

# Upravljanje prikazom sadržaja na zaslonu pomoću gesti ruku

---

**Maričević, Mario**

**Master's thesis / Diplomski rad**

**2023**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:190:713790>

*Rights / Prava:* [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2024-07-14**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Sveučilišni diplomski studij računarstva

Diplomski rad

**UPRAVLJANJE PRIKAZOM SADRŽAJA NA ZASLONU  
POMOĆU GESTI RUKU**

Rijeka, ožujak 2023.

Mario Maričević

0069082450

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Sveučilišni diplomski studij računarstva

Diplomski rad

**UPRAVLJANJE PRIKAZOM SADRŽAJA NA ZASLONU  
POMOĆU GESTI RUKU**

Mentor: izv. prof. dr. sc. Sandi Ljubić

Rijeka, ožujak 2023.

Mario Maričević

0069082450

Rijeka, 3. ožujka 2022.

Zavod: **Zavod za računarstvo**  
Predmet: **Napredna korisnička sučelja**  
Grana: **2.09.01 arhitektura računalnih sustava**

## ZADATAK ZA DIPLOMSKI RAD

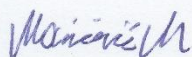
Pristupnik: **Mario Maričević (0069082450)**  
Studij: **Diplomski sveučilišni studij računarstva**  
Modul: **Računalni sustavi**

Zadatak: **Upravljanje prikazom sadržaja na zaslonu pomoću gesti ruku / Controlling the Content Display on a Screen via Hand Gestures**

### Opis zadatka:

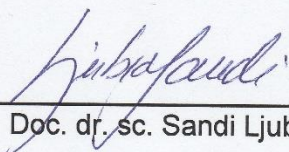
U sklopu diplomskog rada potrebno je implementirati sustav za raspoznavanje različitih gesti ruku koje se izvode ispred zaslona računala. Sustav treba podržati više različitih modaliteta interakcije koji se zasnivaju na zračnim gestama i omogućiti upravljanje prikazom sadržaja na zaslonu. Operacije koje treba podržati modalitetima interakcije su promjena veličine sadržaja (engl. zooming) i navigiranje sadržajem (engl. panning). Predložene modalitete interakcije treba komparativno analizirati, provodeći odgovarajući eksperiment sa stvarnim korisnicima. Posebno pažnju treba usmjeriti na učinkovitost i opterećenje pri radu s predloženim modalitetima interakcije.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.



Zadatak uručen pristupniku: 21. ožujka 2022.

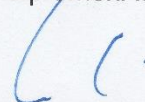
Mentor:



---

Doc. dr. sc. Sandi Ljubić

Predsjednik povjerenstva za  
diplomski ispit:



---

Prof. dr. sc. Kristijan Lenac

## **IZJAVA**

Ja, Mario Maričević, izjavljujem da sam diplomski rad „Upravljanje prikazom sadržaja na zaslonu pomoću gesti ruku“ izradio samostalno, koristeći literaturu i primjenjujući usvojeno znanje koje sam stekao pohađajući sveučilišni prijediplomski i diplomski studij računarstva na Tehničkom fakultetu u Rijeci, uz pomoć mentora izv. prof. dr. sc. Sandija Ljubića.



# SADRŽAJ

|  |    |
|--|----|
| 1. UVOD .....  | 1  |
| 2. GESTOVNA SUČELJA .....  | 2  |
| 2.1 Općeniti princip rada gestovnih sučelja.....                                 | 2  |
| 2.2 Karakteristike gestovnih sučelja .....                                       | 3  |
| 2.3 Primjeri .....   | 5  |
| 2.4 Prednosti i nedostaci.....   | 10 |
| 3. SUSTAV ZA UPRAVLJANJE PRIKAZOM SADRŽAJA NA ZASLONU POMOĆU<br>GESTI RUKU ..... | 12 |
| 3.1 Modaliteti upravljanja.....  | 13 |
| 3.1.1 MH modalitet .....   | 14 |
| 3.1.2 SH modalitet.....  | 17 |
| 3.2 Implementacija .....   | 20 |
| 3.2.1 MediaPipe.....   | 20 |
| 3.2.2 Image-viewer.....  | 24 |
| 3.2.3 Komunikacija između modula .....   | 26 |
| 3.3 Instalacija sustava i postavljanje u rad .....                               | 28 |
| 3.3.1 Linux .....  | 28 |
| 3.3.2 Windows.....   | 32 |
| 4. HCI EKSPERIMENT .....   | 39 |
| 4.1 Dizajn eksperimenta .....  | 40 |
| 4.2 Zadaci upravljanja prikazom sadržaja na zaslonu .....                        | 41 |
| 4.3 Ispitni korisnici .....  | 45 |
| 4.4 Oprema i uvjeti za provedbu eksperimenta .....                               | 46 |
| 4.5 Procedura provedbe eksperimenta.....   | 47 |
| 5. REZULTATI.....  | 51 |
| 5.1 Modalitet.....   | 52 |
| 5.2 Složenost.....   | 53 |
| 5.3 Modalitet i složenost.....   | 54 |
| 5.4 Upotrebljivost, radno opterećenje i preferirani modalitet .....              | 55 |
| 6. ZAKLJUČAK .....   | 60 |
| 7. LITERATURA .....  | 62 |
| 8. SAŽETAK .....   | 65 |
| 9. POPIS OZNAKA I KRATICA.....   | 67 |

## POPIS SLIKA

|   |    |
|---|----|
| Slika 2.1. Princip rada sustava s gestovnom interakcijom .....                                  | 2  |
| Slika 2.2. Slikoviti gestovni dijagrami kao upute za korištenje sustava, preuzeto iz [2] .....  | 5  |
| Slika 2.3. Samoposlužujući kiosk u McDonalds-u, preuzeto iz [3] .....                           | 5  |
| Slika 2.4. „The Clapper“, preuzeto iz [4] .....   | 6  |
| Slika 2.5. Slavina s infracrvenim senzorom, preuzeto iz [5] .....                               | 6  |
| Slika 2.6. Nintendo Wii igrača konzola, preuzeto iz [6] .....                                   | 7  |
| Slika 2.7. Nintendo Wii igrača konzola kao dio procesa rehabilitacije, preuzeto iz [7] .....    | 7  |
| Slika 2.8. Gestovno sučelje za rehabilitaciju, preuzeto iz [8] .....                            | 8  |
| Slika 2.9. Microsoft Kinect u operacijskoj sali, preuzeto iz [9] .....                          | 8  |
| Slika 2.10. Gestovno upravljanje slikovnim sadržajem na zaslonu, preuzeto iz [10] .....         | 9  |
| Slika 2.11. Gestovno sučelje u automobilu, preuzeto iz [11] .....                               | 9  |
| Slika 2.12. Usporedba dodirne tipkovnice na zaslonu i fizičke na mobilnom uređaju .....         | 10 |
| Slika 2.13. Višekorisnički rad s gestovnim sučeljem, preuzeto iz [12] .....                     | 11 |
| Slika 2.14. Gestovno upravljanje automobilom unutar videoigre, preuzeto iz [13] .....           | 11 |
| Slika 3.1. Princip rada Pan&Zoom sustava .....  | 12 |
| Slika 3.2. Početak i završetak akcija skaliranja i pomicanja .....                              | 13 |
| Slika 3.3. Dijagram stanja MH modaliteta .....  | 14 |
| Slika 3.4. Stanje mirovanja u MH modalitetu .....   | 14 |
| Slika 3.5. Stanje pomicanja u MH modalitetu .....   | 15 |
| Slika 3.6. Stanje skaliranja u MH modalitetu .....  | 16 |
| Slika 3.7. Umanjivanje slikovnog sadržaja u MH modalitetu .....                                 | 16 |
| Slika 3.8. Uvećavanje slikovnog sadržaja u MH modalitetu .....                                  | 16 |
| Slika 3.9. Dijagram stanja SH modaliteta .....  | 17 |
| Slika 3.10. Stanje mirovanja u SH modalitetu .....  | 18 |
| Slika 3.11. Stanje pomicanja u SH modalitetu .....  | 18 |
| Slika 3.12. Uvećavanje slikovnog sadržaja u SH modalitetu .....                                 | 19 |
| Slika 3.13. Umanjivanje slikovnog sadržaja u SH modalitetu .....                                | 19 |
| Slika 3.14. Gotova rješenja strojnog učenja unutar radnog okvira MediaPipe, preuzeto iz [14] .. | 21 |
| Slika 3.15. Ključne točke/orijentiri dlana ruke, preuzeto iz [17] .....                         | 22 |
| Slika 3.16. Grafičko sučelje MediaPipe modula .....   | 23 |
| Slika 3.17. Grafičko sučelje Image-viewer modula .....  | 25 |
| Slika 3.18. Primjeri predefiniраниh macro oznaka za različite operacijske sustave .....         | 27 |
| Slika 3.19. Instalacija Qt softvera na Linux operacijskom sustavu .....                         | 31 |



|   |    |
|---|----|
| Slika 3.20. Prikaz sistemskih varijabli okoline .....                                     | 33 |
| Slika 3.21. Instalacija potrebnih Visual C++ alata u operacijskom sustavu Windows.....    | 35 |
| Slika 3.22. Instalacija Qt softvera u Windows operacijskom sustavu .....                  | 37 |
| Slika 4.1. Dizajn HCI eksperimenta.....   | 40 |
| Slika 4.2. Primjer slike jednostavnog zadatka.....  | 41 |
| Slika 4.3. Primjer slike umjereno složenog zadatka.....                                   | 41 |
| Slika 4.4. Primjer slike složenog zadatka.....  | 42 |
| Slika 4.5. Tekstualna datoteka s RGB definicijom korištenih boja u odnosnom zadatku ..... | 43 |
| Slika 4.6. Prikaz mape pictures u Pan&Zoom sustavu .....                                  | 43 |
| Slika 4.7. Tekstualna datoteka koja definira poredak zadataka .....                       | 44 |
| Slika 4.8. Iskustvo ispitnih korisnika s gestovnom interakcijom (N=30) .....              | 45 |
| Slika 4.9. Problem položaja ruku izvan područja kamere.....                               | 46 |
| Slika 4.10. Uspješno skaliran pravokutnik na slici zadatka .....                          | 47 |
| Slika 4.11. Vizualne oznake za praćenje tijeka eksperimenta.....                          | 48 |
| Slika 4.12. Ispisivanje rezultata o izvođenju zadatka u komandnu liniju .....             | 49 |
| Slika 4.13. Korisnička CSV datoteka s rezultatima završenih zadataka .....                | 49 |
| Slika 5.1. Prosječno vrijeme izvršavanja zadatka po modalitetima .....                    | 52 |
| Slika 5.2. Prosječno vrijeme izvršavanja kompletnog ciklusa zadataka po modalitetima..... | 52 |
| Slika 5.3. Prosječno vrijeme izvršavanja zadataka s obzirom na složenost.....             | 54 |
| Slika 5.4. Prosječno vrijeme izvršavanja zadataka po modalitetima i složenosti .....      | 55 |
| Slika 5.5. Atributi upotrebljivosti korištenih modaliteta.....                            | 56 |
| Slika 5.6. Radno opterećenje interakcije (NASA TLX).....                                  | 57 |
| Slika 5.7. Preferirani modalitet (N = 30).....  | 59 |

## **POPIS TABLICA**

|   |    |
|---|----|
| Tablica 5.1. Post hoc analiza s Bonferronijevom korekcijom - modalitet .....                    | 53 |
| Tablica 5.2. Post hoc analiza s Bonferronijevom korekcijom - složenost .....                    | 54 |
| Tablica 5.3. Prosječno vrijeme izvršavanja zadataka prema modalitetima i složenosti .....       | 55 |
| Tablica 5.4. Prosječne ocjene i rezultati Wilcoxon's Signed Rank testova - upotrebljivost ..... | 57 |
| Tablica 5.5. Prosječne ocjene i rezultati Wilcoxon's Signed Rank testova – radno opterećenje .. | 58 |

# 1. UVOD

U današnje vrijeme dolazi do znatnog napretka i razvoja u području tehnologije. Pojavljuju se razni novi uređaji poput pametnih robotskih usisavača (npr. Roomba) i naprednih pametnih telefona. Tehnologija virtualne stvarnosti (VR, engl. Virtual Reality) postaje sve naprednija, a aplikacije i sustavi velikih kompanija poput Google-a, Microsoft-a, Mete i drugih svakodnevno se ažuriraju kako bi njihove usluge bile brže, jednostavnije, interesantnije i intuitivnije za korištenje.

Uspješnim razvijanjem novih značajki (engl. features) sustava ili ažuriranjem postojećih, kompanije prolaze kroz vremenski period prikupljanja podataka i povratnih informacija njihovih korisnika. Iz prikupljenih podataka dobivaju odgovore na razna interesantna pitanja poput: „Ispunjava li nova značajka/verzija softvera korisnička očekivanja?“, „Što je potrebno promijeniti kako bi korisnici bili zadovoljniji sustavom?“, „Postoje li greške u radu softvera koje je potrebno ispraviti?“ i sl. Nad prikupljenim podacima kompanije provode i razne statističke analize kako bi utvrdile je li nova značajka/verzija softvera značajno bolja i učinkovitija u odnosu na prethodnu te hoće li se ona tada zadržati ili izbaciti kako bi sustav, u konačnici, bio što prilagođeniji korisniku (engl. user friendly). Jedna od popularnijih značajki novijih sustava/aplikacija zasigurno je i gestovna interakcija čovjeka i računala, odnosno uvođenje gestovnih sučelja.

U ovome radu osmišljen je i realiziran sustav temeljen na gestovnoj interakciji. Implementirani sustav nosi ime Pan&Zoom, a njegov zadatak je upravljanje prikazom sadržaja na zaslonu pomoću gesti ruku kroz dvije moguće akcije: pomicanje (engl. pan) i skaliranje (engl. zoom). Sustav je namijenjen stolnim i prijenosnim računalima s operacijskim sustavima Linux ili Windows, a sastoji se od dvaju modula: MediaPipe i Image-viewer, koji međusobno komuniciraju putem utičnice (engl. socket). MediaPipe modul koristi se za prepoznavanje i kontrolu gesti, dok Image-viewer predstavlja grafičko sučelje za prikaz slikovnog sadržaja kojim se želi upravljati. Za izvedbu navedenih akcija osmišljena su i implementirana dva različita modaliteta. Proveden je HCI eksperiment u kojem su ispitni korisnici (N = 30) trebali izvršiti određene zadatke različitih složenosti u što kraćem vremenu koristeći oba modaliteta te popuniti odgovarajuće upitnike. Provedbom eksperimenta i popunjavanjem upitnika dobiveni su podaci koji su podvrgnuti statističkoj analizi s ciljem izlučivanja zaključaka po pitanju rada sustava, korištenih modaliteta i složenosti zadataka.

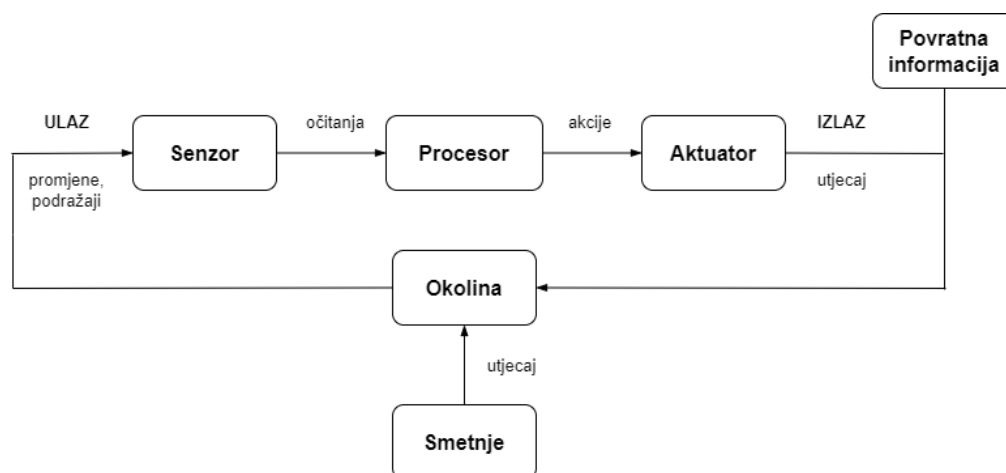
## 2. GESTOVNA SUČELJA

Gestovna sučelja podvrsta su korisničkih sučelja u kojima se interakcija između čovjeka i računala odvija odgovarajućim gestama. U ovome kontekstu, gesta je bilo kakva fizička aktivnost koju računalo percipira i na nju odgovara provedbom sukladne akcije. Neki od primjera gesti mogu biti pomicanje ruke, noge, glave, dodir, podizanje obrve, određeni pokret tijelom ili sl. S obzirom na široki izbor gesti, a time i raznih interakcija sa sustavom, gestovna sučelja sve se više razvijaju te postaju vrlo korisna i popularna u društvu.

Postoji mnogo različitih gestovnih sučelja, no većina njih može se svrstati u iduće dvije kategorije: gestovna sučelja zaslona osjetljivih na dodir (TUI, engl. Touch User Interface) i gestovna sučelja slobodnog oblika (FFGUI, engl. Free-form Gestural User Interface). Gestovna sučelja zaslona osjetljivih na dodir zahtijevaju fizičku interakciju zbog čega imaju ograničen izbor gesti za kontrolu sustava (npr. pritisak i pomaci prstiju), dok kod gestovnih sučelja slobodnog oblika takvo ograničenje ne postoji. U gestovnim sučeljima slobodnog oblika mogu se koristiti gestikulacije čitavim tijelom, razni kontroleri ili pametne rukavice za interakciju sa sustavom što omogućuje veliki izbor gesti odnosno mogućih akcija [1].

### 2.1 Općeniti princip rada gestovnih sučelja

Svi sustavi koji koriste gestovnu interakciju većinom se sastoje od nekoliko povezanih elemenata prikazanih slikom 2.1, a to su: okolina, smetnje, senzori, procesor, aktuatori te povratne informacije.



Slika 2.1. Princip rada sustava s gestovnom interakcijom

Radni ciklus sustava s gestovnom interakcijom započinje osluškivanjem okoline pomoću senzora. Senzori (ili osjetila) su elektroničke komponente koje služe za detekciju promjena koje se događaju u okolini sustava. U ovakvim sustavima najviše se koriste idući senzori: senzor tlaka za detekciju pritiska (dodira) na zaslonu, senzori blizine, nagiba, pokreta i orijentacije koji otkrivaju prisutnost, položaj i brzinu objekata u prostoru (npr. pokret ruke) te senzori jačine zvuka i intenziteta svjetla. Okolinu sustava predstavljaju ljudi koji gestama ili dodirima žele njime upravljati.

Uspješnom detekcijom promjene u okolini (npr. gesta, dodir), senzor prenosi informaciju/očitanje prema procesoru. Zadatak procesora je odlučiti što će napraviti s dobivenom informacijom odnosno interpretirati određenu gestu ili dodir u odgovarajuću akciju. Odgovarajuću akciju izvršava aktuator te tada sustav prelazi u novo stanje (izlaz). Na temelju novog stanja sustava, čovjek odlučuje koja će promjena biti idući ulaz sustava te se tako dobiva jedna povezana cjelina. U digitalnim sustavima procesor i aktuator objedinjeni su unutar softvera, dok je kod mehaničkih sustava aktuator odvojen od procesora (npr. sklopka, električni motor).

Broj radnih ciklusa ovisi o složenosti sustava. Jednostavniji gestovni sustavi mogu imati, u idealnom slučaju, samo jedan radni ciklus kako bi se izvršio određeni zadatak (npr. pljesak rukama) dok je kod složenijih sustava potrebno i do nekoliko ciklusa (npr. kupnja hrane na samoposlužnim kioscima). Na broj radnih ciklusa utječe i mnogo drugih faktora poput vanjskih smetnji, ispravnosti rada i očitavanja senzora te čovjekovog poznavanja rada sustava. Kako bi se smanjila mogućnost pogreške rada sustava, a time i broj dodatnih radnih ciklusa, potrebno je provjeriti ispravnost rada senzora te provesti njihovu kalibraciju ovisno o namjeni sustava. Također, čovjek bi se trebao bolje upoznati s radom sustava kako bi ga koristio na ispravan način odnosno uz što manje pogrešaka.

## **2.2 Karakteristike gestovnih sučelja**

U današnje vrijeme razvija se veliki broj gestovnih sučelja različitih namjena. U procesu razvoja poželjno je poznavati i uzeti u obzir karakteristike koje će gestovno sučelje učiniti boljim, jednostavnijim i intuitivnijim za korištenje. Ponekad je vrlo teško zadovoljiti sve tražene karakteristike s obzirom na namjenu i složenost sustava, međutim uvijek je dobro njima težiti.

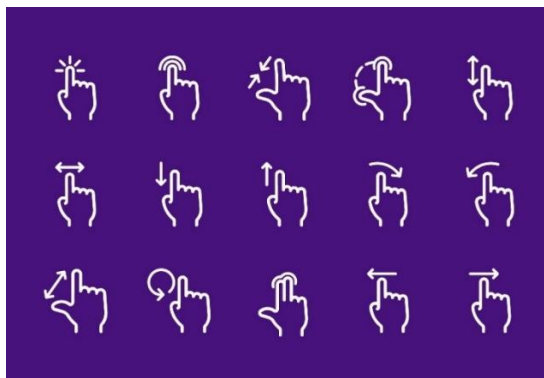
Namjena odnosno svrha sustava je karakteristika o kojoj je potrebno razmišljati u početku razvoja. Definiranje problema koje će sustav rješavati i stvaranje plana implementacije vrlo su važni koraci pri samom stvaranju sustava. Bez jasno definiranih zadataka sustav, odmah u početku, gubi svoju svrhu, a njegova buduća upotreba dovodi se u pitanje.

Iduća stavka na koju je potrebno obratiti pozornost prije razvijanja i postavljanja samog sustava u rad jest situacija odnosno kontekst u kojem će se on koristiti. Gestovni sustavi koji uključuju mahanje rukama te velike pomake tijelom zasigurno nisu prikladni za manje prostore, gdje se kreće veliki broj ljudi poput hodnika, bolnica ili trgovina brzom hranom. Također, geste u različitim kulturama ili zemljama imaju različito značenje. Zanimljivo je kako gesta kojom se nekome želi reći da je sve u redu, a izvodi se formiranjem kruga pomoću palca i kažiprsta ruke, ima različito značenje u pojedinim zemljama. U Turskoj, Grčkoj ili Rusiji navedena gesta predstavlja uvredu, te je stoga važno prilikom izrade gestovnog sustava obratiti pozornost i na izbor gesti kojima će se upravljati sustavom.

Ljudske geste odnosno željene akcije, posebice u digitalnim sustavima, trebale bi biti popraćene povratnom informacijom kako bi čovjek znao je li sustav uspješno primio i izvršio akciju te u skladu s ishodom planirao idući korak u radu. Povratne informacije mogu biti različiti vizualni ili zvukovni efekti, poruke na zaslonima sustava i sl. Poželjno je da povratne informacije pristižu u što kraćem roku, a ne s velikim zakašnjenjem jer će korisnici u nedostatku povratnih informacija pokušati ponoviti željene akcije (geste, dodire na zaslonu), čime mogu izazvati neočekivane probleme u radu sustava. Za vremenski duže akcije bilo bi poželjno da sustav sadrži neku vrstu trake za napredak (engl. progress bar) kako bi se korisniku dalo do znanja da se ona trenutno obrađuje.

Brzina odgovora ili odziv sustava jedna je od najvažnijih karakteristika gestovnih sučelja. Odziv sustava znatno ovisi o odzivu senzora pa je važno izvršiti kalibraciju i provjeru ispravnosti. Spora reakcija sustava dovodi do nezadovoljstva i lošeg iskustva korisnika. Na odziv sustava utječe i njegova implementacija. Pametni digitalni sustavi brzo računaju i donose odluke te pamte i automatski prepoznaju geste koje korisnik pokušava izvesti i prije nego su one uspješno izvedene. Prilikom implementacije treba uzeti u obzir i estetski dizajn sučelja koji također doprinosi zadovoljstvu korisnika. Kako postoji veliki broj različitih gestovnih sustava vrlo je važno priložiti i upute za njegovo korištenje. Što je sustav složeniji to su upute važnije i potrebnije kako bi se korisnik u što kraćem vremenu naučio koristiti sustavom.

Upute za korištenje mogu biti prikazane u obliku obogaćenog teksta ili malih slikovitih gestovnih dijagrama kao na slici 2.2. Slikoviti dijagrami također doprinose estetskom dizajnu sučelja.



Slika 2.2. Slikoviti gestovni dijagrami kao upute za korištenje sustava, preuzeto iz [2]

### 2.3 Primjeri

Gestovna sučelja ubrzano se razvijaju i pojavljuju u raznim područjima, a njihova široka primjena biti će prikazana kroz idućih nekoliko primjera. Primjeri se, osim u načinu korištenja, razlikuju i u samoj vrsti gestovnog sučelja.

Na slici 2.3 prikazan je samoposlužujući kiosk za kupnju hrane u McDonalds-u, jednom od najvećih lanaca brze hrane. Od uvođenja ovih gestovnih sučelja sa zaslonom osjetljivim na dodir, McDonalds bilježi znatno veći profit, gužve u njihovim restoranima znatno su smanjene, usluga je brža, a broj kupaca u znatnom je porastu.



