času. Na základě těchto poznatků se Vicon Tracker jeví jako jasná volba pro použití jako referenční systém při měření přesnosti lokalizace Oculus Quest 2.

3.2 Experiment ověření Oculus Quest 2

Po ověření lokalizace pomocí Vicon Trackeru můžeme přistoupit k ověřování přesnosti lokalizace headsetu Oculus Quest 2 a sledování jeho ovladačů. V této pod-

kapitole se zaměří na návrh experimentu. Začne se vytvořením standardizovaného prostředí a poté návrhem aplikace v Unity [43]. Tato aplikace bude využita během měření jako klíčová součást procedury experimentu.

3.2.1 Návrh standardizovaného prostředí

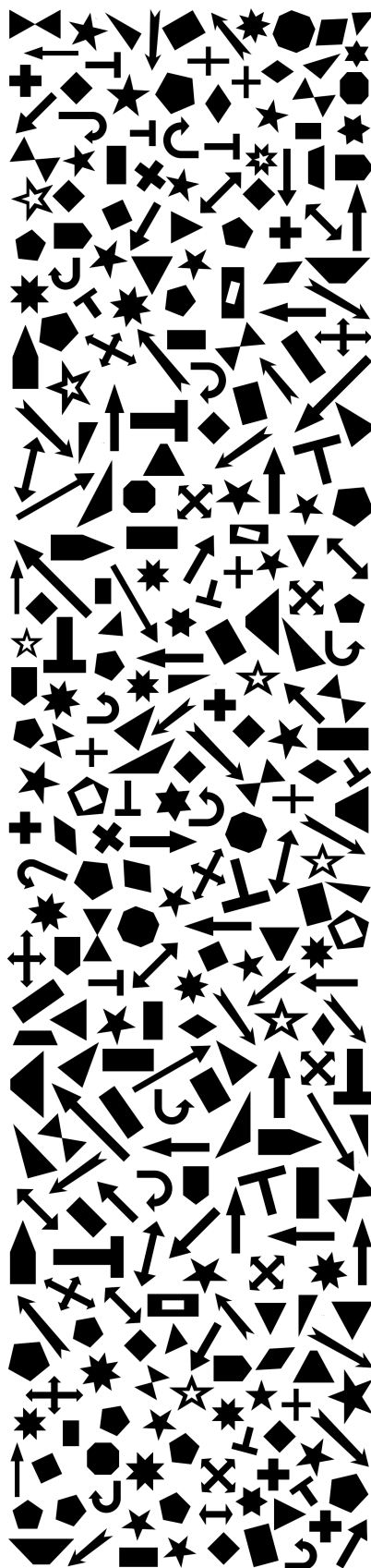
Prostředí vytvořené pro první experiment, jak je popsáno v kapitole 3.1.1, bylo zde znovu využito. Na rozdíl od dříve zmíněného manipulátoru, tentokrát v bodě T bude stát člověk s nasazeným HMD.

Je nutné vytvořit prostředí, které zajistí spolehlivou lokalizaci Oculus Quest 2. Podle teoretického rozboru 1.1.1, Oculus Quest 2 pro svoji lokalizaci využívá technologii Visual SLAM. Nejsou však známy podrobnosti o tom, jak přesně tento SLAM funguje. Z analýzy ostatních vizuálních SLAM systémů vyplývá, že ideální prostředí pro maximalizaci přesnosti a spolehlivosti by mělo obsahovat jasně rozpoznatelné, vysoce kontrastní a stabilní orientační body. Na základě těchto kritérií bylo navrženo plátno 3.8, které bylo upevněno na stativy a umístěno pod kameru $C2$. Vizualizaci konstrukce plátna lze nalézt na obrázku 3.7.

Plátno je tvořeno černými geometrickými útvary na bílém pozadí, což zajišťuje vysoký kontrast. Útvary jsou různorodé a rozmístěny i orientovány náhodně, aby odpovídaly požadavku na nehomogenní prostředí a zvýšily pravděpodobnost rozpoznatelnosti klíčových bodů (nikde v obraze nejsou dva zcela identické objekty). Pro zajištění fyzikální stability a efektivní detekci pevných orientačních bodů byly zvoleny tvary obsahující hrany a především rohy, které mají menší tendenci k deformaci a jsou detektory významných bodů považovány za stabilní.



Obr. 3.7: Konstrukce plátna

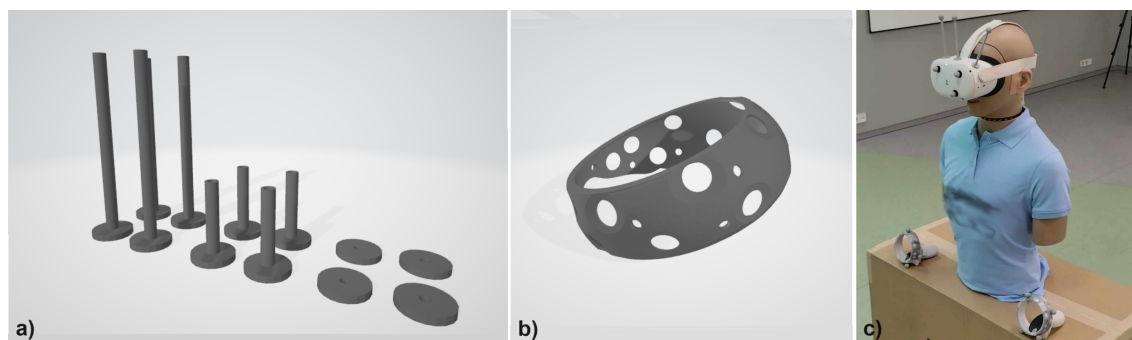


Obr. 3.8: Návrh plátna optimálního prostředí pro lokalizaci HMD

3.2.2 Držáky lokalizačních markerů pro OQ2

Pro možnost sledování HMD a ovladačů bylo nezbytné navrhnout držáky, do kterých by byly integrovány markery pro Vicon. Tyto držáky byly vyrobeny pomocí 3D tisku. Na obrázku 3.9a jsou zobrazeny modely stojánků, které byly přilepeny na HMD. Obrázek 3.9b ukazuje model pravého ovladače, který byl převzat z [44] a upraven tak, aby umožnil instalaci markerů.

Na obrázku 3.9c jsou zobrazeny modely s osazenými markery. Během testování se ukázalo, že pouhé nehomogenní umístění markerů na ploše ovladačů není dostatečné pro splnění podmínky asymetrie, která je klíčová pro správnou lokalizaci. Aby se tento problém vyřešil, byly použity modely stojánků určené pro HMD. Trupy těchto stojánků s různými délkami byly přilepeny na povrch modelů ovladačů, čímž byla dosažena požadovaná asymetrie.



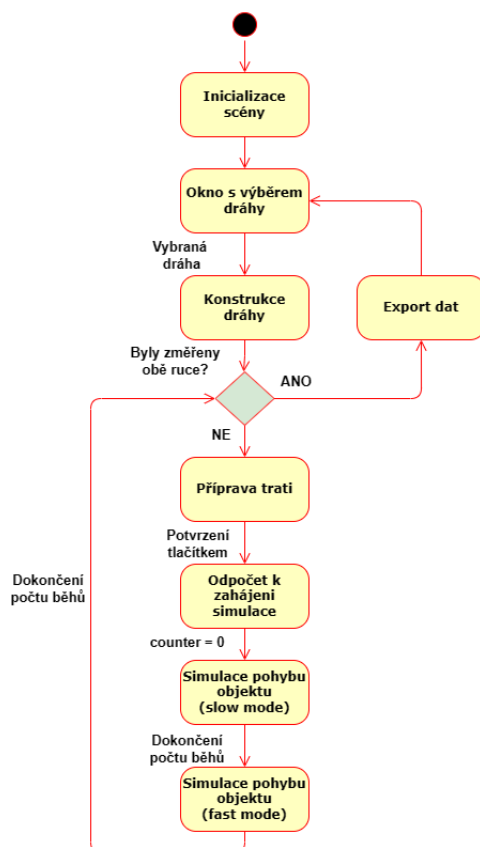
Obr. 3.9: Návrh modelů: **a)** modely pro HMD; **b)** model pro pravý ovladač; **c)** realizace modelů

3.2.3 Aplikace pro získávání dat při měření

Je nutné získat data o poloze a orientaci jednotlivých prvků Oculus Quest 2, a zároveň definovat průběh měření. K tomuto účelu byla vytvořena VR aplikace pomocí Unity, která generuje tři typy tratí s objektem, jež se po nich pohybuje. Úkolem uživatele je tento pohyb objektu sledovat a obkreslovat pomocí jednoho ovladače.

Konfigurace aplikace a způsob získávání dat z HMD jsou podrobněji popsány v kapitole 4. Hlavní funkcionality aplikace je postavena na stavovém automatu, zobrazeném na obrázku 3.10. Po spuštění aplikace dojde k inicializaci a sestavení výchozí scény, která zahrnuje hlavní panel před uživatelem. Panel nabízí výběr ze tří druhů tratí a možnost volby, zda se má simulovaný objekt na trati pohybovat plynule, nebo má jeho pohyb být přerušován s jednosekundovými pauzami. Po výběru trati uživatelem se vygeneruje dráha s objektem, který čeká, až bude uživatel připraven a stiskne libovolné tlačítko na ovladači. Následuje odpočet ke zahájení simulace

objektu, který se začne pohybovat po vybrané trati po skončení odpočtu. Úkolem uživatele je co nejvěrněji kopírovat trajektorii pohybujícího se objektu. Simulace probíhá ve dvou fázích: v první se objekt pohybuje pomaleji a provádí se dvakrát, zatímco ve druhé fázi se objekt pohybuje rychleji a uživatel ji musí absolvovat desetkrát. Po dokončení obou simulací pro jednu ruku se stejný postup opakuje pro druhou ruku. Po absolvování simulací oběma rukama jsou exportována všechna data o pozicích a orientacích HMD a ovladačů. Uživateli je následně nabídnuta možnost pokračovat v dalších tratích bez nutnosti restartovat aplikaci.



Obr. 3.10: Stavový diagram aplikace pro experiment 2

Generování dráhy

Jak již bylo zmíněno, aplikace nabízí tři druhy tratí: přímku, sférickou spirálu a lemniskátu. Pro generování a simulaci pohybu objektu po těchto drahách je nutné použít parametrické rovnice funkcí pro jednotlivé trasy.

První dráha, přímka, je definována rovnicí v 3.8. Tato dráha byla zahrnuta na žádost lékařů, aby simulovala nejběžnější pohyb ruky v rámci aplikace pro analýzu propriocepce, jak je popsáno v kapitole 4. Parametry Δx , Δy a Δz představují rozdíly v pozicích mezi generovanou dráhou a pozicí HMD. Tyto parametry umožňují

nastavit výškové posunutí dráhy tak, aby odpovídalo výšce uživatele a dráha nebyla příliš vysoko nebo nízko vzhledem k jeho postavení. Tento přístup zajišťuje, že simulace bude pro každého uživatele ergonomicky přizpůsobená.

$$\begin{cases} x(t) = t + \Delta x \\ y(t) = -0,4 + \Delta y \\ z(t) = 0,5 + \Delta z, \quad t \in [-0,7; 0,7] \end{cases} \quad (3.8)$$

Druhá trať, kterou si uživatel může vybrat, je sférická spirála. Tato dráha byla zařazena do výběru, aby uživatelé nezůstávali staticky na jednom místě, ale aktivně se pohybovali v prostoru, což přispívá k lepšímu mapování a získání dat o lokalizaci HMD. Parametrická rovnice pro tuto dráhu je specifikována v 3.9, kde poloměr $r = 0,6$, počet otáčení $n = 8$ a ofsety $o_y = 0,4$; $o_z = 0,8$.

$$\begin{cases} x(t) = r \sin(t) \cos(nt) + \Delta x \\ y(t) = r \cos(t) + o_y + \Delta y \\ z(t) = r \sin(t) \sin(nt) + o_z + \Delta z, \quad t \in [0; \frac{\pi}{2}] \end{cases} \quad (3.9)$$

Poslední dostupnou tratí je lemniskáta, která je definována parametrickou rovnicí uvedenou v 3.10 s měřítkem $s = 1.5$. U této dráhy se kromě sledování pohybu zaměřujeme také na opisování rotace objektu, jak je podrobněji popsáno v kapitole o realizaci pohybu.

$$\begin{cases} x(t) = \frac{1}{s} \frac{\cos(t)}{1+\sin^2(t)} + \Delta x \\ y(t) = \frac{1}{s} \frac{\cos(t) \sin(t)}{1+\sin^2(t)} + \Delta y \\ z(t) = 0,5 + \Delta z, \quad t \in [0; 2\pi] \end{cases} \quad (3.10)$$

Na obrázku 3.11 jsou zobrazeny všechny dráhy, jak je vidí uživatel v aplikaci. Tyto dráhy byly vytvořeny pomocí *LineRenderer*, který realizuje spojování bodů dodaných funkcí. Detailnější informace o *LineRenderer* jsou popsány v bakalářské práci [41].



Obr. 3.11: Druhy tratí z pohledu uživatele: **a)** Přímka; **b)** Sférická spirála ; **c)** lemniskáta

Simulace pohybu sledovaného objektu

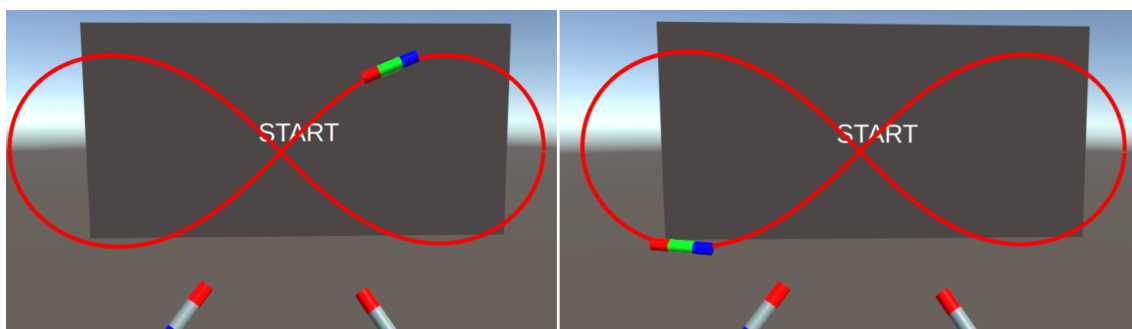
Aby bylo uživateli jasné, jak se mají pohybovat nebo jakou rychlostí, byl do simulace přidán sledovaný objekt. Tento objekt má za úkol simulovat pohyb, který uživatelé musí co nejpřesněji kopírovat. Pro intuitivní rozpoznání je model objektu stejný jako modely ovladačů, avšak jeho trup je zvýrazněn, aby nedošlo k záměně s jinými modely během měření.

