SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 2540

NADOPUNJENA STVARNOST OBJEKTIMA RAČUNALNE GRAFIKE

Marko Vrljičak

Zagreb, svibanj 2012.

Sadržaj

1. Uvod

Razvojem računalne grafike kroz prošlih trideset godina uvelike su se proširile mogućnosti prikaza objekata na računalima. Današnji programi svakodnevno pomiču granice fotorealizma, konstantno se istražuju nove stvari, otvaraju nove opcije i postavljaju nove granice na ovom nepresušnom izvoru novih ideja.

Jedna od ovih novih tehnologija je nadopunjena stvarnost (eng. augmented reality). Od virtualne stvarnosti, koja stvara u potpunosti računalno generirane okoline do stvarnog svijeta, nadopunjena stvarnost ipak je bliže stvarnom svijetu. Ona nadopunjava stvarni svijet grafikom, zvukovima, čak mirisima i opipom, te nam daje povratnu vezu o kombinaciji stvarnog i virtualnog svijeta koji vidimo. Nadopunjena stvarnost se najviše razvija uz računalne igre ili aplikacije za mobilne telefone. Razvojem tehnologije pokušava se omogućiti korištenje nadopunjene stvarnosti svakome, a taj cilj će u skoroj budućnosti možda i biti ostvaren, jer mogućnosti su neiscrpne – od vojnih i medicinskih potreba do područja računalnih igara.

U ovom radu biti će prikazane mogućnosti nadopunjene stvarnosti, neke mogućnosti implementacije na računalima, trenutna ograničenja nadopunjene stvarnosti te ukratko primjena i budućnosti ove tehnologije. Sve će biti nadopunjeno jednim jednostavnim praktičnim primjerom i njegovim opisom što bi trebalo demonstrirati korištenu tehnologiju.

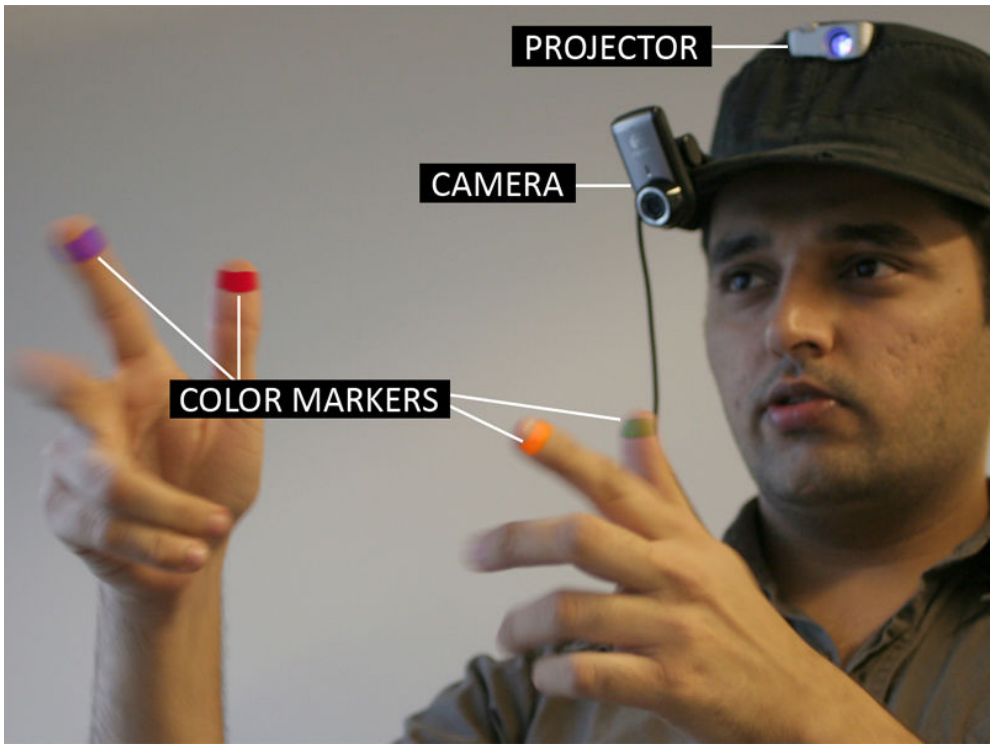
2. Nadopunjena stvarnost

Osnovna ideja nadopunjene stvarnosti je prilično jednostavna – koristiti računalno generiranu grafiku, zvuk ili pojačala za ostala osjetila umjesto dijela stvarnog svijeta u stvarnom vremenu. Iako se ova tehnologija već dulje vrijeme koristi na televiziji, najzanimljiviji radovi dolaze upravo sa sveučilišta. Vjerojatno najbolji primjer je SixthSense [3], projekt Pattie Maes i Pranava Mistryja koji je predstavljen 2009. godine na TED konferenciji u SAD-u. Oni su napravili sustav nadopunjene stvarnosti kao dio laboratorijskog projekta na MIT-u. Demonstracija sustava vidi se na slici 2.1.

Slika 2.1. – Pranav Mistry, demonstracija SixthSense uređaja

Sama izvedba oslanja se na neke osnovne komponente koje se mogu naći u mnogim sličnim sustavima – kameru, mali projektor, *smartphone* i zrcalo. Cijeli sustav je obješen o vrat korisnika, a upravlja ga se obojanim kapicama na prstima (slika 2.2.). Poseban je po tome što je jednostavan, jeftin, a učinkovit. Kamera uz pomoć zrcala pregledava svijet oko sebe te daje sliku mobilnom uređaju koji ju obrađuje. Projektor zatim projicira informacije na površinu koja može biti bilo što – zid, druga osoba ili vlastita ruka su samo neki od primjera. SixthSense vraća informaciju o gotovo bilo čemu, jer sliku stvarnog svijeta

analizira koristeći Internet – na ovaj način moguće je identificirati neki nepoznati objekt, dobiti informacije o proizvodima u dućanu i obavljati slične poslove. Koristeći ruke s prstima različitih boja korisnik može upravljati sadržajem projekcije preko kamere, a omogućeno je i prepoznavanje kompliciranih pokreta – crtanje kruga na zglobu će projicirati sat s trenutnim vremenom [25].



Slika 2.2. – Dijelovi SixthSense uređaja

Ipak, razvoj ovakvih tehnologija još je u povojima, ali slične, jednostavnije implementacije nadopunjene stvarnosti mogu se naći na nekim mobilnim uređajima, osobito u aplikacijama za iPhone i Android operativni sustav. Aplikacija Layar, koja se zasad koristi samo u Nizozemskoj, koristi kameru mobitela i njene GPS mogućnosti da pribavi informacije o okolnom području i prikazuje ih na ekranu, s time da prikazuje informacije o zgradama u smjeru u kojemu je mobitel usmjeren [16].

Komponenta iPhone aplikacije Yelp imenom Monocle, koja se razvila tijekom 2009. godine prikazuje informacije o lokalnim restoranima kako se korisnik kreće [23]. Do danas stvoreno je na stotine aplikacija za iPhone i slične telefone koji koriste nadopunjenu stvarnost, a još ih je više u razvoju. Urbanspoon [24] i Wikitude [9], aplikacije za snalaženje u prostoru, su samo neke od njih, a većina tih aplikacija zasniva se na GPS-u

mobilnog uređaja i kompasu. Iako prepoznavanje slike još nije dovedeno do savršenstva, već je vrlo visoke kvalitete, a razvija se svakodnevno.