Slika 2.3. Samoposlužujući kiosk u McDonalds-u, preuzeto iz [3]

Idući primjer, razvijen davne 1985. godine, koristi se za automatizaciju doma i temelji se na gestovnoj interakciji. Uređaj na slici 2.4 naziva se „The Clapper“, a njime je moguće kontrolirati rad (paljenje/gašenje) dvaju uređaja istovremeno pomoću geste pljeskanja. Dvostrukim pljeskom vrši se kontrola prvog, a trostrukim drugog uređaja. Pljeskanje je potrebno izvoditi naizmjenično s kratkim pauzama kako bi sustav na pravilan način detektirao kojim uređajem se želi upravljati. U ovom mehaničkom sustavu koristi se mikroskop pomoću kojeg kontroler, na temelju broja pljesaka, odlučuje koji prekidač spustiti/podignuti kako bi upalio/ugasio uređaj.



Slika 2.4. „The Clapper“, preuzeto iz [4]

Još jednu vrstu mehaničkog gestovnog sustava moguće je vidjeti na slici 2.5. Slavine koje koriste infracrveni senzor za detekciju ruku znatno smanjuju nepotrebnu potrošnju vode te omogućuju bolje higijenske uvjete, posebice na javnim mjestima, jer korisnici ne ostvaruju direktan dodir sa slavinom.



Slika 2.5. Slavina s infracrvenim senzorom, preuzeto iz [5]



Gestovna sučelja mogu biti vrlo interesantna i zabavna za korištenje. Primjer na slici 2.6 prikazuje dječaka kako se zabavlja igrajući golf na poznatoj igraćoj konzoli Nintendo Wii. U dječakovim rukama nalazi se igraća palica pomoću koje se vrši interakcija s konzolom. Dječak svojom gestom simulira udarac golf palicom koji se zatim prenosi i prevodi u odgovarajuće akcije na konzoli.



*Slika 2.6. Nintendo Wii igraća konzola, preuzeto iz [6]*

Videoigre se, osim zabave, na ovoj konzoli mogu koristiti i u rehabilitacijske svrhe. Osim golfa moguće je igrati i druge različite sportove poput košarke, tenisa, stolnog tenisa i sl. Na slici 2.7 nalazi se čovjek starije dobi koji u sklopu rehabilitacije pokušava vratiti svoju pokretljivost igrajući tenis na Wii konzoli, a pritom izgleda prilično raspoložen i zadovoljan.



*Slika 2.7. Nintendo Wii igraća konzola kao dio procesa rehabilitacije, preuzeto iz [7]*

Još jedan, malo drugačiji, primjer korištenja sustava s gestovnom interakcijom u svrhu rehabilitacije moguće je vidjeti na slici 2.8. Naginjanjem svoga tijela osoba pokušava uspješno sletjeti padobranom u za to predviđeno područje. Interakcija između čovjeka i konzole odvija se korištenjem posebne crvene rukavice.



*Slika 2.8. Gestovno sučelje za rehabilitaciju, preuzeto iz [8]*

Svoju primjenu, osim u fizikalnoj terapiji, gestovna sučelja pronalaze i u preostalim područjima medicine. Vrlo se često koriste u operacijskim salama kako osoblje ne bi dodirivalo zaslone i time ugrožavalo sterilnost koja je od iznimne važnosti za zdravlje pacijenta. Na slici 2.9 može se vidjeti kako medicinska djelatnica gestama ruku pomiče sliku na zaslonu pomoću uređaja naziva Kinect. Navedeni uređaj proizvod je tvrtke Microsoft i služi upravo za kontrolu i prevođenje gestovnih pokreta u adekvatne akcije za upravljanje sadržajem na zaslonu.



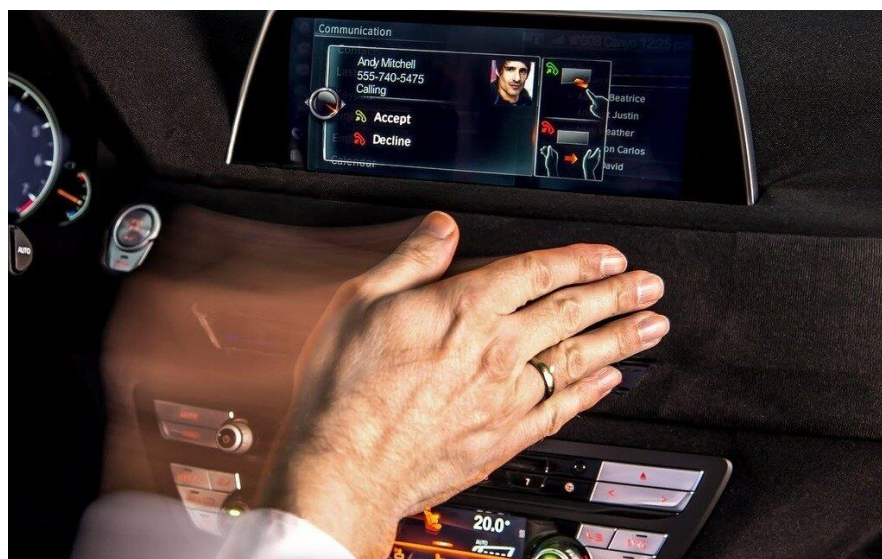
*Slika 2.9. Microsoft Kinect u operacijskoj sali, preuzeto iz [9]*

Primjer na slici 2.10 prikazuje osobu koja pomoću geste oponašanja rada dvogleda uvećava sliku na zaslonu. Ovakvi primjeri upravljanja sadržajem na zaslonu mogu se koristiti u detaljnim slikovnim analizama ili u prezentacijama kako bi dodatno privukle ljudsku pozornost.



*Slika 2.10. Gestovno upravljanje slikovnim sadržajem na zaslonu, preuzeto iz [10]*

Posljednji izdvojeni primjer dolazi iz automobilske industrije. Gestovna sučelja zasigurno su budućnost automobila jer omogućuju bržu i jednostavniju interakciju s njihovim sustavom te na taj način ostavljaju fokus vozača na samoj vožnji. U primjeru na slici 2.11 može se vidjeti kako vozač jednostavnom gestom ruke odbija dolazni poziv. Gestama u automobilu moguće je mijenjati radio kanale, uvećavati, umanjivati i pomicati navigacijsku kartu i sl.



*Slika 2.11. Gestovno sučelje u automobilu, preuzeto iz [11]*

## 2.4 Prednosti i nedostatci

Sustavi s gestovnom interakcijom, kao i svi ostali, imaju svojih prednosti i nedostataka. Iz navedenih primjera može se zaključiti kako je njihova velika prednost brza i jednostavna interakcija korisnika i računalnog sustava. Navedenu prednost najbolje je opisati kroz zadatak utipkavanja poruke na mobilnom uređaju koristeći tipkovnice na slici 2.12. Utipkavanje poruke pomoću tipkovnice na zaslonu osjetljivim na dodir znatno je brže i efikasnije nego utipkavanje poruke pomoću pritiskanja fizičkih tipki na mobilnom uređaju. S većom brzinom i jednostavnošću sustava raste i zadovoljstvo njegovim korištenjem.



*Slika 2.12. Usporedba dodirne tipkovnice na zaslonu i fizičke na mobilnom uređaju*

Ljudi uobičajeno preferiraju direktnu interakciju (dodir) s fizičkim objektima u prostoru. Digitalni gestovni sustavi omogućuju takvu vrstu prirodne interakcije s objektima na zaslonu pomoću dodira. Za rad takvih sustava nije potrebno koristiti dodatno sklopovlje poput miševa, tipkovnica, gumbova i drugih ulaznih uređaja. Minimalni hardverski zahtjevi omogućuju im primjenu na mjestima gdje bi upotreba tradicionalnih računala bila nepraktična poput automobila, zračnih luka, muzeja i drugih javnih površina gdje se kreće veliki broj ljudi.

Gestovne sustave moguće je izmijeniti na različite načine ovisno o željenim funkcionalnostima. Moguće je promijeniti veličinu zaslona osjetljivog na dodir, izgled grafičkog sučelja i korištene senzore te na taj način u potpunosti prenamijeniti sustav. Primjer navedene fleksibilnosti može biti softverska promjena pozicije i izgleda gumba na grafičkom sučelju sustava koja je znatno jednostavnija od zamjene fizičkih tipki za interakciju u tradicionalnim sustavima.

Korištenje digitalnih gestovnih sustava vrlo je zabavno i interesantno, što se najbolje može vidjeti iz primjera igranja videoigara s društvom na slici 2.13. Igranje videoigara direktnim fizičkim pokretima odnosno gestama znatno je zanimljivije od igranja videoigara korištenjem tipkovnice ili igraće palice.



*Slika 2.13. Višekorisnički rad s gestovnim sučeljem, preuzeto iz [12]*

Preveliko oslanjanje na vizualnost jedan je od nedostataka gestovnih sučelja. Pri izvođenju akcija gestama izostaje fizički osjećaj kao potvrda interakcije sustavom (npr. pritiskanje i otpuštanje fizičkog gumba). Povratne informacije gestovnih sustava u velikoj većini su vizualnog tipa i na njih je potrebno čekati određeno vrijeme koje ovisi o implementaciji sustava i kalibraciji senzora. Također, isključivo vizualne povratne informacije mogu predstavljati problem slabovidnim osobama prilikom korištenja sustava.

Za dugotrajan rad sa složenijim gestovnim sučeljima slobodnog oblika (interakcija gestama) potreban je veći fizički i psihički napor za razliku od tradicionalnih sustava gdje se koriste ulazni periferni uređaji poput miša i tipkovnice, što može predstavljati jedan od nedostataka ovakvih sustava. Slika 2.14 prikazuje osobu koja gestama upravlja automobilom u videoigri. Upravljanje može postati vrlo naporno nakon dužeg vremena igre zbog konstantnog držanja ruku u zraku.



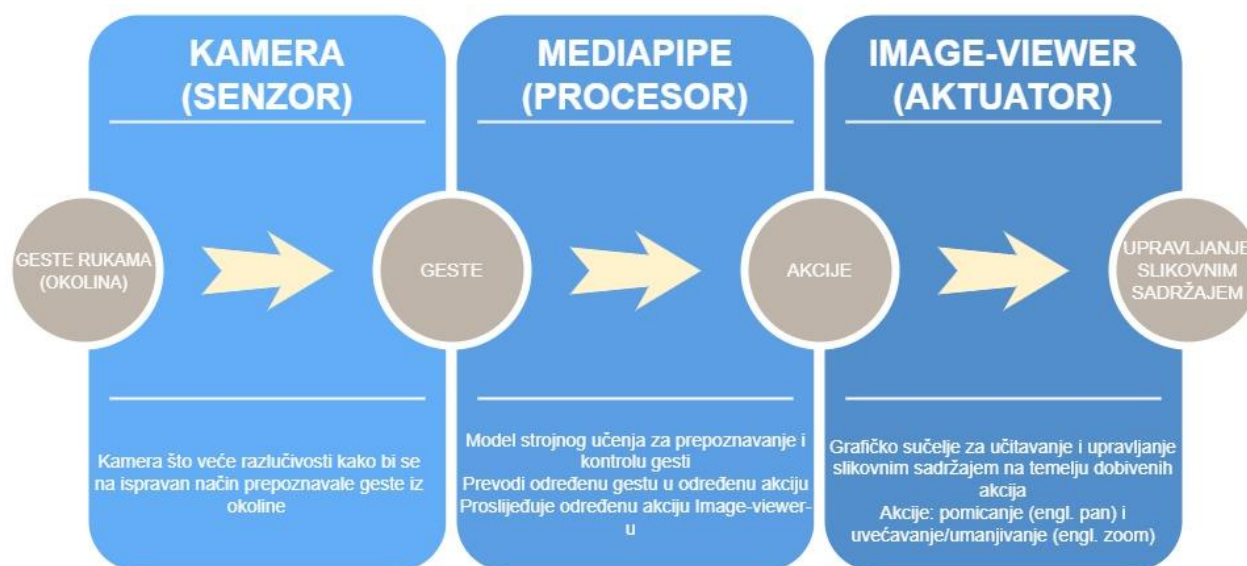
*Slika 2.14. Gestovno upravljanje automobilom unutar videoigre, preuzeto iz [13]*

Geste u ovakvim sustavima nisu univerzalne odnosno ne predstavljaju istu akciju u različitim sustavima. Iz tog razloga, korisnicima je potrebno dodatno vrijeme prilagodbe i upoznavanja s novim sustavom.

### 3. SUSTAV ZA UPRAVLJANJE PRIKAZOM SADRŽAJA NA ZASLONU POMOĆU GESTI RUKU

Široka primjena, lagana i brza interakcija te zanimljivi princip rada potaknuli su razvoj digitalnog gestovnog sustava slobodnog oblika naziva Pan&Zoom, s ciljem upravljanja prikazom sadržaja na zaslonu pomoću gesti ruku. Sustav se sastoji od kamere koja prati korisničke geste i dvaju programskih modula: MediaPipe i Image-viewer koji međusobno komuniciraju putem utičnice (engl. socket). MediaPipe modul koristi se za prepoznavanje i kontrolu gesti dok Image-viewer sadrži grafičko sučelje za učitavanje i upravljanje slikovnim sadržajem. Kao što je moguće zaključiti iz samog naziva sustava, slikovnim sadržajem, unutar grafičkog sučelja, moguće je upravljati akcijama pomicanja (engl. Pan) i skaliranja (engl. Zoom), a sam princip rada sustava moguće je vidjeti na slici 3.1.

Svaka akcija odvija se izvršavanjem određene geste. Jednom kada MediaPipe modul prepozna određenu gestu, prenosi se informacija o odgovarajućoj akciji putem komunikacijskog kanala Image-viewer modulu koji, na temelju akcije, upravlja slikovnim sadržajem. Akcije pomicanja i skaliranja slikovnog sadržaja provode se simulacijom radnja mišem. Pomicanje po slici uobičajeno se odvija pritiskom i zadržavanjem lijevog klika miša i njegovim pomicanjem po podlozi, dok se skaliranje vrši pomicanjem kotačića na mišu prema gore ili dolje (engl. Scroll up/down). Navedeni princip rada sustava konstantno se ponavlja dok se ne terminiraju navedeni programski moduli.

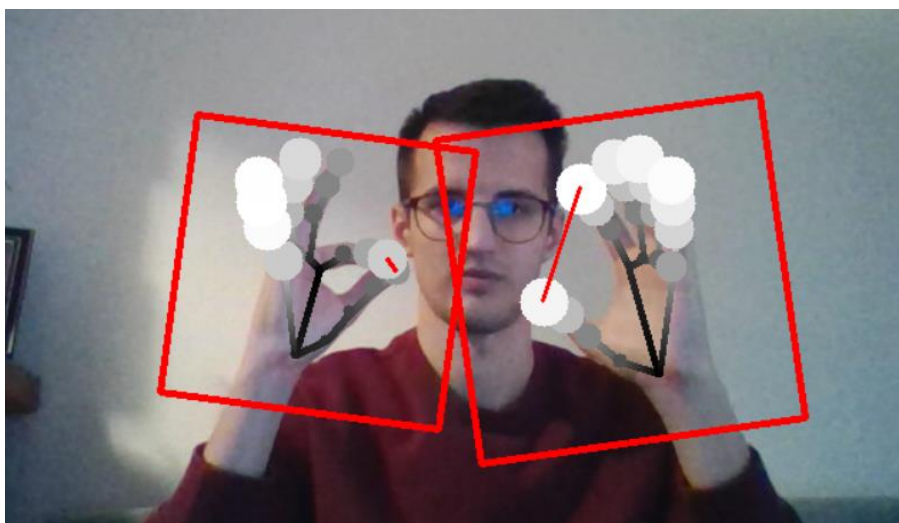


Slika 3.1. Princip rada Pan&Zoom sustava

### 3.1 Modaliteti upravljanja

U sklopu razvoja Pan&Zoom sustava implementirana su dva različita modaliteta odnosno načina upravljanja slikovnim sadržajem: jednoručno skaliranje (SH, engl. Single hand zoom) i dvoručno skaliranje (MH, engl. Multi hand zoom). Modaliteti se, osim u potrebnom broju ruku, razlikuju i u načinu gestikulacije za akciju skaliranja. Akcija pomicanja odvija se na isti način odnosno istom gestom u oba modaliteta. Za izvođenje navedene geste se u MH modalitetu može koristiti bilo koja ruka, dok kod SH modaliteta to mora biti isključivo lijeva ruka.

Način započinjanja i završavanja akcija skaliranja i pomicanja prikazan je na slici 3.2. U oba modaliteta akcije skaliranja i pomicanja uvijek započinju spajanjem jagodica palca i kažiprsta na jednoj ili obje ruke (lijevi dio slike), ovisno o modalitetu, a završavaju njihovim razdvajanjem (desni dio slike). Započete akcije kontroliraju se pomicanjem ruku u području kamere. Vrlo je važno voditi računa o pravilnom započinjanju i završavanju akcija kako bi sustav upravljao slikovnim sadržajem na ispravan način.

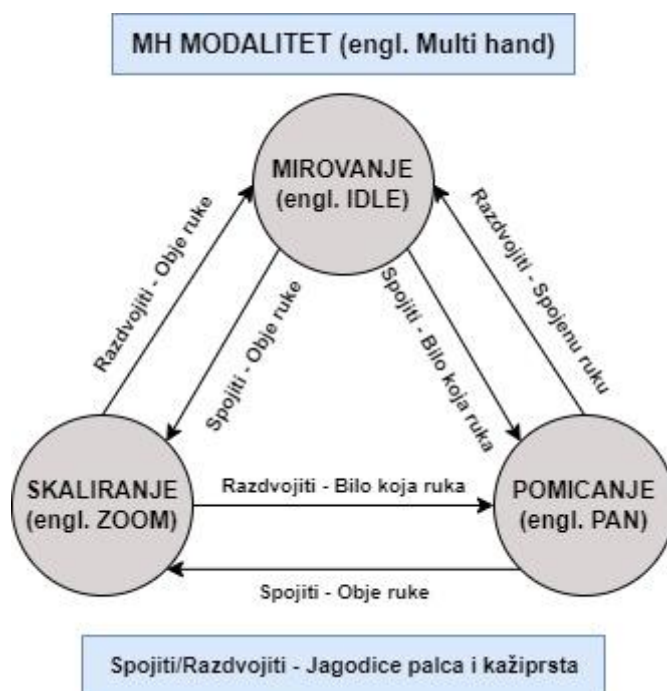


*Slika 3.2. Početak i završetak akcija skaliranja i pomicanja*

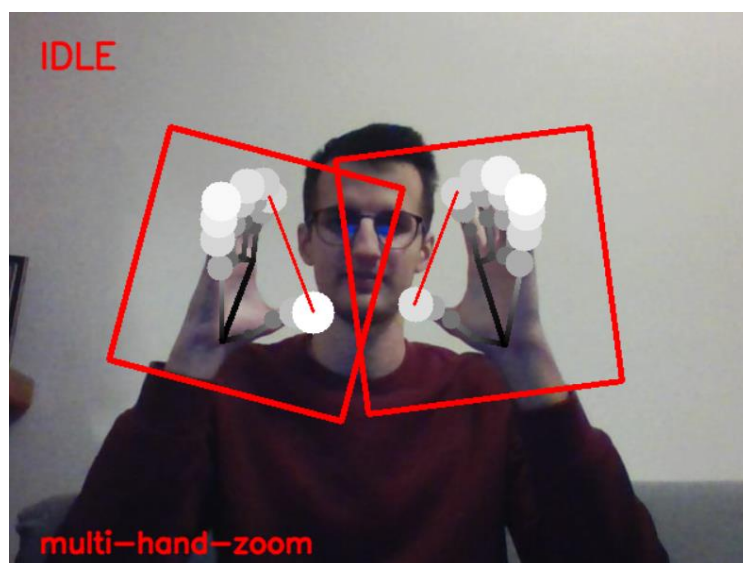
Sustav se tijekom njegova korištenja može naći u tri moguća stanja: stanje mirovanja (engl. Idle) - kada se ne izvodi niti jedna akcija, stanje skaliranja (engl. Zoom) - kada se izvodi akcija skaliranja te stanje pomicanja (engl. Pan) kada se izvodi akcija pomicanja. Iz svakog stanja moguće je prijeći u preostala dva stanja izvođenjem određene geste ovisno o trenutno korištenom modalitetu.

### 3.1.1 MH modalitet

Trenutno stanje u MH modalitetu određuje se temeljem broja ruku na kojima su spojene jagodice palca i kažiprsta. Analizirajući dijagram stanja MH modaliteta prikazanog na slici 3.3, može se zaključiti kako će se sustav nalaziti u stanju mirovanja kada su palac i kažiprst u potpunosti razdvojeni na obje ruke kao što to prikazuje slika 3.4. Stanje mirovanja inicijalno je stanje sustava odnosno s njime započinje njegov rad.



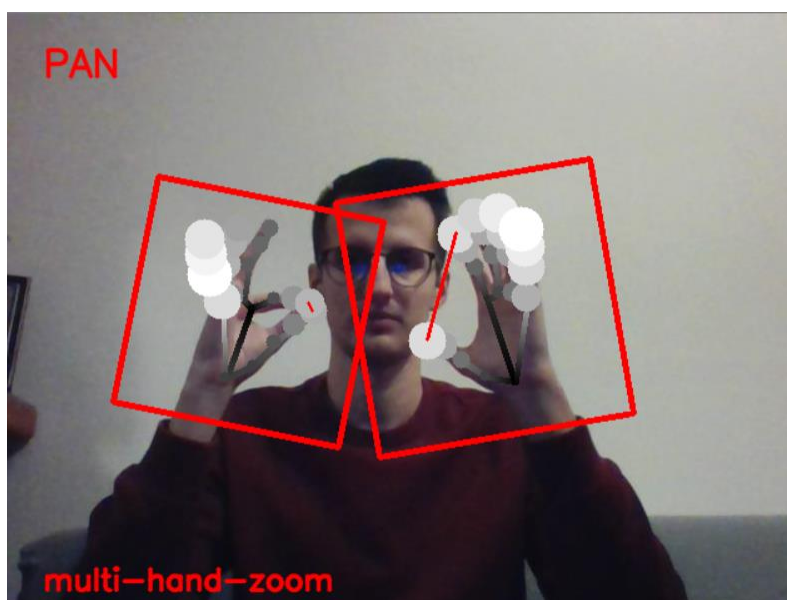
Slika 3.3. Dijagram stanja MH modaliteta



Slika 3.4. Stanje mirovanja u MH modalitetu

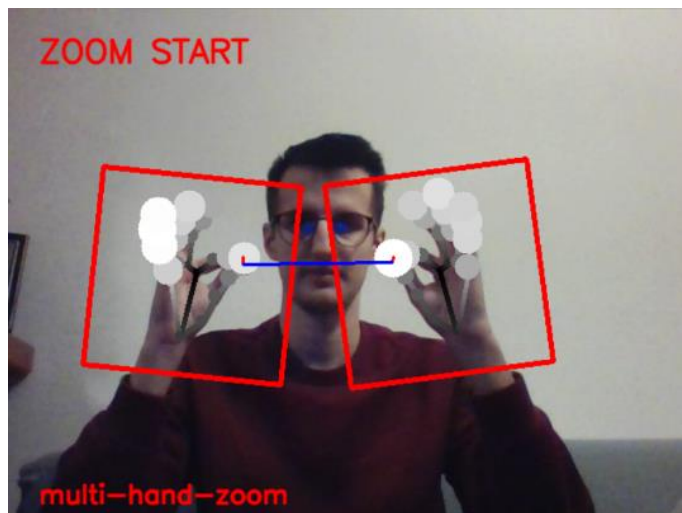


Jednom kada se na bilo kojoj ruci spoje jagodice palca i kažiprsta kao na slici 3.5, sustav prelazi u stanje pomicanja odnosno aktivira se akcija pomicanja slikovnog sadržaja. Jednom kada je akcija aktivirana, pomakom ruke po zraku moguće je pomicati sliku. Razdvajanjem spojenih prstiju završava akcija pomicanja slike. U stanje pomicanja moguće je doći i iz stanja skaliranja tako da se na jednoj ruci razdvoje palac i kažiprst, dok na drugoj ostanu povezani. Gesta spajanja palca i kažiprsta predstavlja dohvaćanje slike na određenom mjestu poput nekakve niti. Pomicanje slike uobičajeno se odvija uzastopnim prijelazom iz stanja mirovanja u stanje pomicanja i obrnuto pa se pomicanje slike doima kao nekakva igra nitima. Važno je napomenuti kako je smjer pomicanja ruke u zraku suprotan od smjera pomicanja slikovnog sadržaja na ekranu odnosno, ukoliko se želi pomaknuti slika prema dolje, ruku je potrebno pomaknuti prema gore.