Modelu je v každém framu přidělena nová pozice, která je určena pomocí dříve zmíněných parametrických rovnic 3.8, 3.9 a 3.10. Hodnota t_f pro každý frame je vypočítána následovně:

$$t_f = \Delta t \cdot v + t_{f-1}, \quad (3.11)$$

Hodnota t_f pro každý frame se vypočítává podle času t_{f-1} v předchozím framu, přičemž Δt reprezentuje časový rozdíl mezi posledním a aktuálním snímkem a v je rychlost pohybu objektu. Pro simulaci rychlejšího pohybu je nastavena rychlost $v = 0,5$ a pro pomalejší pohyb $v = 0,2$.

U tratí přímka a sférická spirála dochází pouze ke změně polohy, ne k změně orientace objektu. Naproti tomu u dráhy lemniskáty dochází k rotaci objektu, tak aby jeho orientace v každém framu odpovídala směru pohybu podél trajektorie. Tato rotace se realizuje výpočtem úhlu mezi vektorem tečny k trajektorii a vektorem určujícím orientaci objektu. Na obrázku 3.12 je zobrazen průběh simulace pohybu objektu pro třetí dráhu lemniskátu.



Obr. 3.12: Simulace pohybu objektu u třetí dráhy - lemniskáta. Dochází ke změně pozice i orientace

3.2.4 Postup měření experimentu

Postup měření v tomto experimentu je podobný jako při ověřování lokalizace pomocí Viconu. Vicon je předem zkalibrován, a pro každé ze tří zařízení Oculus jsou vytvořeny odpovídající objekty. Vzniká však problém, jelikož není přesně známo, kde

Oculus měří pozici a orientaci. Tento nedostatek byl řešen experimentálním nastavením lokálních souřadnicových systémů v Unity. Tyto systémy byly pootočený tak, aby při orientaci $rot = (0, 0, 0)^\circ$ v Unity odpovídala rotace ve Viconu také nulové hodnotě. V Unity pak nulová rotace lokálního souřadnicového systému koresponduje s globálním souřadnicovým systémem, což umožňuje srovnání i u dat získanými ve Viconu. Kalibrace pozice byla provedena za pomoci funkce *obnovení lokalizace* v bodě T s cílem analyzovat i chybu kalibrace Oculus Quest 2.

Po dokončení kalibrace a nastavení systému je spuštěna aplikace a zahájeno nahrávání dat. Následně jsou postupně provedeny všechny tři dráhy, aby bylo možné analyzovat výsledky a získat potřebné informace o přesnosti a spolehlivosti lokalizace zařízení Oculus Quest 2.

3.2.5 Předzpracování změřených dat

Po získání dat je nezbytné tyto data předzpracovat. To zahrnuje změnu orientace os, synchronizaci a kalibraci. Data z Unity jsou zaznamenána v levotočivém souřadnicovém systému, zatímco Vicon používá systém pravotočivý. Aby bylo možné data sjednotit, je potřeba u pozic VR prohodit osy Y a Z.

Podobná úprava se týká i orientací objektů, která se provádí přehozením komplexních složek kvaternionů a inverzí jejich skalární složky, jak je specifikováno v [45]. Provedená transformace kvaternionů je následující:

$$q = (w, x, y, z) \rightarrow q' = (-w, x, z, y) \quad (3.12)$$

Synchronizace datasetů

Aplikace načítá měřená data z zařízení v každém framu, což znamená, že vzorkovací frekvence je přímo úměrná *snímkové frekvenci* (Frame Rate) – tedy kolikrát dojde k průchodu hlavní smyčkou za jednotku času. Tato frekvence však závisí na řadě faktorů, jako je složitost aplikace, úroveň optimalizace, a také na použitém hardwaru a operačním systému. V průběhu měření se snímková frekvence udržovala v rozmezí 40 až 50 *fps*. Naproti tomu systém Vicon má vzorkovací frekvenci 100 *Hz*.

Aby bylo možné dosáhnout synchronizace mezi oběma datovými sadami, byl nejprve identifikován počáteční bod obou trajektorií. Poté byla na data z Viconu aplikována interpolace, aby se dosáhlo souladu s nižší snímkovou frekvencí získanou z VR.

Kalibrace dat

Stejně jako v prvním experimentu byl použit ICP algoritmus pro zarovnání prvních bodů a nalezení transformační matice. Bohužel bylo zjištěno, že je nutné mít podobné korespondující pozice a orientace, aby bylo možné určit správnou transformační matici. Zde se objevuje několik problémů, jako je například samotná interpolace nebo výběr metody synchronizace. Možným řešením tohoto problému by mohlo být vytvoření nové trati, na které by se měřily komponenty v klidovém stavu, tedy když jsou komponenty položeny na povrchu, nebo před začátkem a na konci experimentu umístit komponenty na nějakou plochu. Byla také vyvinuta možnost s pauzami, která by tento problém mohla vyřešit, avšak tato metoda nebyla otestována.

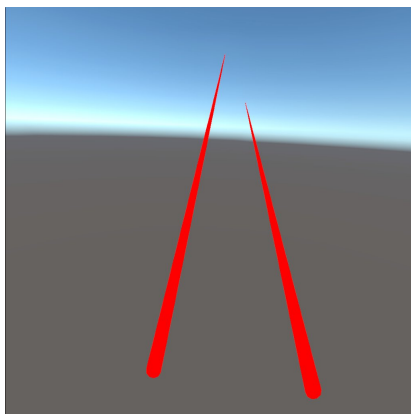
4 Aplikace pro analýzu propriocepce ve VR

Tato kapitola se bude zabývat implementací nové aplikace pro analýzu propriocepce ve virtuální realitě. Aplikace vychází z obdobné instalace aplikace, která neobsahovala možnost přístupu k jejím zdrojovým souborům. To znamená, že nová aplikace byla vytvořena zcela od počátku. Dále byly přidány nové požadavky a funkce. Tyto požadavky na aplikaci byly popsány v kapitole 2. Porovnání nové a původní aplikace lze nalézt na konci této kapitoly.

4.1 Propojení VR a Unity

Herní prostředí Unity nabízí minimálně tři *pluginy*, které umožňují vytvořit funkcionalitu VR s editorem Unity. Nejčastěji používané jsou *Oculus*, *OpenXR* a *Mock HMD*. Oculus [46] poskytuje řadu pomocných *assetů*, jako jsou modely rukou s animacemi a fyzika rukou. Při implementaci bylo zjištěno, že tento plugin obsahuje mnoho chyb kompatibility v Editoru, když se nejedná o Android Standalone aplikaci¹. Proto byla celá aplikace vytvořena pomocí OpenXR [47], který nabízí pouze základní funkcionalitu. To zahrnuje pouze *XR origin* (střed sledovacího prostoru ve scéně XR) se základním skriptem pro získávání dat o pozici a rotaci HMD a ovladačů, které jsou zvýrazněny jednoduchým paprskem. Počáteční pohled scény lze vidět na 4.1.

Hlavní výhodou OpenXR je jeho univerzálnost, díky které aplikace funguje i na ostatních VR zařízeních. Na rozdíl od Oculus pluginu, který je speciálně vyvinut pro zařízení Meta (Oculus Quest, Rift S, . . .), je OpenXR vhodnější pro aplikace běžící na PC a umožňuje realizovat komunikaci s HMD.



Obr. 4.1: Pohled HMD na scénu a zvýrazněné ovladače za pomoci paprsků

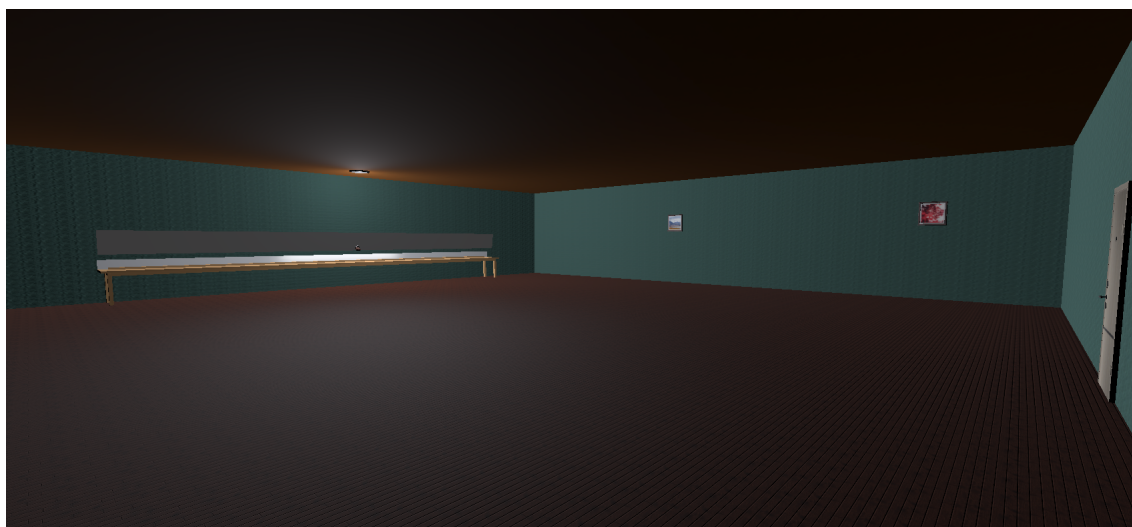
¹Android aplikace běžící přímo na HMD

Aplikace byla koncipována tak, aby byla spouštěna na PC připojeném k HMD. Tato konfigurace poskytuje výhodu dvou vstupních kanálů – z počítače a z HMD. Uživatel přítomný u PC má možnost zasahovat do průběhu experimentu nebo jej monitorovat. Je možné provádět zásahy tak, aby si jich uživatel s nasazeným brýlemi nevšiml.

4.2 Návrh scény

Scéna byla navržena podle předchozí aplikace. Konečný návrh místnosti je možné zhlédnout na obrázku 4.2. Jedná se o uzavřený prostor obsahující dvě hlavní objekty – *stůl* a *černou tabuli* – u kterých uživatel bude řešit úkoly. Místnost a uvedené objekty byly vytvořeny skládáním primitivních tvarů (krychle, sféry a další), které jsou k dispozici v Unity. Pro barevné provedení tabulí bylo využito jednoduchých materiálů. Proces tvorby objektů a materiálů je podrobně popsán v bakalářské práci [41].

Pro dosažení realističtějšího vzhledu místnosti byly použity texturey pro zdi a podlahu [48][49]. Dále byly použity modely obrazů [50] a dveří [51], které doplňují celkovou estetiku prostoru.



Obr. 4.2: Výsledná scéna místnosti

Dalším klíčovým prvkem návrhu je osvětlení místnosti. Bylo použito jedno hlavní bodové světlo, které bylo posunuto blíže k místu řešení úloh. Toto uspořádání vytváří dojem, že místo zájmu je nejvíce osvětleno, zatímco zbytek místnosti zůstává temnější díky definované intenzitě a dosahu světelného zdroje, jak je ilustrováno na obrázku.

Porovnání s původní aplikací z pohledu uživatele je uvedeno na konci kapitoly.

4.3 Realizace modelů rukou a jejich animací

Aby uživatel neřešil úlohy za pomoci červených paprsků, jak bylo ukázáno na 4.1, bylo nezbytné vytvořit modely rukou. Tyto modely musí splňovat několik kritérií: měly by uchopovat ovladače Oculus, vypadat realisticky (včetně barvy kůže a nehtů) a umožňovat realizaci pohybů při interakci s tlačítky nebo joystickem na ovladači. Postupná realizace těchto modelů bude podrobně popsána v následujících subkapitolách.

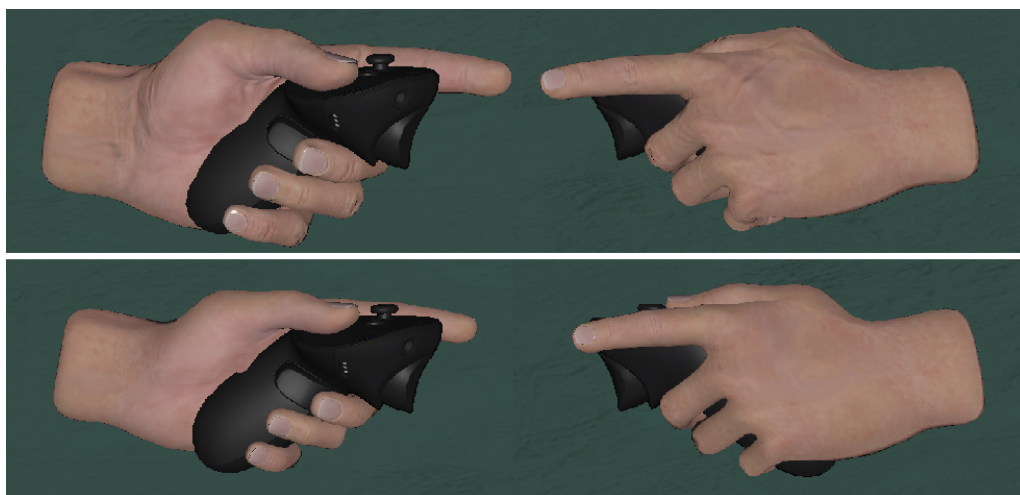
4.3.1 Tvorba uživatelových rukou

Modely ovladačů společně s texturami jsou přímo poskytovány společností Oculus [52], a pro modely rukou byl využit zdroj [53].

Všechny modely rukou využívají princip *kostry* (skeleton). Ruce jsou modelovány pomocí kostí a kloubů, podobně jako lidské ruce. Struktura kostry umožňuje definovat délku jednotlivých kostí, rotaci kloubů a další parametry, což umožňuje vytvořit libovolnou pózu reálné ruky.

Translace a rotace jednotlivých kloubů a kostí kostry je možné provádět v editoru Unity stejně jako s jakýmkoliv primitivním objektem. Výsledná póza levé mužské ruky je zobrazena na obrázku 4.3. Tento proces byl realizován pouze pro jednu ruku. Pravá ruka byla vytvořena jako zrcadlově otočená kopie levé ruky.

