

Pilotna rešitev za prikazovanje komplementarnih vsebin s pomočjo nadgrajene resničnosti

Matej Kopljan, Klemen Pečnik, Matevž Pogačnik

Univerza v Ljubljani, Fakulteta za elektrotehniko
E-pošta: matkopljan@gmail.com

A pilot augmented reality solution for displaying complementary content

Abstract. This paper presents an implementation of a pilot augmented reality solution which allows users to view complementary content as holograms using a Microsoft HoloLens headset. The main advantage of the solution is being able to display complementary content without modifying the original, thus preserving the experience for the users without the headset.

In this way we have also solved the ergonomic problems that come hand-in-hand with second-screen solutions such as the gorilla arm syndrome.

There are many potential use cases for this type of a solution, such as complementary information about/tourist attractions, museums, guided tours, education, cinema, and television channels

Although the application of this solution was developed for television channels, it can be applied to any of the listed options.

As a pilot project we have made an additional video stream for the deaf and hard of hearing. If applied by internet service providers, this solution would remove the need for duplicate video channels for the deaf, since the only difference is the sign language interpreter, effectively halving the bandwidth needed for each such channel.

The solution consists of a web application, containing a video player for the main content, a Microsoft HoloLens application, displaying complementary content and a NodeJS server, synchronising the two.

1 Uvod

1.1 Cilj projekta

Cilj tega projekta je bil izdelava aplikacije za prikaz dodatnih vsebin, ki nima slabosti trenutnih rešitev, in sicer s pomočjo nove tehnologije – z Microsoft HoloLens očali za mešano resničnost, ki prikazujejo holograme, s katerimi obogatimo prostor, v katerem se nahajamo.

Želeli smo rešiti več težav, ki se pojavljajo pri uporabi trenutnih rešitev za dodatno vsebino. Tovrstni pristopi k prikazovanju dodatnih vsebin so uporabni za številne scenarije, kot so uporaba v muzejih, galerijah in v turizmu, prikazovanje dodatnih vsebin v športu ali za izobraževalne namene, ne glede na področje uporabe pa tudi za nadomeščanje zvočnih informacij za gluhe in naglušne uporabnike.

Prva težava teh rešitev je 'ločenost' komplementarnih vsebin od glavne vsebine zaradi fizične razdalje med njimi.

Druga težava so ergonomske težave, ki pridejo z roko v roki skupaj z rešitvami tipa "second-screen", saj je velikokrat treba napravo držati v roki dalj časa neprekinjeno, kar predstavlja ergonomski problem. Pod to točko spadajo tudi vse ostale ergonomske težave, do katerih pride pri uporabi fizičnih rešitev.

Tretja s tem povezana težava je žrtvovanje glavne izkušnje za namene prikaza dodatnih informacij.

Vse znane rešitve imajo vsaj eno izmed naštetih težav, naš cilj pa je bil razvoj aplikacije, s katero bi se jim ognili.

1.2 Obstoječe rešitve

Nekaj rešitev z dodatnimi vsebinami že obstaja, a menimo, da ima vsaka vsaj kakšen problem, ki ga ne moremo prezreti, če hočemo uporabniku ponuditi kvalitetno izkušnjo. Te težave so večinoma ergonomske narave ali pa žrtvujejo kvaliteto izkušnje za uporabnike, ki dodatne vsebine ne potrebujejo.

Identificirali smo tri glavne obstoječe rešitve:

Prva rešitev so kanali [1], kjer je video v načinu slika v sliki (ang. picture in picture video). To je način, ki se trenutno uporablja v televizijskih programih. Ta rešitev prekrije del zaslona z osnovno vsebino, hkrati pa ga še zmanjša. Tako lahko gledalec izgubi pomembne informacije, ki bi jih na navadnem kanalu videl. Tako ta rešitev poslabša izkušnjo za vse gledalce, tudi za tiste, ki dodatno vsebino potrebujejo. Tovrstna rešitev velikokrat zahteva prenos dveh televizijskih kanalov, enega za navadne gledalce in enega za slušno prizadete. S tem operaterjem učinkovito podvoji zahtevano pasovno širino za prenos televizijskega kanala.



Slika 1. Rešitev slika v sliki. [1]

Druga rešitev je rešitev tipa drugi zaslon (ang. second-screen) [2]. Takšna rešitev večinoma uporablja telefonske in tablične aplikacije na drugi fizični napravi. Le-te prikazujejo dodatne informacije, kot je statistika o

trenutni igri športa, dodatne informacije o artiklu, ki ga reklama predstavlja ipd.