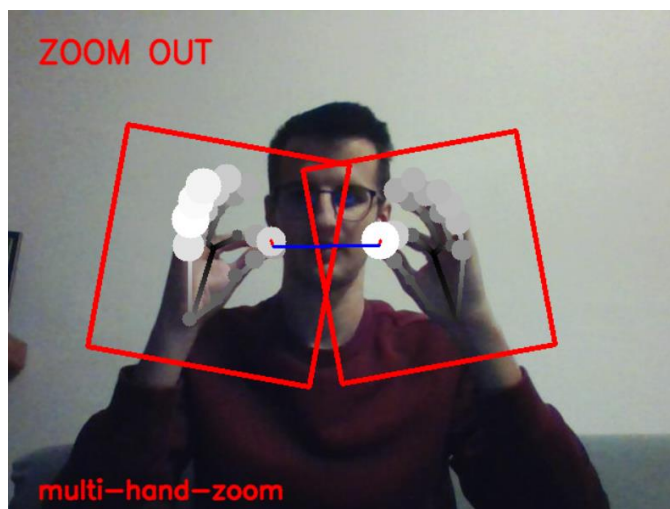


*Slika 3.5. Stanje pomicanja u MH modalitetu*

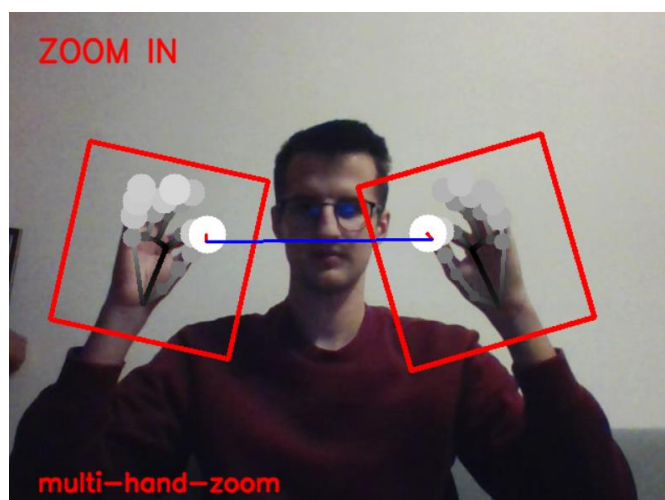
U MH modalitetu, akcija skaliranja izvršava se gestikulacijom u kojoj se koriste obje ruke. Iz stanja mirovanja u stanje skaliranja prelazi se na način da se povežu jagodice palca i kažiprsta na obje ruke istovremeno kao na slici 3.6. U stanje skaliranja može se prijeći i iz stanja pomicanja tako da se i na drugoj ruci povežu jagodice navedenih prstiju. Prijelazom sustava u stanje skaliranja pojavljuje se plava linija koja predstavlja udaljenost između spojenih prstiju obje ruke. Smanjivanjem udaljenosti odnosno približavanjem ruku (slika 3.7) umanjuje se slikovni sadržaj (engl. zoom-out), dok se povećavanjem udaljenosti odnosno udaljavanjem ruku (slika 3.8) uvećava slikovni sadržaj (engl. zoom-in). Ovakav princip skaliranja sadržaja temeljem udaljenosti između dvije točke već se koristi u nekim aplikacijama mobilnih uređaja poput Google Maps-a i sl.



*Slika 3.6. Stanje skaliranja u MH modalitetu*



*Slika 3.7. Umanjivanje slikovnog sadržaja u MH modalitetu*

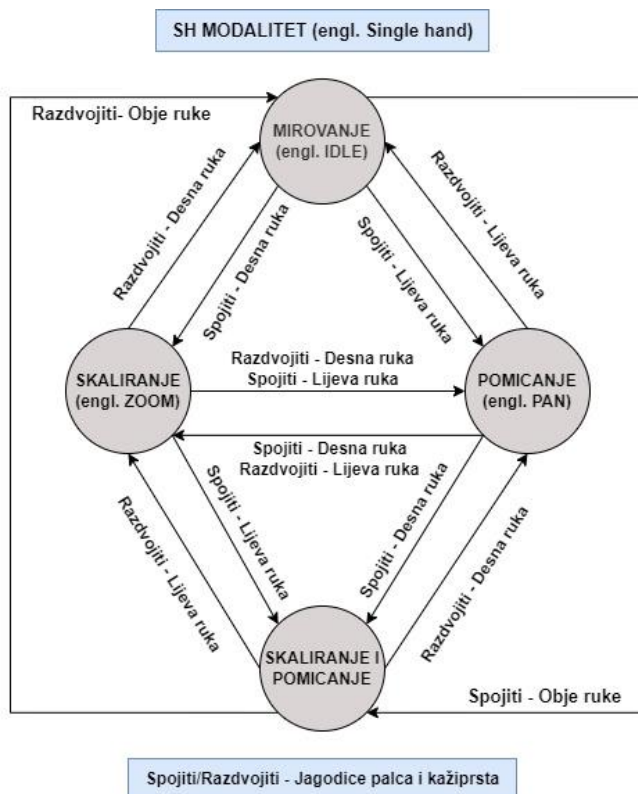


*Slika 3.8. Uvećavanje slikovnog sadržaja u MH modalitetu*

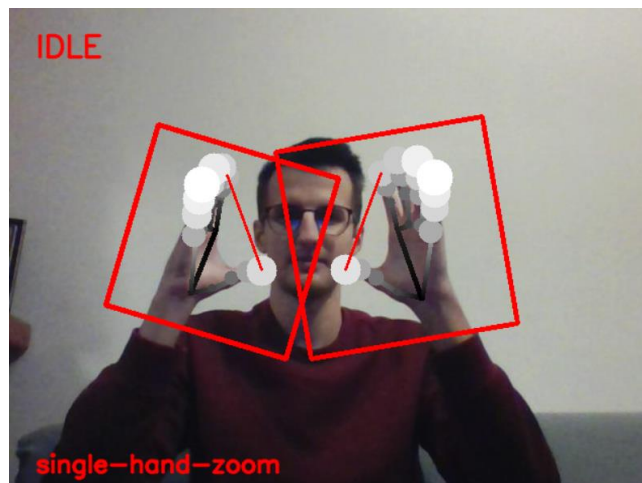
Akcija skaliranja završava razdvajanjem palca i kažiprsta obje ruke, čime sustav prelazi u stanje mirovanja ili razdvajanjem palca i kažiprsta na samo jednoj ruci, čime sustav prelazi u stanje pomicanja. Vrlo je važno da se prilikom prelaska iz stanja mirovanja u stanje skaliranja prsti na obje ruke spajaju istovremeno. Ukoliko se spajanje prstiju na obje ruke ne događa istovremeno, sustav na vrlo kratak period ulazi u stanje pomicanja te zatim u stanje skaliranja, što rezultira prvo pomicanjem, a tek onda skaliranjem slike. Navedeni problem predstavlja jedan od nedostataka ovoga modaliteta. Također, ovaj modalitet zna predstavljati problem osobama kojima je teže uskladiti istovremene pokrete ruku. S druge strane, njegova velika prednost je mogućnost korištenja bilo koje ruke za pomicanje sadržaja pa osobe mogu koristiti svoju dominantnu ruku ili čak obje naizmjenice kako bi si olakšali rad sa sustavom.

### 3.1.2 SH modalitet

U ovome modalitetu za izvršavanje akcija pomicanja i skaliranja koristi se samo jedna ruka. Za akciju pomicanja moguće je koristiti isključivo lijevu ruku, dok je za akciju skaliranja moguće koristiti isključivo desnu ruku. Pregledom dijagrama stanja SH modaliteta (slika 3.9) može se primijetiti kako će se sustav nalaziti u stanju mirovanja u istom slučaju kao i kod MH modaliteta tj. kada su jagodice palca i kažiprsta razdvojene na obje ruke kao na slici 3.10.

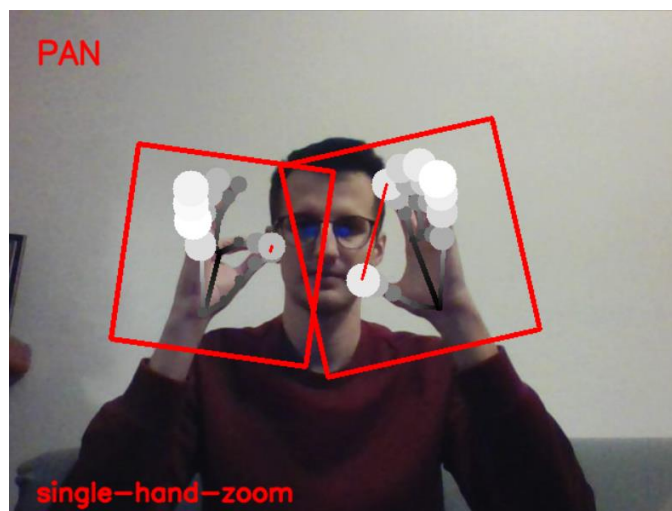


Slika 3.9. Dijagram stanja SH modaliteta



*Slika 3.10. Stanje mirovanja u SH modalitetu*

Iz dijagrama stanja SH modaliteta može se vidjeti kako se sustav može naći u čak četiri stanja. Razlog tome je što su akcije pomicanja i skaliranja u ovome modalitetu neovisne jedna o drugoj jer se koriste različite ruke za njihovo pokretanje. Neovisnost akcija omogućava izvršavanje obiju akcija istovremeno. Paralelno izvršavanje akcija najveća je prednost ovoga modaliteta, međutim potrebno je na pravilan način uskladiti pokrete ruku kako bi ova prednost zaista došla do izražaja. Spajanjem jagodica palca i kažiprsta lijeve ruke, kao na slici 3.11, sustav prelazi iz stanja mirovanja u stanje pomicanja. Pomicanje slikovnog sadržaja odvija se po istom principu kao i kod MH modaliteta, dakle pomicanjem ruke po zraku. U stanje pomicanja moguće je doći i iz novog stanja u kojem se obje akcije paralelno izvode i to na način da se prekine akcija skaliranja. Akcija pomicanja završava razdvajanjem spojenih prstiju na lijevoj ruci. Također, kao i kod MH modaliteta, pomicanje ruke po zraku suprotno je pomicanju slikovnog sadržaja na zaslonu pa je i na to potrebno obratiti pažnju.

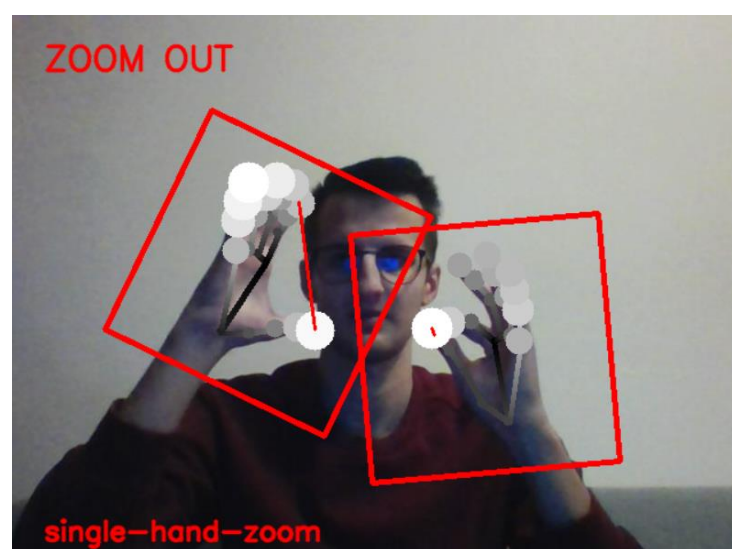


*Slika 3.11. Stanje pomicanja u SH modalitetu*

Glavna razlika MH i SH modaliteta leži u načinu pokretanja akcije skaliranja. Za akciju skaliranja unutar SH modaliteta potrebna je samo jedna ruka i to isključivo desna. Skaliranje započinje spajanjem palca i kažiprsta desne ruke. Jednom kada su prsti spojeni te akcija zumiranja pokrenuta, sustav očitava i sprema visinu kažiprsta na kameri tj. y koordinatu. Pomicanjem desne ruke prema gore (slika 3.12) povećava se visina kažiprsta na kameri i uvećava slikovni sadržaj, dok se pomicanjem ruke prema dolje (slika 3.13) visina kažiprsta smanjuje, a time i slikovni sadržaj. Akcija skaliranja traje sve dok su prsti na desnoj ruci povezani. Razina skaliranja mijenja se svaki puta kada y koordinata prijeđe postavljeni prag (engl. threshold). Ovaj prag sprječava preveliku osjetljivost sustava odnosno nekontrolirano smanjivanje i uvećavanje slikovnog sadržaja.



Slika 3.12. Uvećavanje slikovnog sadržaja u SH modalitetu



Slika 3.13. Umanjivanje slikovnog sadržaja u SH modalitetu

## 3.2 Implementacija

Pan&Zoom gestovni sustav sastoji se od dvaju programskih modula: MediaPipe i Image-viewer koji međusobno komuniciraju putem utičnice (engl. socket) i kamere koja prati korisničke geste. Sustav je implementiran i testiran na prijenosnom računalu Lenovo Legion Y520. Prilikom razvoja sustava korištena je integrirana 720p HD kamera s rezolucijom od 1280 x 720 točaka.

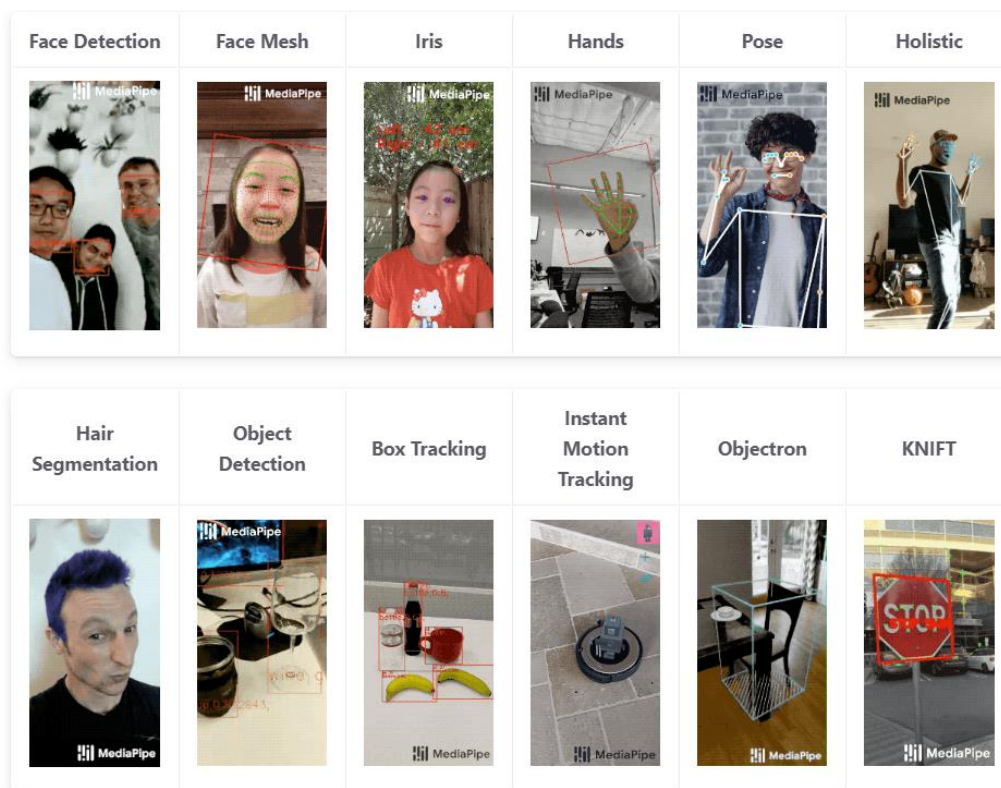
### 3.2.1 MediaPipe

MediaPipe je radni okvir (engl. framework) otvorenog koda za razvoj prilagodljivih rješenja strojnog učenja u području računalnog vida [14]. Računalni vid jedno je od područja umjetne inteligencije koje uključuje metode za obradu, analiziranje i razumijevanje slike. Prepoznavanje određenih objekata u prostoru ili na slikama i analiza pokreta jedni su od mnogih zadataka računalnog vida [15].

Svaki videozapis ili prijenos uživo (engl. livestream) zapravo je niz okvira (engl. frame) odnosno slika koje se prikazuju u vrlo kratkim vremenskim intervalima. Svaku sliku moguće je obraditi temeljem metoda računalnog vida i primjenom parametara treniranih modela strojnog učenja kako bi se iz njih izvukle željene informacije poput odgovora na pitanje: „*Nalazi li se na slici traženi objekt?*“. Kako bi se dobio odgovor na navedeno pitanje potrebno je trenirati modele strojnog učenja na velikom broju slika koje sadrže traženi objekt. Slike bi trebale biti različitih veličina, tekstura i boja, a objekti raznih veličina i postavljeni pod različitim kutem. Različitost slika omogućuje treniranom modelu veću preciznost u svojoj kasnijoj predikciji. Pomoću dobivenih parametara modela, moguće je dobiti odgovor na navedeno pitanje. Za treniranje što boljih i preciznijih modela potrebna je vrlo velika količina ulaznih podataka (nekoliko tisuća raznovrsnih slika) i vrlo jaka računalna moć kako proces treniranja ne bi trajao poprilično dugo. Zbog navedenih preduvjeta većinom se koriste trenirani modeli velikih kompanija kojima takvi zahtjevniji preduvjeti ne predstavljaju problem.

MediaPipe radni okvir sadrži pregršt gotovih rješenja strojnog učenja u području računalnog vida. Unutar svakoga rješenja nalaze se vrlo kvalitetni i precizni modeli trenirani od strane Google-a koji omogućuju odlično prepoznavanje objekata poput lica, ruku, prometnih znakova, automobila i sl. Gotova rješenja moguće je mijenjati i prilagoditi onome što se želi postići [16]. Upravo ta fleksibilnost razlog je korištenja ovog radnog okvira pri razvoju Pan&Zoom sustava.

Razne primjere rješenja strojnog učenja koje sadrži MediaPipe radni okvir moguće je vidjeti na slici 3.14. Rješenja su vrlo zanimljiva i omogućuju prepoznavanje lica, šarenica očiju, ruku, poza, prometnih znakova, objekata u prostoru i sl.



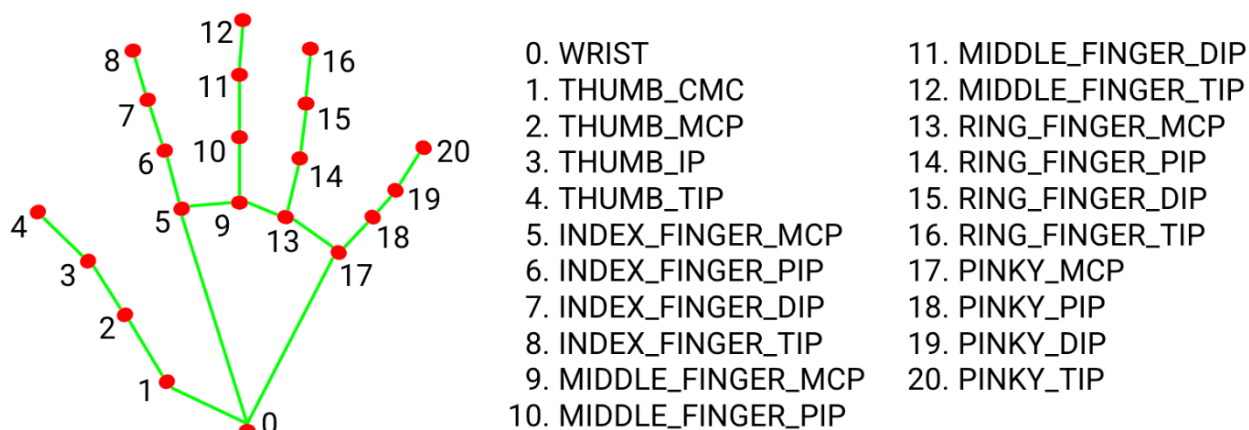
Slika 3.14. Gotova rješenja strojnog učenja unutar radnog okvira MediaPipe, preuzeto iz [14]

Kako se Pan&Zoom sustav temelji na izvođenju gesti rukama, pri njegovoj implementaciji korišteno je MediaPipe rješenje koje uključuje uspješno prepoznavanje dlanova i njihovih ključnih točaka odnosno orijentira. Programski kod ovoga rješenja pretežito je pisan u C++ programskom jeziku. Podloga rada ovoga rješenja su modeli i tehnike strojnog učenja.

U ovome rješenju korištena su dva modela strojnog učenja: model za detekciju dlana ruke (engl. Palm Detection Model) koji na slici pronalazi dlanove ruku te oko njih formira pravokutan okvir crvene boje i model koji na tom pravokutnom području detektira i lokalizira čak dvadeset i jednu ključnu točku dlana (engl. Hand Landmark Model).

Svaka ključna točka opisana je s tri vrijednosti  $x$ ,  $y$  i  $z$ , gdje  $x$  i  $y$  predstavljaju koordinate točke po širini i visini područja kamere dok  $z$  predstavlja dubinu odnosno udaljenost točke od kamere. Princip rada navedenih modela poprilično je zanimljiv, ali i složen te stoga neće biti opisan u sklopu ovoga rada [17].

Ključne točke dlana ili orijentire (engl. landmarks) moguće je grafički predstaviti slikom 3.15. U implementaciji gesti za akcije pomicanja i skaliranja slikovnog sadržaja korištene su jedino točke pod brojevima 4 i 8 koje predstavljaju jagodice palca i kažiprsta jer u oba modaliteta akcije započinju i završavaju spajanjem ili razdvajanjem tih prstiju.



Slika 3.15. Ključne točke/orijentiri dlana ruke, preuzeto iz [17]

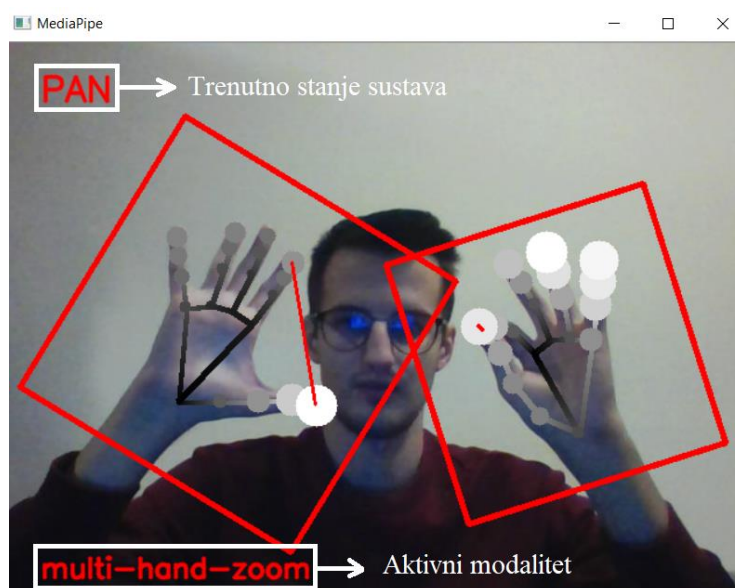
Izvorni programski kod za prevođenje i pokretanje opisanog MediaPipe<sup>1</sup> rješenja preuzet je s GitHub-a. Pokretanjem rješenja, na zaslonu se pojavljuje prikaz područja kamere. Jednom kada se dlan ruke postavi u područje kamere, na zaslon se ispisuje o kojoj ruci je riječ odnosno je li model uspješno prepoznao dlan lijeve ili desne ruke. Ako se u području kamere nađu dlanovi obje ruke istovremeno, model ih uspijeva prepoznati, ali se na zaslonu ne pojavljuje nikakav ispis.

Navedeno rješenje dodatno je unaprijeđeno kako bi se uspješno mogle prepoznati geste kojima se pokreću akcije pomicanja i skaliranja slikovnog sadržaja. Implementirane su dvije funkcije koje predstavljaju nova dva modaliteta. Inicijalni modalitet je MH, a modalitete je moguće mijenjati pritiskom slova „s“ na tipkovnici (engl. switch). Prilikom obrade slike odnosno okvira s kamere, MediaPipe modul detektira o kojem se dlanu ili dlanovima radi (lijevi, desni ili oba), prema pozicije njihovih ključnih točaka, provjerava u kojoj se poziciji nalaze jagodice palca i kažiprsta u odnosu na prethodni okvir (spojeni ili razdvojeni) i koji je trenutno aktivni modalitet. Prikupljene informacije i zaključci spremaju se u poruku koja se na kraju obrade okvira šalje Image-viewer modulu putem utičnice kako bi se uspješno ažuriralo stanje sustava. Između jagodica palca i kažiprsta odnosno ključnih točaka dlana 4 i 8 (slika 3.15) nacrtana je linija crvene boje koja predstavlja udaljenost među točkama.

<sup>1</sup> <https://github.com/google/mediapipe>



Ako je udaljenost između točaka manja od određenog praga ili jednaka nuli - radi se o pokretanju ili nastavljanju određene akcije, a ako je veća od praga - radi se o završetku određene akcije. U sklopu MH modaliteta, prilikom akcije skaliranja crta se linija plave boje između spojenih palčeva i kažiprsta obje ruke koja predstavlja njihovu udaljenost. Povećanje udaljenosti rezultira uvećavanjem, a smanjivanje udaljenosti umanjivanjem slikovnog sadržaja. Također, u lijevome gornjem kutu grafičkog sučelja navedeno je trenutno stanje u kojem se sustav nalazi, dok je u lijevom donjem kutu naveden trenutno aktivni modalitet kojim se upravlja sustavom. Grafičko sučelje nadograđenog MediaPipe modula može se vidjeti na slici 3.16.



Slika 3.16. Grafičko sučelje MediaPipe modula

Kako model za detekciju i lokalizaciju ključnih točaka dlanova nije idealan, kao ni mirnoća korisničke ruke, detektirane točke često mijenjaju svoju poziciju između okvira odnosno slika kamere iako korisnik zadržava ruku u istoj poziciji. Ukoliko se sustav nalazi u stanju pomicanja, a korisnik drži svoju ruku u istoj poziciji, manji pomaci pozicija ključnih točki mogu rezultirati konstantnim ažuriranjem akcije pomicanja odnosno neželjenim pomacima slikovnog sadržaja. Kako bi se spriječio navedeni problem, ažuriranje akcije pomicanja odbacuje se ako su pomaci  $x$  i  $y$  koordinata točaka manji od određenog praga šuma (0.25% veličine područja slike kamere).

Za korištenje MediaPipe modula potrebno je imati adekvatnu kameru i obratiti pozornost na područje koje ona pokriva. Naime, ukoliko korisnik prilikom upravljanja sustavom izađe svojim rukama izvan područja kamere, gubi se kontrola nad sustavom odnosno prekidaju se akcije jer sustav više ne može detektirati dlanove ruku. Rezultat navedenog problema su neželjeni pomaci slikovnog sadržaja.

Navedeni problem vrlo se često pojavljuje kada je osoba preblizu kameri. S većom udaljenosti od kamere smanjuje se i veličina dlanova pa je sustavu teže prepoznati i lokalizirati ključne točke, što posljedično dovodi do nemogućnosti izvršavanja akcija.