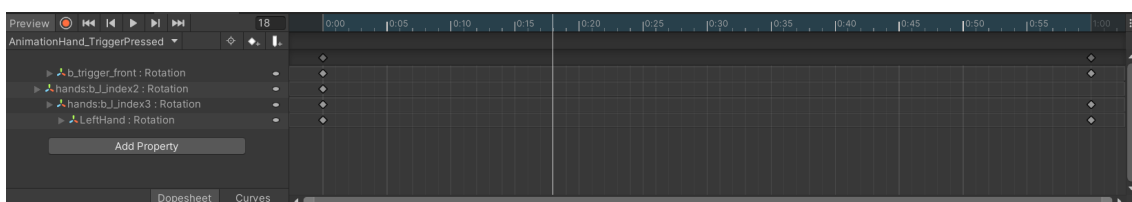
Studie [54] odhalila, že ženy vnímají nižší úroveň imerzivity při používání modelů rukou mužského typu, zatímco muži vnímají nižší úroveň při používání nehumánních modelů rukou. V reakci na tato zjištění byly vytvořeny ženské modely rukou úpravou velikosti dle [55] a textur původního mužského modelu. Ženský model je možné zhlédnout na 4.3.



Obr. 4.3: Mužský (nahore) a ženský (dole) model ruky

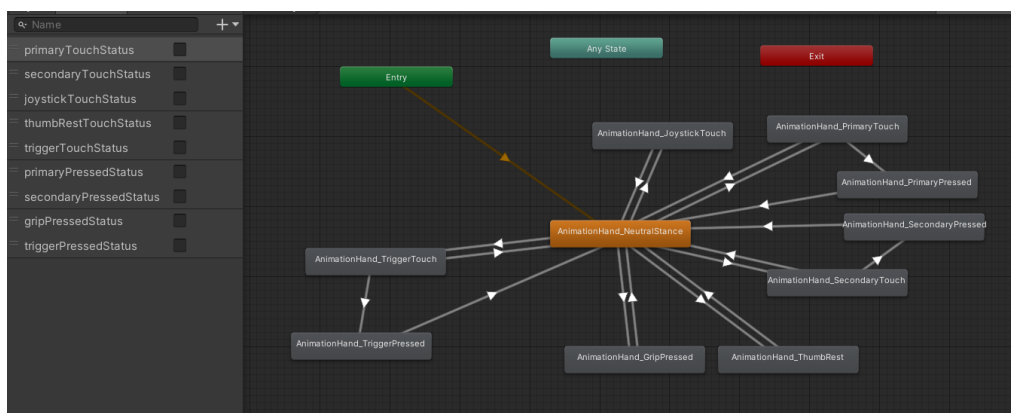
4.3.2 Animace a pohyby rukou

Editor Unity umožňuje jednoduše vytvářet animace pomocí *Animation System*. Tento systém funguje na principu nahrávání, jak je možné vidět na obrázku 4.4. V konkrétním čase je možné definovat pozici jednotlivých objektů (počáteční nebo koncovou), a systém mezi těmito body automaticky vytváří plynulý pohyb. Plynulost tohoto pohybu lze dále upravit pomocí specifických funkcí, ale v tomto případě byla použita základní funkce. Na obrázku 4.4 je zobrazena nahrávka animace, která simuluje stlačení předního tlačítka prstem ukazováčku. V levé části jsou zobrazeny přepočítané translace a rotace jednotlivých kloubů, které definují pózy mezi, kterými je následně vytvořen pohyb. V pravé části jsou uvedeny časové okamžiky, kdy má začít nebo končit pohyb jednotlivých částí modelu.



Obr. 4.4: Animation system pro stlačení předního tlačítka

Pro všechny dotyky a stlačení tlačítek byly vytvořeny animace. K vyvolání těchto animací v reakci na určité vstupní události je nezbytný *Animator Controller*. Jedná se o grafické rozhraní, které umožňuje organizovat a spravovat sadu animací a souvisejících přechodů mezi nimi. Toto rozhraní pracuje na principu *stavového automatu*. Na obrázku 4.5 je zobrazen Animator Controller pro levou ruku, kde jednotlivé stavy jsou reprezentovány obdélníky a přechody šipkami. Podmínky pro tyto přechody lze nastavit přímo v tomto prostředí, ale v tomto případě byly definovány pomocí skriptu, jelikož jsou určovány příchozími boolovskými hodnotami z ovladače.

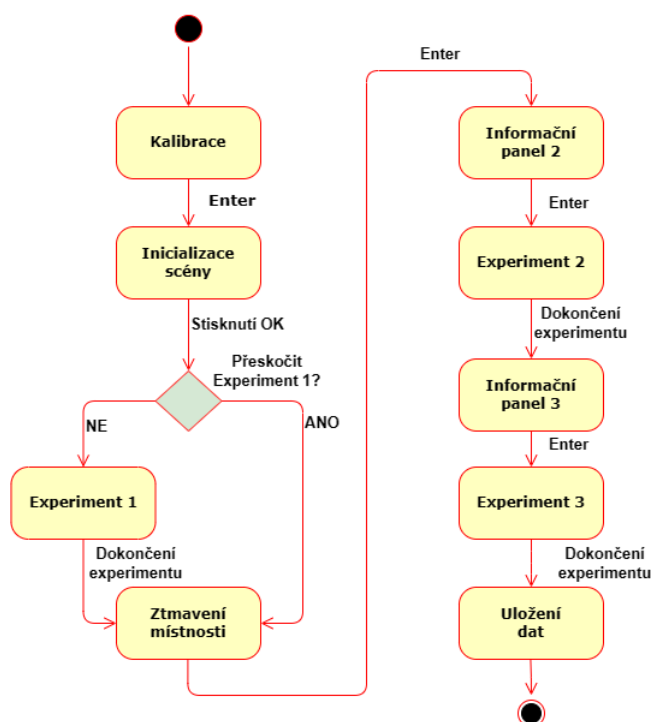


Obr. 4.5: Animator controller pro levou ruku

Tento přístup má určitou nevýhodu. Když probíhá animace, dojde k uzamčení pozice objektu, což zabraňuje další manipulaci s ním. Pro animace tlačítek je tento způsob vhodný, avšak pro realizaci pohybu joysticku a prstu při manipulaci s joystickem se tento přístup nejeví jako výhodný. Z tohoto důvodu je pohyb joysticku a prstu realizován matematicky.

4.4 Implementace hlavní funkce aplikace

Uživatel musí projít třemi experimenty – *experimentem 1, 2 a 3*. Během nich by měl být průběžně informován o tom, kde se nachází a co bude v rámci experimentů provádět. Je nutné, aby tyto experimenty uživatel absolvoval postupně, v předem definovaném pořadí. S ohledem na tyto požadavky je nejvhodnější využít stavový diagram přímo v hlavní smyčce programu. V Unity se pro tvorbu stavových automatů obvykle používají *korutiny* (Coroutines), které umožňují rozdělit úkoly do definovaných časových rámců běhu programu a nabízejí také užitečné funkce jako je *yield*. Bohužel v současné verzi Unity Editoru pro VR aplikace nejsou s tímto nástrojem plně kompatibilní, což může způsobit, že HMD přestane na cokoliv reagovat.



Obr. 4.6: Hlavní smyčka programu

Princip činnosti programu je znázorněn v diagramu 4.6. Po spuštění aplikace probíhá inicializace všech proměnných a vytvoření scény. Uživateli je na obrazovce počítače zobrazeno vstupní uživatelské rozhraní, které můžete vidět na obrázku 4.7.

Vytvoření tohoto uživatelského rozhraní, jeho funkcí a propojení se skripty bylo detailně popsáno v bakalářské práci [41]. Uživatelské rozhraní se skládá ze dvou hlavních částí - *dotazníku* pro osobu provádějící experiment a *nastavení* parametrů experimentu a kalibrace.

Obr. 4.7: Vstupní uživatelské rozhraní

Po vyplnění údajů je provedena *kalibrace systému*, která je detailně popsána v kapitole 4.5. Po stisknutí klávesy *Enter* je vstupní uživatelské rozhraní deaktivováno, nastavené parametry aktualizovány a nakonec jsou informace z dotazníku a nastavené parametry uloženy do textového souboru. V této fázi je uživateli s nasazeným HMD umožněno vidět místnost a pohybovat se v ní.

Před uživatelem se objeví *informační černá tabule*, viz obrázek 4.8, která mu pomáhá se zorientovat a udává, co má dělat. Uživatel je na začátku požádán, aby si prohlédl místnost a seznámil se s plněním úkolů, například dotykem ukazováčku s tlačítky.

Implementace dotyků dvou modelů byla provedena pomocí *kolizí*. Způsob implementace kolizí je obdobný tomu, který byl popsán v bakalářské práci [41], s jedním rozdílem. Modely rukou jsou připojeny k ovladačům, což Unity bere jako dynamické modely, a standardní způsob kolizí s nimi nefunguje. Proto je nutné přidat komponentu *RigidBody* ke modelům rukou. Nevýhodou tohoto řešení je, že dochází pouze k kolizím se statickými objekty. Pro aplikaci to však není překážkou, protože se v ní vyskytují pouze kolize se statickými objekty, ale musí se to vzít v úvahu při budoucích změnách.

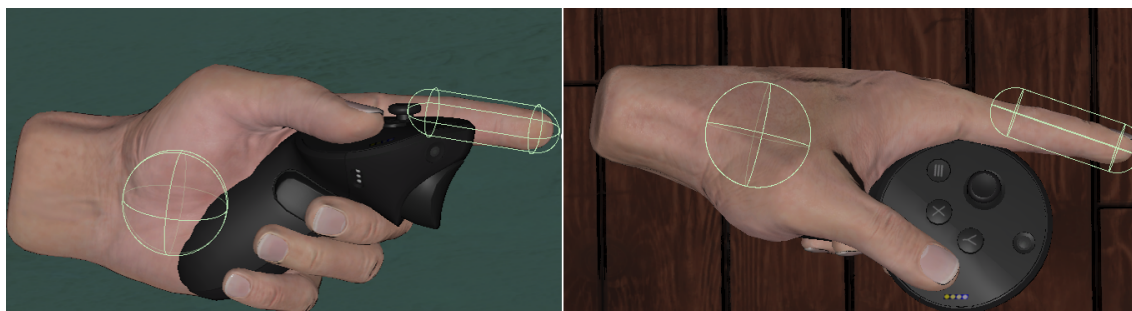
Na obrázku 4.9 je návrh *kolidorů* pro mužskou ruku. Pozice kolidorů u obou rukou jsou stejné - *ukazováček a dlaně*. Tento návrh se neliší ani u ženských modelů



Obr. 4.8: Prvotní pohled uživatele - informační tabule a modely rukou na stole

ruky. Kolidor ukazováčku je realizován pomocí *kapsle* a slouží k označení odpovědí. Na druhou stranu kolidor dlaně slouží k potvrzování odpovědí (položení ruky na stůl) a určení výšky ruky nad stolem.

Kolidor dlaně má tvar *sféry* a vrací události (eventy) pouze při kolizi s dvěma prahy. Tyto prahy jsou neviditelné roviny s minimální tloušťkou, která je nutná k získání událostí. Tyto roviny jsou rovnoběžné se stolem, ale jsou ve výšce odlišné od stolu.

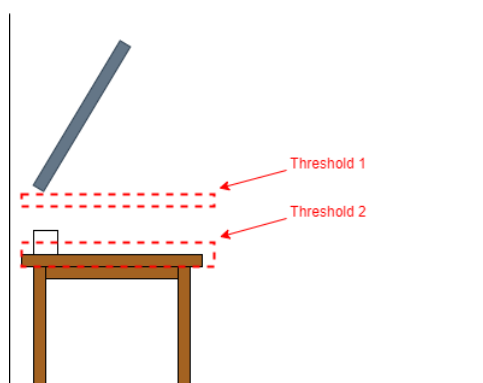


Obr. 4.9: Kolidory modelu ruky: kolidor ukazováček a dlaň

Pozice rovin jsou znázorněny na obrázku 4.10. Práh *Threshold 1* se využívá k určení pozice modelů rukou, například k ověření, zda je používána správná ruka pro řešení úlohy (což aktualizuje text v upozorňujícím okně) nebo k zastínění hodnocené ruky v rámci experimentu 2. Využívá výhodu sférického kolidoru, který umožňuje jednoduché zjištění pomocí globálních souřadnic, zda se kolidor nachází nad nebo pod prahem, když kolize skončí.

Práh *Threshold 2* se používá k simulaci kolize modelu s povrchem stolu. Jedním z požadavků návrhu bylo, aby se model ruky nevnořil do povrchu stolu. Proto běžná podmínka ohledně tloušťky pro zachycení kolize nepostačovala, tedy kolidor nemůže být na povrchu stolu. V průběhu byly implementovány dvě metody. První metoda využívala události, zda aktivní ruka je pod tímto prahem. Bohužel tato metoda byla významně závislá na výšce stolu, a to explicitně na velikosti ruky. To způsobovalo problémy s nalezením kompromisní výšky, aby nedocházelo ani k jednomu z dvou možných problémů. Člověk s větší rukou měl problém uskutečnit kolizi, protože kolidor byl nízko, zatímco u uživatele s menší rukou se model ruky zafixoval nad virtuálním stolem, protože práh byl vysoko. Další problém této metody přišel s univerzálností aplikace, protože závisela na přesnosti a stabilitě trakování ovladačů. Oculus Rift S vykazoval problémy s touto metodou na rozdíl od Oculus Quest 2.

Nakonec byla tato metoda thresholdu 2 modifikována. Výška prahu byla zvýšena tak, aby se eliminovala možnost, že by člověk měl ruku větší než je výška prahu. Prah v tuto chvíli představuje informativní charakter, že aktivní ruka splnila úlohu a přibližuje se ke stolu. Při kolizi s prahem se začne sledovat rychlost pohybu ruky. Pokud se rychlost sníží pod práh $v_T = 0.05 \text{ m.s}^{-1}$, dojde k potvrzení ruky. Tato metoda pracuje na principu, že rychlost pohybu ruky ležící na stole je nulová a vždy nastane. Práh v_T nebyl nastaven na nulu kvůli citlivosti snímačů a zpoždění získávání dat, ale také kvůli možnému třepání ruky u uživatele. Nevýhody první metody se touto metodou zredukovaly. Nevýhodou této metody je stále nutnost definovat výšku prahu kvůli možnosti zastavení v průběhu pohybu mezi prahem a stolem. Tento problém byl vyřešen tím, že výška prahu není nastavena příliš vysoko, což odstraňuje možné kolize při interakci během experimentu. S ohledem na charakter pohybu člověka při pokládání ruky je velmi málo pravděpodobné, že by se pohyboval s rychlostí menší než je nastavený práh.