Največja težava tega tipa rešitve je razmik med televizijo in dodatno napravo, kar oteži hkraten ogled obeh vsebin. To lahko rešimo s tem, da napravo držimo pred seboj, vendar je to za uporabnika precej utrujajoč položaj, ki ne more biti dolgo vzdrževan in lahko povzroči sindrom goriline roke (ang. gorilla arm syndrome [8]).



Slika 2. Rešitev tipa drugi zaslon podjetja 2Immerse. [3]

Tretja rešitev [3], ki jo je naredil laboratorij IRT, pa temelji na Microsoft HoloLens očalih. Pojavila se je šele v končnih fazah izdelave naše rešitve in smo jo vzeli kot potrditvev, da gremo v pravo smer.

Ta rešitev prikazuje hologram kretalca znakovnega jezika poleg televizijskega zaslona.

Osebi, ki še ni uporabljala HoloLens očal, lahko izgleda ta rešitev ustrezna, čeprav ima skrit problem. To je omejen kot vida. Takšna rešitev zahteva, da mora uporabnik gledati televizijo od daleč ali pa mora imeti glavo usmerjeno v hologram kretalca znakovnega jezika. Če uporabnik tega ne naredi, bo hologram ob robu odrezan ali pa sploh ne bo viden. Zaradi implementacije rešitve je okno kretalca tudi nemogoče zožiti.

Iz teh razlogov je ta rešitev precej neudobna za dolgotrajno uporabo, hkrati pa se je težko osredotočiti na obe vsebini hkrati, saj je med njima veliko prostora.



Slika 3. Rešitev laboratorija IRT. [3]

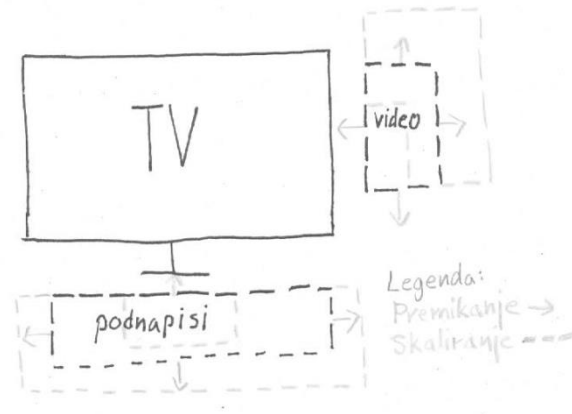
1.3 Zasnova naše rešitve

V projektu smo se osredotočili na nudenje dodatne vsebine za video gluhih in naglušnih.

Ugotovili smo, da potrebujemo dva holograma. Enega za video kretalca, drugega pa za podnapise.

Vsakemu smo določili kontrole za povečavo, premikanje in izbris.

Odločili smo se, da bomo gledalcu omogočili prilagoditev velikosti in lokacije teh dveh hologramov, saj niso vse televizije enako velike. Poleg tega smo tudi omogočili, da uporabnik posamezen hologram izklopi, če ga ne potrebuje.



Slika 4. Prva zasnova naše rešitve.

Slika 4 predstavlja naš začetni koncept izgleda aplikacije. Na sredini je televizor z glavno vsebino, zraven pa holograma z dodatno vsebino. Razporeditev je podobna sedanjim izkušnjam, ki jih imajo uporabniki – podnapisi na dnu slike so sedaj spodaj, dodatna vsebina, ki je bila prej v desnem predelu slike, pa je sedaj na desni strani zaslona.

Med zasnovno smo pazili na omejeno območje prikaza hologramov in na lahkost interakcije, saj lahko natančno klikanje majhnih elementov zelo hitro postane utrujajoče. Pri kliku je namreč potrebno element centrirati z obratom glave, ki pa se vedno nekoliko premika.

2 Izdelava rešitve

Izdelava je potekala v več fazah. Pri vsaki smo se soočili z različnimi izzivi.