Nadopunjena stvarnost također se pojavljuje u računalnim igricama [22]. Razni proizvođači već su odavno krenuli u smjeru istraživanja ovih tehnologija. Američki Total Immersion stvara aplikacije koje primjenjuju metode nadopunjene stvarnosti na kartice baseball igrača [21]. Takvi potezi su samo početak – nakon raširenijeg prijenosa ove tehnologije na mobilne uređaje, biti će moguće igrati igre posvuda. Jedan od primjera je scavenger hunt, potraga za blagom, koja će se moći igrati s virtualnim objektima i mobitelima kao uređajima uz pomoć kojih se ti, naoko nevidljivi predmeti, nalaze. Nadopunjena stvarnost korištena je i u vojne svrhe, upotrebom prijenosnih ekrana kojima se vojnicima prikazuju podaci korisni za trenutnu situaciju, poput nacrta zgrada [18].

Ipak, nadopunjena stvarnost ima svojih nedostataka i ograničenja. Na primjer, preciznost GPS-a je 9 metara, a najčešće ne radi dobro u zgradama. Daljnje razvijanje tehnologije za prepoznavanje slika moglo bi riješiti ovaj problem.

Iako im popularnost raste iz dana u dan, pretpostavlja se da pametni mobiteli kakvi postoje danas nikada neće biti korišteni u neke svrhe zbog svojih nedostataka (najčešće premali ekran). Neki drugi uređaji, poput SixthSensea, mogli bi zamijeniti ili nadograditi mobitele, nakon čega veličina ekrana, tipkovnice i mogućnosti uređaja više ne bi bile ograničene veličinom uređaja.

Jedan od većih problema u daljnjem razvoju nadopunjene stvarnosti je poticanje asocijalnog ponašanja i ovisnosti u toj mjeri da bi život u virtualnom svijetu mogao biti omogućen. Ipak, ne vjeruje se da bi ovo bio toliko veliki problem koliko nedostatak privatnosti. Uređaji s nadopunjenom stvarnosti svojim bi korisnicima omogućavali ispis svih dostupnih podataka o osobi koju vide [18].

Unatoč ovim nedostacima, mogućnosti nadopunjene stvarnosti su nevjerojatne – kao tehnologija može se primjeniti u gotovo bilo kojem zanimanju i situaciji – od arheologa, građevinara, arhitekata, preko liječnika, do običnog turista.

2.1. Razvoj nadopunjene stvarnosti

Sve je započelo 1957. godine kada je Morton Heilig, kinematograf, započeo razvijati simulator zvan Sensorama sa slikom, zvukom, vibracijom i mirisom. Nekoliko godina kasnije, 1966. godine, Ivan Sutherland je na Sveučilištu u Harvardu osmislio prvi uređaj za prikaz slike koji korisnik nosi na glavi (eng. Head Mounted Display) i na njega primjenio metode nadopunjene stvarnosti. Myron Krueger je prvi korisnicima omogućio interakciju s virtualnim objektima 1975. godine. Četrnaest godina kasnije Jaron Lanier osmišlja naziv virtualna stvarnost, a već godinu dana kasnije Tom Caudell osmišlja naziv nadopunjena stvarnost. L. B. Rosenberg 1992. razvija jedan od prvih sustava koji koriste nadopunjenu stvarnost, koji je nazvao Virtual Fixtures i pokazuje njegov doprinos ljudskom učinku u vojsci. Iste godine Steven Feiner, Blair MacIntyre i Doree Seligmann pišu prvi rad na temu nadopunjene stvarnosti, KARMA (Knowledge-based Augmented Reality Maintenance Assistant). To je jedan od najpoznatijih članaka na tu temu i vrlo je često citiran. Tijekom slijedećih nekoliko godina, popularnost i broj praktičnih projekata nadopunjene stvarnosti je rastao. Ipak, tek 1999. godine Hirokazu Kato uz pomoć M. Bilinghursta razvija ARToolkit, koji kasnije dovršavaju drugi znanstvenici iz HIT laboratorija Sveučilišta u Washingtonu. Ovo je potaklo nagli porast zainteresiranih za to područje jer je nadopunjena stvarnost napokon bila dostupna svim korisnicima računala, s minimalnim zahtjevima. Prva računalna igra koja koristi nadopunjenu stvarnost, ARQuake, napravljena je već godinu dana kasnije, a zanimljiva je po tome što je mobilna i igra se u prirodi. Nakon toga razvoj novih opcija u nadopunjenoj stvarnosti polako stagnira, ali se stare verzije nadopunjuju i nove izdaju redovito. Tek 2008. godine ponovo počinje ozbiljniji rad na modifikaciji ARToolkita i stvaranju novih programa koji bi otvorili nove mogućnosti programiranju aplikacija nadopunjene stvarnosti. Dolaskom Android platforme dolaze i mnogi programi koji pokušavaju prilagoditi nadopunjenu stvarnost mobilnim telefonima. Nova nada javlja se godinu dana kasnije, prebacivanjem tehnologije u Adobe Flash (FLARToolkit), a time i na internet, i obećavajućom prezentacijom SixthSensea, MIT projekta za nosive uređaje nadopunjene stvarnosti. Od tog trenutka započinje povećana proizvodnja aplikacija i drugih proizvoda dostupnih svima. [2]

3. Alati nadopunjene stvarnosti

U ovom poglavlju biti će opisani najpoznatiji alati koji se koriste za izradu aplikacija koje koriste elemente nadopunjene stvarnosti. Postoje alati za razne platforme s različitim mogućnostima. Navedeni alati su oni najpoznatiji ili najčešće korišteni, iako lista nije potpuna jer se konstantno širi.

3.1. ARToolkit

ARToolkit je jedan od najpopularnijih alata otvorenog koda, napravljen kao biblioteka za programski jezik C. Razvoj na programu je započeo japanski programer Hirokazu Kato s Instituta za znanost i tehnologiju u Nari kao projekt na Sveučilištu u Washingtonu u HIT laboratoriju. Neke od njegovih mogućnosti su praćenje orijentacije i položaja jedne kamere, praćenje jednostavnih markera unutar crnih kvadrata s opcijom definiranja vlastitih markera i jednostavno podešavanje kamere, a njegova brzina dovoljna je za izradu aplikacija nadopunjene stvarnosti u stvarnom vremenu. Trenutno je dostupan na platformama SGI IRIX, MS Windows, Mac OS X te sustave na osnovi Linuxa. Nedavno je također prenesen na iPhone, Android i Windows Phone.

Od ideje iza ARToolkita napravljene su mnoge inačice programa za rad u raznim programskim jezicima i za širenje na još više platformi. U nastavku su navedene samo neke od njih.

OSGART je kombinacija ARToolkita i OpenSceneGrapha, popularnog sučelja za programiranje aplikacija koje koriste 3D grafiku, najčešće korištenog za simulacije, računalne igre, virtualnu stvarnost, znanstvene vizualizacije i modeliranje [27].

ARTag je alternativa ARToolkitu koja koristi kompleksnije metode procesuiranja slike s ciljem otpornosti na negativne podražaje poput svjetla i veće pouzdanosti [28].

Mixed Reality Toolkit je projekt Sveučilišta u Londonu s jednostavnijom implementacijom i instalacijom od ARToolkita, a zadržava većinu funkcija nadopunjene stvarnosti [2].

FLARToolKit (Flash-based Augmented Reality Toolkit) je verzija ARToolKita koja služi za izradu Flash aplikacija s nadopunjenom stvarnosti. Flash se inače, suprotno imenu, vrlo sporo izvodi, iako je i dalje u mogućnosti izvoditi programe u stvarnom vremenu [29].