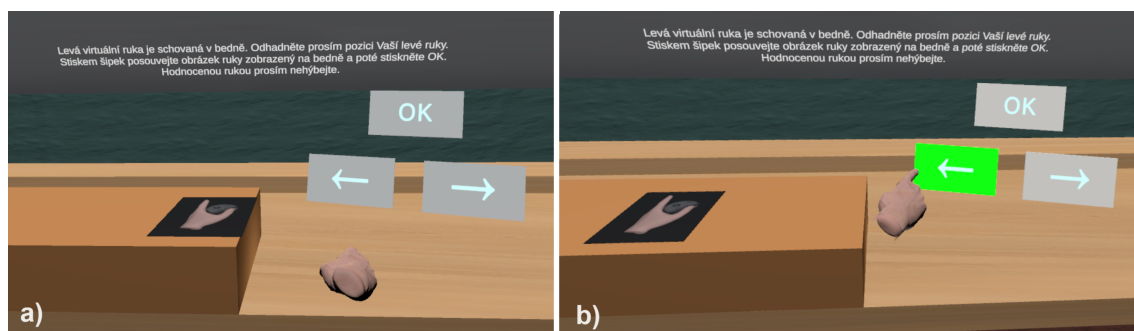
Obr. 4.10: Pozice neviditelných prahů ve scéně (bokoryst)

Experimenty budou probrány samostatně v podkapitolách. Po dokončení všech

experimentů jsou hodnoty získané během nich uloženy do souboru CSV. Tento soubor obsahuje údaje o průchodu hlavní smyčkou (začátek i konec experimentu) a jednotlivá data z experimentu, jako jsou například pozice ruky nebo řešení distrakčních úloh.

4.4.1 Experiment 1 a 3 - Box

Experimenty 1 a 3 jsou identické. Jedná se o typ úlohy, který se podobá popsanému v kapitole 1.3.1. Uživateli je zakryta jedna z rukou boxem (poloha této ruky je zafixována). Druhou rukou se snaží posunout, za pomoci šipek, obrázek na bedně na místo zafixované ruky. Tedy je využívána zcela proprioceptivní zpětná vazba pro odhad pozice ruky pod bednou, když je odebrána vizuální zpětná vazba. Na obrázcích 4.11 je znázorněna scéna experimentu a interakce s ní.

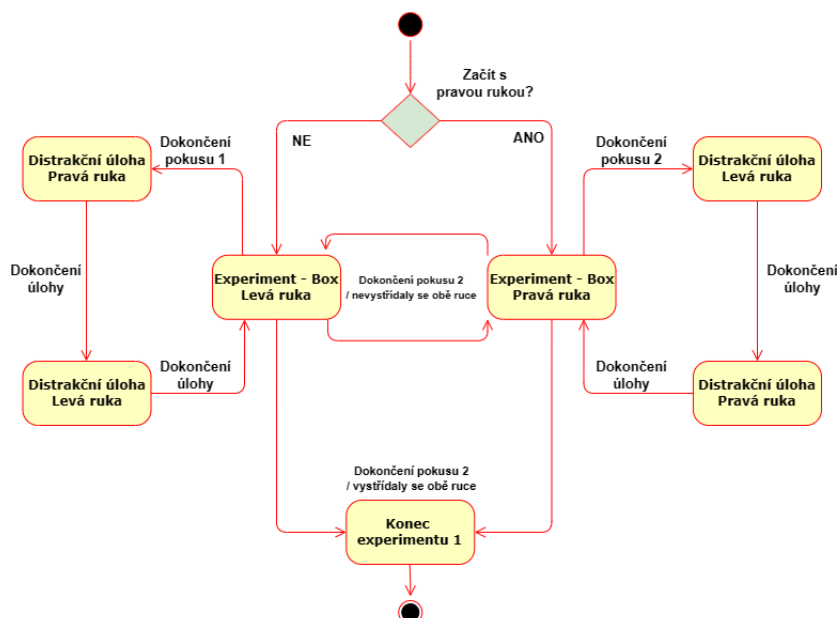


Obr. 4.11: Experiment 1 a 3 - box: a) scéna pro experiment; b) interakce uživatele se scénou

Experimenty jsou implementovány podle stavového diagramu 4.12. Experiment s bednou musí být proveden pro obě ruce s definovaným počtem pokusů podle vstupních parametrů. Na začátku tohoto stavu dochází k vytvoření potřebné scény pro daný experiment (viz obrázek 4.11a). Model ruky je zafixován a čeká se, dokud uživatel nepotvrdí pozici snímku. Snímek ruky je obyčejný model roviny s texturou, kterému je přidána komponenta Rigidbody, umožňující definovat rychlost, zrychlení a další parametry pohybu. V případě držení tlačítka pro pohyb rigidbody je nastaven směr pohybu a konstantní rychlost.

Pokus o experiment s bednou musí být proveden opakovaně pro obě ruce. Aby nebyly tyto pokusy hned za sebou, což by způsobovalo, že uživatel se začne více soustředit na ruce a adaptovat se, tak jsou přidány mezi pokusy experimentů *distrakční úlohy*. Tyto distrakční úlohy budou podrobněji diskutovány později v kapitole 4.4.3. Pro ilustraci se jedná o dva druhy jednoduchých úkolů, jejichž cílem je odvést pozornost uživatele od hlavního experimentu.

Během experimentu jsou ukládány hodnoty pokusů s bednou a také *chybovost* distrakčních úloh do datové struktury. Po dokončení stanoveného počtu pokusů pro obě ruce je experiment ukončen, scéna je deaktivována a parametry jsou přepsány pro opakování tohoto experimentu, pokud se nejednalo o experiment 3. Počet úloh a pořadí rukou při jejich plnění jsou předem určeny.



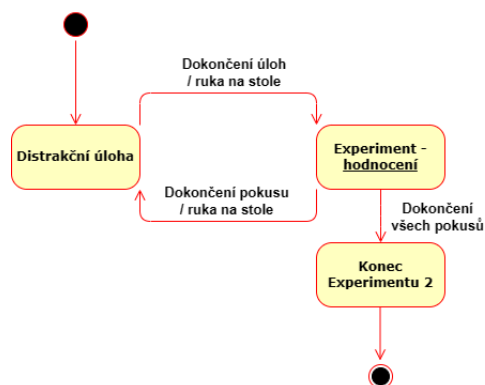
Obr. 4.12: Stavový diagram Experimentu 1 a 3

4.4.2 Experiment 2 - Hodnocení

Experiment 2 vychází z typu úlohy, který byl popsán v kapitole 1.3.2. Hlavním principem tohoto experimentu je, aby uživatel byl schopen rozpoznat, zda pozice modelu ruky ve virtuální realitě koresponduje s pozicí jeho skutečné ruky. Model ruky je během každého pokusu posunut o určitý *posun* (shift). Tento posun může být i nulový, tedy nedojde k posunutí.

Diagram experimentu 2 je znázorněn na obrázku 4.13. Není tak složitý jako u Experimentů 1 a 3, protože zde je vše určováno náhodně. Počet distrakčních úloh, které uživatel musí po sobě splnit, je náhodně vybrán z množiny 1, 2, 3. Pravděpodobnost výskytu dvou úloh je 0,6, jedné úlohy 0,3 a třech úloh 0,1. Výběr aktivní ruky (řeší distrakční úlohu) je také náhodný (*Levá*, *Pravá*) s pravděpodobnostmi 0,5 pro každou ruku. Distrakční úlohy jsou popsány v kapitole 4.4.3. Po každé sekvenci distrakčních úloh je vygenerován experiment - *hodnocení* (*judgment*).

U experimentu - hodnocení v původní aplikaci se objevil zásadní problém. Když byl k modelu ruky přidán posun (shift), uživatel si této změny všiml - ruka přeskočila. Proto byla nově implementována tzv. *černá scéna*. Tato funkce funguje na



Obr. 4.13: Stavový diagram Experimentu 2

principu ztmavení scény a pomocí zdrojů světla eliminuje předchozí problém. Celý proces tohoto experimentu bude podrobně vysvětlen pomocí grafického znázornění na obrázku 4.14.

Před začátkem experimentu 2 dochází k postupnému ztmavení scény, aby uživatel měl čas adaptovat se na prostředí a byl informován o průběhu experimentu (viz diagram hlavní smyčky 4.6). Po seznámení s experimentem započne samotný průběh experimentu 2 (dle diagramu 4.13). Uživateli jsou kladeny sekvence distrakčních úloh, jak je znázorněno na obrázku 4.14a. Během těchto úloh je scéna (tabule i stůl s modely rukou) osvětlena pomocí bodového světla (*zdroj světla 3*). Po splnění jednotlivých úloh musí uživatel vždy vrátit ruku na stůl, kde reaguje práh (*threshold 2*), a dle předdefinovaného počtu se rozhodne, zda pokračovat v úkolu, nebo přejít k pokusu experimentu - hodnocení.

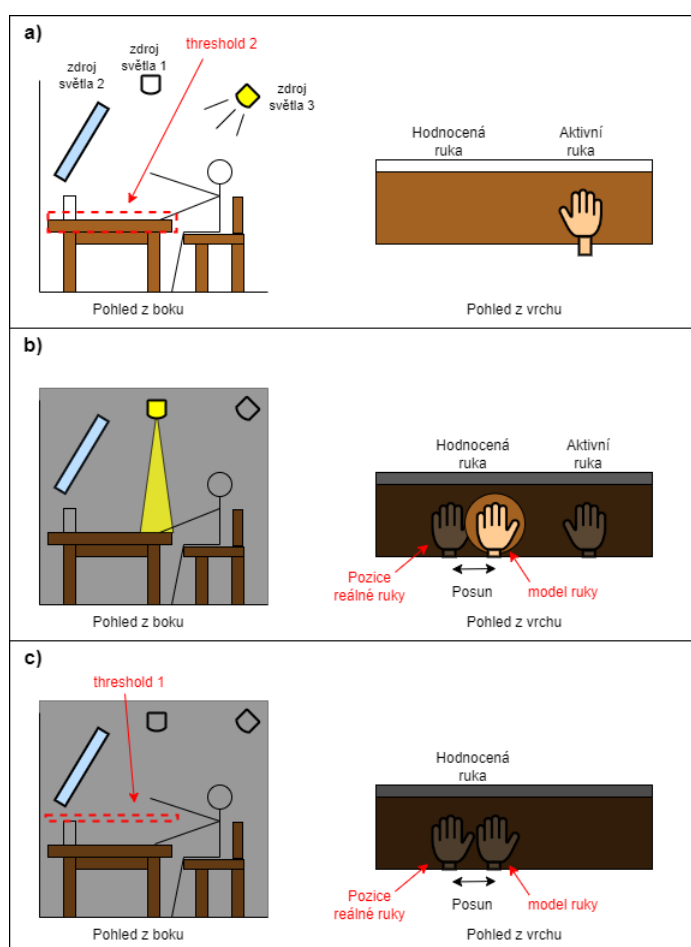
Pokud sekvence končí po vyřešení distrakční úlohy, *zdroj světla 3* je deaktivován po zakliknutí poslední odpovědi sekvence úloh. Až uživatel položí ruku zpět na stůl, je aktivován *zdroj světla 1*, který zvýrazní *hodnotící model ruky*, který byl v průběhu zafixován a jeho pozice posunuta o shift. Toto lze vidět na obrázku 4.14b. Pokud si uživatel je jistý správnosti odpovědi, natažením aktivní ruky k tabuli je zaznamenána událost za pomoci *threshold 1* a *zdroj světla 1* je vypnut. Tedy uživatel nemá druhou možnost si rozmyslet svou odpověď a musí zakliknout odpověď. Tento proces je ilustrován na obrázku 4.14c.

Tabule funguje na principu monitoru, který vyzařuje světlo směrem k uživateli. Bohužel Unity nenabízí přímou podporu tohoto způsobu osvětlení, na rozdíl od Unreal Engine. Proto byl tento princip simulován za pomoci tří bodových světél umístěných za tabulí s definovaným poloměrem dosvitu a intenzitou. Tabule je nastavena jako transparentní na světlo a aby nereagovala na stíny. Tento přístup umožňuje simulovat realistickou funkci svítícího monitoru a vytváří požadovaný dojem, když aktivní ruka překročí práh (*threshold 1*). Aktivní ruka postupně vynořuje

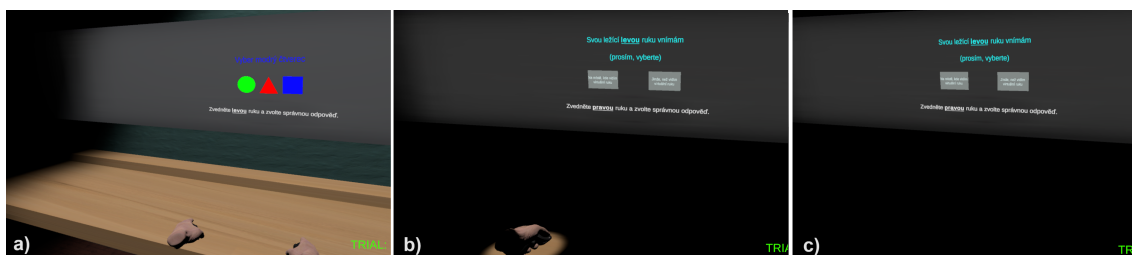
z tmy, čím blíže je k tabuli.

Po vrácení aktivní ruky na stůl je její pozice odfixována a posunuta o příslušný shift zpět. Poté je aktivován *zdroj světla 3* a uživateli je vygenerována nová sada distrakčních úloh. Tento postup se opakuje v souladu s definovaným parametrem ve vstupním uživatelském rozhraní. Na obrázku 4.15 je zobrazen průběh přímo z HMD.

V poslední řadě je důležité zmínit, jak je určen posun ruky (*shift*). Pro tento účel byl využit generátor náhodných čísel podle normálního rozdělení, jak je popsáno v [56]. Parametry normálního rozdělení jsou definovány v uživatelském rozhraní. Následně je tento posun omezen z obou stran tak, aby mohl nabývat hodnot mezi $\langle -10, 20 \rangle$, kde záporné znaménko zde označuje posun směrem k tělu, a kladné od těla. Tyto hodnoty reprezentují posun v centimetrech.