Iz tog razloga, preporučljivo je da korisnik pri početku korištenja sustava pronađe poziciju odnosno udaljenost od kamere na kojoj se navedeni problemi neće događati. Također, prije korištenja ovoga sustava potrebno je osigurati dovoljnu količinu svjetla u prostoriji kako bi model uspješno detektirao dlanove i ključne točke. Vrlo je važno da dlanovi ruku pri korištenju sustava budu okrenuti prema kameri, a ne od nje jer će se i tada pojaviti problemi s interakcijom. Implementirani MediaPipe modul moguće je nadograditi u budućnosti stvaranjem novoga modaliteta ili izmjenom postojećih. S obzirom na detekciju dvadeset i jedne ključne točke dlana, postoji vrlo širok spektar gesti koje se mogu prevesti u određene akcije za upravljanje sadržajem na zaslonu.

### 3.2.2 Image-viewer

Kako bi se testirali implementirani modaliteti, s Github-a je preuzeto grafičko sučelje za prikaz slikovnog sadržaja pod nazivom Image-viewer<sup>2</sup>. Grafičko sučelje izrađeno je korištenjem C++ programskog jezika i Qt GUI radnog okvira. Image-viewer modul također je nadograđen kako bi se mogle primati poruke poslane od strane MediaPipe modula te na temelju njih izvoditi akcije pomicanja i skaliranja slikovnog sadržaja.

Akcija pomicanja provodi se u tri dijela odnosno događaja: početak (engl. pan-start), pomicanje (engl. pan-update) i završetak (engl. pan-finish). Akcija skaliranja provodi se kao događaj uvećavanja ili umanjivanja slikovnog sadržaja. Primljene poruke predstavljaju navedene događaje. Akcije pomicanja i skaliranja slikovnog sadržaja odnosno navedeni događaji izvode se simuliranjem rada miša.

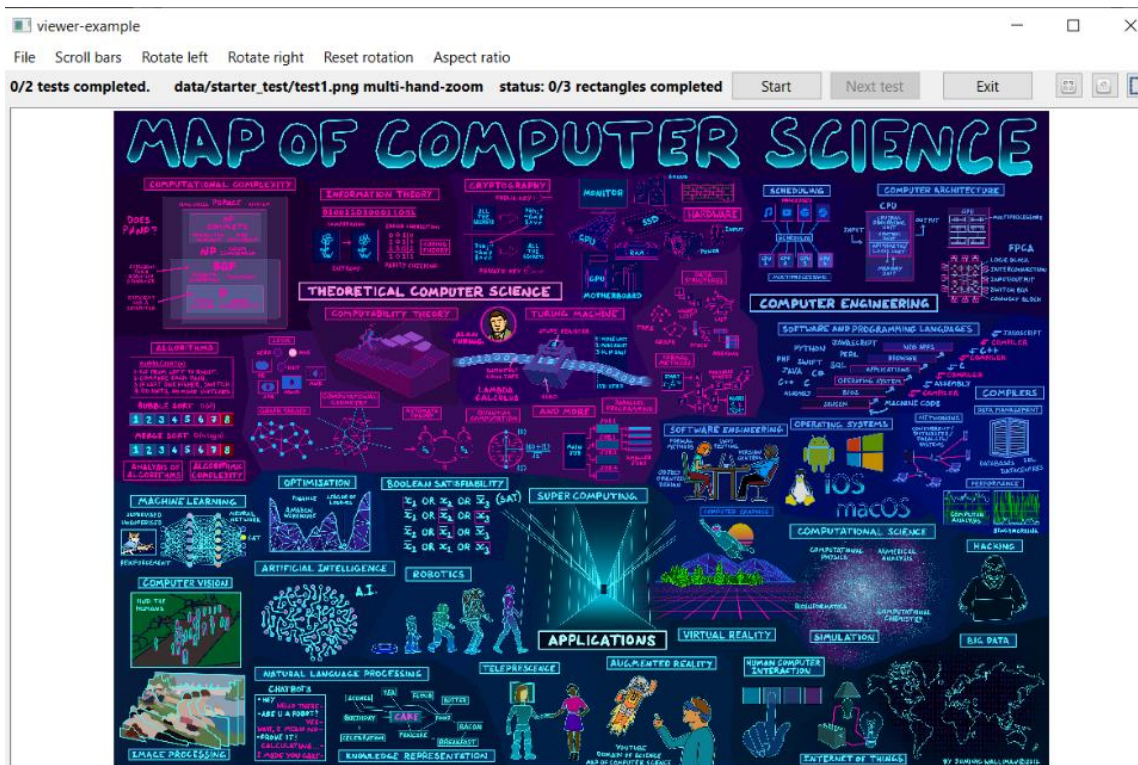
Početak akcije pomicanja simulira se pritiskom lijevog klika miša nad slikom, pomicanje slikovnog sadržaja odvija se simulacijom pomicanja miša po podlozi ovisno o primljenom x i y pomaku, a završetak akcije pomicanja otpuštanjem lijevog klika miša. Akcija skaliranja odvija se simulacijom pomicanja kotačića miša prema gore (uvećanje) ili dolje (umanjeno). Sve navedene simulacije izvršavaju se pomoću Qt funkcija.

---

<sup>2</sup> <https://github.com/palacaze/image-viewer>

Za potrebe provedbe HCI eksperimenta, ažurirano je grafičko sučelje Image-viewer modula i dodana podrška za izvršavanje zadataka kojima se želi utvrditi učinkovitost dvaju implementiranih modaliteta. Grafičko sučelje Image-viewer modula prikazano je na slici 3.17. Sučelje se sastoji od 3 dijela: okvira u koji se učitava slikovni sadržaj, izbornika kojim se upravlja i prati napredak HCI eksperimenta te alatne trake za upravljanje slikovnim sadržajem.

Na samome vrhu sučelja nalazi se alatna traka kojom je moguće učitati slikovni sadržaj s računala, upravljati prikazom trake za pomicanje slikovnog okvira (engl. scroll bar), rotirati učitane slike i dr. Ispod alatne trake nalazi se izbornik za upravljanje i praćenje tijeka HCI eksperimenta. Na lijevoj strani izbornika nalaze se 3 oznake crne boje. Prva oznaka prikazuje koliko je zadataka eksperimenta uspješno odrađeno, druga oznaka predstavlja putanju do trenutno učitane slike zadatka i modalitet kojim ga je potrebno izvršiti dok treća oznaka predstavlja broj uspješno skaliranih stavki na trenutnoj slici zadatka. Nakon navedenih oznaka slijede 2 gumba kojima se pokreće vrijeme trenutnog zadatka i omogućuje prijelaz na idući te oznaka koja prikazuje vrijeme potrebno za izvršavanje zadatka u milisekundama (pojavljuje se na kraju svakog zadatka). Na desnoj strani izbornika nalazi se gumb za izlaz iz Image-viewer modula, oznaka koja predstavlja trenutnu poziciju pokazivača na slikovnom okviru i boju na kojoj se on nalazi u RGB formatu (pojavljuje se simulacijom odabira unutar slikovnog okvira) te 3 gumba za skaliranje slike unutar okvira.



Slika 3.17. Grafičko sučelje Image-viewer modula

Akciju pomicanja moguće je izvršavati sve dok se slika svojim rubom ne nađe na kraju okvira u kojem je učitana. Kada slika svojim rubom dotakne kraj okvira, akcija pomicanja slike prema toj strani neće biti moguća stoga je potrebno prvo uvećati sliku, a tek onda nastaviti s pomicanjem. Navedena poteškoća javlja se barem jednom prilikom upravljanja slikovnim sadržajem i to na samome početku kada se rubovi učitane slike skaliraju i izjednačavaju s okvirom, pa je upravljanje slikovnim sadržajem potrebno započeti uvijek akcijom skaliranja. Problem nemogućnosti pomicanja slike češće se javlja kod slika manjih dimenzija jer se navedena pojava češće događa.

### 3.2.3 Komunikacija između modula

Akcije pomicanja i skaliranja slikovnog sadržaja započinju s određenim gestama i pokretima ruku unutar MediaPipe modula, a završavaju njihovom izvedbom nad slikom unutar Image-viewer modula. Kako bi Image-viewer znao o kojoj se akciji radi i što učiniti sa slikovnim sadržajem, između ovih dvaju modula mora postojati nekakav oblik komunikacije u stvarnom vremenu. Za komunikaciju između modula korištena je UNIX porodica utičnica (engl. socket) koja omogućuje brzu komunikaciju između procesa koji se izvršavaju na istom računalu (IPC, engl. Inter-Process Communication). Jedan od procesa ponaša se kao poslužitelj (engl. server), a drugi kao klijent (engl. client). Poslužitelj zauzima i povezuje utičnicu s određenom adresom i čeka na zahtjev za povezivanje od strane klijenta. Jednom kada klijent zatraži uslugu od poslužitelja, povezuje se na utičnicu i ostvaruje uspješnu komunikaciju među procesima [18].

Pan&Zoom sustav moguće je koristiti na Linux i Windows platformama. Implementacija komunikacije među modulima razlikuje se s obzirom na navedene platforme. Na Linux operacijskom sustavu, UNIX porodicu utičnica moguće je koristiti putem *sys/socket* i *sys/un* knjižnica koje dolaze s GNU prevoditeljem, a za koje Windows platforma nema podršku [19]. Korištenje UNIX porodice utičnica na Windows operacijskim sustavima nije bilo moguće sve dok Windows kompanija nije razvila svoj Winsock API koji koristi *Ws2\_32* knjižnicu i *afunix.h* zaglavlje za rad s UNIX porodicom utičnica [20]. Navedena knjižnica dolazi s MSVC prevoditeljem i ima ponešto drugačiju sintaksu od knjižnica koje se koriste na Linux sustavima. Uzevši u obzir dvije različite knjižnice i njihove sintakse, za pisanje C++ rješenja namijenjenog različitim prevoditeljima i operacijskim sustavima unutar jedne datoteke koriste se tzv. *macro* oznake i naredbe uvjetnog prevođenja [21].

Svaki C++ prevoditelj sadrži predefinirane pretprocesorske *macro* oznake koje su definirane C++ standardima. *Macro* predefinirane oznake za određene operacijske sustave moguće je vidjeti na slici 3.18.

|  |  |
|--|--|
| Linux and Linux-derived                                      | <code>__linux__</code>                                     |
| Android  | <code>__ANDROID__</code> (implies <code>__linux__</code> ) |
| Linux (non-Android)  | <code>__linux__</code> && ! <code>__ANDROID__</code>       |
| Darwin (Mac OS X and iOS)                                    | <code>__APPLE__</code>                                     |
| Akaros ( <a href="http://akaros.org">http://akaros.org</a> ) | <code>__ros__</code>                                       |
| Windows  | <code>_WIN32</code>  |
| Windows 64 bit   | <code>_WIN64</code> (implies <code>_WIN32</code> )         |
| NaCL   | <code>__native_client__</code>                             |
| AsmJS  | <code>__asmjs__</code>                                     |
| Fuchsia  | <code>__Fuchsia__</code>                                   |

Slika 3.18. Primjeri predefiniranih *macro* oznaka za različite operacijske sustave

Naredbe uvjetnog prevođenja omogućavaju preskakanje prevođenja određenog dijela programskog koda na temelju određenih uvjeta. Primjer takve naredbe je *#ifdef* naredba koja se uobičajeno koristi s predefiniranim *macro* oznakama. Na ovaj način moguće je napisati programski kod koji će prevoditelj prevesti odnosno uzeti u obzir samo kada se radi o npr. Windows platformi. Koristeći se navedenom naredbom i *macro* oznakama, omogućena je uspješna implementacija komunikacije među modulima unutar jedne C++ datoteke koristeći različite knjižnice i sintakse.

Komunikacija među modulima odvija se u zasebnoj dretvi kako ne bi narušavala brzinu rada sustava. U Pan&Zoom sustavu ulogu poslužitelja obavlja Image-viewer modul, a ulogu klijenta MediaPipe modul. Image-viewer modul prvi se pokreće, učitava slikovni sadržaj i stvara *socket* datoteku u privremenoj mapi (engl. temporary) na računalu, a zatim čeka na povezivanje od strane MediaPipe modula. MediaPipe modul se nakon pokretanja povezuje na stvorenu datoteku i time se uspješno ostvaruje međusobna komunikacija.

Prestankom rada sustava zatvara se i međusobna komunikacija. Vrlo je važno prvo zatvoriti Image-viewer modul i to pritiskom na gumb „Exit“, jer će se tada izbrisati stvorena *socket* datoteka i na pravilan način zatvoriti međusobna komunikacija. Nakon zatvaranja Image-viewer modula potrebno je zatvoriti i MediaPipe modul kombinacijom tipki „Ctrl+C“ u terminalu računala. Ako se terminiranje sustava ne napravi na iznad navedeni način potrebno je ručno ukloniti *socket* datoteku prije ponovnog pokretanja sustava.

### 3.3 Instalacija sustava i postavljanje u rad

Pan&Zoom sustav, kao što je već spomenuto, namijenjen je stolnim i prijenosnim računalima s operacijskim sustavima Linux ili Windows. Sustav je uspješno testiran na trima operacijskim sustavima: 64-bitnom operacijskom sustavu Windows 10, Arch Linux-u i Ubuntu-u. Za uspješno prevođenje programskog koda i jednostavnije pokretanje sustava napisane su *bash* skripte. Kako bi se programski kodovi modula uspješno preveli u izvršne datoteke potrebno je instalirati potrebne alate koji variraju u ovisnosti u operacijskom sustavu. Cjelokupan programski kod Pan&Zoom<sup>3</sup> sustava postavljen je na Github i javno je dostupan kako bi ga svatko mogao preuzeti i dorađivati.

#### 3.3.1 Linux

Kako bi se Pan&Zoom sustav uspješno pokrenuo na Linux sustavima potrebno je instalirati nekoliko alata. Na samome početku potrebno je preuzeti mapu s programskim kodom Pan&Zoom sustava s GitHub-a ručno ili korištenjem *git clone* naredbe u terminalu. Naredbe za preuzimanje koda putem terminala su sljedeće:

- *sudo apt-get update*
- *sudo apt-get upgrade*
- *sudo apt-get install git*
- *git clone https://github.com/MMaricevic64/Pan\_Zoom.git*

Za korištenje *git* naredbi u terminalu potrebno je prvo instalirati *git* sustav za verzioniranje. Tada je moguće, pomoću naredbe *git clone*, preuzeti kod s navedene poveznice. Za prevođenje programskog koda MediaPipe modula potrebno je instalirati Python, Bazel, OpenCV i OpenGL alate. Modeli strojnog učenja za detekciju dlanova i njihovih ključnih točaka se, na Linux sustavu, mogu izvršavati putem grafičke kartice umjesto putem procesora. Izvršavanje ovakvih modela na grafičkoj kartici može biti puno brže zbog korištenja velikog broja jezgri čime je omogućeno rastavljanje složenog problema na više manjih simultanih izračuna. Navedena brzina dolazi do izražaja kada se koriste naprednije grafičke kartice s više jezgri. Ako se za izvršavanje modela koristi računalo s integriranom grafičkom karticom i snažnijim procesorom, vrlo je vjerojatno da će tada izvršavanje biti brže na samom procesoru nego na grafičkoj kartici.

---

<sup>3</sup> [https://github.com/MMaricevic64/Pan\\_Zoom](https://github.com/MMaricevic64/Pan_Zoom)

Zbog navedene mogućnosti bržeg izvršavanja, MediaPipe modeli na Linux sustavu predviđeni su za izvršavanje na grafičkoj kartici. Za potrebe prevođenja i pokretanja MediaPipe modula potrebno je prvo instalirati Python verzije 3 i knjižnicu *numpy*. Prilikom implementacije Pan&Zoom sustava korištena je verzija 3.10., a naredbe za instalaciju Python-ovih osnovnih knjižnica i knjižnice *numpy* su:

- `sudo apt-get install python3.10`
- `sudo apt install python3-pip`
- `pip3 install numpy`

Bazel je besplatan softverski alat otvorenog koda razvijen od strane Google-a koji se koristi za prevođenje i testiranje programskih rješenja, a u sklopu ovoga rada koristi se za prevođenje izvornog koda MediaPipe modula. Izvorni kod preuzetog MediaPipe modula namijenjen je prevođenju Bazel-om verzije 5.2, zbog čega je potrebno instalirati baš navedenu verziju. Za uspješnu instalaciju Bazel-a verzije 5.2.0 potrebno je slijediti upute na idućim poveznicama:

- <https://bazel.build/install/ubuntu?fbclid=IwAR3JMFT8iRm7a5aAlMoVkQ589eLbki41jBaZCNuLkPlzyOnh--wpI1K-hw#binary-installer>
- <https://github.com/bazelbuild/bazel/releases/tag/5.2.0> - *.sh* skripta za poveznicu iznad

U idućem koraku potrebno je preuzeti knjižnicu otvorenog koda za računalni vid, strojno učenje i obradu slike naziva OpenCV. Funkcionalnosti MediaPipe modula temelje se na ovoj knjižnici pa je tako i unutar *mediapipe* mape dana *bash* skripta koja preuzima ovu knjižnicu i instalira njene zavisne pakete poput Cmake-a o kojem će biti više riječi nešto kasnije u ovome poglavlju. Bash skripta dodatno konfigurira tzv. *build* datoteke MediaPipe modula kako bi prevoditelj znao putanju do OpenCV knjižnica. Za pokretanje navedene skripte iz terminala potrebno je slijediti iduće naredbe:

- `cd Pan_Zoom` – pozicioniranje u mapu Pan&Zoom sustava
- `cd mediapipe` – pozicioniranje u mapu MediaPipe modula
- `chmod +x setup_opencv.sh` – dodavanje ovlasti za izvršavanje skripte
- `./setup_opencv.sh` – pokretanje skripte

Za izvršavanje modela strojnog učenja na grafičkoj kartici potrebno je instalirati i postaviti OpenGL ES alat pomoću iduće poveznice:

- [https://google.github.io/mediapipe/getting\\_started/gpu\\_support.html#opengl-es-setup-on-linux-desktop](https://google.github.io/mediapipe/getting_started/gpu_support.html#opengl-es-setup-on-linux-desktop)

Uspješnom instalacijom i konfiguracijom navedenih alata može se prijeći na završni korak, a to je prevođenje programskog koda MediaPipe modula. Putem terminala potrebno je upisati iduće naredbe:

- `cd Pan_Zoom`
- `cd mediapipe`
- `chmod +x build_pan_zoom.sh linux`
- `./build_pan_zoom.sh linux`

Nakon uspješnog prevođenja MediaPipe-a, potrebno je instalirati još jedan softver prije prevođenja Image-viewer-a. Qt sadrži alate za uspješno stvaranje grafičkih korisničkih sučelja na različitim platformama u C++ programskom jeziku. Knjižnice ovoga softvera omogućuju učitavanje i upravljanje slikovnim sadržajem unutar Image-viewer modula. Na idućoj poveznici: <https://www.qt.io/download-open-source> potrebno je otići na odjeljak „*Get Started*“, preuzeti *Qt Online Installer*, postaviti ga na radnu površinu te pokrenuti instalaciju iz terminala idućim naredbama:

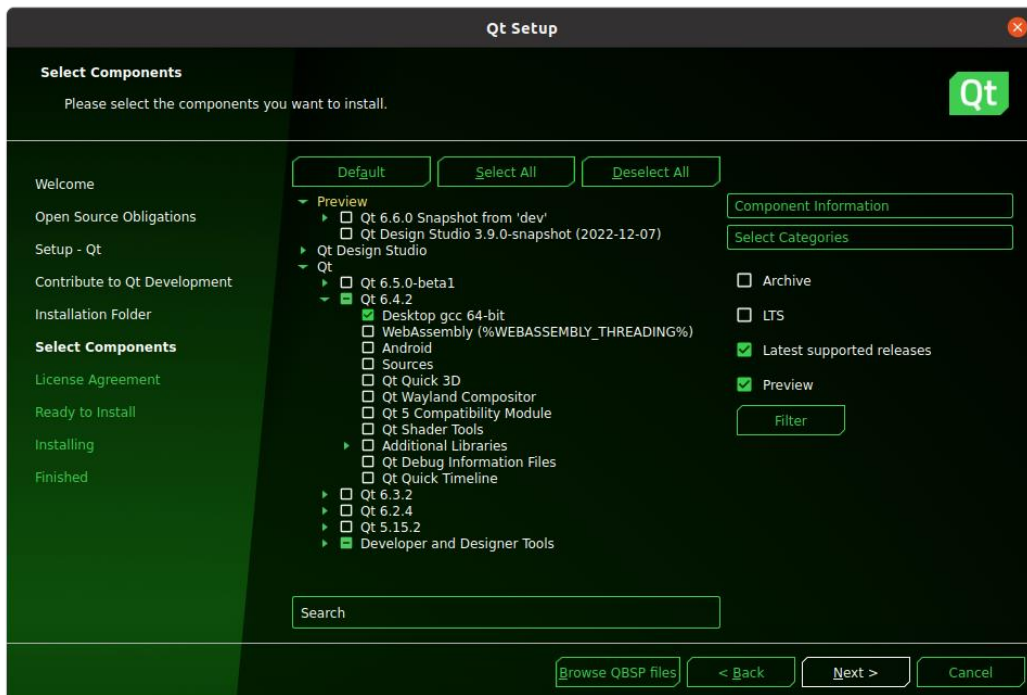
- `cd Desktop`
- `chmod +x <ime_instalacijskog_paketa>.run`
- `./<ime_instalacijskog_paketa>.run` – npr. `qt-unified-linux-x64-4.5.1-online`
- Ako se instalacija ne pokreće, potrebno je napisati iduću naredbu te zatim ponovno pokrenuti naredbu iznad:
- `sudo apt install --reinstall libxcb-xinerama0`

Na početku instalacije potrebno je obaviti registraciju kako bi se moglo nastaviti s instalacijom i korištenjem Qt softvera. Nakon obavljene registracije potrebno je nastaviti instalaciju sljedećim koracima:

- Odabrati opciju „*Custom Installation*“ te bilo koju Qt 6 verziju
- Iz odabrane verzije označiti samo opciju „*Desktop gcc 64-bit*“ kao na slici 3.19
- Nastaviti i dovršiti instalaciju slijedeći upute na zaslonu

Jednom kada je Qt softver instaliran, može se prijeći na prevođenje izvornog koda Image-viewer modula.





Slika 3.19. Instalacija Qt softvera na Linux operacijskom sustavu

Za prevođenje programskog koda Image-viewer modula koristi se Cmake alat koji bi već trebao biti instaliran, ukoliko su na pravilan način slijeđene sve navedene upute. Cmake se koristi za prevođenje programskog koda pomoću jednostavnih konfiguracijskih datoteka (engl. makefiles) neovisnih o platformi i prevoditelju. Kako bi se programski kod Image-viewer modula uspješno preveo potrebno je koristiti sljedeće naredbe u terminalu:

- `cd Pan_Zoom`
- `cd image-viewer`
- `mkdir build`
- `cd build`
- `cmake .. -DCMAKE_PREFIX_PATH=" <putanja do gcc_64 mape unutar Qt> "`  
primjer putanje: `/home/mariopc/Qt/6.4.2/gcc_64/`
- `make`

Za potrebe pokretanja Pan&Zoom sustava napravljene su dvije *bash* skripte: *train-experiment.sh* koja služi za upoznavanje s principom korištenja sustava i *run-experiment.sh* koja se koristi tijekom provedbe HCI eksperimenta. Skripte prilikom pokretanja primaju argument u kojem se navodi o kojoj se platformi radi te se s obzirom na platformu izvršavaju određene naredbe za pokretanje oba modula. Prilikom pokretanja *run-experiment.sh* skripte, drugi argument predstavlja jedan od mogućih poredaka zadataka HCI eksperimenta.

Pokretanje sustava na Linux operacijskom sustavu odvija se idućim naredbama:

- `cd Pan_Zoom`
- `chmod +x train-experiment.sh` ili `chmod +x run-experiment.sh`
- `./train-experiment.sh linux` ili `./run-experiment.sh linux test-sequence-1.txt`

Terminiranje sustava odvija se pritiskom na gumb „Exit“ grafičkog sučelja Image-viewer modula kako bi se na pravilan način zatvorila komunikacija među modulima.

### 3.3.2 Windows

Pokretanje Pan&Zoom sustava na Windows operacijskom sustavu također ovisi o određenim alatima. Popis potrebnih alata za prevođenje programskog koda sustavnih modula na Windows platformi ponešto je duži, a postavljanje okoline nešto složenije. Prije instalacije potrebnih alata i prevođenja sustavnih modula poželjno je isključiti bilo koji oblik antivirusne zaštite. Nakon uspješne instalacije i pokretanja sustava, antivirusnu zaštitu može se ponovno uključiti. Važno je napomenuti kako sustav ne sadrži nikakav oblik virusa te se isključivanje anti-virusa odvija isključivo radi jednostavnije instalacije.

Kako s Windows platformom ne dolazi podrška i prevoditelj za izvršavanje *bash* skripti i naredbi putem terminala prvo je potrebno instalirati MSYS2 skupinu alata. S ovom skupinom alata dolazi GCC prevoditelj koji se koristi na Linux operacijskim sustavima i razne druge knjižnice. Instalacijski paket MSYS2 skupine alata potrebno je preuzeti s iduće poveznice: <https://www.msys2.org/> i pokrenuti instalaciju u ulozu administratora. Jednom kada je instalacija dovršena, potrebno je pokrenuti MSYS2 MSYS terminal iz mape *msys64* ili upisivanjem imena u Windows tražilicu. U terminalu je potrebno upisati iduće naredbe kako bi se instalirao iznad navedeni prevoditelj i potrebne knjižnice:

- `pacman -Syu`
- `pacman -Su`
- `pacman -S mingw-w64-x86_64-toolchain`

Sljedeći korak je dodavanje putanja (engl. Path) do *bin* mapa MSYS2 alata u sistemske varijable okoline (engl. Environment variables).