SLARToolKit (Silverlight and Windows Phone Augmented Reality Toolkit) je biblioteka za Microsoftovu alternativu Flashu – Silverlight i za Windows Phone, a zasnovan je na

principima ARToolkita i NyARToolkita. Također nije jedan od najbržih opcija za programiranje nadopunjene stvarnosti, ali je popularan zbog dobre dokumentacije i nekih dodatnih opcija povezanih sa samim Silverlightom [31].

NyARToolkit je ARToolkit biblioteka za virtualne strojeve, specifično one koji pokreću Javu, C# ili Android. Predstavlja jedan od ranijih pokušaja prilagodbe ARToolkita na druge programske jezike [30].

ARDesktop je biblioteka za trodimenzionalnu upotrebu na radnim površinama s raznim widgetima [32]. AndAR je Java API kojime se upravlja nekim elementima nadopunjene stvarnosti na Android operativnom sustavu [2].

ATOMIC Authoring Tool je jedan od najjednostavnijih alata za korištenje jer je napravljen za ljude s malo iskustva u programiranju, s jednostavnim sučeljem, ali i ograničenim funkcijama. ATOMIC Web Authoring Tool je dio ovog projekta, a služi za prebacivanje aplikacija nadopunjene stvarnosti na bilo koju web stranicu. Razvijen je kao grafičko sučelje za FLARToolkit.

ArUco je vrlo mala biblioteka za aplikacije pokretane uz OpenCv, koji se koristi na širokom spektru platformi i programskih jezika [33].

3.1.1. Svojstva i korištenje ARToolkita

Prilikom korištenja ARToolkita neizbježno moramo koristiti markere – praćene oznake koje predstavljaju poziciju i postavljaju koordinatni sustav prikaza. Algoritam za praćenje markera i obradu slike je slijedeći [1]:

1. Kamera snima video stvarnog svijeta i šalje ga u računalo.
2. Video se u računalu obrađuje sliku po sliku u potrazi za kvadratnim oblicima.
3. Ako je kvadrat pronađen i prepoznat kao marker (ako je u bazi markera i ima crne rubove), matematički se računa položaj kamere u odnosu na crni kvadrat.
4. Čim je definiran položaj kamere, crta se grafički model s mjesta tog markera u koordinatnom sustavu kamere.
5. Model se iscrtava na video iz stvarnog svijeta tako da izgleda kao da je na markeru.
6. Konačni izlaz na zaslonu je video s modelom nalijepljenim na početni video iz stvarnog svijeta.

3.2. Goblin XNA

Goblin XNA je platforma za istraživanje trodimenzionalnih korisničkih sučelja, uključujući nadopunjenu stvarnost i virtualnu stvarnost, s naglaskom na igre. Pisan je u C# i zasnovan na Microsoftovom XNA Game Studiu, a koristi ARTag za upravljanje kamerom pri pronalaženju markera. Razvijen je na Sveučilištu Columbia, a razvili su ga Ohan Oda i Steven Feiner. Prednost izrade programa u Goblin XNA radnom okviru je dvojaka – kod je relativno brz i pouzdan, a stalno izlaze nove verzije programa s novim mogućnostima.
Goblin XNA je dijelom bio financiran od strane Microsoft Researcha zbog povezanosti s Game Studiom, a povezan je i s projektom Goblin, prema čemu je i dobio ime. Brojne mogućnosti, stalno poboljšavanje i jednostavnost izrade računalnih igara su najvažnije značajke ovog alata [34].

3.3. Ostali alati

Postoje još mnogi drugi alati za razne platforme. Većina novih alata su biblioteke napravljene prvenstveno za mobilne uređaje Android ili iPhone. U nastavku su navedeni neki od njih. Njihova posebnost je u tome što zbog specijalizacije za mobilne uređaje nisu uvijek nužni markeri i zbog toga im je spektar mogućnosti puno širi od već navedenih aplikacija. Unatoč mnogim mogućnostima već navedenih aplikacija, ove biblioteke imaju vrlo jednostavnu implementaciju svih svojih funkcija jer mogu koristiti razne funkcionalnosti uređaja na kojemu se nalaze.
QCAR (Qualcomm Augmented Reality SDK) je napravljen za oba sustava, a programiranje se vrši u Javi [4]. Popcode je zanimljiv, jednostavniji program koji također podržava oba sustava [5]. Armsk (Augmented Reality Markerless Support Kit) je API koji je razvijan samo za Android platformu, a poseban je po tome što podržava prepoznavanje slika i objekata bez potrebe za crno-bijelim markerima, što se može vidjeti na slici 3.1. [6].

Slika 3.1. Nadopunjena stvarnost bez markera uz Armsk

SATCH je japanski projekt također napravljen isključivo za mobilne aplikacije, a podržava prepoznavanje slika i lica [7]. Ovaj projekt napravljen je iz već spomenutog Total Immersion projekta i na njemu rade stručnjaci koji već više godina programiraju u takvim sustavima [8]. SGAREnvironment je slabo podržan i relativno kompliciran radni okvir za korištenje nastao kao dio SimpleGeo projekta [10] koji je uključivao izradu aplikacija koje koriste nadopunjenu stvarnost u koordinaciji sa GPS uređajem i senzorima za lokaciju za iOS. Sličan, ali noviji i bolje održavan je ARView, radni okvir koji služi za upravljanje senzorima na iOS-u i prikazivanje nadopunjene stvarnosti. Moodstocks je softver koji se zasniva na kamerama mobilnih uređaja, s pristupom internet [11]. Qoncept i Kudan su neki od najpopularnijih proizvoda koji se koriste na Japanskom tržištu [12]. Na popisu se još nalazi nevjerojatno velik broj novih programa, od kojih je velik broj korišten posvuda u svijetu – Junaio [14], Magnitude [13], AndAR [15], Layar [20] i mnogi drugi. Praktičnost novih sustava poput onih koji se koriste na *smartphone* uređajima već je privukla tisuće programera diljem svijeta, a sudeći po broju korisnika, ovo područje je zanimljivo i onima koji se ne bave programiranjem.

4. Primjer nadopunjene stvarnosti u ARToolKitu

Prilikom izrade primjera korišten je programski jezik C (Microsoft Visual Studio 2010), GLUT biblioteka inačice 3.7.6. te ARToolKit inačice 2.27.1 za operativni sustav Windows. Još je korištena web kamera ugrađena u prijenosno računalo, a program se izvodi na Windowsima 7 i na Windowsima XP.
Za rad u Windows operativnom sustavu potrebna je DS video biblioteka koja se brine o komunikaciji s kamerom i DirectX inačice 9.0b ili noviji. Za rad u Windows XP operativnom sustavu potreban je i DirectX SDK koji podržava korištenu inačicu programa Microsoft Visual Studio.

4.1. Program simpleTest

Ovo je jedan od jednostavnijih programa koji vrlo dobro demonstrira mogućnosti ARToolKita te uz minimalne preinake može obrađivati složenije primjere iz računalne grafike. Program se sastoji od funkcije main i nekoliko funkcija za iscrtavanje grafike.

Kao i OpenGL, ARToolKit obrađuje podatke u petlji koja se izvršava kroz cijelo trajanje programa. Ta petlja poziva razne funkcije obrade koje obrađuju unos tipkovnicom, mišem ili neke druge promjene sustava ili promjene u načinu rada programa.