2.1 Uporabljene tehnologije

Pri razvoju smo uporabili naslednjo opremo in tehnologije:

- Unity3D razvojno orodje za izdelavo video igrer, ki je hkrati uradno razvojno orodje za Microsoft HoloLens aplikacije.
- Microsoft HoloLens SDK (ang. software development kit), ki v kombinaciji z Unity3D omogoči izgradnjo (ang. build) aplikacije in prenos le-te na očala.
- Visual Studio 2017 in Windows 10, saj sta zahtevana za razvoj HoloLens aplikacij [6].
- NodeJS strežnik s procesnim upravljalnikom PM2.
- Spletne tehnologije za implementacijo video predvajalnika (HTML, CSS, JavaScript).
- HoloLens emulator in fizično napravo.

2.2 Interakcija z napravo in omejitve naprave

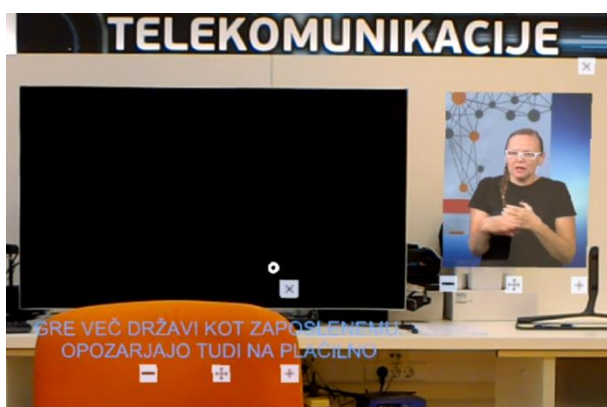
Med razvojem smo se spopadali z dvema glavnima omejitvama.

Prva omejitev je bila novost Microsoft HoloLens. To je pomenilo, da še ni bilo na razpolago veliko programskih knjižnic, ki bi pomagale pri razvoju takšnih aplikacij, tiste, ki pa so že bile, pa so bile še vedno v povojih. Poleg tega je bilo pri razvoju treba upoštevati, da HoloLens SDK ne podpira vseh funkcionalnosti razvojnega orodja Unity3D. Posledično smo morali veliko stvari implementirati sami.

Druga težava je bila omejen vidni kot, ki ga očala omogočajo. To pomeni, da elementi niso smeli biti preveliki, saj ne želimo prekriti glavne vsebine.

2.3 Implementacija uporabniškega vmesnika in HoloLens različica

Najprej smo naredili osnovo uporabniškega vmesnika v Unity3D brez HoloLens SDK in implementirali interakcijo, potem pa smo jo prenesli na očala.



Slika 5. Hologram podnapisov in video predvajalnika.

Med prenosom aplikacije na očala HoloLens smo opazili, da navadne 'onClick' (koda, ki se izvede ob kliku elementa) funkcije, ki jih Unity3D ponuja, ne delujejo na njej, čeprav se je v Unity3D emulatorju vse obnašalo po pričakovanjih.

Za rešitev te težave smo implementirali "poslušalca" (ang. "listener", programerski izraz za element v kodi, ki čaka na določen dogodek in ko pride do njega, aktivira drugo programsko kodo). S tem smo dosegli, da so se gumbi obnašali po pričakovanjih tudi na napravi. V tej fazi smo še vedno reševali problem ozkega vidnega kota, ki ga naprava omogoča.

Holograme smo postavili fiksno v prostor z možnostjo, da jih uporabnik po želji prilagodi. Pri takšni implementaciji se efektiven prostor za holograme še dodatno zmanjša, saj je treba preprečiti, da se hologrami odrežejo, ko uporabnik nekoliko premakne glavo. To pomeni, da morajo biti hologrami postavljeni z nekaj rezervnega prostora, saj uporabniki vedno vsaj malo premikajo glavo. Primerna razdalja od roba vidnega polja je odvisna od uporabnika in jo je možno tudi prilagoditi s povečavo ali premikanjem elementov.

2.4 Predvajanje video posnetka in podnapisov ter sinhronizacija

V tej fazi smo implementirali prenos podnapisov in videa.

Prenos podnapisov smo naredili s preprostim HTTP zahtevkom. Prikazujemo jih z uporabo interpreterja, ki

smo ga implementirali sami. Zaradi razširjenosti smo uporabili format WebVTT [7].

Vsebinsko polja za podnapise nadzira sinhronizacijska koda, ki prejema informacije od kontrolnega strežnika.