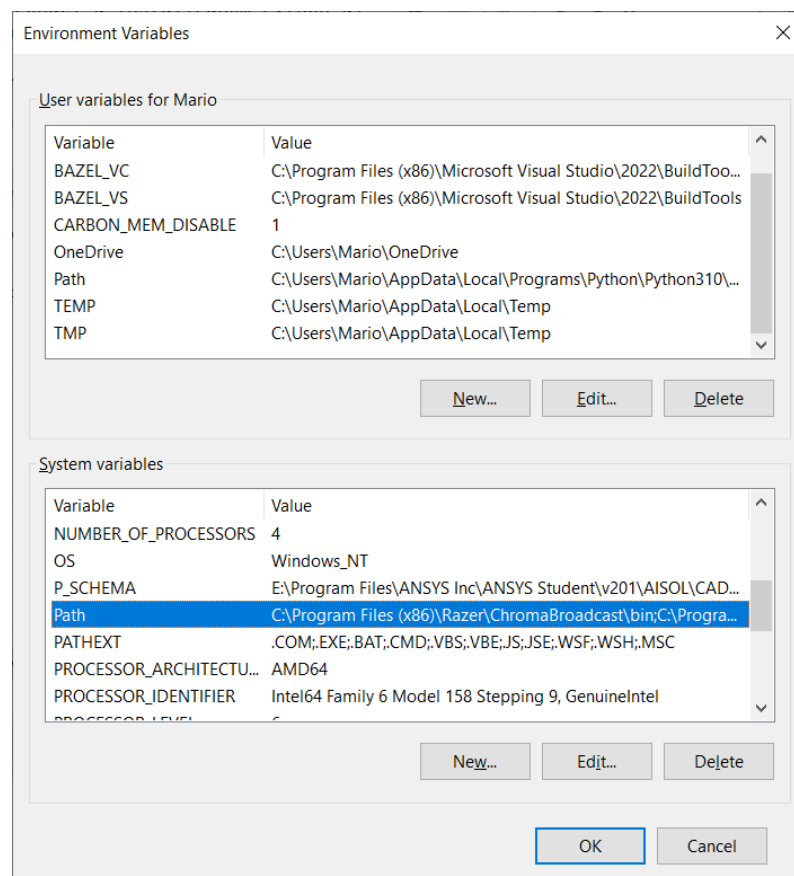
Sistemske varijable okoline sadrže putanje do određenih izvršnih datoteka i omogućuju njihovo pokretanje putem komandne linije (engl. Command Prompt) na bilo kojem mjestu bez potrebe za prethodnim pozicioniranjem u mapu gdje se one nalaze.

Dodavanje putanje u sistemske varijable okoline odvija se na idući način:

- Windows tražilica - „variables“ - „Edit the system environment variables“ - „Environment Variables...” – dobiva se prikaz kao na slici 3.20
- pod „System Variables“ - dvostruki klik na „Path“ varijablu – „New“ - dodati putanje do `msys64\usr\bin` i `msys64\mingw64\bin` mapa - „OK“
- Primjer putanja:

`C:\msys64\usr\bin`

`C:\msys64\mingw64\bin`



Slika 3.20. Prikaz sistemskih varijabli okoline

Mapu s programskim kodovima modula sustava Pan&Zoom moguće je, kao i kod Linux platforme, preuzeti ručno s GitHub-a ili putem `git clone` naredbe.

Sljedeći korak je instalacija Python-a verzije 3 i njegovih skripti. S iduće poveznice: <https://www.python.org/downloads/windows/> potrebno je preuzeti *Windows Installer (64-bit)* za Python verziju 3 (npr. 3.10.8) i pokrenuti instalaciju. Prilikom instalacije potrebno je odabrati opciju da se putanje do Python izvršne datoteke i njegovih skripti automatski dodaju u sistemske varijable okoline: „*Add python.exe to PATH*“, inače je to potrebno učiniti ručno na iznad opisan način.

Primjeri putanja do Python izvršne datoteke i njegovih skripti su:

- *C:\Users\Mario\AppData\Local\Programs\Python\Python310\*
- *C:\Users\Mario\AppData\Local\Programs\Python\Python310\Scripts\*

Jednom kada je Python uspješno instaliran, a putanje dodane u sistemske varijable potrebno je otvoriti komandnu liniju i instalirati *numpy* knjižnicu putem naredbe: *pip install numpy*.

Za prevođenje programskog koda *MediaPipe* modula na Windows platformi putem *Bazel* alata potrebno je prvo instalirati *Visual C++* alate koji uključuju i *MSVC* prevoditelj. Ovaj prevoditelj koristi se zato što sadrži skripte i *Windows* knjižnice za razvoj *UNIX* utičnica na *Windows* platformi. Sa sljedeće poveznice:

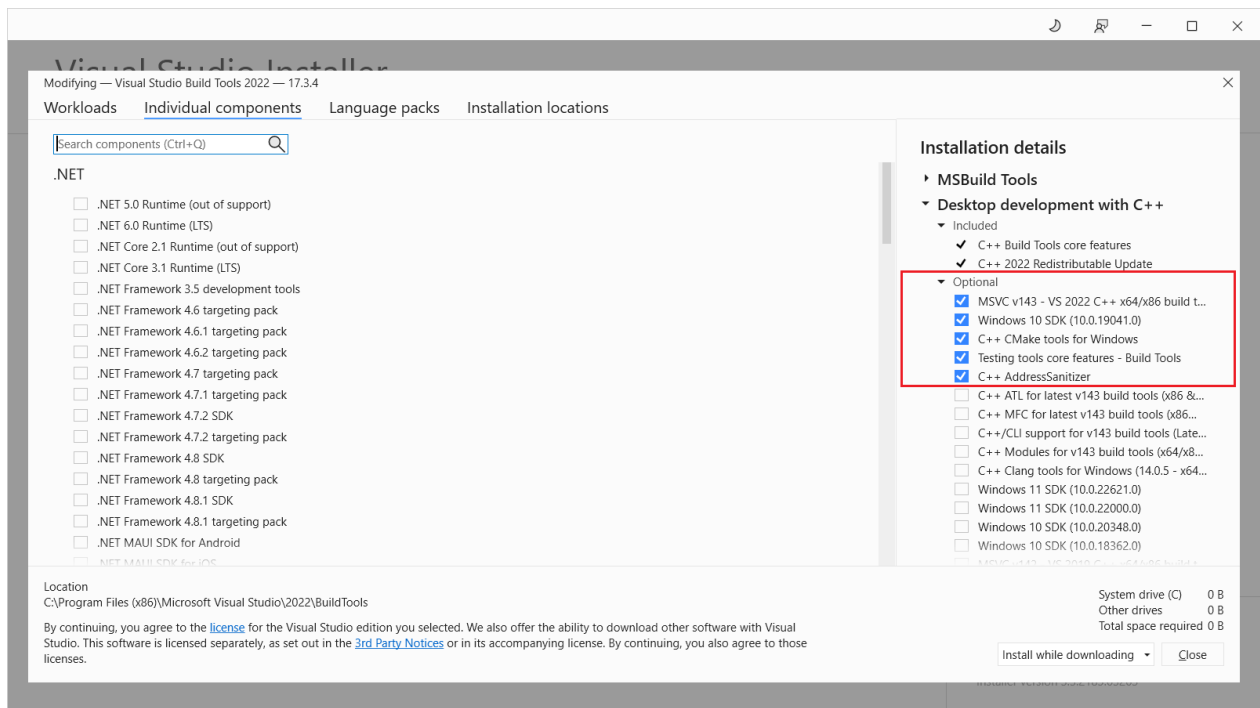
- <https://visualstudio.microsoft.com/visual-cpp-build-tools/>

potrebno je preuzeti i pokrenuti *Microsoft Visual Studio* instalaciju. U otvorenom izborniku „*Workload*“ odabrati opciju "*Desktop development with C++*" te otići u izbornik „*Individual components*“.

U izborniku „*Individual components*“ potrebno je odabrati iduće opcije:

- *MSVC v143 – VS 2022 C++ x64/x86 build tools (v14.33-17.3)* umjesto zadanog
- *C++ Cmake tools for Windows* (ako nije već odabrano)
- *Windows 10 SDK (10.0.19041.0)* ili *Windows 11 SDK (10.0.22000.0)* umjesto zadane

S ovim verzijama alata provjerena je uspješnost prevođenja programskog koda *MediaPipe* modula. Ako su odabrane sve komponente prema uputama zaslon bio trebao izgledati slično kao na slici 3.21. Važno je napomenuti kako su navedeni alati na slici 3.21 već prethodno instalirani na računalu na kojem se razvijao *Pan&Zoom* sustav pa se pokretanjem instalacijskog paketa otvara zaslon za modifikaciju instaliranih stavki. Ako stavke u crvenom okviru odgovaraju navedenim uputama može se nastaviti s instalacijom.



Slika 3.21. Instalacija potrebnih Visual C++ alata u operacijskom sustavu Windows

Sljedeći korak je preuzimanje Bazel softverskog alata za prevođenje MediaPipe programskog koda. Na idućoj poveznici: <https://github.com/bazelbuild/bazel/releases/tag/5.2.0> potrebno je preuzeti datoteku *bazel-5.2.0-windows-x86\_64.exe*, preimenovati datoteku u *bazel.exe* i dodati putanju do mape u kojoj se nalazi u sistemске varijable okoline prema već navedenim uputama.

Kako bi Bazel prilikom prevođenja programskog koda MediaPipe modula znao gdje se nalaze potrebni Visual C++ alati potrebno je generirati 3 nove sistemске varijable okoline:

- Ime varijable (engl. Variable name): **BAZEL\_VC**  
Vrijednost varijable (engl. Variable value): putanja do VC mape u instaliranom MS Build Tools-u  
Npr. *C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools\VC*
- Ime varijable: **BAZEL\_VS**  
Vrijednost varijable: putanja do mape *Build Tools* u instaliranom MS Build Tools-u  
Npr. *C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools*
- Ime varijable: **BAZEL\_VC\_FULL\_VERSION**  
Vrijednost varijable: 14.33.31629 (instalirana verzija MS Build Tools-a)

Posljednji korak prije prevođenja programskog koda MediaPipe modula je preuzimanje OpenCV knjižnice koja se koristi za računalni vid, strojno učenje i obradu slike. S iduće poveznice: <https://opencv.org/releases/> potrebno je preuzeti Windows instalaciju verzije 3.4.10 OpenCV knjižnice ili bilo koje druge verzije treće generacije. Za verzije četvrte generacije OpenCV-a potrebno je ručno mijenjati pojedine MediaPipe datoteke stoga je preporučljivo da se koristi neka od verzija treće generacije. Ukoliko je instalirana neka druga verzija treće generacije, a ne 3.4.10, potrebno je u datoteci *Pan\_Zoom/mediapipe/third\_party/opencv\_windows.BUILD* promijeniti liniju koda `OPENCV_VERSION`. Također, ukoliko OpenCV knjižnica nije instalirana na *C:/* disku potrebno je promijeniti 274. liniju koda u datoteci *Pan\_Zoom/mediapipe/WORKSPACE* na idući način:

- `path = "C:\\opencv\\build"`, - ovdje je potrebno staviti putanju do *build* mape u instaliranoj knjižnici OpenCV

Uspješnom instalacijom OpenCV knjižnice može se prijeći na prevođenje programskog koda MediaPipe modula. Pomoću komandne linije (engl. Command Prompt) potrebno se pozicionirati u mapu sustava *Pan\_Zoom* te izvršiti iduće naredbe:

- `cd mediapipe`
- `bash build_pan_zoom.sh windows <putanja do python.exe datoteke>`
- Primjer putanje do *python.exe* datoteke:

`C://Users//Mario//AppData//Local//Programs//Python//Python310//python.exe`

Vrlo je važno prilikom specificiranja putanje do *python.exe* datoteke koristiti `//` umjesto `/`. Ako prevođenje programa ne prolazi uspješno do kraja, što se zna nekada dogoditi, potrebno je zaustaviti prevođenje kombinacijom tipki „*Ctrl+C*“ u komandnoj liniji, ponovno pokrenuti računalo, u *Pan\_Zoom/mediapipe* mapi putem komandne linije pokrenuti naredbu: `bazel clean --expunge` kojom će se očistiti podaci vezani uz posljednje prevođenje programa te ponovno pokrenuti `bash` skriptu `build_pan_zoom.sh`. Modeli strojnog učenja MediaPipe modula se na Windows platformi izvršavaju na procesoru, a ne na grafičkoj kartici kao kod Linux platforme. Razlog tome je što Windows još uvijek nema podršku za izvršavanje MediaPipe softverskih rješenja na grafičkoj kartici [22].

Za prevođenje programskog koda Image-viewer modula na Windows platformi potrebno je instalirati još dva alata: Cmake i Qt. Cmake alat omogućuje prevođenje programskog koda pomoću jednostavnih konfiguracijskih datoteka neovisnih o platformi i prevoditelju. S iduće poveznice: <https://cmake.org/download/> potrebno je preuzeti *Cmake Windows x64 Installer* izvršnu datoteku i pokrenuti njegovu instalaciju.

Prilikom instalacije je potrebno ostaviti opciju da se putanja do *bin* mape automatski dodaje u sistemske varijable okoline.

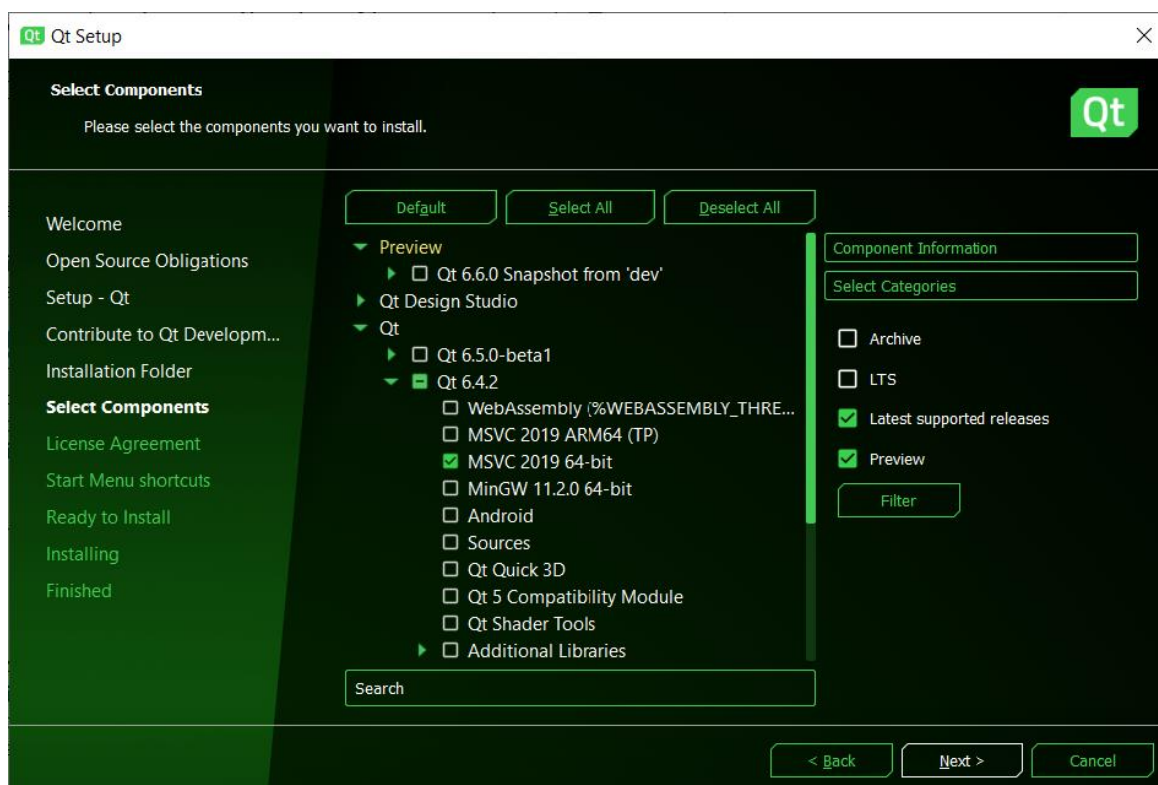
Princip instalacije Qt softvera na Windows platformi sličan je instalaciji na Linux platformi. Qt softver sadrži alate za stvaranje i upravljanje grafičkim sučeljima u C++ programskom jeziku. Na sljedećoj poveznici: <https://www.qt.io/download-open-source> potrebno je pronaći odjeljak „Get started“, preuzeti *Qt Online Installer* izvršnu datoteku za Windows platformu i pokrenuti instalaciju. U početku instalacije treba se stvoriti Qt korisnički profil i nastaviti s instalacijom. Nakon uspješne registracije potrebno je nastaviti instalaciju idućim koracima:

- Odabrati opciju „*Custom Installation*“ i bilo koju Qt 6 verziju
- Iz odabrane verzije označiti samo opciju „*MSVC 2019 64-bit*“ kao na slici 3.22
- Nastaviti i dovršiti instalaciju slijedeći upute na zaslону
- Nakon instalacije ručno dodati putanje do *bin* i *lib* mapa iz MSVC 2019 prevoditelja u sistemske varijable okoline prema već navedenim uputama

- Primjeri putanja:

*E:\Qt\6.4.0\msvc2019\_64\bin*

*E:\Qt\6.4.0\msvc2019\_64\lib*



Slika 3.22. Instalacija Qt softvera u Windows operacijskom sustavu

Nakon uspješno instaliranih Cmake i Qt alata može se prijeći na posljednji korak instalacije Pan&Zoom sustava, a to je prevođenje programskog koda Image-viewer modula. Za to je potrebno pokrenuti „*Development Command Prompt for VS*“ komandu liniju te slijediti iduće korake:

- *cd Pan\_Zoom*
- *cd image-viewer*
- *mkdir build*
- *cd build*
- *cmake .. -DCMAKE\_PREFIX\_PATH="<putanja do MSVC prevoditelja unutar Qt softvera>"*

Primjer putanje:

*E:\Qt\6.4.0\msvc2019\_64\* - ovdje se nalaze sve potrebne Qt knjižnice, obavezno koristiti *\\* umjesto *\* ili */*

- *msbuild PallImageViewer.sln -p:Configuration=Release*

Izvršna datoteka Image-viewer modula pojavljuje se u mapi s putanjom *Pan\_Zoom/image-viewer/build/example/Release*, a u svome radu koristi *PallImageViewer.dll* datoteku koja se nalazi u *Pan\_Zoom/image-viewer/build/src/Release*. Putanju do *PallImageViewer.dll* datoteke potrebno je dodati u *Path* sistemsku varijablu okoline kako bi ju izvršna datoteka mogla pronaći.

Pokretanje Pan&Zoom sustava na Windows platformi odvija se, kao i na Linux platformi, pomoću dviju *bash* skripti: *train-experiment.sh* i *run-experiment.sh*. Navedene skripte pokreću oba modula. Za pokretanje Pan&Zoom sustava potrebno je koristiti iduće naredbe:

- *cd Pan\_Zoom*
- *bash train-experiment.sh windows* – za upoznavanje s korištenjem sustava
- *bash run-experiment.sh windows data/test-sequence-1.txt* – za pokretanje HCI eksperimenta s određenim poretком zadataka

Za pravilno terminiranje sustava potrebno je prvo pritisnuti gumb „*Exit*“ na grafičkom sučelju Image-viewer modula, a zatim pomoću kombinacije tipki „*Ctrl+C*“ zatvoriti MediaPipe modul za upravljanje gestama ruku. Navedeni postupak terminiranja sustava potrebno je slijediti kako bi se komunikacija među programima zatvarala na ispravan način.



## 4. HCI EKSPERIMENT

HCI je multidisciplinarno područje istraživanja usmjereno na dizajn računalne tehnologije s ciljem ostvarivanja što uspješnije interakcije između čovjeka i računala. Primarni cilj ovoga područja je stvaranje upotrebljivih, sigurnih i funkcionalnih sustava koji će omogućiti što jednostavniju i bržu interakciju čovjeka i računala. Iz navedenih tvrdnji može se zaključiti kako ovo područje stavlja naglasak na zadovoljstvo i efikasnost čovjekova korištenja sustavom. Ljudi ne bi trebali mijenjati svoj način upravljanja sustavom, već bi sustav trebao biti dizajniran da odgovara njihovim potrebama [23].

Kako bi računalna tehnologija konstantno napredovala te time pružala bolju uslugu korisnicima, potrebno je provoditi stalna testiranja, anketiranja i prikupljanja podataka o njenom radu. Temeljem prikupljenih podataka i povratnih informacija korisnika moguće je unaprijediti postojeće sustave i znati o čemu voditi računa prilikom razvoja novih značajki [24].

Jedan od načina prikupljanja podataka o radu sustava i povratnih informacija korisnika je i provedba HCI eksperimenta. HCI eksperiment obično se sastoji od neke vrste zadatka kojeg korisnici moraju izvršiti koristeći se sustavom. Podaci prikupljeni HCI eksperimentom mogu se podvrgnuti statističkoj analizi kako bi se dobila važna saznanja o radu sustava. Temeljem tih saznanja sustav se može unaprijediti kako bi težio onom primarnom cilju, a to je uspješna interakcija čovjeka i računala. Ovakvi eksperimenti vrlo su popularni i prilikom razvoja i testiranja novih značajki sustava velikih kompanija.

Sustavi zasnovani na gestovnoj interakciji, kao što je moguće vidjeti po već navedenim primjerima u poglavlju 2., sustavi su budućnosti. Jednostavnost korištenja prednost je ovakvih sustava koja im donosi raznovrsnu primjenu u različitim područjima. Zbog ove prednosti, ali i mnogih drugih, ovakve vrste sustava konstantno se razvijaju i nadograđuju. Zbog svojeg, svijetu još relativno novog pristupa rješavanju problema (korištenjem gestikulacija), nad ovakvim sustavima vrlo se često provode razni eksperimenti i testiranja kako bi se utvrdilo koliko su oni zapravo prilagođeni korisniku i na kakve sve poteškoće korisnici nailaze prilikom njihove upotrebe.

U sklopu ovoga rada proveden je HCI eksperiment s ciljem dobivanja raznih zaključaka po pitanju upotrebljivosti implementiranih modaliteta Pan&Zoom gestovnog sustava, složenosti eksperimentalnih zadataka i zadovoljstva korisnika.

## 4.1 Dizajn eksperimenta

Prilikom razvoja Pan&Zoom gestovnog sustava implementirana su dva različita modaliteta: MH i SH. Prvi cilj provedbe HCI eksperimenta bio je utvrditi postoji li signifikantna razlika u učinkovitosti korištenja sustavom između dvaju navedenih modaliteta. Učinkovitost korištenja predstavljena je vremenom koje je potrebno da se izvrše određeni zadaci upravljanja slikovnim sadržajem. Smatra se da će modalitet s kojim je potrebno manje vremena za obavljanje takvih zadataka biti učinkovitiji od onoga s kojim je potrebno više vremena. Prvotna pretpostavka je bila da će SH modalitet biti učinkovitiji jer omogućuje paralelno izvršavanje akcija pomicanja i skaliranja koristeći različite ruke, dok se kod MH modaliteta za akciju skaliranja moraju koristiti obje ruke istovremeno. Za naučiti i ovladati paralelnim izvršavanjem akcija pomicanja i skaliranja obično je potrebno neko duže vrijeme pa se može dogoditi da vrijeme izvršavanja zadataka bude ipak sporije korištenjem SH modaliteta. U konačnici, sve ovisi o ispitnim korisnicima i njihovim sposobnostima da nauče koristiti implementirane modalitete u što kraćem vremenu. U sklopu zadatka pomicanja i uvećavanja slikovnog sadržaja koristile su se generirane slike kao reprezentanti zadataka triju različitih složenosti (jednostavno, umjereno i složeno).

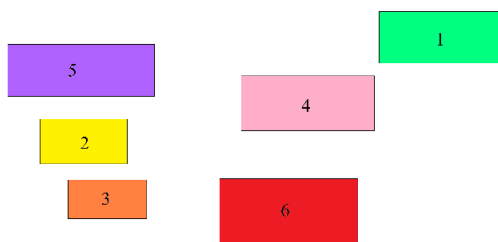
Drugi cilj ovoga eksperimenta bio je utvrditi postoji li signifikantna razlika između vremena potrebnog za pomicanje i skaliranje stavki u zadacima različitih složenosti. Pretpostavka je bila da će najmanje vremena biti potrebno za rješavanje jednostavnih zadataka, a najviše za rješavanje složenih zadataka i to pri korištenju oba modaliteta. Temeljem dvaju navedenih ciljeva, HCI eksperiment uključivao je dvije nezavisne varijable: modalitet (engl. modality) i složenost zadatka (engl. complexity) te jednu zavisnu varijablu, a to je vrijeme potrebno za obavljanje zadatka. Opisani dizajn eksperimenta moguće je, radi lakšeg razumijevanja, grafički sažeti i prikazati slikom 4.1.