ARToolKit slijedi ove korake prilikom rada, odnosno, ove se naredbe na neki način nalaze u funkciji main:

1. Inicijalizacija – pokretanje snimanja videa te učitavanje markera i parametara kamere. Ovom koraku odgovara funkcija init.
2. Dohvat slike – ovaj korak započinje obradu slike koja se događa više puta u sekundi i prvi je dio glavne petlje nazvane mainLoop. Funkcija koja vrši dohvat je arVideoGetImage.
3. Prepoznavanje markera – u dohvaćenoj slici prepoznaju se svi markeri i prepoznatljive oznake. Ovaj korak vrši funkcija arDetectMarker.
4. Izračun transformacija kamere – ovisno o veličini i položaju markera određuje se koordinatni sustav u kojemu će objekti biti iscrtani. Funkcija koja radi ove izračune je arGetTransMat.

5. Iscrtavanje virtualnih objekata – ovo je funkcija koja se najčešće mijenja, a više ovisi o OpenGLu nego o ARToolKitu. Ujedno je i zadnji korak glavne petlje mainLoop. U ovoj implementaciji ova funkcija zove se draw.
6. Završetak procesuiranja slike – ovaj korak najčešće obilježava kraj izvođenja programa jer se događa nakon glavne petlje i zatvara video izlaz. Funkcija koja odgovara ovom koraku je funkcija cleanup.

Kod programa sastoji se od još nekih dijelova, poput postavljanja postavki kamere i uključivanja potrebnih biblioteka.

Blok za uključivanje biblioteka na Windows operativnom sustavu uključuje slijedeće linije:

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <GL/gl.h>
#include <GL/glut.h>
#include <AR/gsub.h>
#include <AR/video.h>
#include <AR/param.h>
#include <AR/ar.h>
```

Prvih nekoliko linija standardni su za većinu C programa i uključuju standardne biblioteke. Nakon toga uključuju se GL/gl.h i GL/glut.h koji predstavljaju GLUT biblioteku koja se koristi za iscrtavanje modela i općenito korištenje računalne grafike. Zatim slijede četiri biblioteke ARToolKita u kojima su zapisane sve funkcije i neki parametri koji se koriste u programu.

Prije definiranja funkcija postavljeni su neki osnovni parametri. Oni uglavnom uključuju imena datoteka koje se kasnije koriste u programu – datoteke za definiranje kamere i parametara kamere, te datoteke u kojoj se nalaze podaci o markeru koji koristimo. Također se posebno navodi veličina markera, koordinate njegovog središta i stvara se matrica koja će kasnije transformirati objekte u sustav kamere.

Prva bitna funkcija je ona osnovna: funkcija main.

Ovo je izgled funkcije za ovaj primjer:

```
int main(int argc, char **argv) {
```

```
        glutInit(&argc, argv);
        init();
        arVideoCapStart();
        argMainLoop( NULL, keyEvent, mainLoop );
        return (0);
    }
```

Redom, prvo nailazimo na glutInit funkciju. To je funkcija GLUT biblioteke koja argumente programa predaje na korištenje glut funkcijama. Slijedi funkcija init, koja je prva funkcija ARToolKita. Ona sadrži kod za pokretanje snimanja videa, čitanje markera i parametara kamere te pokretanje prozora za postavljanje opcija grafike. Ova funkcija odgovara prvom koraku izvođenja programa te se izvodi samo jednom, a biti će detaljnije objašnjena kasnije. Nakon nje dolazimo do funkcija koje se izvršavaju u stvarnom vremenu, prva od kojih je arVideoCapStart. Ona pokreće video u novom prozoru. Nakon nje, argMainLoop obrađuje sve daljnje unose u program. Ta funkcija također asocira funkciju keyEvent s događajima pritiska tipke na tipkovnici, a mainLoop s glavnom petljom koja obrađuje svu grafiku. Definicija funkcije argMainLoop nalazi se u biblioteci gsub.

Unutar funkcije main nalazi se poziv funkcije init koja služi za inicijalizaciju snimanja i učitavanje nekih početnih parametara ARToolKita.

```
    static void init( void ) {
        ARParam  wparam;
        if( arVideoOpen( vconf ) < 0 ) exit(0);
        if( arVideoInqSize(&xsize, &ysize) < 0 ) exit(0);
        printf("Image size (x,y) = (%d,%d)\n", xsize, ysize);
        if( arParamLoad(cparamName, 1, &wparam) < 0 ) {
            printf("Camera parameter load error!\n");
            exit(0);
            }
        arParamChangeSize( &wparam, xsize, ysize, &cparam );
        arInitCparam( &cparam );
        printf("*** Camera Parameter ***\n");
        arParamDisp( &cparam );
```

```
if( (pattId = arLoadPatt(pattName)) < 0 ) {
      printf("Pattern load error!\n");
      exit(0);
      }
argInit( &cparam, 1.0, 0, 0, 0, 0 );
}
```

Prvo se otvara spremnik za video funkcijom arVideoOpen te pronalazi veličina prozora funkcijom arVideoInqSize. Varijabla vconf sadrži ime datoteke s početnom konfiguracijom kamere, a definirana je na početku programa. Ova datoteka specifična je za sustav na kojemu se nalazi, iako je većinom automatski namještena.

Nakon toga postavljaju se parametri programa funkcijom arParamLoad. Najvažniji su markeri, virtualni objekti kojima ovi markeri odgovaraju, te karakteristike kamere koja se koristi.

U ovom slučaju, koriste se datoteke definirane na početku programa, gdje je cparam_name datoteka s parametrima kamere – Data/camera_para.dat, a patt_name datoteka s izgledom markera – Data/patt.hiro.

Čak i kada se koristi ista kamera, neki parametri kamere se mijenjaju, poput trenutne veličine slike. Takve promjene obrađuje funkcija arParamChangeSize.

Parametri se zatim postavljaju funkcijom arInitCparam, te ispisuju na ekran funkcijom arParamDisp.

Slijedećom funkcijom, arLoadPatt, učitavamo izgled markera i dobivamo njegov identifikacijski broj. Na kraju otvaramo prozor u kojemu ćemo prikazivati svu grafiku pomoću dosad navedenih parametara funkcijom argInit. Time smo prošli kroz funkciju init.

Funkcija koja slijedi je vjerojatno ona najvažnija – mainLoop, funkcija koja u petlji obrađuje sliku i sve zahtjeve koje program dobiva. Funkcija sadrži i ponavlja korake 2 do 5 standardnog izvođenja ARToolKit programa.

Zbog duljine, funkcija će biti predstavljena u dijelovima. Prvi dio uključuje drugi korak već navedenog algoritma, dohvat slike.

```
if( (dataPtr = (ARUint8 *)arVideoGetImage()) == NULL ) {
      arUtilSleep(2);
      return;
```

```
}
```

Za to koristimo funkciju arVideoGetImage. Nakon toga na ekranu se prikazuje slika. Ovisno o postavkama, slika može biti iskrivljena ili neiskrivljena. Iskrivljavanje slike može stvoriti puno bolju sliku, ali može i značajno smanjiti broj obrađenih slika u sekundi. U ovom primjeru koristi se neiskrivljena slika. To namještamo slijedećim funkcijama:

```
argDrawMode2D();
argDispImage( dataPtr, 0,0 );
```

Zatim prelazimo na treći korak algoritma – prepoznavanje markera. Za to koristimo slijedeći dio koda:

```
if( arDetectMarker(dataPtr, thresh, &markerInfo, &markerNum)
< 0 ) {
     cleanup();
     exit(0);
}
```

U ovom kodu koristimo funkciju arDetectMarker kojoj predajemo podatke koje smo do tad nakupili o video ulazu i markeru. Broj markera koji je pronađen vraća se u varijabli marker_num, a u varijabli marker_info spremljen je pokazivač na popis markera u kojem se nalaze podaci o koordinatama i pouzdanosti prepoznavanja te identifikator za svaki marker.

U tom trenutku slika je prikazana i analizirana, tako da se odmah počinje obrađivati slijedeća. Prijelaz na slijedeću sliku određuje se funkcijom arVideoCapNext, a obrada se nastavlja. Odmah uspoređujemo pouzdanosti prepoznavanja prepoznatih markera te prikazujemo onaj s najvećom pouzdanosti. Prikazani marker određujemo kroz petlju koja redom prolazi kroz identifikatore markera i ispituje ih. Ukoliko nije pronađena niti jedan marker, međuspremnik u kojemu se sprema slika se odmah postavlja, kako se ništa ne bi iscrtalo.