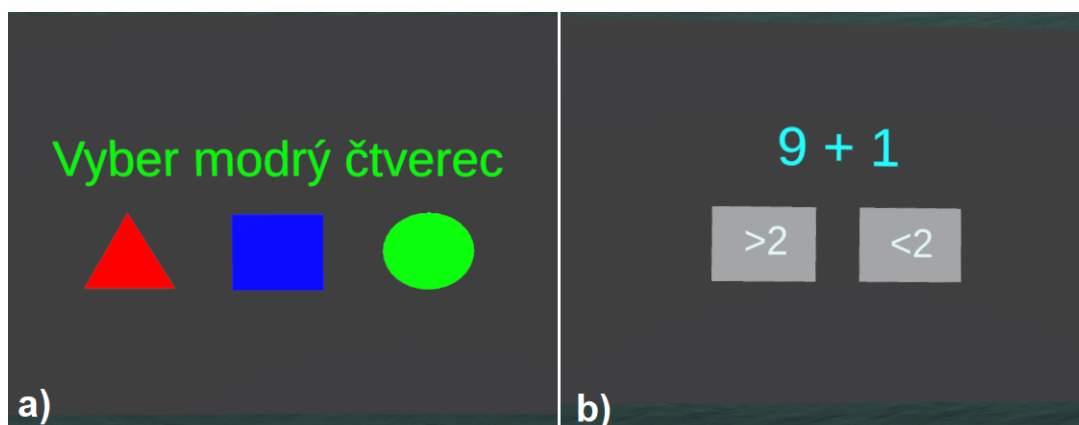
Obr. 4.14: Algoritmus experimentu 2 z více pohledů: **a)** řešení distrakčních úloh; **b)** zvýraznění hodnocené ruky po dokončení distrakčních úloh; **c)** vypnutí zvýraznění ruky po překročení prahu a výběru odpovědi



Obr. 4.15: Průběh experimentu 2 z pohledu uživatele: **a)** distrakční úloha; **b)** zvýraznění hodnocené ruky; **c)** vypnutí zvýrazněné ruky při výběru odpovědi

4.4.3 Distrakční úlohy

Distrakční úlohy jsou jednoduché typy úloh navržené k odvádění pozornosti uživatele od sledování polohy ruky při jejím položení. Byly vytvořeny dva druhy úloh - *matematické* a *geometrické*. Tyto úlohy jsou zobrazeny na obrázku 4.16. Generování typu úloh, zadání a odpovědi probíhá zcela náhodně, ale vždy je jedna z odpovědí správná.



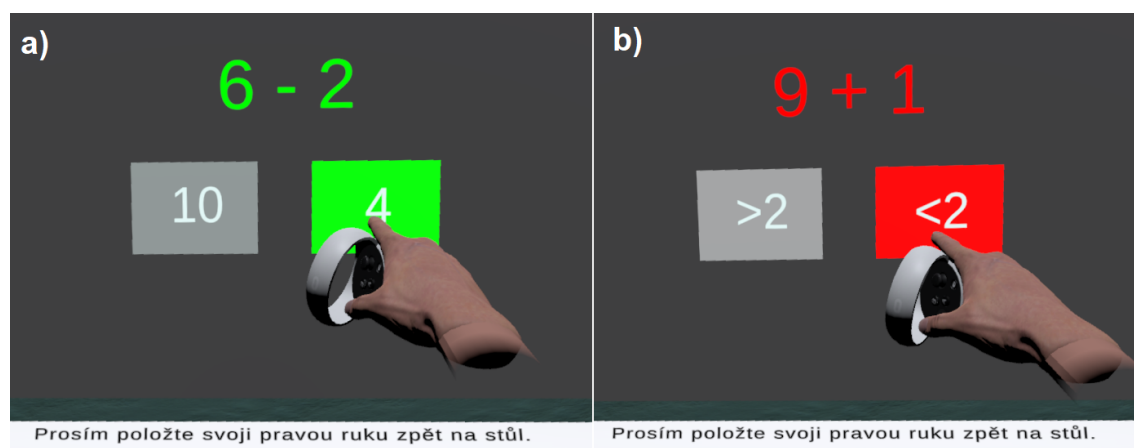
Obr. 4.16: Druhy distrakčních úloh: **a)** geometrická úloha; **b)** matematická úloha

Při generování geometrické úlohy dochází ke změně polohy tří základních rovinných útvarů - *červeného trojúhelníku*, *modrého čtverce* a *zeleného kruhu* - mezi sebou. Dále se mění text a barva zadání. Jak je patrné na obrázku 4.16a, kde je zadání zaměřeno na výběr *modrého čtverce* a text je zelený. Na rozdíl od matematické úlohy, zde máme pouze jeden typ úlohy, kde dochází k prohazování útvarů a změně zadání.

Matematická úloha se může ptát na dvě různé matematické problémy, a to čemu se výraz rovná anebo jestli je větší nebo menší. Výraz je vždy součet či rozdíl dvou čísel. Čísla se generují tak, aby správná hodnota byla z množiny $\{0, 1, 2, \dots, 19, 20\}$. Nesprávná hodnota je náhodná hodnota z této množiny. U úloh s porovnáním musí

platit stejné kritérium a generovaná odpověď nikdy nesmí být shodná s výsledkem výrazu.

Po vygenerování zadání je vždy uložena správná odpověď. Po kliknutí na odpověď je uživateli vizuálně zobrazena správnost výsledku. U matematické úlohy dojde ke změně barvy plochy vybrané odpovědi a textu výrazu, což je zřejmé na obrázku 4.17. U geometrické úlohy by změna barvy odpovědi a textu nebyla tak zřetelná, a proto je správnost odpovědi zobrazena pomocí symbolu uvnitř útvaru (fajfka nebo křížek). Pokud uživatel odpoví nesprávně, je vygenerována další úloha stejného typu. Tento proces se opakuje, dokud uživatel neodpoví správně.



Obr. 4.17: Vizuální zobrazení správnosti úlohy: **a)** výběr správné odpovědi; **b)** výběr špatné odpovědi

Po každé distrakční úloze jsou vždy uloženy do souboru informace o tom, jaký typ úlohy byl řešen, a jaká byla odpověď na ni. Tyto údaje jsou ukládány do stejného souboru jako data z hlavních experimentů.

Během validace aplikace bylo požadováno přidání možnosti nastavení distrakčních úloh. Toto bylo implementováno pomocí konfiguračního souboru nazvaného *experiment.config*. Tento soubor je vytvořen při prvním spuštění aplikace v adresáři *Inputs*, který se nachází ve složce této aplikace. V tomto konfiguračním souboru je ve formátu XML a lze za pomoci něho nastavit pět parametrů. Jedná se o čtyři logické hodnoty, které určují: zobrazení vizualizace špatné odpovědi, zobrazování odlišné barvy textu u geometrických úloh, opakování distrakční úlohy po špatné odpovědi a zahrnutí úloh s nerovnostmi. Posledním parametrem je definována maximální hodnota pro výsledky úloh. Aplikace čte tento konfigurační soubor při každém spuštění, a proto je po provedení změn nutné restartovat aplikaci.

4.5 Kalibrace systému

Každé HMD provádí při každém zapnutí vlastní centralizaci a obnovu směru pohledu. U Oculus Quest 2 dochází k tomuto procesu dokonce i při opětovném nasazení HMD. Pokud je HMD sundáno, systém detekuje nepřítomnost uživatele a aplikace se automaticky pozastaví. U jiných zařízení, jako je Oculus Rift S, se při sundání HMD opětovná kalibrace systému neprovádí.

Pro zajištění stejných podmínek pro všechny uživatele je vhodné systém zkalibrovat před začátkem používání aplikace. Na obrázku 4.18 vlevo je možné vidět, že při spuštění aplikace nejsou modely rukou na stole. V takovém případě je XR origin špatně nastaven – je pootočený a případně posunutý.



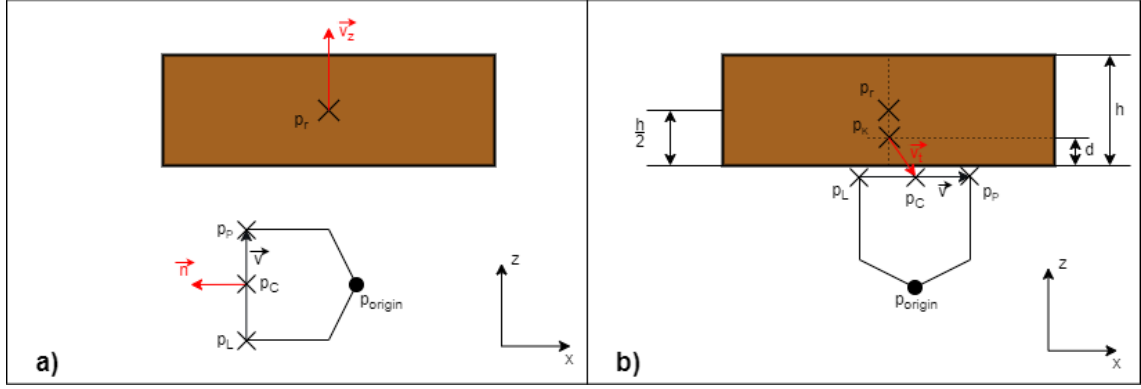
Obr. 4.18: Kalibrace pohledu HMD v aplikaci

Hlavním cílem kalibrace je zajistit, aby hrana virtuálního stolu korespondovala s hranou reálného stolu a aby směr pohledu HMD byl orientován na černou tabuli.

Kalibrace se provádí pomocí obou ovladačů, které jsou položeny na stůl směrem od uživatele tak, že konce rukojetí jsou zarovnané s hranou stolu. Ovladače by měly být ideálně umístěny ve stejné vzdálenosti od HMD. Nevýhodou této metody je potenciální chyba vzniklá špatným umístěním ovladačů. Tuto nepřesnost lze však kompenzovat umístěním ovladačů co nejdále od sebe, čímž se zvýší přesnost kalibrace.

Pro lepší pochopení výpočtu kalibrace je na obrázku 4.19 zobrazen zjednodušený náčrt pohledu shora. Na obrázku je vidět stůl, jehož střed je označen jako p_r , a pozice XR originu p_{origin} . K originu jsou vztaženy pozice ovladačů p_P (pravý ovladač) a p_L (levý ovladač). Stůl má šířku h a tloušťku l . Dále je důležitá délka ovladače d , která měří od konce rukojeti až po pozici měřicí jednotky v ovladači, a je stanovena na $d = 0,13\text{ m}$.

Prvním krokem v procesu kalibrace je vypočítat rotaci tak, aby směr pohledu byl orientován ve směru souřadnicové osy z . Dle obrázku 4.19a je možné vidět, že tuto



Obr. 4.19: Náskres výpočtu kalibrace: **a)** výpočet rotace; **b)** výpočet translace

úlohu lze zjednodušit na rovinný problém, při kterém je možné zanedbat souřadnici y . Prvně je zapotřebí získat normalizovaný vektor pohledu $\hat{\mathbf{n}}$, který je normálou k vektoru $\mathbf{v}$ mezi ovladači.

$$\mathbf{v} = p_P - p_L \quad (4.1)$$

$$\mathbf{n} = \mathbf{R} \left(\frac{\pi}{2} \right) \cdot \mathbf{v} \quad (4.2)$$

$$\hat{\mathbf{n}} = \frac{\mathbf{n}}{\|\mathbf{n}\|} \quad (4.3)$$

Místnost byla navržena tak, že její lokální souřadnicový systém má počátek v místě těžiště stolu, označeného jako p_r . Orientace os lokálního systému odpovídá orientaci globálního (World) souřadnicového systému. Díky tomuto návrhu je zjednodušen výpočet potřebný k nalezení rotace mezi normalizovaným vektorem pohledu $\hat{\mathbf{n}}$ a vektorem $\hat{\mathbf{v}}_z = (0, 0, 1)'$. Rotace je nejprve určena ve *Angle-axis* reprezentaci, která vyjadřuje rotaci pomocí vektoru $\mathbf{r}$ a úhlu θ , o který je potřeba se kolem tohoto vektoru otočit. Vektor $\mathbf{r}$ je možné získat vektorovým součinem a úhel θ skalárním součinem. Tato reprezentace je poté převedena na *kvaterniony*, protože právě s nimi pracuje systém Unity. Kvaterniony jsou v Unity preferovány pro jejich stabilitu a efektivitu při interpolaci rotací.

$$\mathbf{r} = \hat{\mathbf{v}}_z \times \hat{\mathbf{n}} \quad (4.4)$$

$$\theta = \arccos(\hat{\mathbf{v}}_z \cdot \hat{\mathbf{n}}) \quad (4.5)$$

Dále je nutné převést vypočítanou rotaci na kvaterniony a aplikovat tuto rotaci na XR origin. Před samotnou konverzí a aplikací rotace je však třeba normalizovat vektor $\mathbf{r}$ stejným způsobem, jaký je popsáný v 4.3.

$$q_{add} = (w; \mathbf{u}) = \left(\cos \left(\frac{\theta}{2} \right); \hat{\mathbf{r}} \sin \left(\frac{\theta}{2} \right) \right) \quad (4.6)$$

$$q_{origin_new} = q_{origin_old} \cdot q_{add} \quad (4.7)$$

Po aplikaci vypočtené rotace na origin je nutné spočítat translaci, která se uplatní na souřadnicový systém místnosti, aby se hrana virtuálního stolu shodovala s hranou skutečného stolu. Dále je třeba znovu spočítat vektor $\mathbf{v}$ podle . Poté je potřeba určit centrální bod mezi ovladači, což je místo, kde se pravděpodobně bude nacházet uživatel s nasazeným HMD:

$$p_C = p_L + \frac{\mathbf{v}}{2} \quad (4.8)$$

Pomocí tohoto bodu a bodu těžiště je možné vypočítat vektor, který představuje první translaci:

$$\mathbf{t} = p_C - p_r \quad (4.9)$$

Díky této translaci by se těžiště stolu posunulo na místo středu mezi ovladači. Nicméně je potřeba vypočítat takovou pozici, aby stůl korespondoval se skutečným stolem. K tomu lze využít délku ovladačů d a jejich způsob napolohování. K vektoru $\mathbf{t}$ je tedy přičtena další část translace, čímž je získána výsledná translace $\mathbf{v}_t$, která je následně aplikována pro posunutí pozice místnosti:

$$\mathbf{v}_t = \mathbf{t} + \left[0; -\frac{l}{2}; \frac{h}{2} - d \right]' \quad (4.10)$$

$$p_{r_new} = p_{r_old} + \mathbf{v}_t \quad (4.11)$$

4.6 Další funkce

Během implementace byly přidány další pomocné funkce, které byly vyžádány od lékařů. Tyto funkce přispívají k plynulému průběhu experimentu a zvyšují vnoření uživatele do samotného experimentálního procesu.

4.6.1 Validace pokusu experimentu 2

Pro experiment 2 byla přidána funkce pro testujícího (sedícího před počítačem), který je schopen za pomoci klávesnice odebrat aktuální nebo předchozí pokus v

případě, že byl nesprávný. Tato možnost je využita v situacích, kdy osoba provádějící experiment neúmyslně zvedne ruku, která byla hodnocena, nebo se zmýlí při hodnocení samotného pokusu.

Za pomoci kláves *P* (předchozí pokus byl invalidní) a *I* (aktuální pokus je invalidní) je znehodnocen daný pokus. Informace o neplatném pokusu je zaznamenána do výstupního souboru a je odečtena z celkového počtu platných pokusů. Testovaná osoba je následně vyzvána k opakování daného pokusu.