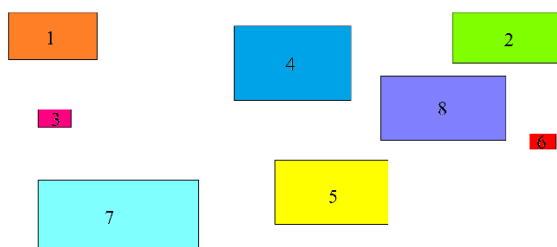
Slika 4.1. Dizajn HCI eksperimenta

## 4.2 Zadaci upravljanja prikazom sadržaja na zaslonu

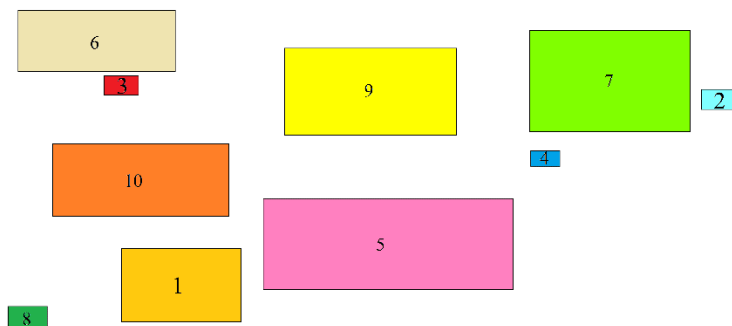
Za provedbu HCI eksperimenta bilo je potrebno osmisliti adekvatne zadatke upravljanja slikovnim sadržajem. Testni ciklus uključivao je 6 različitih zadataka ravnomjerno raspoređenih u 3 kategorije prema složenosti na jednostavne, umjereno složene i složene. Svaki zadatak bio je predstavljen određenom slikom. Na slikama su se nalazili nasumično raspoređeni pravokutnici različitih boja i veličina. Na slikama jednostavnih zadataka (engl. Easy) nalazilo se ukupno 6 većih osnovnih pravokutnika približno jednakih veličina (slika 4.2). Na slikama umjereno složenih zadataka (engl. Medium) su se uz 6 navedenih pravokutnika nalazila i dodatna 2 manja pravokutnika (slika 4.3), dok su se na slikama složenih zadataka (engl. Hard) uz 6 osnovnih pravokutnika nalazila dodatna 4 manja pravokutnika (slika 4.4). Pravokutnici na slikama bili su numerirani rastućim redoslijedom. Zadatak ispitnog korisnika bio je uspješno skalirati sve pravokutnike numeričkim redoslijedom na 6 različitih slika koristeći oba modaliteta (ukupno 12 zadataka tj. 2 testna ciklusa).



Slika 4.2. Primjer slike jednostavnog zadatka



Slika 4.3. Primjer slike umjereno složenog zadatka

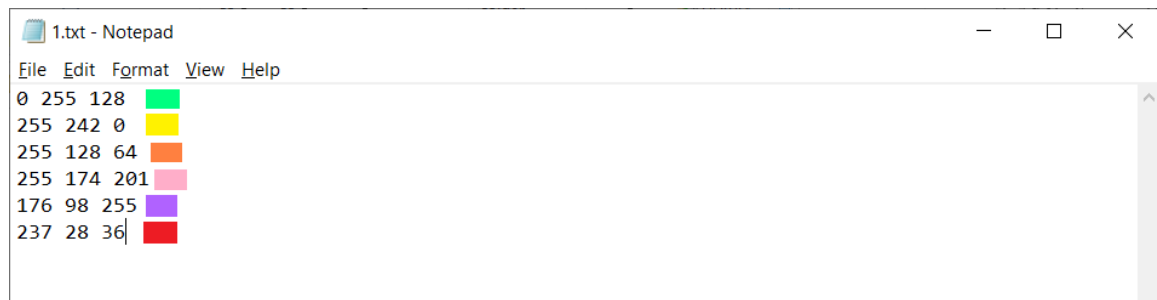


Slika 4.4. Primjer slike složenog zadatka

Slike su generirane u MS Paint aplikaciji na Windows platformi te se tijekom provedbe eksperimenta učitavaju u okvir grafičkog sučelja Image-viewer modula. Boje pravokutnika koriste se kako bi Image-viewer mogao odrediti kada je pravokutnik skaliran do potrebne razine te obavijestiti korisnika da može prijeći na idući. Slike korištene u ovome eksperimentu nalaze se u mapi na putanji *Pan\_Zoom/data/pictures* zajedno s pripadajućim *.txt* datotekama. Unutar *.txt* datoteka nalaze se boje pravokutnika u RGB formatu poredane istim numeričkim redoslijedom kao i pravokutnici na slici. Imena slika i pripadajućih *.txt* datoteka moraju biti jednake kako bi modul znao koje RGB vrijednosti boja pravokutnika odgovaraju kojoj slici. Ukoliko se eksperiment želi provesti korištenjem drugačijih slika, postupak za dodavanje nove slike odnosno novog zadatka je sljedeći:

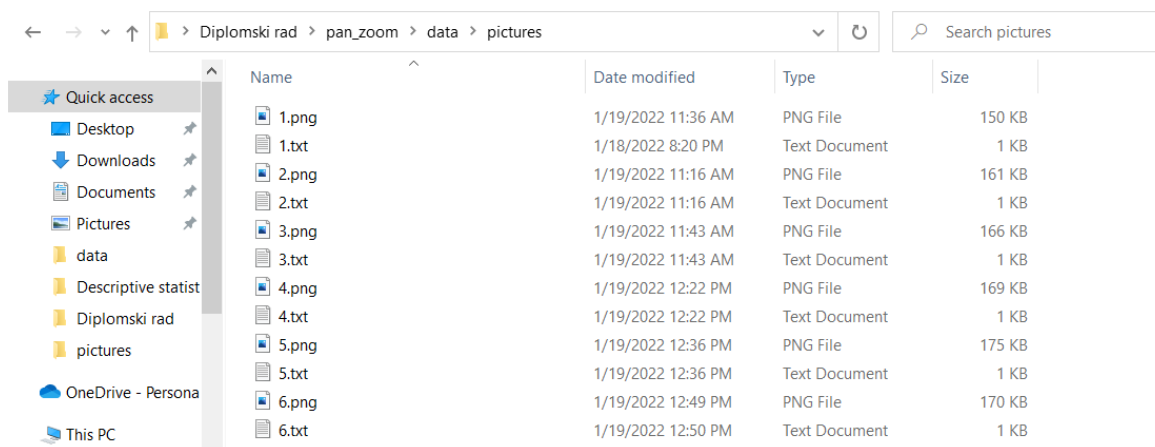
1. Stvoriti proizvoljnu sliku s pravokutnicima različitih boja putem MS Paint aplikacije ili nekog drugog alata (poželjno je da bude što veće rezolucije)
2. Pravokutnike numerirati proizvoljno u rastućem redoslijedu kao na iznad prikazanim primjerima
3. Spremiti sliku pod određenim imenom (npr. <ime\_slike>.png)
4. Dodati sliku u mapu *Pan\_Zoom/data/pictures*
5. Stvoriti *.txt* datoteku, dodijeliti joj ISTO ime kao i slici te ju spremiti u navedenu mapu
6. Prolazeći pravokutnike na slici numeričkim redoslijedom, potražiti odgovarajuću boju u RGB formatu i dodati ju u stvorenu *.txt* datoteku
7. Nakon posljednje dodane RGB vrijednosti spremiti sadržaj *.txt* datoteke

Na slici 4.5 prikazana je jedna od *.txt* datoteka koja je povezana s već iznad prikazanom slikom 4.2. Može se vidjeti kako su RGB vrijednosti boja unutar *.txt* datoteke poredane točno redosljedom kojim su numerirani i pravokutnici na slici. Pored RGB vrijednosti prikazani su mali pravokutnici tih boja radi lakše usporedbe s povezanom slikom. Ove tekstualne datoteke koristi Image-viewer modul kako bi znao je li pravokutnik skaliran do određene razine te koji pravokutnik je idući na redu u tijeku eksperimentalnog zadatka.



Slika 4.5. Tekstualna datoteka s RGB definicijom korištenih boja u odnosnom zadatku

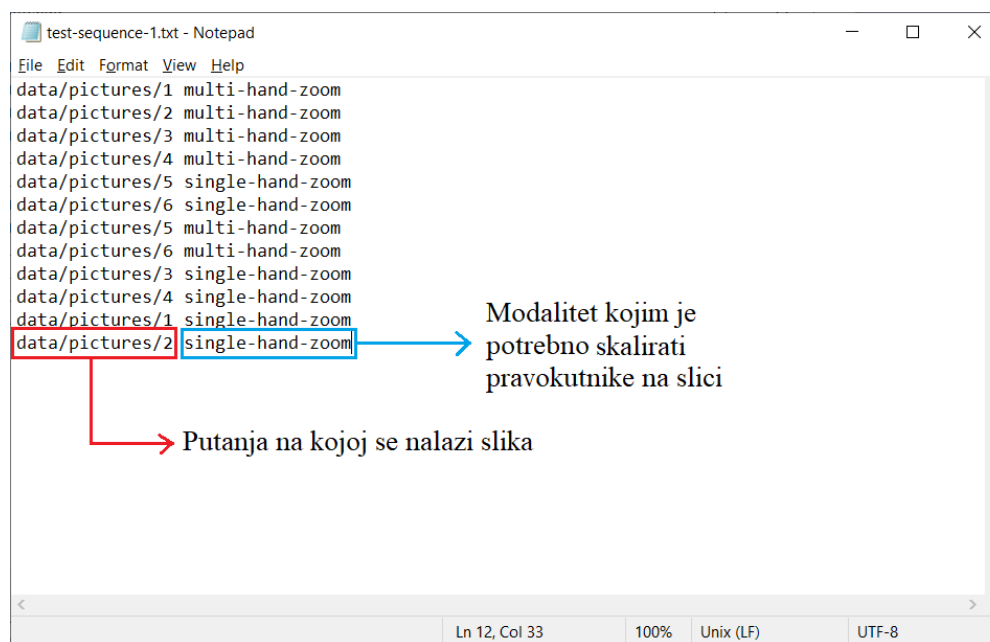
Mapa *pictures* u putanji *Pan\_Zoom/data/pictures* trebala bi izgledati kao na slici 4.6 na kojoj se može vidjeti kako su imena slika i odgovarajućih *.txt* datoteka u potpunosti jednaka. Mapa naziva *starter\_test* u putanji *Pan\_Zoom/data/starter\_test* izgleda jednako kao i mapa *pictures*, a u njoj se nalaze dvije slike s pripadajućim *.txt* datotekama koje se koriste za izvođenje probnog eksperimenta jednom kada korisnik ovlada načinom korištenja implementiranih MH i SH modaliteta.



Slika 4.6. Prikaz mape *pictures* u *Pan&Zoom* sustavu

Svaki zadatak uključivao je odgovarajuću sliku na kojoj se trebaju skalirati pravokutnici i određeni modalitet kojim je to potrebno učiniti. Poredak zadataka nije bio jednak svim ispitnim korisnicima, već je napravljeno 6 različitih poredaka po principu uravnoteženih latinskih kvadrata (engl. Balanced latin squares) [25]. U niti jednom od poredaka nisu se pojavljivali zadaci s dvije iste slike zaredom. Ovaj princip onemogućava učenje pozicija i redosljeda pravokutnika na slici korištenjem jednog modaliteta i ostvarivanju boljih performansi na idućem zadatku s istom slikom korištenjem drugoga modaliteta.

U svakom poretku, zadaci jednake složenosti dolazili su jedni za drugim i bilo ih je potrebno odraditi koristeći isti modalitet. Svaki poredak bio je korišten 5 puta ( $N = 30$ ), a tekstualne datoteke poretka zadataka nalazile su se na putanji *Pan\_Zoom/data*. Primjer jedne tekstualne datoteke s poretkom zadataka moguće je vidjeti na slici 4.7.



```
test-sequence-1.txt - Notepad
File Edit Format View Help
data/pictures/1 multi-hand-zoom
data/pictures/2 multi-hand-zoom
data/pictures/3 multi-hand-zoom
data/pictures/4 multi-hand-zoom
data/pictures/5 single-hand-zoom
data/pictures/6 single-hand-zoom
data/pictures/5 multi-hand-zoom
data/pictures/6 multi-hand-zoom
data/pictures/3 single-hand-zoom
data/pictures/4 single-hand-zoom
data/pictures/1 single-hand-zoom
data/pictures/2 single-hand-zoom
```

Slika 4.7. Tekstualna datoteka koja definira poredak zadataka

Ukoliko se želi dodati novi zadatak i ponoviti ovakav eksperiment, potrebno je, uz pripadajuću istoimenu tekstualnu datoteku s poredanim RGB vrijednostima boja, dodati i tekstualna datoteka poretka zadataka kao na slici iznad u mapu *data*.

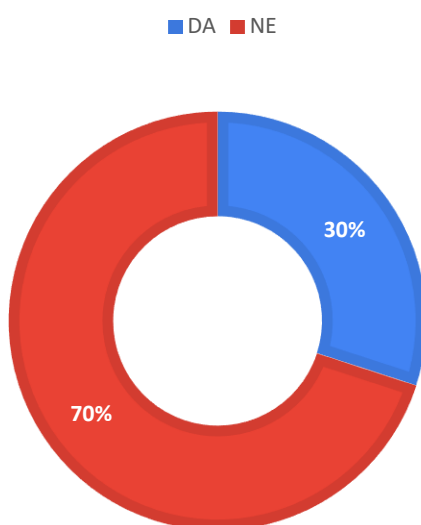
U novoj datoteci poretka zadataka, uz već navedenih 12 zadataka, moraju se nadodati još dva zadatka koja uključuju novu sliku i dva modaliteta kojima će se odvijati skaliranje pravokutnika. Također, nova dva zadatka potrebno je dodati i u preostalim 6 poredaka poštujući princip uravnoteženih latinskih kvadrata.

### 4.3 Ispitni korisnici

U provedbi HCI eksperimenta sudjelovalo je 30 ispitnih korisnika što se smatra dovoljno velikim brojem za predstavljanje istinitih, a ne slučajnih eksperimentalnih rezultata. Od 30 korisnika njih 25 (83%) radi ili studira u IT sektoru tj. području tehničkih znanosti. Sve osobe koriste računalo u svome poslu ili studiranju, a njih 18 (60%) vezano je konkretno uz računarstvo i programiranje. Među ispitnim korisnicima prevladavaju muške osobe (87%). Prosječna starost ispitnih korisnika iznosi  $24.8 \pm 5.97$  godina. Valja napomenuti kako je najstarija osoba imala 48 godina, a najmlađa 20 iz čega se može zaključiti kako zadatak HCI eksperimenta nije previše složen te ga mogu odraditi osobe različitih dobnih skupina.

Samo dvije osobe su ljevaci što iznosi 7%. Prilikom provedbe eksperimenta nije uzeta u obzir različita efikasnost pri skaliranju pravokutnika na slikama između ljevaka i desnjaka koristeći SH modalitet. Naime, kod SH modaliteta akciju skaliranja moguće je izvesti jedino koristeći desnu ruku, dok kod MH modaliteta dominantna ruka ne igra nikakvu ulogu jer se akcija skaliranja odvija koristeći obje ruke istovremeno.

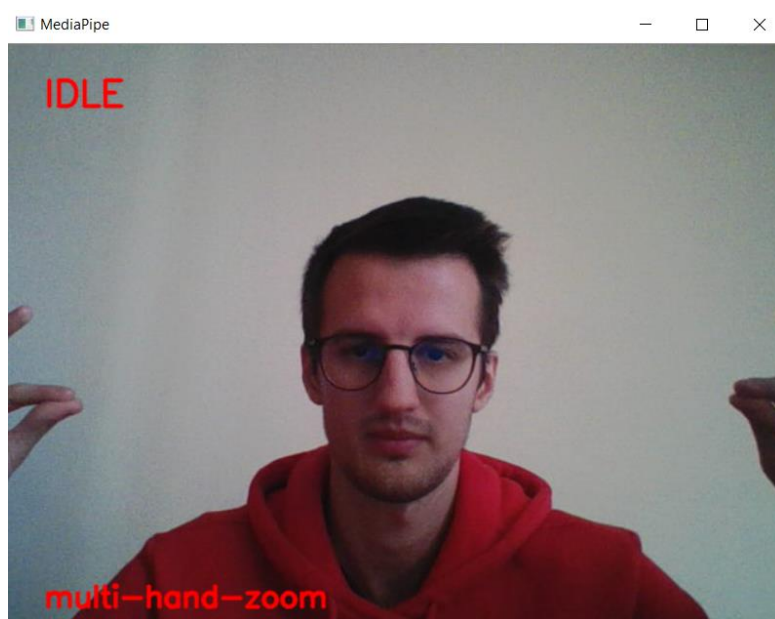
Slika 4.8 prikazuje prethodno iskustvo ispitnih korisnika s digitalnim gestovnim sustavima slobodnog oblika kod kojih se akcije izvode gestama. 9 osoba (30%) imalo je prethodnog iskustva s korištenjem ovakvih sustava poput igranja Nintendo igraće konzole, VR tehnologije i sl. dok je za 21 osobu (70%) ovo bilo jedno posve novo iskustvo.



Slika 4.8. Iskustvo ispitnih korisnika s gestovnom interakcijom (N=30)

#### 4.4 Oprema i uvjeti za provedbu eksperimenta

Za izvođenje ovoga eksperimenta korišteno je prijenosno računalo Lenovo Legion Y520 s ugrađenom 720p HD kamerom rezolucije 1280 x 720 točaka i 64-bitnim operacijskim sustavom Windows 10. Vrlo je važno koristiti kameru ovakve ili veće rezolucije kako bi MediaPipe modul uspješno prepoznao geste u radu sustava. Prije same provedbe eksperimenta bilo je potrebno pronaći ispravnu poziciju ispitnog korisnika pred kamerom računala. Ako se ispitni korisnik nalazi preblizu kameri postoji mogućnost da mu ruke prilikom izvođenja akcija pomicanja i skaliranja iziđu izvan područja kamere (slika 4.9) što rezultira gubljenjem kontrole nad radom sustava, prekidanjem spomenutih akcija te u konačnici dužim vremenom izvršavanja eksperimentalnog zadatka. S udaljavanjem od kamere smanjuje se veličina dlanova ruku. Ako se ispitni korisnik previše udalji od kamere, sustav neće biti u mogućnosti prepoznati i lokalizirati ključne točke dlanova što dovodi do nemogućnosti izvođenja akcija. Za rješavanje ovih problema može se koristiti širokokutna kamera visoke rezolucije. Uspješna detekcija dlanova i njihovih ključnih točaka ovisi i o količini svjetla u prostoriji. Ukoliko je korisnik pronašao ispravnu poziciju pred kamerom, a događa se da sustav ne prepoznaje ključne točke dlanova ili ih svako malo „izgubi“, razlog tome je nedovoljna količina svjetla u prostoriji. Također, važno je da se prilikom izvršavanja zadatka u području kamere nalazi samo ispitni korisnik jer pojava druge osobe može „zbuniti“ modele za prepoznavanje dlanova što ponovno rezultira dužim izvođenjem eksperimentalnog zadatka.



Slika 4.9. Problem položaja ruku izvan područja kamere



## 4.5 Procedura provedbe eksperimenta

Prije početka provedbe eksperimenta, ispitnom korisniku je objašnjen koncept eksperimenta i zadaci koje će morati odraditi te mu je ostavljeno vrijeme za dodatna pitanja. Jednom kada je korisniku bilo sve jasno u vezi zadataka, trebao je ispuniti obrazac kojim svojevolumno pristaje biti dio eksperimenta te popuniti upitnik s osobnim podacima. Navedeni upitnik uključivao je ime i prezime korisnika, broj godina, dominantnu ruku, spol, zanimanje i prethodno iskustvo s gestovnom interakcijom sustava slobodnih oblika.

Idući korak zahtijevao je ispunjavanje potrebnih uvjeta za uspješno izvođenje zadataka eksperimenta. Kako je već prethodno navedeno, bilo je potrebno pronaći dovoljno osvjetljeno mjesto u prostoriji gdje se ne kreće veliki broj ljudi kako bi sustav mogao raditi na ispravan način. Ispitni korisnici trebali su pronaći ispravnu poziciju pred kamerom kako ne bi imali poteškoća prilikom izvođenja akcija za vrijeme trajanja eksperimenta. Kada su navedeni uvjeti bili zadovoljeni pokrenuo bi se svojevrsni trening koji je sadržavao dva zadatka. Svaki zadatak uključivao je po jednu pripadajuću sliku na kojoj su se nalazila tri veća pravokutnika koje je bilo potrebno skalirati numeričkim redosljedom u što kraćem vremenu. U prvom zadatku bilo je potrebno koristiti MH modalitet, a u drugom SH modalitet. Mjerenje vremena za izvršavanje zadatka započinjalo je pritiskom na gumb „*Start Test*“, a završavalo uspješnim skaliranjem svih pravokutnika na slici. Prije pokretanja mjerenja vremena, ispitnim korisnicima je objašnjeno i pokazano kako koristiti modalitete te im je dano određeno vrijeme treniranja kako bi ovladali principom njihova korištenja. Jednom kada su bili spremni, pokrenulo bi se vrijeme za skaliranje pravokutnika na trenutnoj slici. Skaliranje pojedinog pravokutnika bilo bi uspješno kada bi se u sva četiri kuta okvira u kojem je učitana slika nalazila ista boja (slika 4.10) i kada bi numerički redosljed bio ispravan.

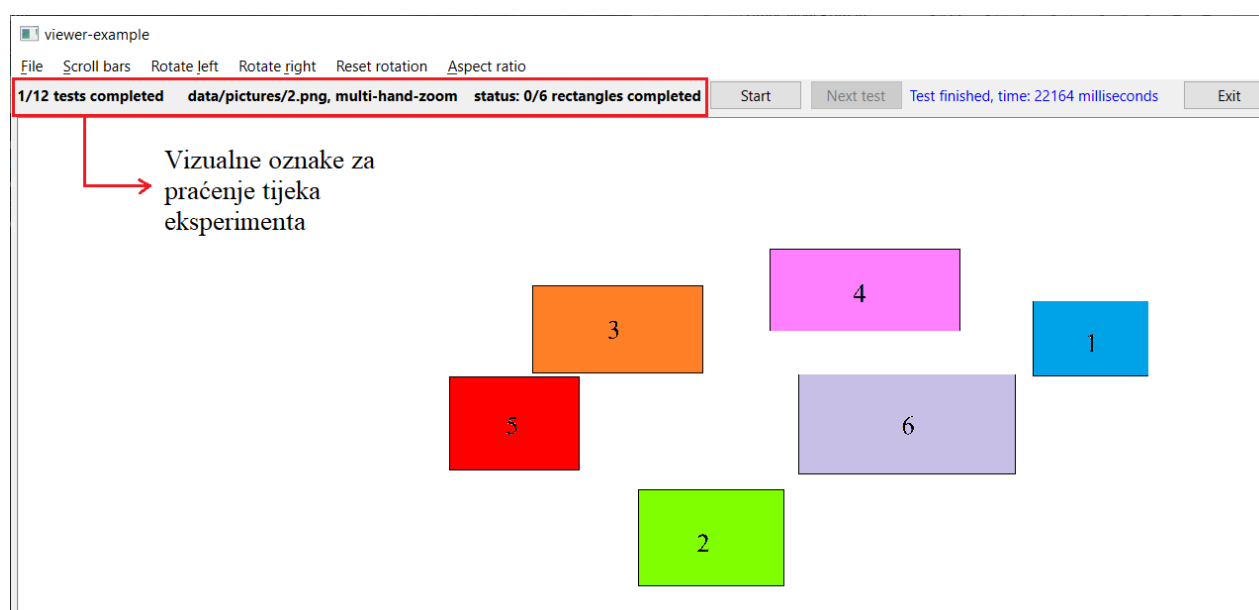


Slika 4.10. Uspješno skaliran pravokutnik na slici zadatka

Važno je napomenuti kako su brojevi na slikama navedeni samo kako bi korisnik znao kojim redoslijedom skalirati pravokutnike. Oni se ne koriste za uspješnu detekciju skaliranog pravokutnika i ne moraju uopće biti u skaliranom području. Image-viewer modul prati redoslijed boja u RGB formatu tekstualne datoteke slike (.txt) i na taj način „zna“ koji je idući pravokutnik po redu kojeg ispitni korisnik mora skalirati.

Zadatak bi bio uspješno odrađen kada bi svi pravokutnici na slici bili zumirani ispravnim redoslijedom koristeći zadani modalitet. Tada se pritiskom na gumb „Next test“ prelazi na idući zadatak s novom slikom kojeg je potrebno odraditi s odgovarajućim modalitetom i tako sve do posljednjeg. Modalitete se, ako je to potrebno, između zadataka mijenja putem slova „s“ na tipkovnici. Uspješnim odrađivanjem treninga korisnik je uvježbao korištenje modaliteta te je mogao prijeći na izvođenje konkretnih eksperimentalnih zadataka.

Tijek eksperimentalnih zadataka korisnici su mogli pratiti putem oznaka na vrhu grafičkog sučelja Image-viewer modula kao na slici 4.11. Oznake pružaju informacije o zadatku koji je trenutno na redu, putanji na kojoj se nalazi trenutna slika zadatka, modalitetu kojim se treba izvršiti trenutni zadatak te broju pravokutnika koje se uspješno skaliralo na trenutnoj slici. Ispitni korisnik je, osim vizualne informacije o trenutnom broju skaliranih pravokutnika na slici, imao i zvučnu povratnu informaciju o uspješnom skaliranju pravokutnika koja bi mu sugerirala da može prijeći na idući.