```
k = -1;
for( j = 0; j < markerNum; j++ ) {
     if( pattId == markerInfo[j].id ) {
          if( k == -1 ) k = j;
          else if( markerInfo[k].cf < markerInfo[j].cf )
```

```
                k = j;
        }
    }
    if( k == -1 ) {
        argSwapBuffers();
        return;
    }
```

Slijedeći dio je vjerojatno najzanimljiviji iz matematičkog pogleda na računalnu grafiku, a sastoji se od transformacije iz sustava kamere u sustav markera, što se u ARToolkitu određuje vrlo jednostavno, a koriste se neki parametri koji su definirani na početku programa:

```
arGetTransMat(&markerInfo[k], pattCenter, pattWidth, pattTrans);
```

Stvarni položaj kamere i njegova orijentacija u odnosu na marker nalaze se u matrici patt_trans veličine 3x4.

Na samome kraju, virtualni objekti iscrtavaju se pozivom funkcije draw, koja predstavlja peti korak algoritma.

Ta funkcija dijeli se u tri dijela – inicijalizacija obrade, postavljanje matrice i sama obrada objekta. Prvi korak obavlja se postavljanjem osnovnih postavki OpenGL-a:

```
argDrawMode3D();
argDraw3dCamera( 0, 0 );
glClearDepth( 1.0 );
glClear(GL_DEPTH_BUFFER_BIT);
glEnable(GL_DEPTH_TEST);
glDepthFunc(GL_LEQUAL);
```

Nakon toga potrebno je prilagoditi dobivenu matricu patt_trans obliku koji koristi OpenGL – listi sa 16 vrijednosti. Ugrađena funkcija argConvGlpara obavlja taj posao. Pošto te vrijednosti predstavljaju položaj i orijentaciju stvarne kamere, korištenjem njih da se postavi virtualna kamera zapravo će postaviti iscrtane objekte na položaj pripadnog markera. Prilagodba matrice i nastavak obrade prikazan je u nastavku.

```
argConvGlpara (pattTrans, glPara);
glMatrixMode (GL_MODELVIEW);
glLoadMatrixd( glPara );
```

Položaj virtualne matrice postavlja se OpenGL funkcijom glLoadMatrixd. Poslijednji dio

koda je obrada i prikaz 3D objekta. U ovom jednostavnom slučaju to je kocka. Ovo je dio koji se najčešće mijenja kako bi se prikazali različiti objekti. U našem slučaju prikaz je definiran funkcijom glutSolidCube koja kao argument prima duljinu stranice kocke.
Konačno, neke varijable OpenGL-a namjestimo na njihove pretpostavljene vrijednosti pozivima:

```
glDisable (GL_LIGHTING);
glDisable (GL_DEPTH_TEST);
```

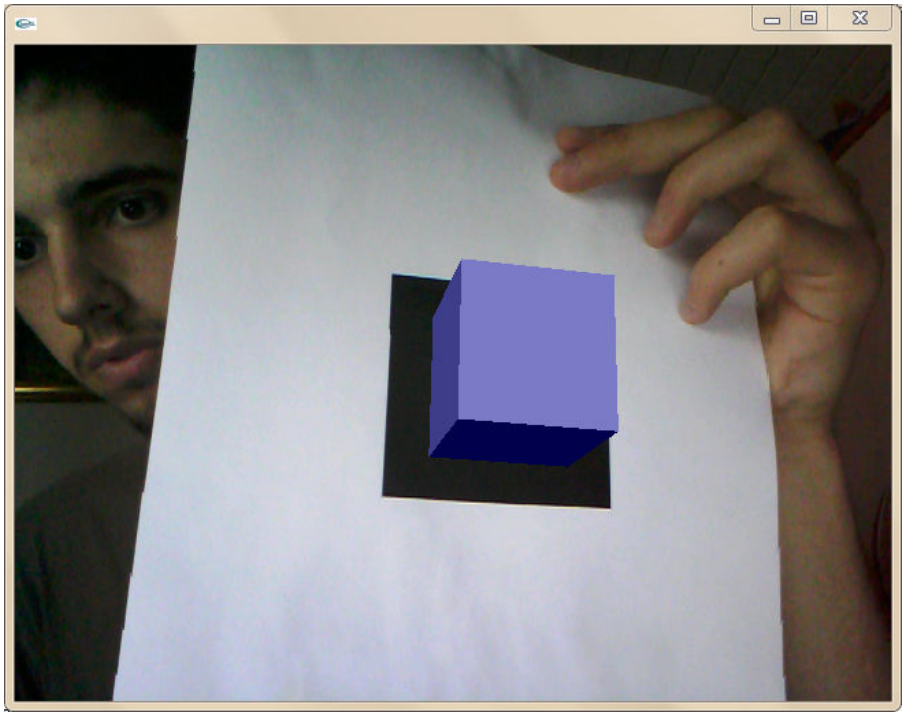
Ovime završava funkcija draw, a time i petlja koja se vrti za svaku sliku koju praćeni video stvara.
Preostala je još samo funkcija cleanup, koja zaustavlja obradu slika i zatvara spremnik za video. To čini pozivom funkcija arVideoCapStop, koja zaustavlja obradu slika, arVideoClose, koja zatvara spremnik za video, te argCleanup, koja završava rad. [1]

4.1.1. Opcije prikaza objekata

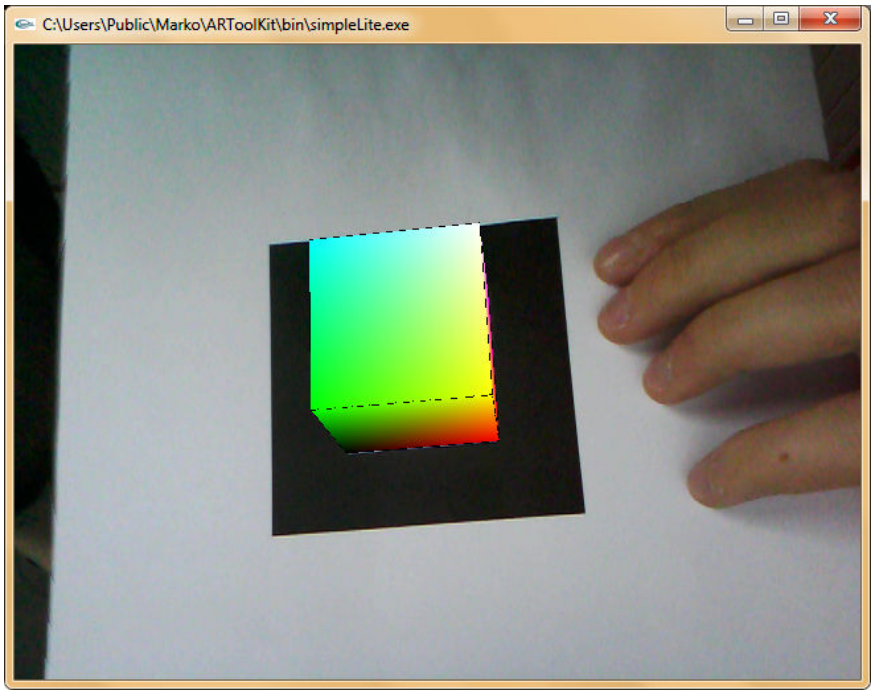
Kao što je već spomenuto, draw funkcija umjesto pretpostavljene kocke može iscrtavati gotovo bilo koji virtualni objekt koji se može iscrtati preko neke od funkcija OpenGL-a. U ovom poglavlju biti će dani neki primjeri objekata. Na slici 4.1.nalazi se rezultat izvođenja programa simpleTest, opisanog u prošlom poglavlju, koristeći osnovni model kocke.

Na slikama 4.2. i 4.3. nalaze se primjeri nekih drugih jednostavnih objekata – kocke s raznobojnim vrhovima napravljene preko glVertex funkcija OpenGL-a te kugla napravljena ugrađenom funkcijom glutSolidSphere. Na slikama 4.4. i 4.5. nalazi se primjer obrade slike pri kojoj ARToolKit ne skalira veličinu objekta na koordinatni sustav markera, nego ga ostavlja u jednoj veličini, neovisno o njegovoj udaljenosti od kamere.

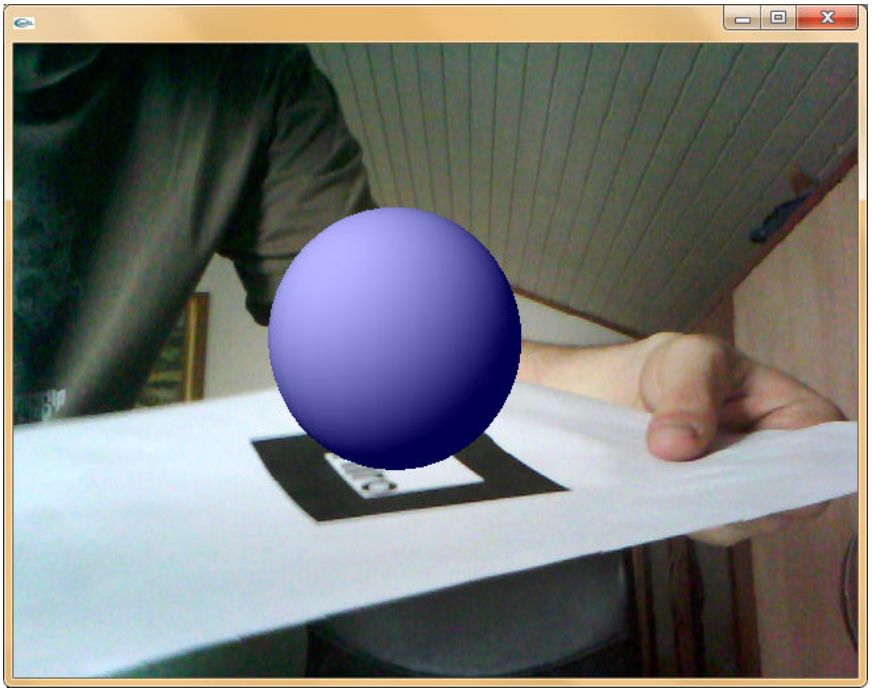
Na samome kraju poglavlja prikazan je izgled modela nekih objekata učitanih iz datoteka u Wavefront zapisu (.obj datoteke). To prikazuju slike 4.6. – 4.8., a sve su i osjenčane konstantnim sjenčanjem [35].



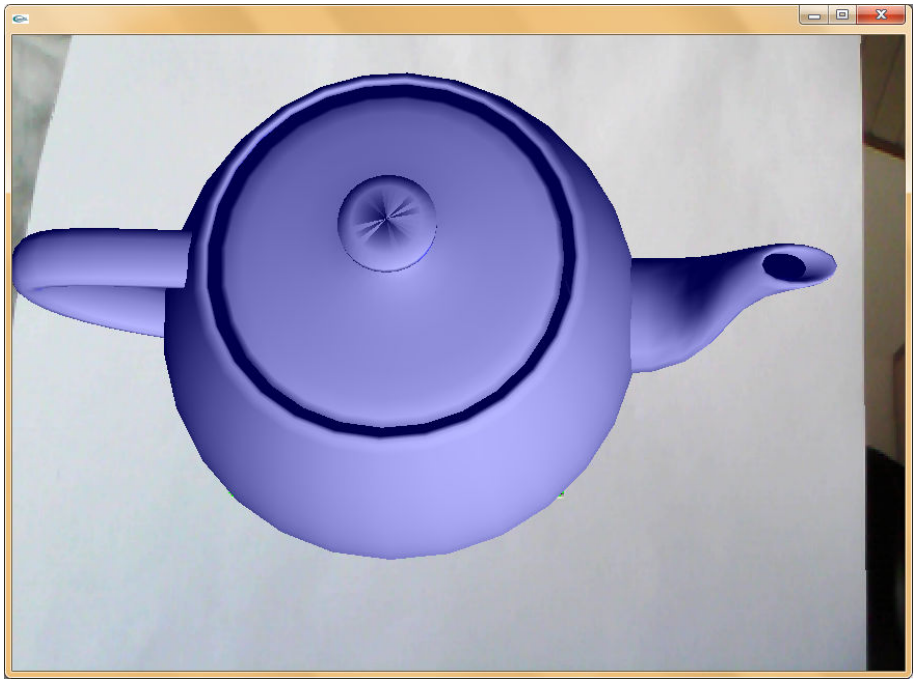
Slika 4.1. – Primjer projekcije objekta kocke



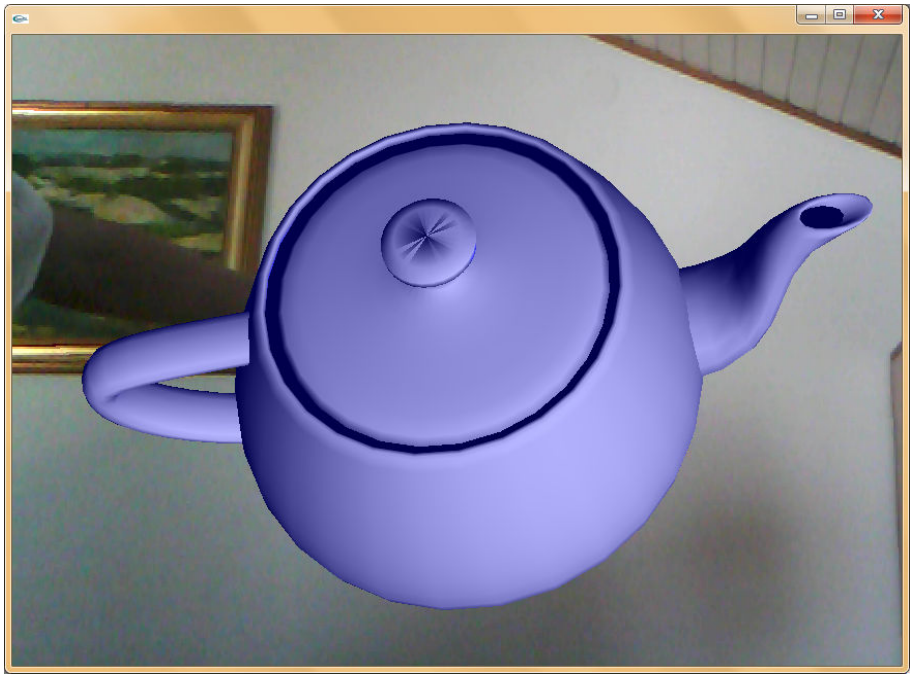
Slika 4.2. – Varijacija primjera simpleTest s obojanom kockom



Slika 4.3. – Primjer drugačijeg objekta – ispis funkcijom glutSolidSphere



Slika 4.4. – Iscrtavanje čajnika na udaljenosti od 40 cm



Slika 4.5. – Iscrtavanje čajnika na udaljenosti od 150 cm

Zbog velike udaljenosti markera vidljiva je mala pogreška u određivanju orijentacije markera – na slici 4.5. čajnik je lagano pomaknut u odnosu na sliku s bližim markerom.

Slika 4.6. – Model žabe

Slika 4.7. Model bika

Slika 4.8. Model lubanje

4.2. Ograničenja ARToolKita

Unatoč velikom rasponu mogućnosti, ARToolKit također posjeduje i mnoga ograničenja. Ta ograničenja se trenutno mogu primjeniti za veliku većinu programa za stvaranje nadopunjene stvarnosti, jer velikim dijelom ovise o sklopovskoj opremi.

Prvo i vjerojatno najveće ograničenje ARToolKita je korištenje markera određenog oblika. Ovaj problem je namjerno implementiran na taj način s ciljem jednostavnijeg i bržeg izvođenja programa. Neki programi, ranije navedeni, podržavaju rad sa markerima koji su u potpunosti definirani od strane korisnika, bez ograničenja boja i crnog kvadratnog obruba, a neki (većinom za mobilne uređaje) mogu stvarati nadopunjenu stvarnost i bez upotrebe markera.

Slijedeći problem proizlazi iz činjenice da će se virtualni objekti prikazivati samo kada su markeri vidljivi kameri. To bi moglo ograničiti veličinu i pomicanje virtualnih objekata, a problem je i djelomično prekrivanje markera raznim objektima.

Slika 4.9. – Spriječavanje iscrtavanja djelomičnim prekrivanjem markera

Za primjer uzmimo objekt veći od markera koji se pomiče prema rubovima slike. U

jednom trenutku dio markera više neće biti vidljiv kameri i virtualni objekt će nestati, što znači da nećemo moći vidjeti izlazak objekta sa slike. Ovo ograničenje moguće je zaobići iscrtavanjem manjeg dijela prostora od onoga koji kamera snima i program analizira. Ipak, neka ograničenja markera nije moguće izbjeći u stvarnom vremenu.

Slika 4.10. – Marker na neravnoj podlozi