Aby testující věděl, kolik zbývá platných pokusů a zda byla zaznamenána neplatnost pokusu, bylo vytvořeno druhé uživatelské rozhraní (obrázek 4.20). To je zobrazeno pouze testujícímu na monitoru PC a pouze během řešení experimentu 2. Na pravé straně je zeleně zvýrazněn počet provedených pokusů a celkový počet pokusů nastavený podle vstupního uživatelského rozhraní. Červeně je zvýrazněn počet neplatných pokusů, které je třeba zopakovat. Dále na obrázku lze vidět pole upozornění, zda byla neplatnost pokusu zaznamenána. Na obrázku 4.20a je zobrazena neplatnost aktuálního pokusu a za 4.20b je zobrazena neplatnost předchozího pokusu.

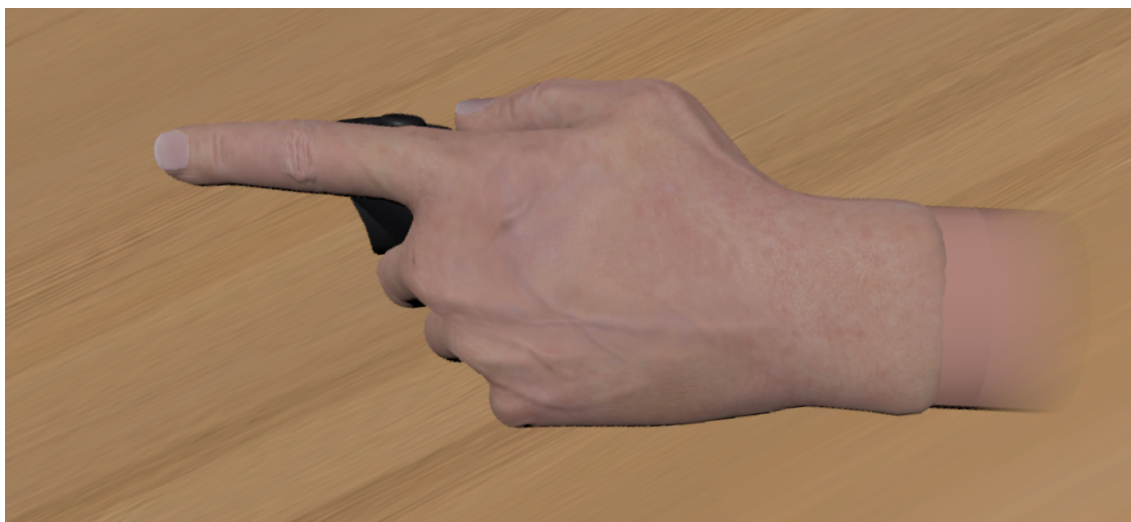


Obr. 4.20: Uživatelské rozhraní pro experiment 2 se zobrazeným upozorněním o invalidním pokusu: **a)** aktuální pokus je invalidní; **b)** předchozí pokus byl invalidní

4.6.2 Blednutí modelu ruky

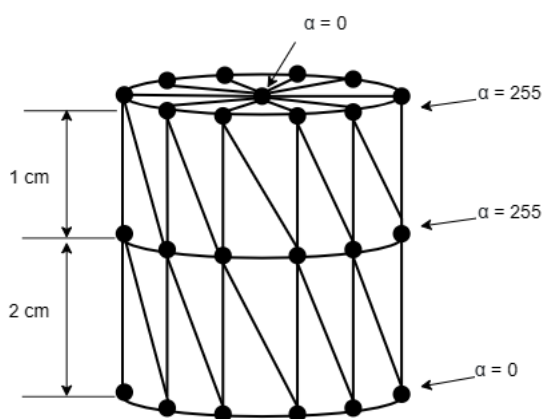
Modelům rukou byl postupně přidán objekt válce, který se blednoucím způsobem simuluje pokračování končetiny. Tento objekt má za cíl snížit pocit oddělení dlaně od zbytku paže, což byl problém ovlivňující pocit vnoření u uživatele. Realizovaný objekt je zobrazen na obrázku 4.21.

Tento útvar byl vytvořen spojením dvou primitivních válců do jednoho meshe. Tento přístup umožnil použití interpolace barev a průhlednosti po celém povrchu objektu. Postup spojování meshů a jejich interpolace barev na povrchu byl popsán v bakalářské práci [41]. Návrh objektu je zobrazen na obrázku 4.22.



Obr. 4.21: Blednutí ruky v aplikaci

Po spojení dvou primitivních válců jsou vytvořeny tři vrstvy vertexů podél výšky válce. Pozice vertexů (výška) a jednotlivé barvy byly definovány pro každý vertex samostatně. Aby se dosáhlo dojmu postupného blednutí, byla pro všechny vertexy stejné úrovně nastavena hodnota průhlednosti α dle obrázku. Mezi prvními úrovněmi nedochází k blednutí barev, aby ruka vypadala, že je spojená s paží. Mezi druhou a třetí úrovní dochází k postupnému blednutí, až je druhá strana úplně průhledná.



Obr. 4.22: Návrh objektu pro vytvoření blednutí ruky

4.7 Hodnocení aplikace a její srovnání

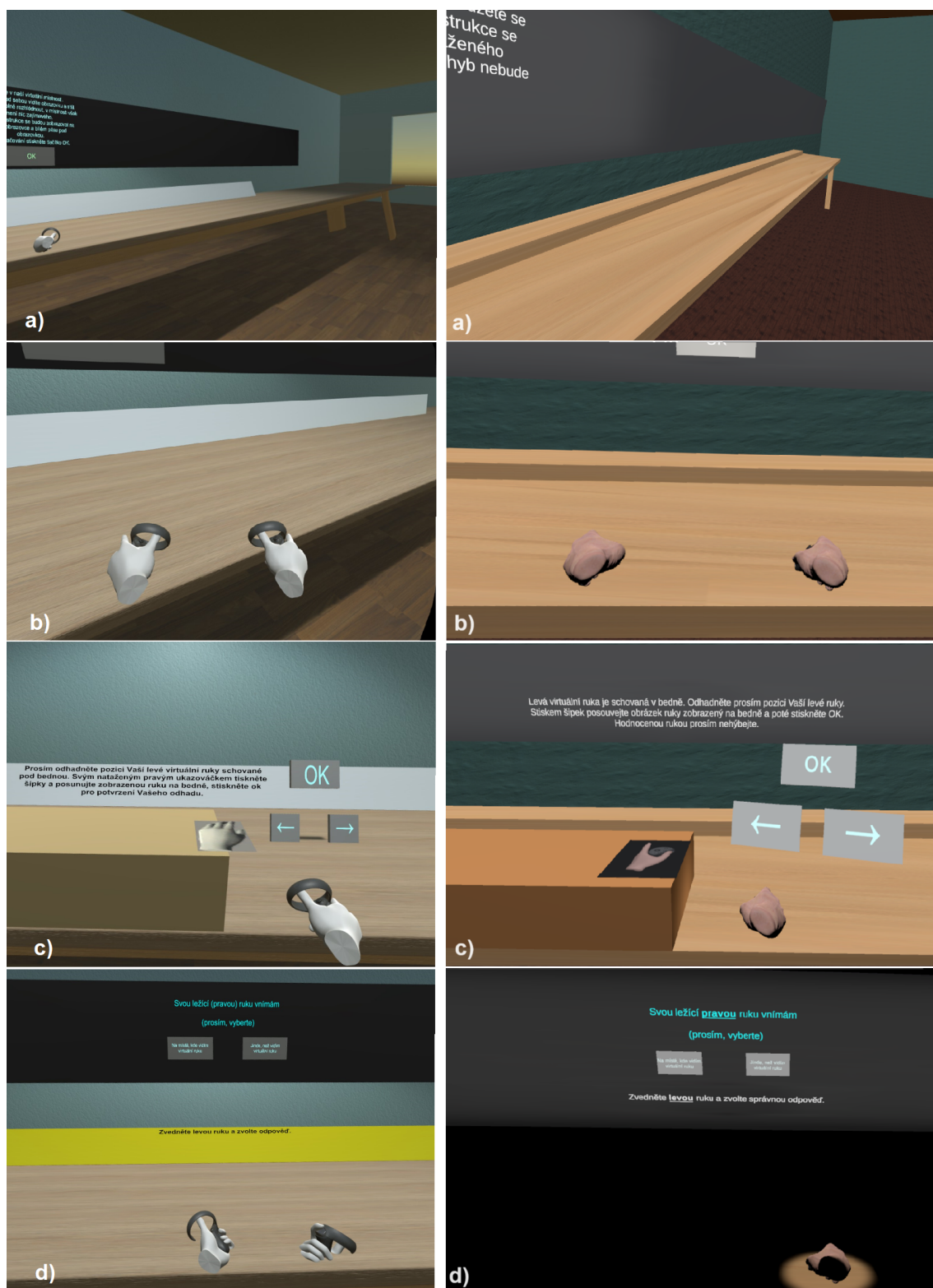
V této podkapitole proběhne hodnocení funkčnosti nově vyvinuté aplikace a k srovnání jejích výsledků s předchozí. Předchozí aplikace byla dostupná pouze jako spustitelný soubor bez přístupu k zdrojovým souborům, což ji činilo pouze jako výchozí předloha pro vývoj nové aplikace, která byla vytvořena úplně od začátku.

Aplikace byla vyvinuta v úzké kolaboraci s expertními lékaři z Lékařské fakulty Masarykovy univerzity. Od počátku vývoje probíhaly systematické a pravidelné konzultace, které sloužily k optimalizaci funkcionalit a implementaci nových požadavků specifikovaných odborníky. Každá navrhovaná změna v aplikaci byla pečlivě ověřována týmem pěti zaměstnanců a doktorandů, což zajišťovalo její vědeckou přesnost a praktickou aplikovatelnost.

Výsledná aplikace obdržela od lékařského týmu vysoké hodnocení a je již plně připravena k nasazení v klinické praxi. Významně byly oceněny jak základní prvky, jako jsou černá scéna a spolehlivá kalibrace, tak i nově přidané funkce, které výrazně zvyšují imerzivní zážitek uživatelů, ale také její univerzálnost. Tato imerzivita byla rozšířena o sofistikované vizuální a interaktivní prvky, včetně speciálně navržených 3D modelů ruk pro muže a ženy. Nově integrované prvky (nové uživatelské rozhraní, validace a černá scéna) nejenže zlepšují uživatelský komfort, ale také značně usnadňují práci lékařů, což aplikaci činí výjimečnou ve své kategorii a rozšiřuje její potenciál pro další výzkum a terapeutické využití.

Na obrázku 4.23 jsou zobrazeny srovnání návrhů scény, výsledky funkce kalibrace, a také záběry z experimentů 1 a 2 z perspektivy uživatele s nasazeným HMD. Na dalším obrázku 4.25 jsou prezentovány oba typy distrakčních úloh - geometrický a matematický.

Z obrázku 4.24 je patrné, že design vstupního uživatelského rozhraní zůstal téměř nezměněn, což umožňuje uživatelům, v tomto případě lékařům, pracovat v již známém prostředí bez nutnosti adaptace na nové rozhraní.



Obr. 4.23: Srovnání aplikací: **a)** návrh scény; **b)** výsledek kalibrace; **c)** experiment 1 - box; **d)** experiment 2 - hodnocení polohy ruky. Původní aplikace (**vlevo**) a nová aplikace (**vpravo**).

VR experiment - dotazník před experimentem

Účastník:

Věk:

Výška:

Hmotnost:

Pohlaví: Muž

Rukost: Pravá

Povolání:

Dosažené vzdělání:

Pravidelně užívané léky:

Délka užívání léků:

Ruka ovládání PC myši: Pravá

Zkušenost s VR: Žádná

Zájemové / sportovní / taneční aktivity + jejich trvání:

Poznámky:

Potvrdit údaje a přejít ke kalibraci.

☐ Vypnout VR na kalibraci
☐ Začít úkolem 2

Před stiskem tlačítka usadte účastníka do správné polohy ke stolu a nasadte mu HMD na hlavu.

Po kliknutí na tlačítko započne kalibrace VR prostředí. Scéna se bude kalibrovat pomocí levého ovladače. Levý ovladač musí být umístěn v přirozeném směru, konec rukojeti na bližší hraně stolu. Kalibruje se výška a vzdálenost od/k účastníkovi, levoprávní osa je automaticky centrována (nekalibruje se).

Následným stiskem klávesy Enter se zafixuje pozice a načte se VR scéna.

Úkol 1 (běh 1):

Úkol 1 (běh 2):

Trialy levá

Trialy pravá

2

2

1

1

☐ Začít odhadem pravé ruky

Úkol 2:

Trialy/ruka

Max. posun dovnitř

Max. posun ven

30

0,1

0,2

Location

Scale

0

0,125

Parametry pro distribuci:

0

0,125

Všechny jednotky jsou v metrech.

Čas. limit (v sekundách) pro úkoly v úkolu 2: (0 = neomezený čas)

10

VR experiment - dotazník před experimentem

Účastník:

Věk:

Výška:

Hmotnost:

Pohlaví: Muž

Rukost: Pravá

Povolání:

Dosažené vzdělání:

Pravidelně užívané léky:

Délka užívání léků:

Ruka ovládání PC myši: Pravá

Zkušenost s VR: Žádná

Zájemové / sportovní / taneční aktivity + jejich trvání:

Poznámky:

Potvrdit údaje a přejít ke kalibraci

☐ Vypnout VR na kalibraci
☐ Začít úkolem 2

Následným stiskem klávesy Enter započne experiment

Model rukou (pohlaví): Muž

☐ Zapnout ostatní animace rukou

Úkol 1 (běh 1):

Úkol 1 (běh 2):

Trialy levá

Trialy pravá

2

2

1

1

☐ Začít odhadem pravé ruky

Úkol 2:

Trialy/ruka

Max. posun dovnitř

Max. posun ven

30

0,1

0,2

Location

Scale

0

0,125

Parametry pro distribuci:

0

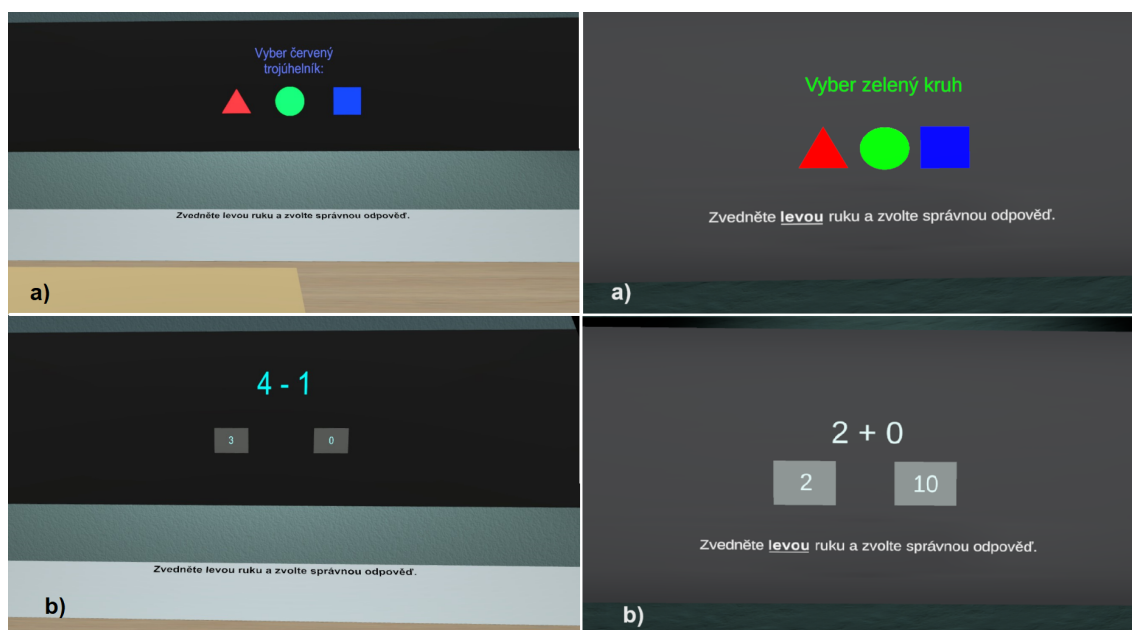
0,125

Všechny jednotky jsou v metrech

Čas. limit (v sekundách) pro úkoly 2: (0 = neomezený čas)

20

Obr. 4.24: Srovnání vstupního uživatelského rozhraní: **a)** původní aplikace; **b)** nová aplikace.