Slika 4.11. Vizualne oznake za praćenje tijeka eksperimenta

Prilikom pokretanja ciklusa eksperimentalnih zadataka bilo je potrebno *bash* skripti proslijediti argument u kojem se navodi putanja do tekstualne datoteke koja definira jedan od mogućih šest poredaka zadataka. Poredci zadataka korišteni su ciklično rastućim redoslijedom. Na samom početku izvođenja zadataka bilo je potrebno unijeti korisničko ime u obliku prvog slova imena i punog prezimena bez korištenja dijakritičkih znakova (npr. mmaricevic). Pomoću korisničkog imena, navedena *bash* skripta stvorila bi CSV datoteku u koju će se zapisivati rezultati svakog pojedinog zadatka ispitnog korisnika. Također, prilikom pokretanja ciklusa zadataka s prvim ispitnim korisnikom stvorena je CSV datoteka imena *concatenated-log* u koju su se redom zapisivali rezultati zadataka svih ispitnih korisnika u eksperimentu. Sve CSV datoteke nalazile su se u mapi *results* u putanji *Pan\_Zoom/results*. Ispitni korisnik izvršavao je zadatke redom s mogućnošću odmora od oko 20 do 30 sekundi nakon svakog pojedinog zadatka. Rezultati su se ispisivali u komandnu liniju kao na slici 4.12. te automatski prosljeđivali i zapisivali u korisničku CSV datoteku i datoteku sa svim dobivenim rezultatima (*concatenated-log.csv*).

```

Command Prompt - bash run-experiment.sh windows data/test-sequence-1.txt

E:\Desktop\pan_zoom>bash run-experiment.sh windows data/test-sequence-1.txt
Enter username (npr. iivic za Ivo Ivić): mmaricevic
Accepting connections on: C:/Users/Mario/AppData/Local/Temp/image-viewer-server.sock
Accepted a connection to mediapipe (client).
mmaricevic, data/test-sequence-1.txt, data/pictures/1.png, multi-hand-zoom, 22164

```

Slika 4.12. Ispisivanje rezultata o izvođenju zadatka u komandnu liniju

Primjer jedne korisničke CSV datoteke s rezultatima svih završenih zadataka moguće je vidjeti na slici 4.13. Svaki zapis izvršenog zadatka sastojao se od 5 dijelova: korisničko ime, korišteni poredak zadataka, putanja do slike korištene u zadatku, modalitet kojim se zadatak trebao izvršiti te vrijeme izvršavanja zadatka izraženo u milisekundama.

|    | A          | B                        | C                   | D                | E                 |
|----|------------|--------------------------|---------------------|------------------|-------------------|
| 1  | username   | test-sequence            | test-image          | modality         | time-milliseconds |
| 2  | mmaricevic | data/test-sequence-1.txt | data/pictures/1.png | multi-hand-zoom  | 40461             |
| 3  | mmaricevic | data/test-sequence-1.txt | data/pictures/2.png | multi-hand-zoom  | 53767             |
| 4  | mmaricevic | data/test-sequence-1.txt | data/pictures/3.png | multi-hand-zoom  | 61497             |
| 5  | mmaricevic | data/test-sequence-1.txt | data/pictures/4.png | multi-hand-zoom  | 62675             |
| 6  | mmaricevic | data/test-sequence-1.txt | data/pictures/5.png | single-hand-zoom | 78280             |
| 7  | mmaricevic | data/test-sequence-1.txt | data/pictures/6.png | single-hand-zoom | 69771             |
| 8  | mmaricevic | data/test-sequence-1.txt | data/pictures/5.png | multi-hand-zoom  | 83544             |
| 9  | mmaricevic | data/test-sequence-1.txt | data/pictures/6.png | multi-hand-zoom  | 84923             |
| 10 | mmaricevic | data/test-sequence-1.txt | data/pictures/3.png | single-hand-zoom | 55142             |
| 11 | mmaricevic | data/test-sequence-1.txt | data/pictures/4.png | single-hand-zoom | 51288             |
| 12 | mmaricevic | data/test-sequence-1.txt | data/pictures/1.png | single-hand-zoom | 27861             |
| 13 | mmaricevic | data/test-sequence-1.txt | data/pictures/2.png | single-hand-zoom | 29045             |

Slika 4.13. Korisnička CSV datoteka s rezultatima završenih zadataka

Nakon uspješnog izvršavanja zadataka i dobivenih rezultata ispitni korisnik je morao popuniti još tri preostala upitnika. Prvi upitnik bio je vezan uz tvrdnje o upotrebljivosti korištenih modaliteta. Odgovori su bili na Likertovoj skali od 1 (u potpunosti se ne slažem s tvrdnjom) do 5 (u potpunosti se slažem s tvrdnjom). Tvrdnje o upotrebljivosti korištenih modaliteta su bile sljedeće:

- Lakoća korištenja (engl. Efficiency) – „Jednom kada se modalitet nauči, jednostavno je koristiti ga.“
- Lakoća učenja (engl. Learnability) – „Lagano je naučiti princip rada modaliteta i primijeniti ga na postavljeni zadatak.“
- Zadovoljstvo (engl. Satisfaction) – „Zadovoljan sam korištenjem ovog modaliteta u kontekstu obavljanja postavljenih zadataka.“

Drugi upitnik naziva NASA TLX provodio se radi subjektivne procjene radnog opterećenja interakcije uz korištenje navedenih modaliteta. Odgovori su bili na skali između 1 i 21, a pitanja o opterećenju korisnika su bila iduća:

- Mentalni napor (engl. Mental demand) – „Koliko mentalnog napora (prisjećanje, razmišljanje) morate uložiti za rješavanje zadatka?“
- Fizički napor (engl. Physical demand) – „Koliko fizičkog napora (zamor ruku, zglobova, ...) morate uložiti za rješavanje zadatka?“
- Razina frustracije (engl. Frustration) – „Koliko razinu frustracije (stres, nesigurnost, uznemirenost) vam uzrokuje dotični modalitet?“
- Učinkovitost (engl. Performance) – „Prema vlastitom osjećaju, koliko ste brzi i uspješni pri izvođenju zadatka dotičnim modalitetom?“
- Ukupni napor (engl. Effort) – „Prema vlastitom osjećaju, koliki je ukupni napor (kumulativno, mentalni i fizički) potrebno uložiti za izvođenje zadatka zadanim modalitetom?“

Posljednji upitnik tražio je od ispitnih korisnika da odaberu modalitet kojega preferiraju, napišu očekivane primjere primjene ovakve vrste interakcije te daju kratak osvrt na cjelokupni eksperiment. Uspješnim popunjavanjem triju navedenih upitnika ispitni korisnik je uspješno završio eksperiment. Za provedbu cijelog eksperimenta bilo je predviđeno oko 35 minuta po osobi.

## 5. REZULTATI

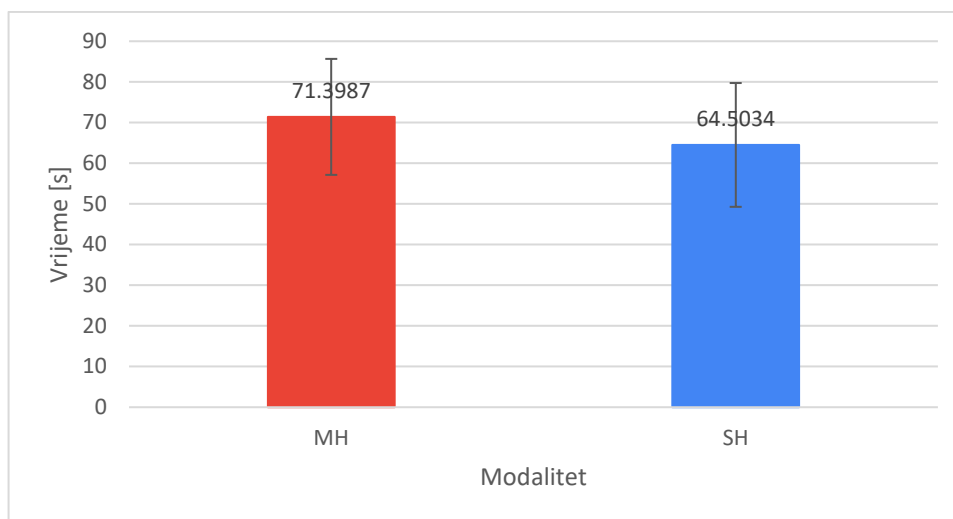
Uspješno provedenim eksperimentima dobivena su vremena izvođenja zadataka svih ispitnih korisnika te popunjeni upitnici vezani uz osobne podatke, upotrebljivost i opterećenje prilikom izvođenja zadataka određenim modalitetima te odabir preferiranog modaliteta. Dobiveni rezultati statistički su analizirani koristeći SPSS i Microsoft Excel alate. SPSS je moćan alat za statističku analizu razvijen od strane IBM-a koji omogućuje pregršt naprednih statističkih procedura i testova koji pomažu u osiguravanju visoke točnosti i kvalitete pri donošenju odluka [26]. SPSS alat koristio se za provedbu statističkih testova, a Microsoft Excel za deskriptivnu statistiku tj. za prikaz rezultata putem grafova.

Dobiveni podaci o vremenima za obavljanje zadataka iz CSV datoteke *concatenated-log* sortirani su prema modalitetu korištenja i složenosti radi lakše provedbe statističkih testova. Kako su svi ispitni korisnici prolazili kroz sve eksperimentalne situacije, odnosno svih 12 mogućih zadataka, ovaj eksperiment spadao je u HCI eksperimente ponavljanog mjerenja (engl. Repeated Measures). Iz tog razloga, za statističku analizu korišten je 2x3 dvosmjerni ANOVA test ponovljenih mjerenja (engl. Two-way Repeated Measures ANOVA test) [27]. Test je uključivao dvije nezavisne varijable: modalitet s dvije moguće vrijednosti (MH i SH) i složenost zadatka s tri moguće vrijednosti (jednostavan, umjereno složen i složen) te jednu zavisnu varijablu, a to je vrijeme potrebno za obavljanje zadatka. Prije provedbe RM ANOVA testa koristi se Mauchlyjev test sferičnosti (engl. Mauchly's sphericity test) kojim se utvrđuje jednakost varijanci između vrijednosti svake nezavisne varijable. Ako je sferičnost narušena ( $p < 0.05$ ), potrebno je na rezultate RM ANOVA testa primijeniti određene korekcije. RM ANOVA testom moguće je utvrditi postoje li statistički signifikantne razlike u vremenu izvršavanja zadataka s obzirom na vrijednosti dviju nezavisnih varijabli (modalitet, složenost), međutim nije moguće odrediti između kojih točno vrijednosti ta signifikantna razlika postoji. Za utvrđivanje signifikantnih razlika među vrijednostima nezavisne varijable koristi se post hoc analiza. Post hoc analiza uključuje korištenje jednog od statističkih testova za višestruku usporedbu vrijednosti nezavisne varijable. U ovome radu korištena je post hoc međusobna usporedba s Bonferronijevom korekcijom (engl. Pairwise comparisons using the Bonferroni correction).

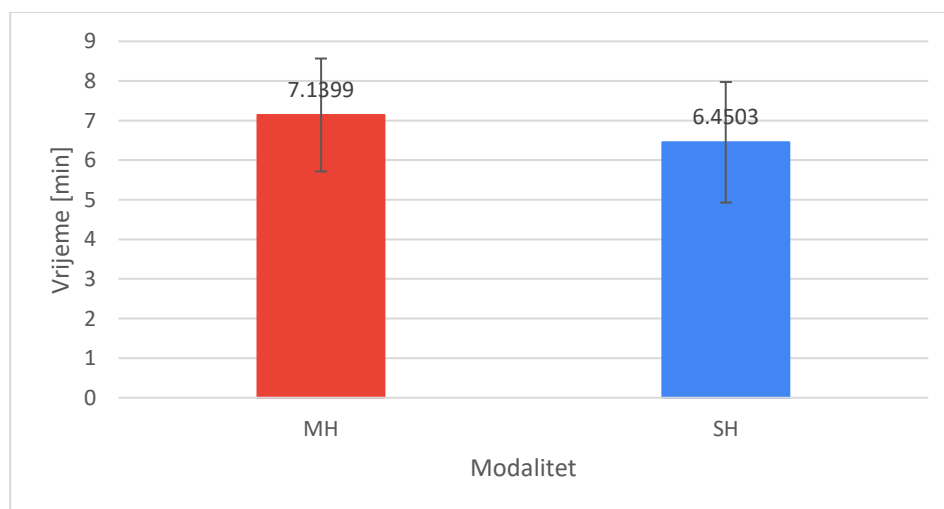
Za analizu podataka dobivenih upitnicima o upotrebljivosti korištenih modaliteta i radnog opterećenja interakcije koristio se Wilcoxon's Signed Rank test. Ovaj test primjer je neparametarskog testa jer ne podliježe pretpostavci o podatkovnoj distribuciji, odnosno koristi se za podatke koji potječu iz rednih, izmišljenih i kategorijskih skala baš kao što su podaci iz upitnika, a temelji se na usporedbi medijana dvaju uzoraka [28].

## 5.1 Modalitet

Ciklus zadataka uključivao je skaliranje pravokutnika na 6 različitih slika, što je ukupno 6 zadataka po modalitetu. Dobivena vremena izvršavanja zadataka sortirana su prema modalitetima te je izračunato prosječno vrijeme izvršavanja zadatka po modalitetu. Prosječno vrijeme izvršavanja zadatka (slika 5.1) korištenjem SH modaliteta manje je ( $64.50 \pm 15.22$  sekundi) nego korištenjem MH modaliteta ( $71.40 \pm 14.26$  sekundi). Također, cijeli ciklus zadataka bilo je prosječno brže odraditi koristeći SH ( $6.45 \pm 1.52$  minuta) modalitet nego MH modalitet ( $7.14 \pm 1.43$  minuta), što je moguće vidjeti i iz slike 5.2.



Slika 5.1. Prosječno vrijeme izvršavanja zadatka po modalitetima



Slika 5.2. Prosječno vrijeme izvršavanja kompletnog ciklusa zadataka po modalitetima

Mauchlyjev test nije pokazao narušavanje sferičnosti podataka ( $p > 0.05$ ) kod nezavisne varijable modaliteta pa nije bilo potrebno primijeniti nikakve korekcije na rezultate dobivene RM ANOVA testom. Sferičnost niti ne može biti narušena kada nezavisna varijabla sadrži samo dvije vrijednosti (u ovom slučaju MH i SH).

RM ANOVA test pokazao je statistički signifikantnu razliku u vremenu izvršavanja zadataka korištenjem modaliteta MH i SH:  $F(1,29) = 10.0136$ ,  $p < 0.05$ . Post hoc analizom s Bonferronijevom korekcijom (tablica 5.1) utvrđeno je kako je prosječno vrijeme za izvršavanje zadatka koristeći MH modalitet značajno veće u odnosu na SH modalitet ( $p = 0.003$ ).

Iz navedenog, može se zaključiti kako je SH modalitet učinkovitiji od MH modaliteta. Može se reći kako su ovakvi rezultati očekivani s obzirom na mogućnost paralelnog izvršavanja akcija skaliranja i pomicanja slikovnog sadržaja kod SH modaliteta.

*Tablica 5.1. Post hoc analiza s Bonferronijevom korekcijom - modalitet*

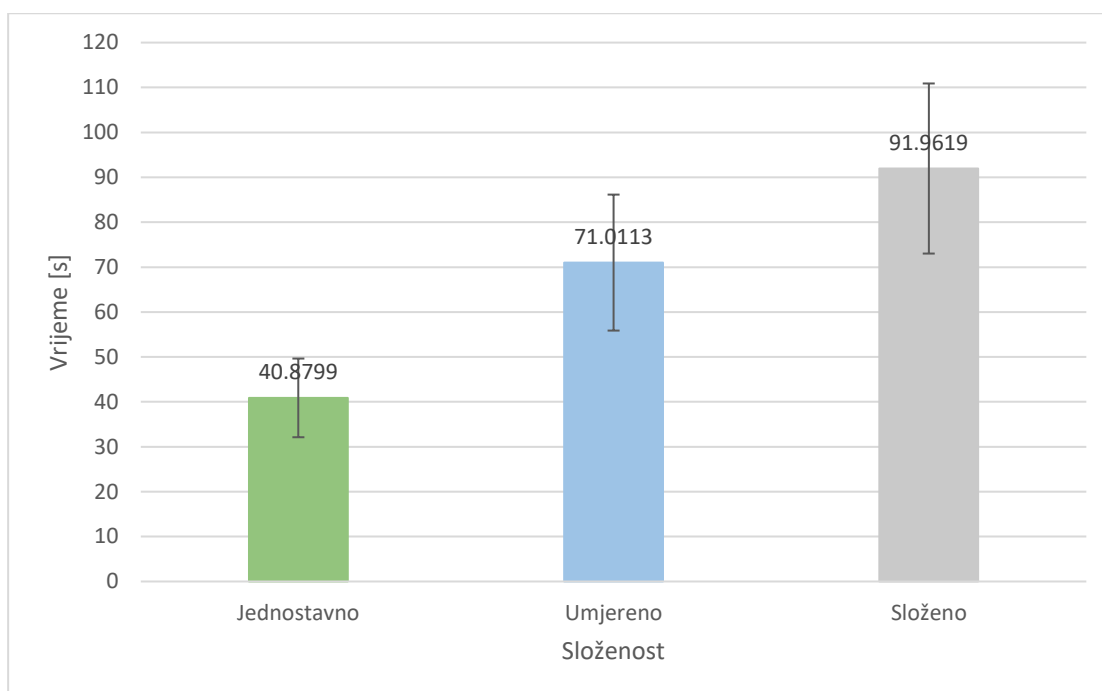
| Modalitet (A) | Modalitet (B) | Srednja razlika (A – B) | Standardna pogreška | p            |
|---------------|---------------|-------------------------|---------------------|--------------|
| MH            | SH            | 6.895                   | 2.166               | <b>0.003</b> |

## 5.2 Složenost

Vremena izvršavanja zadataka sortirana su i prema složenosti te su izračunata prosječna vremena po svakoj složenosti zadatka. Prosječna vremena pokazala su kako je izvršavanje jednostavnijih zadataka kraće od vremena izvršavanja složenijih zadataka (slika 5.3).

Mauchlyjev test nije pokazao narušavanje sferičnosti ( $p = 0.115$ ) kod nezavisne varijable složenosti zadatka stoga nije bilo potrebno provoditi nikakve korekcije na rezultate dobivene RM ANOVA testom.

RM ANOVA testom utvrđeno je da postoji statistički signifikantna razlika u vremenu izvršavanja zadataka različitih složenosti:  $F(2,58) = 336.713$ ,  $p < 0.001$ . Post hoc analiza s Bonferronijevom korekcijom (tablica 5.2) pokazala je kako postoji statistički signifikantna razlika u prosječnom vremenu za obavljanje zadataka između svih parova složenosti ( $p < 0.001$ ). Kao što je i bilo za pretpostaviti, za izvršavanje složenijih zadataka uvijek je bilo potrebno prosječno više vremena nego za izvršavanje jednostavnijih zadataka.



Slika 5.3. Prosječno vrijeme izvršavanja zadataka s obzirom na složenost

Tablica 5.2. Post hoc analiza s Bonferronijevom korekcijom - složenost

| Složenost (A) | Složenost (B) | Srednja razlika (A-B) | Standardna pogreška (engl. std. error) | p              |
|---------------|---------------|-----------------------|--|----------------|
| Jednostavno   | Umjereno      | -30.131               | 1.656                                  | < <b>0.001</b> |
| Jednostavno   | Složeno       | -51.082               | 2.299                                  | < <b>0.001</b> |
| Umjereno      | Složeno       | -20.951               | 1.929                                  | < <b>0.001</b> |

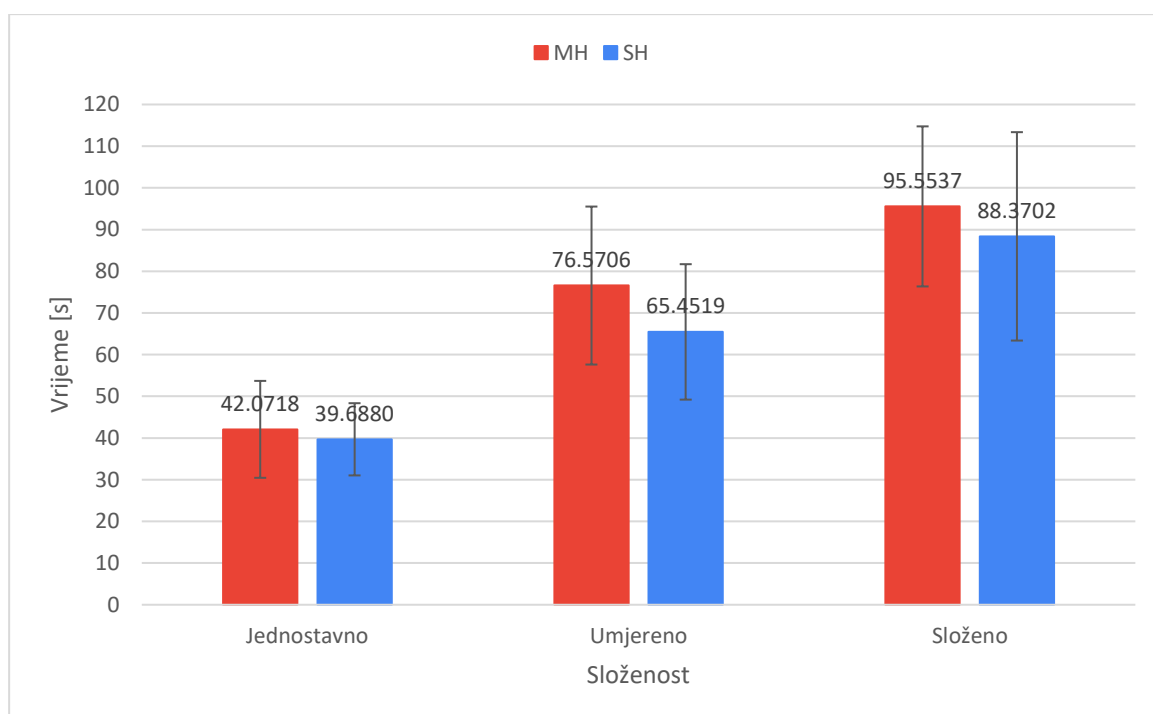
### 5.3 Modalitet i složenost

Analizom vremena za izvršavanje zadataka i po složenosti i po modalitetima dobiveni su kumulativni rezultati u tablici 5.3. Rezultati su i grafički prikazani slikom 5.4, gdje je moguće vidjeti kako je izvršavanje zadataka SH modalitetom bilo učinkovitije kroz sve tri složenosti. Zanimljivo je za primijetiti kako je najveća razlika u prosječnom vremenu izvršavanja zadataka različitim modalitetima ipak ostvarena kod umjereno složenih zadataka umjesto kod složenih zadataka kao što je bilo za pretpostaviti. Najmanja razlika ostvarena je u jednostavnim zadacima što je bilo i za očekivati s obzirom na mali broj velikih pravokutnika na slikama.



Tablica 5.3. Prosječno vrijeme izvršavanja zadataka prema modalitetima i složenosti

| Složenost                 | Modalitet   |          |         |             |          |         |
|---------------------------|-------------|----------|---------|-------------|----------|---------|
|                           | MH          |          |         | SH          |          |         |
|                           | Jednostavno | Umjereno | Složeno | Jednostavno | Umjereno | Složeno |
| Prosječno vrijeme [s]     | 42.0718     | 76.5706  | 95.5537 | 39.6880     | 65.4519  | 88.3702 |
| Standardna devijacija [s] | 11.6226     | 18.9479  | 19.1985 | 8.6615      | 16.2442  | 24.9945 |

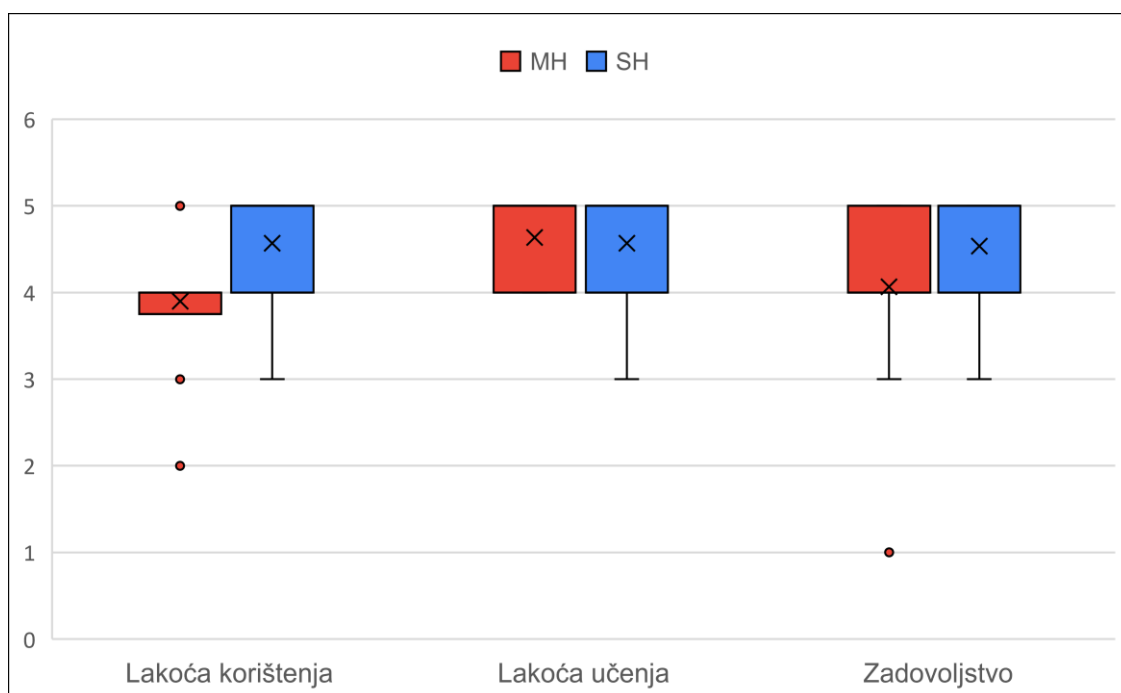


Slika 5.4. Prosječno vrijeme izvršavanja zadataka po modalitetima i složenosti

#### 5.4 Upotrebljivost, radno opterećenje i preferirani modalitet

Analiza podataka dobivenih upitnikom o upotrebljivosti korištenja dvaju modaliteta prikazana je kutijastim dijagramom (engl. Box and Whisker Plot) na slici 5.5. Iz dijagrama se može primijetiti kako su ispitni korisnici bili zadovoljniji korištenjem SH modaliteta te su smatrali kako ga je bilo jednostavnije koristiti u odnosu na MH modalitet. Također, korisnici su zaključili kako se oba modaliteta mogu vrlo brzo naučiti i primijeniti na postavljene eksperimentalne zadatke tj. nije im bilo teže naučiti koristiti jedan modalitet u odnosu na drugi.