Jedno od takvih ograničenja je i loš prikaz virtualnih objekata na deformiranim markerima. Deformacija markera uključuje sve promjene na originalnom markeru, od savijanja papira i micanja markera na veću udaljenost, do bojanja markera bojama ili ispis markera na nekom tamnijem papiru. ARToolKit koristi vlastiti algoritam za olakšano pronalaženje markera, tako da umanjuje utjecaj ovih smetnji, no već i sam taj algoritam uvodi neka nova ograničenja. Na primjer, zbog slobodnije interpretacije markera i računanja na pogreške, kamera kao marker može prepoznati i objekte koji su slični markeru. ARToolKit zbog toga na slikama često detektira veći broj markera, što usporava postupak usporedbe pronađenog markera s traženim. Savijanje papira može u potpunosti onemogućiti prepoznavanje uzorka, ili ga samo učiniti nepouzdanim. Markeri koji se ne prikazuju pouzdano zbog neke od deformacija, ali se i dalje mogu prepoznati često mijenjaju koordinate točaka koje nisu

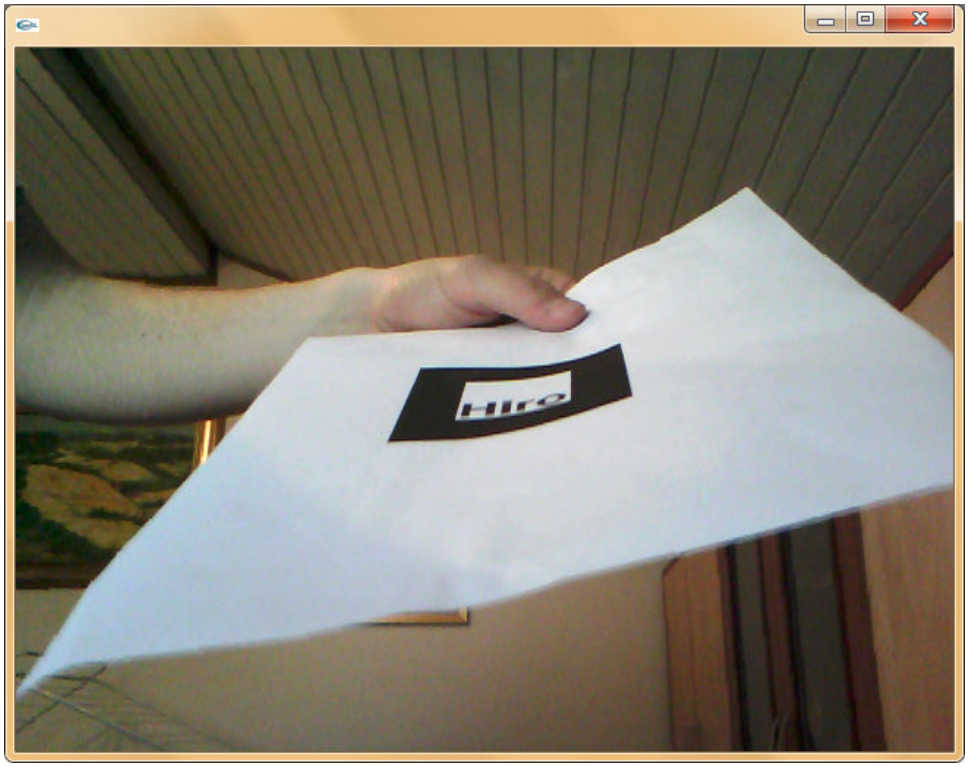
položene na marker. To se događa zbog prepoznavanja drugačijih koordinatnih sustava u svakoj slici, što najviše utječe na točke virtualnog objekta udaljene od samog markera.

Pomicanje markera na veću udaljenost utječe na sposobnost alata da ga detektira. Iako to često nije praktično, moguće je napraviti veće markere kako bi bili prepoznati na većim udaljenostima. Najveća udaljenost na kojoj se marker može prepoznati također ovisi o složenosti uzorka – jednostavni markeri s velikim crnim ili bijelim površinama mogu biti prepoznati na puno većim udaljenostima. Markeri iste veličine ali složenijeg uzorka mogu prepoloviti maksimalnu udaljenost prepoznavanja markera. Pri optimalnim uvjetima (dobrom osvjetljenju, ravnom papiru, nepomičnom markeru), marker hiro.patt (korišten u primjerima) ponekad je moguće prepoznati s udaljenosti od 150 centimetara. Uzevši u obzir da je taj marker veličine 8x8 cm, on spada među markere s jednostavnijim uzorcima.

U slijedećoj tablici dane su vrijednosti veličina markera s prosječno složenim uzorcima i udaljenosti na kojoj se mogu pouzdano prepoznati.

Veličina markera (cm)	Udaljenost od kamere (cm)
7	40
9	65
11	87
19	130

Prepoznavanje također može biti otežano lošom orijentacijom markera u odnosu na kameru. Kako se markeri naginju u stranu, njihovo prepoznavanje ubrzo postaje nepouzdano. Već pod kutem od 45 stupnjeva u odnosu na kameru nije moguće prepoznati neke uzorke, a pod kutem od 75 stupnjeva moguće je prepoznati samo one najjednostavnije, i to u jako dobrim uvjetima. Kod horizontalno položenih markera identifikacija je nemoguća. Na slici 4.8. prikazana je granica ispod koje se hiro.patt marker više neće prepoznavati.



Slika 4.11. Neprepoznati marker pod velikim kutem

Osvjetljenje također utječe na prepoznavanje uzoraka – prvenstveno refleksija svjetlosti od bijele površine ispod markera, koja u nekim slučajevima može spriječiti prepoznavanje svih markera. Pri dobrom osvjetljenju ili uz površinu koja nije refleksivna, ovaj problem moguće je gotovo u potpunosti izbjeći [1].

U konačnici, prepoznavanje markera ovisiti će i o kvaliteti same kamere i njenoj brzini prilagodbe, te o sustavu na kojemu se program pokreće. Brzo pomicanje markera će često biti pogrešno interpretirano.

Neki od ovih problema, poput brzine sustava, korištenje bržih algoritama umjesto preciznih i smanjena detekcija markera na većim udaljenostima, mogli bi se izbjeći korištenjem alata koji ne rade u stvarnom vremenu, a cilj im je ostvariti vjeran prikaz – ne istovremen. Najveća udaljenost markera od kamere ovisi o kvaliteti kamere, odnosno, razvojem tehnologije ovo ograničenje biti će moguće ublažiti.

5. Zaključak

Istraživanje mogućnosti i granica nadopunjene stvarnosti pokazalo se opširnim, ali jednostavnim poslom, jer se literatura širi svakodnevno. Veći problem predstavljala je instalacija i pokretanje raznih biblioteka potrebnih za izvođenje ovih programa zbog složenog postupka instalacije i problema s kompatibilnosti, unatoč velikoj prenosivosti biblioteka. Sama implementacija je relativno jednostavna, jer uključuje tek upoznavanje s nadopunjenom stvarnosti, odnosno najjednostavnijim primjerima koji se mogu napraviti, a da pokazuju dosege tehnologije.

Rad predstavlja uvod u nadopunjenu stvarnost s popisom brojnih mogućnosti, ali i ograničenja. Kao takav predstavlja pregled podloge za složenije projekte, te analizu trenutnog napretka na ovom području.

Neke od mogućnosti poboljšanja prikazanih primjera su [19]:

- Prijenos na mobilnu platformu