Obr. 4.25: Srovnání distrakčních úloh: **a)** geometrická úloha; **b)** matematická úloha. Původní aplikace (**vlevo**) a nová aplikace (**vpravo**).

Závěr

V rámci této práce byla vyvinuta nová aplikace pro analýzu poruch propiocepce využívající prostředí virtuální reality na platformě Unity. Aplikace byla zkonstruována od základů s využitím předchozí instalace aplikace, k jejichž zdrojovým kódům nebyl přístup, jako zadání. Vývoj probíhal ve spolupráci s odborníky z Lékařské fakulty Masarykovy univerzity, což umožnilo přizpůsobit aplikaci specifickým požadavkům klinické praxe. Celý proces vývoje byl charakteristický pravidelnými iterativními konzultacemi, díky kterým aplikace reflektovala nejnovější vědecké poznatky i praktické potřeby lékařů.

Aplikace, původně zaměřená na propiocepci, nabízí široké možnosti využití v dalších oblastech zdravotnictví, což otevírá nové cesty pro její implementaci ve fyzikální terapii, psychologii a rehabilitaci. Potenciál aplikace přesahuje její původní účel a naznačuje možnosti jejího využití v širším spektru klinických a terapeutických oblastí.

Experimentální část práce poskytla cenná data o přesnosti lokalizačních systémů, jako je Vicon Tracker a Oculus Quest 2. Tyto data jsou klíčová pro další výzkum a je plánováno jejich rozšířené zkoumání a publikace ve vědeckých časopisech.

Vzhledem k flexibilní architektuře aplikace je možné v budoucnosti pokračovat v jejím rozvoji a rozšířit ji o nové funkce a experimenty zaměřené na dynamickou propiocepci. Tyto rozšíření by mohla přispět k lepšímu porozumění dynamické propiocepce a k vývoji nových terapeutických postupů. Rovněž bude důležité zdokonalit metodiku předzpracování dat, především v oblasti synchronizace dat v experimentech zaměřených na zjišťování přesnosti lokalizace headsetu Oculus Quest 2.

Literatura

- [1] WALSH, Kenneth R. a PAWLOWSKI, Suzanne D. Virtual Reality: A Technology in Need of IS Research. Online. *Communications of the Association for Information Systems*. 2002, roč. 8. ISSN 15293181. Dostupné z: <https://doi.org/10.17705/1CAIS.00820>. [cit. 2023-10-05].
- [2] SANCHEZ-VIVES, Maria V. a SLATER, Mel. From presence to consciousness through virtual reality. Online. *Nature Reviews Neuroscience*. 2005, roč. 6, č. 4, s. 332-339. ISSN 1471-003X. Dostupné z: <https://doi.org/10.1038/nrn1651>. [cit. 2023-10-05].
- [3] NILSSON, Niels Christian; NORDAHL, Rolf a SERAFIN, Stefania. Immersion Revisited: A review of existing definitions of immersion and their relation to different theories of presence. Online. *Human Technology*. 2016, roč. 12, č. 2, s. 108-134. ISSN 17956889. Dostupné z: <https://doi.org/10.17011/ht/urn.201611174652>. [cit. 2023-10-05].
- [4] WOHLGENANT, Isabell; SIMONS, Alexander a STIEGLITZ, Stefan. Virtual Reality. Online. *Business & Information Systems Engineering*. 2020, roč. 62, č. 5, s. 455-461. ISSN 2363-7005. Dostupné z: <https://doi.org/10.1007/s12599-020-00658-9>. [cit. 2023-10-05].
- [5] LAVALLE, Steven M. *Virtual Reality*. Online. Cambridge University Press, 2023. ISBN 9781108182874. Dostupné z: <https://doi.org/10.1017/9781108182874>. [cit. 2023-11-07].
- [6] COLDEWEY, Devin. *How Oculus squeezed sophisticated tracking into pipsqueak hardware*. Online. In: TechCrunch, 22.08.2019. Dostupné z: <https://techcrunch.com/2019/08/22/how-oculus-squeezed-sophisticated-tracking-into-pipsqueak-hardware/>. [citováno 2023-11-08].
- [7] CHANDRASHEKHAR, A; SAHEB, Shaik Himam; PANDA, Sandeep Kumar; BALAMURUGAN, S. a PENG, Sheng-Lung (ed.). *Metaverse and Immersive Technologies*. Online. Wiley, 2023. ISBN 9781394174546. Dostupné z: <https://doi.org/10.1002/9781394177165>. [cit. 2023-11-08].
- [8] HESCH, Joel; KOZMINSKI, Anna a LINDE, Oskar. *Powered by AI: Oculus Insight*. Online. In: Meta, 22.08.2019. Dostupné z: <https://ai.facebook.com/blog/powered-by-ai-oculus-insight/>. [citováno 2023-11-08].

- [9] MULVANY, Gerard; TAYLOR, Simeon a GREUTER, Stefan. Target. Online. In: *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play*. New York, NY, USA: ACM, 2020, s. 321-325. ISBN 9781450375870. Dostupné z: <https://doi.org/10.1145/3383668.3419875>. [cit. 2023-12-08].
- [10] HOLZWARTH, Valentin; GISLER, Joy; HIRT, Christian a KUNZ, Andreas. Comparing the Accuracy and Precision of SteamVR Tracking 2.0 and Oculus Quest 2 in a Room Scale Setup. Online. In: *2021 the 5th International Conference on Virtual and Augmented Reality Simulations*. New York, NY, USA: ACM, 2021, s. 42-46. ISBN 9781450389327. Dostupné z: <https://doi.org/10.1145/3463914.3463921>. [cit. 2023-11-08].
- [11] WINKLER, Alexander; WON, Jungdam a YE, Yuting. QuestSim: Human Motion Tracking from Sparse Sensors with Simulated Avatars. Online. In: *SIGGRAPH Asia 2022 Conference Papers*. New York, NY, USA: ACM, 2022, s. 1-8. ISBN 9781450394703. Dostupné z: <https://doi.org/10.1145/3550469.3555411>. [cit. 2024-04-05].
- [12] MELIM, Andrew. *Tracking Technology Explained: LED Matching*. Online. In: Meta Quest, 04.11.2019. Dostupné z: <https://developer.oculus.com/blog/tracking-technology-explained-led-matching/>. [citováno 2024-04-05].
- [13] FISCHLER, Martin A. a BOLLES, Robert C. Random sample consensus. Online. *Communications of the ACM*. 1981, roč. 24, č. 6, s. 381-395. ISSN 0001-0782. Dostupné z: <https://doi.org/10.1145/358669.358692>. [cit. 2024-04-05].
- [14] PEREIRA, Diogo; OLIVEIRA, Vitor; VILAÇA, João L.; CARVALHO, Vítor a DUQUE, Duarte. Measuring the Precision of the Oculus Quest 2's Hand-held Controllers. Online. *Actuators*. 2023, roč. 12, č. 6, s. 257. ISSN 2076-0825. Dostupné z: <https://doi.org/10.3390/act12060257>. [cit. 2023-11-08].
- [15] CARNEVALE, Arianna; MANNOCCHI, Ilaria; SASSI, Mohamed Saifeddine Hadj; CARLI, Marco; DE LUCA, Giovanna De et al. Virtual Reality for Shoulder Rehabilitation: Accuracy Evaluation of Oculus Quest 2. Online. *Sensors*. 2022, roč. 22, č. 15. ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s22155511>. [cit. 2024-04-05].
- [16] HAN, Shangchen; LIU, Beibei; CABEZAS, Randi; TWIGG, Christopher D.; ZHANG, Peizhao et al. MEgATrack. Online. *ACM Transactions on Graphics*.

- 2020, roč. 39, č. 4. ISSN 0730-0301. Dostupné z: <https://doi.org/10.1145/3386569.3392452>. [cit. 2024-04-06].
- [17] ABDULKARIM, Diar; DI LUCA, Massimiliano; AVES, Poppy; MAAROUFI, Mohamed; YEO, Sang-Hoon et al. A methodological framework to assess the accuracy of virtual reality hand-tracking systems: A case study with the Meta Quest 2. Online. *Behavior Research Methods*. 2024, roč. 56, č. 2, s. 1052-1063. ISSN 1554-3528. Dostupné z: <https://doi.org/10.3758/s13428-022-02051-8>. [cit. 2024-04-10].
- [18] MAO, Xinrong; LIU, Kaiming a HANG, Yanfen. Feature Extraction and Matching of Slam Image Based on Improved SIFT Algorithm. Online. In: *2020 2nd Symposium on Signal Processing Systems*. New York, NY, USA: ACM, 2020, s. 18-23. ISBN 9781450388627. Dostupné z: <https://doi.org/10.1145/3421515.3421528>. [cit. 2023-09-02].
- [19] DURRANT-WHYTE, H. a BAILEY, T. Simultaneous localization and mapping: part I. Online. *IEEE Robotics & Automation Magazine*. 2006, roč. 13, č. 2, s. 99-110. ISSN 1070-9932. Dostupné z: <https://doi.org/10.1109/MRA.2006.1638022>. [cit. 2023-09-02].
- [20] KHAN, Misha Urooj; ZAIDI, Syed Azhar Ali; ISHTIAQ, Arslan; BUKHARI, Syeda Ume Rubab; SAMER, Sana et al. A Comparative Survey of LiDAR-SLAM and LiDAR based Sensor Technologies. Online. In: *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*. IEEE, 2021, s. 1-8. ISBN 978-1-6654-2413-4. Dostupné z: <https://doi.org/10.1109/MAJICC53071.2021.9526266>. [cit. 2023-09-02].
- [21] XU, Xiaobin; ZHANG, Lei; YANG, Jian; CAO, Chenfei; WANG, Wen et al. A Review of Multi-Sensor Fusion SLAM Systems Based on 3D LIDAR. Online. *Remote Sensing*. 2022, roč. 14, č. 12. ISSN 2072-4292. Dostupné z: <https://doi.org/10.3390/rs14122835>. [cit. 2023-09-03].
- [22] THEODOROU, Charalambos; VELISAVLJEVIC, Vladan; DYO, Vladimir a NONYELU, Fredi. Visual SLAM algorithms and their application for AR, mapping, localization and wayfinding. Online. *Array*. 2022, roč. 15. ISSN 25900056. Dostupné z: <https://doi.org/10.1016/j.array.2022.100222>. [cit. 2023-09-03].
- [23] MUR-ARTAL, Raul; MONTIEL, J. M. M. a TARDOS, Juan D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. Online. *IEEE Transactions*

- on Robotics*. 2015, roč. 31, č. 5, s. 1147-1163. ISSN 1552-3098. Dostupné z: <https://doi.org/10.1109/TR0.2015.2463671>. [cit. 2023-09-04].
- [24] PIRE, Taihú; FISCHER, Thomas; CASTRO, Gastón; DE CRISTÓFORIS, Pablo; CIVERA, Javier et al. S-PTAM: Stereo Parallel Tracking and Mapping. Online. *Robotics and Autonomous Systems*. 2017, roč. 93, s. 27-42. ISSN 09218890. Dostupné z: <https://doi.org/10.1016/j.robot.2017.03.019>. [cit. 2023-09-05].
- [25] LYU, Lin; DING, Yan; YUAN, Yating; ZHANG, Yutong; LIU, Jinpeng et al. DOC-SLAM: Robust Stereo SLAM with Dynamic Object Culling. Online. In: *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE, 2021, s. 258-262. ISBN 978-1-6654-0469-3. Dostupné z: <https://doi.org/10.1109/ICARA51699.2021.9376418>. [cit. 2023-09-05].
- [26] MUR-ARTAL, Raul a TARDOS, Juan D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. Online. *IEEE Transactions on Robotics*. 2017, roč. 33, č. 5, s. 1255-1262. ISSN 1552-3098. Dostupné z: <https://doi.org/10.1109/TR0.2017.2705103>. [cit. 2023-09-07].
- [27] CAMPOS, Carlos; ELVIRA, Richard; RODRIGUEZ, Juan J. Gomez; M. MONTIEL, Jose M. a D. TARDOS, Juan. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. Online. *IEEE Transactions on Robotics*. 2021, roč. 37, č. 6, s. 1874-1890. ISSN 1552-3098. Dostupné z: <https://doi.org/10.1109/TR0.2021.3075644>. [cit. 2023-09-07].
- [28] JAVAID, Mohd a HALEEM, Abid. Virtual reality applications toward medical field. Online. *Clinical Epidemiology and Global Health*. 2020, roč. 8, č. 2, s. 600-605. ISSN 22133984. Dostupné z: <https://doi.org/10.1016/j.cegh.2019.12.010>. [cit. 2023-11-09].
- [29] WADE, Derick T. What is rehabilitation? An empirical investigation leading to an evidence-based description. Online. *Clinical Rehabilitation*. 2020, roč. 34, č. 5, s. 571-583. ISSN 0269-2155. Dostupné z: <https://doi.org/10.1177/0269215520905112>. [cit. 2023-11-09].
- [30] CHO, Sangwoo; KU, Jeonghun; CHO, Yun Kyung; KIM, In Young; KANG, Youn Joo et al. Development of virtual reality proprioceptive rehabilitation system for stroke patients. Online. *Computer Methods and Programs in Biomedicine*. 2014, roč. 113, č. 1, s. 258-265. ISSN 01692607. Dostupné z: <https://doi.org/10.1016/j.cmpb.2013.09.006>. [cit. 2023-11-10].