Video zaradi performančnih razlogov, ki izvirajo iz prenosa videa, ne sme biti prevelik. Ker Unity3D v kombinaciji z Microsoft HoloLens nativno ne podpira pretočnega videa (ang. video streaming), ga je potrebno prenesti vnaprej. Za tovrsten prenos potrebujemo nekaj sekund, preden je video pripravljen, kar je še v okviru sprejemljivih mej. Upamo, da bo Unity kmalu v svoji programski knjižnici dodal podporo za pretočni video.

2.5 Glavni predvajalnik, strežnik in sinhronizacija

V končni fazi smo implementirali še sinhronizacijo med video predvajalnikom in očali.

V spletni aplikaciji smo uporabili HTML5 video predvajalnik. Napisali smo preprosto JavaScript kodo, s katero je možno ugotoviti status videa in predvajani delež ter posredovati te informacije strežniku.

S to kodo informacije pošljemo strežniku do desetkrat na sekundo, a le v primeru sprememb (preskok v videu ali njegova zaustavitev). Strežnik si zapomni zadnje informacije o statusu in jih ob zahtevi HoloLens aplikacije posreduje naprej.

HoloLens aplikacija s pomočjo sinhronizacijske kode ugotovi delež preteklega videa in status videa (to je pavza ali predvajanje). To naredi tako, da upošteva čas, ko je bila informacija poslana s spletne aplikacije, delež pregledanega posnetka ob času pošiljanja in status predvajanja. Da dosežemo želeno natančnost, je pomembno, da je podan čas naveden v milisekundah.

Čas, ko je bila informacija poslana, se primerja s časom HoloLens očal in s časom videa. S pomočjo tega lahko ugotovimo, kakšen mora biti čas videa na naši strani, in sicer s preprosto formulo (čas HoloLens – poslan čas + čas videa = trenutni čas videa).

Izračunan čas se uporabi za sinhronizacijo podnapisov in videa.

S tem pristopom trajanje zakasnitve prenosa informacij iz predvajalnika do strežnika in naprej do HoloLens naprave nima vpliva na natančnost sinhronizacije. Zanašamo se le na natančnost ur v predvajalniku glavne vsebine (TV-program) in HoloLens napravi.

Ugotovili smo, da je dovolj, če se sinhronizacija izvede le nekajkrat na sekundo.

2.6 Stopnja sinhronizacije

Pri razvoju smo testirali več različnih stopenj sinhronizacije. Pri visokih stopnjah (60 Hz ali višje) smo opazili performančne probleme, pri nizkih (1 Hz ali manjše), pa je bilo skakanje pri zaustavljanju videa preveč opazno. Na osnovi poskusov smo izbrali 10 Hz, kot dovolj dobro stopnjo brez performančnih problemov.

Ker naša sinhronizacija temelji na urah naprav, je možno, da avtomatski popravki ure, ki se na današnjih napravah dogajajo precej pogosto, povzročijo napake. Za rešitev tega problema smo predpostavili, da je popravek ure skoraj vedno majhen (manjši od 100 ms) in razlike v času, ki so manjše od tega, preprosto ignorirali. To je

bistveno izboljšalo gladkost dodatnega videa in podnapisov.

Na specifični rešitvi, ki smo jo razvili, smo ugotovili, da je potrebno upoštevati gibanje kretalca. Za razliko od podnapisov, ki jih lahko beremo tudi med pavzo, ali pa govora, kjer lahko tudi z delom besede ugotovimo, kaj je bilo izrečeno, se pri kretanju ob pavziranju izgubi veliko več pomena. To smo rešili tako, da Hololens, kljub temu da že prejme informacijo o pavzi, dodatno video vsebino nadaljuje še 1.5 sekunde, kar je dovolj, da kretalec konča začeto gesto. Pri nadaljnjem predvajanju se ta del predvaja še enkrat.

2.7 Območje vida

Kot projekciranja hologramov na očala Hololens je zelo omejen in znaša 45 stopinj [5].

Približek vidnega polja po širini si lahko najlažje predstavljamo, če iztegnemo roki predse, razširimo prste in se z enim palcem dotaknemo drugega. Če upoštevamo, da hoče uporabnik imeti v centru vida še kaj drugega, se zavemo, da pravzaprav ni veliko prostora in se moramo omejiti na manjše število elementov, ki jih porazdelimo po vidnem polju tako, da ne zavzamejo preveč prostora.