Navedeni zaključci proizlaze iz dobivenih razlika u srednjim vrijednostima (označena s X) i medijanama ocjena svake od tvrdnji koji se mogu vidjeti na prikazanom dijagramu. Mišljenja korisnika po pitanju lakoće korištenja MH modaliteta bila su vrlo podijeljena odnosno varirala su između 2 i 5, međutim većina ih je ipak odlučila kako je MH modalitet bilo prilično jednostavno koristiti jednom kada bi ga se naučilo. Jedan korisnik bio je u potpunosti nezadovoljan korištenjem MH modaliteta.



Slika 5.5. Atributi upotrebljivosti korištenih modaliteta

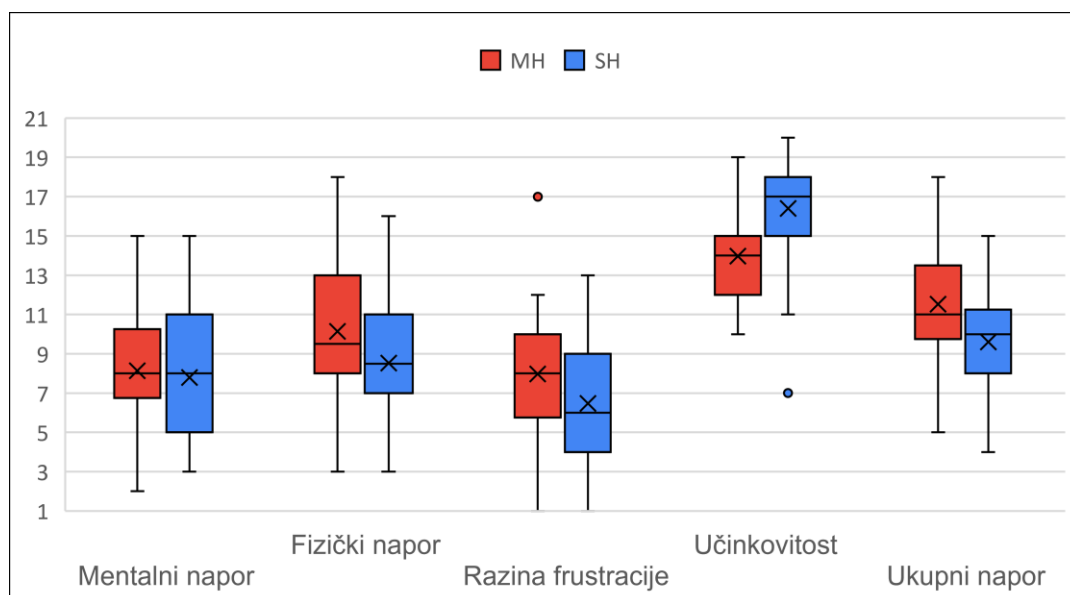
Provedbom Wilcoxon's Signed Rank testova nad podacima o upotrebljivosti utvrđeno je kako postoji statistički signifikantna razlika u lakoći korištenja ( $Z = -3.038$ ,  $p = 0.002$ ) i zadovoljstvu ( $Z = -2.399$ ,  $p = 0.016$ ) pri korištenju navedenih modaliteta. Zaključci izvedeni na temelju deskriptivne statistike bili su tada i formalno potvrđeni.

Dakle, ispitni korisnici zadovoljniji su korištenjem SH modaliteta te ga smatraju jednostavnijim za korištenje pri rješavanju zadataka u odnosu na MH modalitet. Za lakoću učenja ( $Z = -0.577$ ,  $p = 0.564$ ) signifikantna razlika nije pronađena pa nije moguće tvrditi kako je jedan modalitet bilo lakše naučiti koristiti u odnosu na drugi. Prosječne ocjene po svakoj tvrdnji i rezultate statističkih testova moguće je vidjeti u tablici 5.4.

Tablica 5.4. Prosječne ocjene i rezultati Wilcoxon's Signed Rank testova - upotrebljivost

|    | Lakoća korištenja | Lakoća učenja | Zadovoljstvo |
|----|-------------------|---------------|--------------|
| MH | 3.90 ± 0.71       | 4.63 ± 0.49   | 4.07 ± 0.91  |
| SH | 4.57 ± 0.63       | 4.57 ± 0.57   | 4.53 ± 0.63  |
| Z  | -3.038            | -0.577        | -2.399       |
| p  | <b>0.002</b>      | 0.564         | <b>0.016</b> |

Analizom podataka dobivenih upitnikom o radnom opterećenju interakcije (NASA TLX) dobiven je kutijasti dijagram na slici 5.6.



Slika 5.6. Radno opterećenje interakcije (NASA TLX)

Iz prikazanog dijagrama može se zaključiti kako je rješavanje zadataka MH modalitetom zahtijevalo veći fizički napor te uzrokovalo veću razinu frustracije što je bilo i očekivano s obzirom da se za akcije skaliranja i pomicanja slike koriste različite geste koje je potrebno uskladiti na pravilan način. Zanimljivo je za primijetiti kako su medijani ocjena mentalnog napora jednaki za oba modaliteta. Za izvršavanje zadataka bilo je potrebno uložiti ukupno veći napor koristeći MH modalitet. Ispitni korisnici smatrali su kako su bili učinkovitiji u izvođenju zadatka korištenjem SH modaliteta.

Provedenim Wilcoxon's Signed Rank testovima utvrđeno je kako postoji statistički signifikantna razlika u:

- fizičkom naporu ( $Z = -3.344$ ,  $p < 0.001$ )
- razini frustracije ( $Z = -2.373$ ,  $p = 0.018$ )
- učinkovitosti ( $Z = -3.661$ ,  $p < 0.001$ )
- ukupnom naporu ( $Z = -3.763$ ,  $p < 0.001$ )

Rješavanje eksperimentalnih zadataka bilo je učinkovitije koristeći SH modalitet, a MH modalitet zahtijevao je veći fizički i ukupni napor kao i veću razinu frustracije. Kod mentalnog napora ( $Z = -0.827$ ,  $p = 0.408$ ) potrebnog za izvođenje zadataka korištenjem dvaju modaliteta nije pronađena statistički signifikantna razlika. Prosječne ocjene po svakoj tvrdnji i rezultate statističkih testova moguće je vidjeti u tablici 5.5.

*Tablica 5.5. Prosječne ocjene i rezultati Wilcoxon's Signed Rank testova – radno opterećenje*

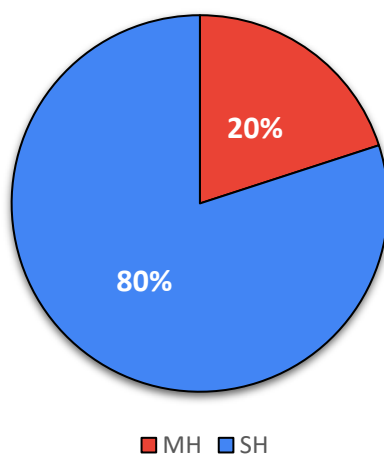
|    | Mentalni napor | Fizički napor     | Razina frustracije | Učinkovitost      | Ukupni napor      |
|----|----------------|-------------------|--------------------|-------------------|-------------------|
| MH | 8.13 ± 3.03    | 10.13 ± 3.78      | 7.97 ± 3.64        | 13.97 ± 2.54      | 11.53 ± 3.19      |
| SH | 7.80 ± 3.27    | 8.53 ± 3.13       | 6.47 ± 3.01        | 16.40 ± 2.84      | 9.60 ± 2.54       |
| Z  | -0.827         | -3.344            | -2.373             | -3.661            | -3.763            |
| p  | 0.408          | <b>&lt; 0.001</b> | <b>0.018</b>       | <b>&lt; 0.001</b> | <b>&lt; 0.001</b> |

U posljednjem upitniku, ispitni korisnici su trebali izabrati modalitet koji im više odgovara na temelju jednostavnosti i učinkovitosti korištenja, napora kojeg je bilo potrebno uložiti, osjećaja zadovoljstva njegovim korištenjem i sl. Prema dijagramu na slici 5.7, čak 24 (80 %) korisnika preferiraju SH modalitet.

Ispitnim korisnicima je provedeni eksperiment bio vrlo interesantan. Kako je za većinu ovakav oblik interakcije bio jedno novo iskustvo, poprilično su se dobro snalazili s gestovnom interakcijom te izvođenjem akcija koristeći implementirane modalitete. Posebno su bili impresionirani pozadinskom tehnologijom koja omogućuje prepoznavanje ruku i ključnih točaka dlanova. Upravljanje sadržajem pomoću gesti bilo im je vrlo zanimljivo i smatraju kako bi ovakav oblik interakcije trebao steći vrlo široku primjenu u budućnosti zbog svoje jednostavnosti.

Ispitni korisnici poprilično su zadovoljni implementiranim modalitetima i principima njihova korištenja. Smatraju kako je modalitete vrlo jednostavno za koristiti. Neki korisnici su na početku njihova korištenja bili frustrirani, no s vremenom i vježbom razina frustracije je opadala, a zadovoljstvo njihovim korištenjem raslo.

Navedene zaključke potvrđuje i jedan od komentara korisnika: „*Tematika je vrlo interesantna, u početku je malo umarajuće, ali nakon savladavanja principa korištenja postaje puno zanimljivije, kako kontrola tako i rješavanje problema odnosno zadatka*“. Naveli su kako korištenje ovakvoga sustava postaje vrlo zamorno nakon dugog vremena jer je potrebno konstantno držati ruke u zraku. Neki korisnici bili su nezadovoljni osjetljivošću akcije pomicanja i malim područjem koje pokriva kamera pa su sugerirali kako bi bilo bolje koristiti širokokutnu kameru. Također, primijetili su da u radu sustava nedostaje povratna informacija prilikom izlaska ruku izvan područja kamere koja bi svakako dodatno doprinijela uspješnijem korištenju sustava.



Slika 5.7. Preferirani modalitet (N = 30)

## 6. ZAKLJUČAK

U ovome radu osmišljen je i implementiran sustav Pan&Zoom koji omogućava upravljanje prikazom sadržaja na zaslonu pomoću gesti ruku, a sastoji se od dvaju modula koji međusobno komuniciraju putem utičnice i kamere. Sustav je namijenjen stolnim i prijenosnim računalima s Linux ili Windows operacijskim sustavom.

Modul zasnovan na MediaPipe radnom okviru koristi se za prepoznavanje ruku odnosno dlanova i njihovih ključnih točaka pomoću treniranih modela strojnog učenja. Ključne točke dlana omogućavaju detekciju izvođenja određene korisničke geste. Svaka gesta povezana je s odgovarajućom akcijom za upravljanje sadržajem na zaslonu. Informacije o određenoj akciji prosljeđuju se Image-viewer modulu putem utičnice.

Image-viewer modul sadrži grafičko sučelje za učitavanje slikovnog sadržaja te upravlja izvođenjem akcija nad učitanim slikom. Za upravljanje slikovnim sadržajem osmišljena su dva različita modaliteta implementirana u MediaPipe modulu: MH i SH. Modalitetima je moguće izvoditi akcije pomicanja i skaliranja slike. Kod SH modaliteta, lijeva ruka se koristi za pomicanje, a desna ruka za skaliranje slike. Kod MH modaliteta, za akciju pomicanja može se koristiti bilo koja ruka, dok se za akciju skaliranja moraju koristiti obje ruke. Za pokretanje Pan&Zoom sustava potrebno je instalirati određene alate.

Za usporedbu učinkovitosti korištenja dvaju modaliteta osmišljen je i proveden HCI eksperiment (N = 30). Učinkovitost korištenja podrazumijevala je rješavanje primjerenih zadataka koristeći određeni modalitet u što kraćem vremenu. Eksperimentalni ciklus zadataka uključivao je 6 različitih slika ravnomjerno raspoređenih u 3 kategorije prema složenosti odgovarajućeg zadatka. Ispitni korisnik trebao je ispravnim redosljedom uvećati pravokutnike na slikama u što kraćem vremenu. Pri tome, zadaci su zahtijevali i odgovarajuća pomicanja slikovnog sadržaja. Za uspješno provođenje eksperimenata bilo je potrebno osigurati uvjete za ispravan rad sustava (dovoljno osvijetljeno mjesto u prostoriji gdje se ne kreće veliki broj ljudi, pravilna pozicija osobe pred kamerom).

Nakon izvođenja zadataka korisnici su morali popuniti tri upitnika o upotrebljivosti korištenih modaliteta, radnom opterećenju interakcijom te izboru preferiranog modaliteta.

Eksperiment je sadržavao dvije nezavisne varijable: modalitet i složenost te jednu zavisnu varijablu: vrijeme izvršavanja zadatka. Nad podacima dobivenim eksperimentom provedena je statistička analiza.

Dvosmjernim ANOVA testom ponovljenih mjerenja i post hoc analizom s Bonferronijevom korekcijom utvrđeno je kako je izvršavanje zadatka bilo brže odnosno učinkovitije korištenjem SH modaliteta u odnosu na MH modalitet. To je bilo i za očekivati s obzirom na mogućnost paralelnog izvođenja akcija pomicanja i skaliranja slikovnog sadržaja kod SH modaliteta.

Na vrijeme izvršavanja zadataka utjecala je i njihova složenost, kao što je to bilo moguće i za pretpostaviti. RM ANOVA testom i post hoc analizom ustanovljeno je kako postoji statistički signifikantna razlika u vremenu izvršavanja zadataka između svih parova složenosti. Za izvršavanje složenijih zadataka uvijek je bilo potrebno prosječno više vremena nego za izvršavanje onih jednostavnijih.

Nad podacima dobivenim iz upitnika o upotrebljivosti korištenih modaliteta i radnog opterećenja interakcijom proveden je neparametarski Wilcoxon's Signed Rank test. Test je pokazao kako su korisnici bili zadovoljniji korištenjem SH modaliteta te su ga smatrali jednostavnijim u odnosu na MH modalitet. Signifikantna razlika u lakoći učenja dvaju modaliteta nije bila pronađena odnosno nije bilo moguće tvrditi da je jedan modalitet bilo lakše naučiti koristiti od drugoga.

Što se tiče radnog opterećenja interakcije, korisnici su bili mišljenja kako je korištenje MH modaliteta bilo fizički i sveukupno napornije od SH modaliteta te da je izazivalo veću razinu frustracije što je kasnije potvrdio i prethodno navedeni test. Statistički signifikantna razlika u mentalnom naporu kojeg je trebalo uložiti pri korištenju dvaju modaliteta nije bila pronađena. Ispitni korisnici su smatrali kako su bili učinkovitiji korištenjem SH modaliteta što je provedeni statistički test i potvrdio. U konačnici, ispitni korisnici više su preferirali SH modalitet (80%).

U svakom interaktivnom sustavu, pa tako i u ovome, postoji prostor za buduću nadogradnju. Bilo bi zanimljivo osmisliti i implementirati još jedan modalitet ili dodati nove zadatke te ponoviti ovakav tip HCI eksperimenta koristeći bolju širokokutnu kameru. Poželjno bi bilo dodati nekakav oblik povratne informacije prilikom izlaska ruku izvan područja kamere te povećati broj ispitnih korisnika. Grafičko sučelje Image-viewer modula moguće je promijeniti kako bi korisnicima bilo što lakše pratiti izvršavanje eksperimentalnih zadataka. Sve u svemu, bilo je vrlo interesantno osmisliti, implementirati i usporediti dva načina (modaliteta) gestovne interakcije kroz HCI eksperiment te se bolje upoznati s ovakvom vrstom interakcije čovjeka i računala. Ovakvi sustavi zasigurno su sustavi budućnosti koji će učiniti upravljanje prikazom sadržaja na zaslonu jednostavnijim i učinkovitijim.

## 7. LITERATURA

- [1] Saffer, D.: "Designing Gestural Interfaces", O'Reilly Media, Sjedinjene Američke Države, 2008.
- [2] Payne, S.: "Understanding Gesture for UI Design", s Interneta, <https://boldist.co/design/gesture-based-interfaces/>, 23. prosinca 2022.
- [3] Omer, C.: "How McDonald's is Changing the Global Restaurant Industry... Again", s Interneta, <https://www.linkedin.com/pulse/how-mcdonalds-changing-global-restaurant-industry-again-choudhary>, 23. prosinca 2022.
- [4] Cappuccino, J.: "The \$15 Clapper Device Is the Original Smart Home Tech We Can't Live Without", s Interneta, <https://spy.com/articles/gadgets/smart-home/clapper-review-clap-on-clap-off-238058/>, 26. prosinca 2022.
- [5] "Sensor Tap", s Interneta, <https://www.amazon.ae/Sensor-Automatic-Hands-Faucet-Bathroom/dp/B081RDHR3M>, 26. prosinca 2022.
- [6] Maher, C.: "The strange and surprisingly intense world of Wii Sports speedrunners", s Interneta, <https://www.theverge.com/2019/8/5/20750161/wii-sports-speedrunning-nintendo>, 27. prosinca 2022.
- [7] NeuRA: "Wii Sports restores movement and fitness to people after a stroke", s Interneta, <https://www.neura.edu.au/news/wii-sports-restores-movement-fitness-people-stroke/>, 27. prosinca 2022.
- [8] Program-Ace: "Virtual Reality in Healthcare: a New Solution for Rehabilitation?", s Interneta, <https://program-ace.com/blog/virtual-reality-in-healthcare-a-new-solution-for-rehabilitation/>, 28. prosinca 2022.
- [9] Dipert B. i dr.: "The Gesture Interface: A Compelling Competitive Advantage in the Technology Race", s Interneta, <https://www.edge-ai-vision.com/2013/04/the-gesture-interface-a-compelling-competitive-advantage-in-the-technology-race/>, 29. prosinca 2022.
- [10] Baker, B.: "The Need for Accesible, Gestural Interfaces", s Interneta, <https://www.bluecadet.com/news/the-need-for-accessible-gestural-interfaces/>, 30. prosinca 2022.



- [11] Raymond, S.: "Controlling Your Car's Interface with a Simple Hand Gesture", s Interneta, <https://www.guideautoweb.com/en/articles/41932/controlling-your-car-s-interface-with-a-simple-hand-gesture/>, 2. siječnja 2023.
- [12] "Group of Friends Playing Wii Music", s Interneta, [https://www.gameclimate.com/games/nintendo-wii/wii-music/i\\_16146/](https://www.gameclimate.com/games/nintendo-wii/wii-music/i_16146/), 2. siječnja 2023.
- [13] Arora, S.: "Gesture Recognition - A new technology to control your devices", s Interneta, <https://c.realm.com/in/post-details/1085801690109575168>, 3. siječnja 2023.
- [14] "MediaPipe", s Interneta, <https://google.github.io/mediapipe/>, 4. siječnja 2023.
- [15] "Računalni vid", s Interneta, [https://hr.wikipedia.org/wiki/Ra%C4%8Dunalni\\_vid](https://hr.wikipedia.org/wiki/Ra%C4%8Dunalni_vid), 4. siječnja 2023.
- [16] Kibalchich, I.: "Drone Control via gestures using MediaPipe Hands", s Interneta, <https://developers.googleblog.com/2021/09/drone-control-via-gestures-using-mediapipe-hands.html>, 4. siječnja 2023.
- [17] "MediaPipe Hands", s Interneta, <https://google.github.io/mediapipe/solutions/hands>, 6. siječnja 2023.
- [18] "Socket tutorial", Oracle Corporation, s Interneta, <https://docs.oracle.com/cd/E19504-01/802-5886/6i9k5sgsl/index.html>, 6. siječnja 2023.
- [19] "Unix - Linux manual page", s Interneta, <https://man7.org/linux/man-pages/man7/unix.7.html>, 6. siječnja 2023.
- [20] Cooley, S.: "AF\_UNIX comes to Windows", s Interneta, [https://devblogs.microsoft.com/commandline/af\\_unix-comes-to-windows/](https://devblogs.microsoft.com/commandline/af_unix-comes-to-windows/), 6. siječnja 2023.
- [21] Kowalczyk, K.: "Guide to predefined macros in C++ compilers", s Interneta, <https://blog.kowalczyk.info/article/j/guide-to-predefined-macros-in-c-compilers-gcc-clang-msvc-etc..html>, 7. siječnja 2023.
- [22] "MediaPipe in C++", s Interneta, [https://google.github.io/mediapipe/getting\\_started/cpp.html#option-2-running-on-gpu](https://google.github.io/mediapipe/getting_started/cpp.html#option-2-running-on-gpu), 7. siječnja 2023.

- [23] "Introduction to HCI", University of Birmingham, s Interneta, <https://www.cs.bham.ac.uk/~rxb/Teaching/HCI%20II/intro.html>, 8. siječnja 2023.
- [24] "What is Human-Computer Interaction (HCI)?", s Interneta, <https://www.getsmarter.com/blog/career-advice/what-is-human-computer-interaction-hci/>, 8. siječnja 2023.
- [25] I. Scott MacKenzie: "Within-subjects vs. Between-subjects Design: Which to Use?", s Interneta, <http://www.yorku.ca/mack/RN-Counterbalancing.html>, 8. siječnja 2023.
- [26] "IBM SPSS Statistics", IBM, s Interneta, <https://www.ibm.com/products/spss-statistics>, 10. siječnja 2023.
- [27] "Two-way repeated measures ANOVA using SPSS Statistics", s Interneta, <https://statistics.laerd.com/spss-tutorials/two-way-repeated-measures-anova-using-spss-statistics.php>, 10. siječnja 2023.
- [28] "Wilcoxon's Signed Rank Test: Definition, How to Run, SPSS", s Interneta, <https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/wilcoxon-signed-rank-test/>, 11. siječnja 2023.

## 8. SAŽETAK

U ovome radu razvijen je sustav Pan&Zoom koji omogućava upravljanje prikazom sadržaja na zaslonu računala pomoću gesti ruku. Sustav se zasniva na dva modula: radni okvir MediaPipe koristi se za detekciju ključnih točaka dlanova i prepoznavanje gesti ruku, a Image-viewer upravlja prikazom te izvođenjem akcija pomicanja i skaliranja grafičkog sadržaja. Za upravljanje sadržajem osmišljena su i implementirana dva modaliteta interakcije: MH (Multi-hand zoom) i SH (Single-hand zoom). Modalitet MH omogućava pomicanje sadržaja bilo kojom rukom, dok je za skaliranje potrebno koristiti obje ruke istovremeno. Nasuprot tome, kod SH modaliteta se za pomicanje koristi isključivo lijeva ruka, a za skaliranje desna ruka. Za usporedbu učinkovitosti i radnog opterećenja navedenih modaliteta proveden je HCI eksperiment s 30 ispitnih korisnika koji je uključivao primjerene zadatke iz tri različite kategorije složenosti. Statističkom analizom dobivenih rezultata utvrđeno je kako se akcije pomicanja i skaliranja sadržaja izvršavaju značajno učinkovitije korištenjem modaliteta SH. Također je pokazano kako i složenost zadatka značajno utječe na vrijeme njegova izvršavanja. Navedeni rezultati dodatno su potkrijepljeni analizom standardiziranih upitnika kojom je utvrđeno kako modalitet MH rezultira značajno većim radnim opterećenjem. Ispitni korisnici preferiraju modalitet SH jer zahtijeva niži fizički napor, izaziva manju razinu frustracije i pruža bolju učinkovitost.

### **Ključne riječi:**

Pan&Zoom, interakcija gestama, prepoznavanje ruku, HCI eksperiment

## **SUMMARY**

In this thesis, the Pan&Zoom system was developed, which enables controlling the display of content on the computer screen using hand gestures. The system is based on two modules: the MediaPipe framework is used to detect the key points of the palms and recognize hand gestures, and the Image-viewer manages the display of graphical content and the execution of pan and zoom actions. Two interaction modalities were designed and implemented for graphical content management: MH (Multi-hand zoom) and SH (Single-hand zoom). The MH modality allows panning with either hand, while both hands have to be used simultaneously for zooming. In contrast, in the SH modality, only the left hand is used for panning and the right hand for zooming. In order to compare the efficiency and interaction workload of the mentioned modalities, an HCI experiment was conducted with 30 test users that included corresponding tasks from three different complexity categories. Statistical analysis of the obtained results showed that the SH modality performs panning and zooming actions significantly more efficiently. It was also shown that the task's complexity significantly impacts the time of its execution. The above results were additionally supported by the analysis of standardized questionnaires, which showed that the MH modality leads to a significantly higher workload. The test users prefer the SH modality because it requires less physical effort, causes a lower level of frustration, and provides better efficiency.

### **Keywords:**

Pan&Zoom, gesture interaction, hand recognition, HCI experiment

## 9. POPIS OZNAKA I KRATICA

|          |  |
|----------|--|
| TUI      | Touch User Interface                   |
| FFGUI    | Free-form Gestural User Interface      |
| MH       | Multi hand zoom                        |
| SH       | Single hand zoom                       |
| HD       | High-definition                        |
| GUI      | Graphical User Interface               |
| UNIX     | UNiplexed Information Computing System |
| IPC      | Inter Process Communication            |
| GNU      | GNU's Not Unix                         |
| MSYS     | Minimal SYStem                         |
| GCC      | GNU Compiler Collection                |
| MSVC     | Microsoft Visual C++                   |
| HCI      | Human-Computer Interaction             |
| MS       | Microsoft                              |
| NASA TLX | NASA Task Load Index                   |