- [31] VLADIMIROVICH ZAKHAROV, Alexander; VLADIMIROVICH KOLSA-
NOV, Alexander; VIKTOROVNA KHIVINTSEVA, Elena; FEDOROVICH
PYATIN, Vasiliy a VLADIMIROVICH YASHKOV, Alexander. Propriocep-
tion in Immersive Virtual Reality. Online. In: A. VEGA, José a COBO, Juan
(ed.). *Proprioception*. IntechOpen, 2021. ISBN 978-1-83968-069-4. Dostupné z:
<https://doi.org/10.5772/intechopen.96316>. [cit. 2023-11-11].
- [32] JUAN, M.-Carmen; ELEXPURU, Julen; DIAS, Paulo; SANTOS, Beatriz Sousa
a AMORIM, Paula. Immersive virtual reality for upper limb rehabilitation:
comparing hand and controller interaction. Online. *Virtual Reality*. 2023, roč.
27, č. 2, s. 1157-1171. ISSN 1359-4338. Dostupné z: <https://doi.org/10.1007/s10055-022-00722-7>. [cit. 2023-11-20].
- [33] MATAMALA-GOMEZ, Marta; DONEGAN, Tony; BOTTIROLI, Sara; SAN-
DRINI, Giorgio; SANCHEZ-VIVES, Maria V. et al. Immersive Virtual Reality
and Virtual Embodiment for Pain Relief. Online. *Frontiers in Human Neu-
roscience*. 2019, roč. 13. ISSN 1662-5161. Dostupné z: <https://doi.org/10.3389/fnhum.2019.00279>. [cit. 2023-11-20].
- [34] MATAMALA-GOMEZ, Marta; DIAZ GONZALEZ, Ana M.; SLATER, Mel a
SANCHEZ-VIVES, Maria V. Decreasing Pain Ratings in Chronic Arm Pain
Through Changing a Virtual Body: Different Strategies for Different Pain Ty-
pes. Online. *The Journal of Pain*. 2019, roč. 20, č. 6, s. 685-697. ISSN 15265900.
Dostupné z: <https://doi.org/10.1016/j.jpain.2018.12.001>. [cit. 2023-11-
20].
- [35] WIEBE, Annika; KANNEN, Kyra; SELASKOWSKI, Benjamin; MEHREN,
Aylin; THÖNE, Ann-Kathrin et al. Virtual reality in the diagnostic and therapy
for mental disorders: A systematic review. Online. *Clinical Psychology Review*.
2022, roč. 98. ISSN 02727358. Dostupné z: <https://doi.org/10.1016/j.cpr.2022.102213>. [cit. 2023-11-20].
- [36] NIEDERRITER, Bryant; RONG, Alice; AQLAN, Faisal a YANG, Hui. Sensor-
Based Virtual Reality for Clinical Decision Support in the Assessment of Mental
Disorders. Online. In: *2020 IEEE Conference on Games (CoG)*. IEEE, 2020,
s. 666-669. ISBN 978-1-7281-4533-4. Dostupné z: <https://doi.org/10.1109/CoG47356.2020.9231896>. [cit. 2023-11-21].
- [37] AMBRON, Elisabetta; MILLER, Alexander; CONNOR, Stephanie a BRANCH
COSLETT, H. Virtual image of a hand displaced in space influences action per-
formance of the real hand. Online. *Scientific Reports*. 2020, roč. 10, č. 1. ISSN

- 2045-2322. Dostupné z: <https://doi.org/10.1038/s41598-020-66348-4>. [cit. 2023-11-21].
- [38] LANILLOS, Pablo; FRANKLIN, Sae; MASELLI, Antonella a FRANKLIN, David W. Active strategies for multisensory conflict suppression in the virtual hand illusion. Online. *Scientific Reports*. 2021, roč. 11, č. 1. ISSN 2045-2322. Dostupné z: <https://doi.org/10.1038/s41598-021-02200-7>. [cit. 2024-01-02].
- [39] VICON MOTION SYSTEMS. *Vicon Documentation*. Online. Dostupné z: <https://help.vicon.com/space/Tracker310>. [citováno 2024-05-09]
- [40] EPSON AMERICA INC. *Epson C3 6-Axis Robots*. Online. Dostupné z: https://epson.com/Support/Robots/6-Axis-Robots/C-Series/Epson-C3-6-Axis-Robots/s/SPT_RC3-A601ST1#manuals. [citováno 2024-05-09]
- [41] ŠOPÁK, Petr. *Porovnání 3D multispektrálních snímků lidského těla*. Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2022, 63 s. Vedoucí práce: Ing. Adam Chromý, Ph.D. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/142278>. [cit. 2023-12-27]
- [42] JOHNSON, DAVID E. *Utah Model Repository*. Online. Dostupné z: <https://users.cs.utah.edu/~dejohnso/models/teapot.html>. [citováno 2024-05-09]
- [43] UNITY TECHNOLOGIES. *Unity Documentation*. Online. Dostupné z: <https://docs.unity.com/>. [citováno 2023-12-20].
- [44] HODGEPODGE_FOXE. *TPU Oculus Quest 2 Controller Cover Protectors*. Online. In: UltiMakerThingiverse, 01.02.2021. Dostupné z: <https://www.thingiverse.com/thing:4745565>. [citováno 2024-05-09].
- [45] DMGREGORY. *Converting a quaternion in a right to left handed coordinate system*. Online. In: Game Development, 28.04.2018. Dostupné z: <https://gamedev.stackexchange.com/questions/157946/converting-a-quaternion-in-a-right-to-left-handed-coordinate-system>. [citováno 2024-05-09].
- [46] META QUEST. *Oculus Documentation*. Online. Dostupné z: <https://developer.oculus.com/documentation/unity/>. [citováno 2023-12-20].

- [47] UNITY TECHNOLOGIES. *OpenXR Plugin*. Online. Dostupné z: <https://docs.unity3d.com/Packages/com.unity.xr.openxr@1.9/manual/index.html>. [citováno 2023-12-20].
- [48] CAPTAINCATSPARROW. *50 Free PBR Materials*. Online. In: Unity Asset Store, 06.02.2023. Dostupné z: <https://assetstore.unity.com/packages/2d/textures-materials/50-free-pbr-materials-242760>. [citováno 2023-11-11].
- [49] CASUAL2D. *Wooden Floor Materials*. Online. In: Unity Asset Store, 18.07.2019. Dostupné z: <https://assetstore.unity.com/packages/2d/textures-materials/wood/wooden-floor-materials-150564>. [citováno 2023-11-11].
- [50] 3DFRK. *Picture frames with photos*. Online. In: Unity Asset Store, 11.07.2018. Dostupné z: <https://assetstore.unity.com/packages/3d/props/interior/picture-frames-with-photos-106907>. [citováno 2023-11-11].
- [51] FERAR, Andrey. *Door Free Pack Aferar*. Online. In: Unity Asset Store, 03.07.2019. Dostupné z: <https://assetstore.unity.com/packages/3d/props/interior/door-free-pack-aferar-148411>. [citováno 2023-11-11].
- [52] META. *Oculus Controller Art*. Online. In: Meta Quest Resources, 26.10.2023. Dostupné z: <https://developer.oculus.com/downloads/package/oculus-controller-art/>. [citováno 2023-11-13].
- [53] RRFREELANCE. *First Person Hand*. Online. In: Unity Asset Store, 15.03.2021. Dostupné z: <https://assetstore.unity.com/packages/3d/characters/humanoids/humans/first-person-hand-134028>. [citováno 2023-11-13].
- [54] SCHWIND, Valentin; KNIERIM, Pascal; TASCI, Cagri; FRAN CZAK, Patrick; HAAS, Nico et al. "These are not my hands!". Online. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017, s. 1577-1582. ISBN 9781450346559. Dostupné z: <https://doi.org/10.1145/3025453.3025602>. [cit. 2024-02-15].
- [55] BURES, M.; GORNER, T. a SEDIVA, B. Hand anthropometry of Czech population. Online. In: *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 2015, s. 1077-1082. ISBN 978-1-4673-8066-9. Dostupné z: <https://doi.org/10.1109/IEEM.2015.7385814>. [cit. 2024-02-15].

- [56] MATH.NET. *Math.NET Numerics*. Online. Dostupné z: <https://numerics.mathdotnet.com/>. [citováno 2023-11-17].

A Obsah elektronické přílohy

V přiloženém médiu jsou k dispozici všechny napsané skripty, instance modelů, materiály, scény a importované soubory pro experiment ověření lokalizace Vicon, pro experiment měření přesnosti Oculus Quest 2 a pro aplikaci diagnostikující poruchy propiocepce. Přehled významných souborů je zobrazen pomocí *Dirtree* na dalších stránkách.

V adresáři *experiment 1* se nacházejí soubory pro ověření přesnosti Viconu. Pomocí skriptu *exp1_preprocess.cs* jsou zpracována naměřená data uložená ve složce *data*, která jsou následně exportována do souboru *exportData*.

V adresáři *experiment 2* jsou umístěny všechny soubory potřebné pro ověření lokalizace Oculus Quest 2, včetně aplikací se skripty, spustitelné aplikace (soubor *.exe*), modelů pro 3D tisk a vzorů pro vytvoření ideálního prostředí.

V posledním adresáři, určeném pro aplikaci pro diagnostiku poruch propiocepčního aparátu, lze najít finální verzi nové aplikace, která se spouští pomocí přidruženého souboru *.exe*. Dále zde naleznete složku *Data*, obsahující výstupy z experimentů a dotazníků, a v souboru *Inputs* konfigurační soubor. Tento adresář také obsahuje skripty, scény, modely a další assety potřebné pro implementaci celé aplikace.

- diplomovaPrace.pdf dokument diplomové práce
- experiment1 adresář obsahující soubory k ověření Viconu
 - data změřená data
 - exportData předzpracovaná data
 - exp1_preprocess.m předzpracování změřených dat
 - alignModels.m zarovnání za pomoci ICP
- experiment2 adresář obsahující soubory k ověření OQ2
 - platno.pdf vzor pro tvorbu ideálního prostředí
 - 3dPrint_HMD.stl 3D model pro HMD
 - 3dPrint_leftHand.stl 3D model pro levý ovladač
 - 3dPrint_rightHand.stl 3D model pro pravý ovladač
 - exp2_preprocess předzpracování dat experimentu 2
 - data změřená data
 - exportData předzpracovaná data
 - exp2_preprocess.m předzpracování změřených dat
 - alignModels.m zarovnání za pomoci ICP
 - exp2App build verze aplikace Exp2
 - data změřená data
 - experimentTest.exe spouštěcí .exe soubor
 - exp2UnityProject Unity projekt aplikace Exp2
 - Assets adresář s skripty, materiály, atd.
 - scripts Implementační soubory
 - ExperimentPrecision.cs Hlavní smyčka programu
 - ExportData.cs export změřených dat
 - InputController.cs čtení dat ze zařízení
 - ButtonValue.cs Výběr tlačítka v UI
 - Simulation Vytvoření tras a simulace
 - ConstMovement.cs pohyb objektu po přímce
 - LemniscateMovement.cs pohyb objektu po lemniskátě
 - SphericalSpiralMovement.cs pohyb objektu po sférické spirále
 - generateTracks.c generování dráhy
 - Aplikace adresář obsahující soubory k aplikaci k diagnostice
 - finalApp finální verze aplikace
 - VRexperiment.exe spouštěcí .exe soubor pro aplikaci
 - Data Složka pro ukládání výstupních souborů
 - Inputs Složka s konfiguračními soubory
 - unityProject adresář s projektem aplikace
 - Animator Animace pro obě ruky
 - Imports Použité 2D obrázky a modely
 - Lighting Instance světel
 - Material Instance materiálů
 - Models Modely rukou
 - LeftHand.fbx Výchozí model mužské ruky
 - Scenes Instance scén

- Aplikace.....adresář obsahující soubory k aplikaci k diagnostice
 - unityProject.....adresář s projektem aplikace
 - Scripts.....Implementační soubory
 - VRExperiment.cs.....Hlavní smyčka programu
 - Calibration.cs.....Kalibrace systému
 - ControllerAnimations.cs.....Animace s tlačítky ovladače
 - HandAnimations.....Pohyby prstu s joystickem
 - ExportExperimentData.cs.....Export dat
 - HandFade.cs.....Blednutí modelu
 - tresholdLayerCollision.cs.....Kolize s thresholdem 1
 - SetHandModels.cs.....Nastavení mužské či ženské ruky
 - LoadConfig.cs.....čtení konfiguračního souboru
 - Experiment1.....implementační soubory pro Experimentu 1
 - FirstExperiment.cs.....Stavový diagram Experimentu 1
 - ControllerFixing.cs.....Fixace modelu ruky
 - ObjectCollision.cs.....Realizace kolize s objekty scény
 - HandImageMove.cs.....Pohyby snímku na bedně
 - HandImagePropertity.cs.....Inicializace snímku ruky
 - Experiment2.....Implementační soubory pro Experimentu 2
 - jugment_task.....Realizace úlohy hodnocení
 - TaskJugment.cs.....Hlavní princip hodnotící úlohy
 - JugmentCollision.cs...Kolize se scénou Experimentu - hodnocení
 - SecondExperiment.cs.....Stavový diagram Experimentu 2
 - LightingBehaviour.cs.....Funkce zdrojů světél
 - TableCollision.cs.....Funkce prahu Threshold 2
 - TaskBehaviour.cs.....Náhodnost distrakčních úloh
 - tasks.....Implementační soubory pro Distrakční úlohy
 - Geometry_Task.....Realizace geometrické úlohy
 - taskGeometry.cs.....Hlavní princip geom. úlohy
 - geometryCollision.cs.....Kolize se scénou geom. úlohy
 - Math_Task.....Realizace matematické úlohy
 - taskMath.cs.....Hlavní princip matem. úlohy
 - MathCollision.cs.....Kolize se scénou matem. úlohy
 - UI.....Implementační soubory pro uživatelské rozhraní
 - DisableButton.cs.....Deaktivace tlačítek
 - outputQuestionaire.cs.....Export dotazníku a parametrů
 - NavigationUI.cs.....Postup vyplňování za pomoci tlačítek
 - InfoCanvasBehaviour.cs.....Změna instancí v UI2