3 Možne izboljšave

V sklopu tega projekta smo naleteli na problem prenosa videa, saj Unity3D ne podpira prenosa videa v živo. To bi se dalo potencialno rešiti na več načinov in bi bilo tudi potrebno v primeru končne rešitve, saj bi lahko čakanje na prenos dolgega videa trajalo predolgo za dobro uporabniško izkušnjo.

Identificirali smo tri možne rešitve, ki bi najbolj in najbolj izvedljivo rešile omenjeno težavo. Vsaka ima svoje prednosti in slabosti.

Prva rešitev je prireditev videa, kot da bi prihajal iz spletne kamere. Jasna slabost je to, da tovrstne kamere niso temu namenjene in bi implementacija zahtevala veliko prilagoditev, saj bi bilo prejete informacije med tekom programa potrebno pretvoriti v teksture, ki bi jih potem 'prilepili' na element v Unity3D, ki predstavlja video predvajalnik, kar pa je zelo zahtevno.

Druga rešitev je uporaba Microsoft Azure pretočnega videa, edinega načina, ki ga HoloLens nativno podpira. Ta je od vseh najbolj preprosta, poleg tega pa se je v zadnjem času tudi bistveno pocenila. V času pisanja tega članka je mesečna cena storitve, ki premore pasovno širino 600 Mb/s, kar zadostuje za do 200 uporabnikov hkrati, 54 € [4], kar predstavlja za veliko ustanov še sprejemljive stroške.

V primeru, da plačevanje za storitev ni možno, pa obstaja tudi tretja možnost, ki je implementacija lastnega protokola za pseudo-pretočni video.

Druge izboljšave zahtevajo izboljšavo vidnega polja naprave.

4 Sklep

Menimo, da je projekt uspešen, saj smo uspeli narediti rešitev, ki:

- je vsebinsko primerljiva obstoječimi rešitvami, saj podobno kot rešitev "slika v sliki" uporabi podnapise in kretalca znakovnega jezika. S tem

zajame vse prednosti omenjene rešitve in lahko postane njena nadgradnja,

- odpravi probleme dvojnega prenosa vsebin in s tem povezanega sočasnega oddajanja dveh programov. Oddaja se le še originalen program, dodatno vsebino pa se lahko prenese tudi drugače,
- ne zniža kvalitete izkušnje gledalcu brez očal, saj osnovna vsebina ostane enaka,
- omogoča, da gledalec zlahka sledi hkrati glavni in dodatni vsebini v udobnem položaju, ker se ne zanaša na naprave, ki jih je treba držati v rokah,
- dobro izkoristi območje za izris hologramov in je uporabna za številne scenarije, ne glede na domeno.

Rešitev bi bilo v prihodnosti, ko se bo vidni kot izboljšal, možno uporabiti tudi v kinu, kjer do sedaj ni bilo cenovno smiselno narediti posebnega predvajanja filma za gluhe in naglušne uporabnike. Z našo rešitvijo, pa to ni več potrebno.

5 Zahvale

Zahvaljujemo se podjetju SportRadar, ki je laboratoriju za multimedijo (LMMFE) na Fakulteti za elektrotehniko Univerze v Ljubljani posodilo očala za razvoj aplikacij, in laboratoriju, ki je omogočil razvoj te rešitve.

Poleg tega se zahvaljujemo vsem, ki so pomagali pri razvoju te rešitve z nasveti in mnenji.

Literatura

- [1] Spletna TV, S kampanjo Zvoke v roke o pomembnosti znakovne govornice, <https://www.youtube.com/watch?v=FsSuWsxmzSU>
- [2] Rešitev 2Immerse, <https://2immerse.eu/>
- [3] Rešitev laboratorija IRT, <https://lab.irt.de/synchronised-holograms/>
- [4] Cenik Microsoft Azure storitev za pretočni video, <https://azure.microsoft.com/en-us/pricing/details/media-services/>
- [5] Specifikacije Microsoft HoloLens očal, <https://docs.microsoft.com/en-us/windows/mixed-reality/Hololens-hardware-details>
- [6] Zahteve za razvoj Microsoft HoloLens aplikacij <https://docs.microsoft.com/en-us/windows/mixed-reality/install-the-tools>
- [7] WebVTT format podnapisov, https://developer.mozilla.org/en-US/docs/Web/API/WebVTT_API
- [8] Gorilla arm syndrome, <https://www.deccanchronicle.com/lifestyle/health-and-wellbeing/230617/beware-of-gorilla-arm-syndrome.html>