- Ugradnja veće interaktivnosti i detekcija pokreta
- Implementacija kompliciranijih algoritama upravljanja virtualnim objektima
- Korištenje sustava prikaza koji ne koristi markere
- Uvođenje prepoznavanja više složenih markera i njihova koordinacija
- Razvijanje sustava prepoznavanja objekata umjesto markera
- Spajanje tehnologije s drugim, trenutno popularnim tehnologijama (SixthSense)

Nadopunjena stvarnost je i dalje mlado i neistraženo područje, kao što pokazuju projekti koji se izvode posvuda u svijetu. Daljnjim istraživanjem ovo područje moglo bi postati vrlo popularno u nadolazećim generacijama zbog postupnog pojednostavljivanja programa za upravljanje nadopunjenom stvarnosti. Također, dostupnost ovih programa se već godinama poboljšava, a trenutno postoje biblioteke za gotovo sve programske jezike i platforme.

6. Literatura

1. Philip Lamb, ARToolKit, 28. studeni 2007., http://www.hitl.washington.edu/artoolkit/, svibanj 2012.
2. Wikipedia, Augmented Reality, http://en.wikipedia.org/wiki/Augmented_reality, svibanj 2012.
3. Pranav Mistry, SixthSense, http://www.pranavmistry.com/projects/sixthsense/, svibanj 2012.
4. Qualcomm Mobile Augmented Reality, https://developer.qualcomm.com/mobile-development/mobile-technologies/augmented-reality, svibanj 2012.
5. Popcode, http://popcode.info/, svibanj 2012.
6. ARmsk, http://armsk.org/, svibanj 2012.
7. SATCH, http://satch.jp/en/, svibanj 2012.
8. Bruno Uzzan, Total Immersion, http://www.t-immersion.com/, svibanj 2012.
9. Wikitude, http://www.wikitude.com/tour/wikitude-world-browser, svibanj 2012.
10. SimpleGeo, http://apps.simplegeo.com/, svibanj 2012.
11. Moodstocks, http://www.moodstocks.com/, svibanj 2012.
12. Kudan, http://www.kudan.eu/, svibanj 2012.
13. Magnitude, http://code.google.com/p/magnitudehq/, svibanj 2012.
14. Junaio, http://www.junaio.com/, svibanj 2012.
15. AndAR, *Android Augmented Reality*, http://code.google.com/p/andar/, svibanj 2012.
16. Layar, http://www.layar.com/, svibanj 2012.
17. ARLab, http://www.arlab.com/, svibanj 2012.
18. Kevin Bonsor, How Augmented Reality Works, http://www.howstuffworks.com/augmented-reality.htm, svibanj 2012.
19. Leslie Berlin, Kicking Reality Up A Notch, http://www.nytimes.com/2009/07/12/business/12proto.html, svibanj 2012.
20. Azadeh Ensha, Another Augmented-Reality App for the iPhone, http://gadgetwise.blogs.nytimes.com/2009/10/15/augmented-reality-apps-continute-to-roll-out/, svibanj 2012.

21. Priya Ganapati, How it Works: Augmented Reality, http://www.wired.com/gadgetlab/2009/08/total-immersion/, svibanj 2012.

22. Ori Inbar, Top 10 augmented reality demos that will revolutionize video games, http://gamesalfresco.com/2008/03/03/top-10-augmented-reality-demos-that-will-revolutionize-video-games/, svibanj 2012.

23. Ben Parr, Easter Egg: Yelp Is the iPhone's First Augmented Reality App, http://mashable.com/2009/08/27/yelp-augmented-reality/, svibanj 2012.

24. Jenna Wortham, UrbanSpoon Makes It Easier to 'Scope' Out Restaurants, http://bits.blogs.nytimes.com/2009/10/14/urbanspoon-makes-it-easier-to-scope-out-restaurants/, svibanj 2012.

25. Pattie Maes, Pranav Mistry, SixthSense Demonstration, http://www.ted.com/talks/pattie_maes_demos_the_sixth_sense.html, svibanj 2012.

26. Kato H., Billinghurst M., Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System, 2nd International Workshop on Augmented Reality, San Francisco, USA (1999)

27. OSGART, http://www.artoolworks.com/community/osgart/, svibanj 2012.

28. ARTag, http://www.artag.net/, svibanj 2012.

29. FLARToolKit, *Flash-based Augmented Reality Toolkit*, http://www.libspark.org/wiki/saqoosha/FLARToolKit/en, svibanj 2012.

30. NyARToolKit, http://www.artoolworks.com/products/desk-top/nyartoolkit, svibanj 2012.

31. SLARToolKit, Silverlight and Windows Phone Augmented Reality Toolkit, http://slartoolkit.codeplex.com/, svibanj 2012.

32. Bruce Sterling, Augmented Reality: ARDeskTop, http://www.wired.com/beyond_the_beyond/2009/09/augmented-reality-ar-toolkit/, svibanj 2012.

33. ArUco, http://www.uco.es/investiga/grupos/ava/node/26, svibanj 2012.

34. Goblin XNA, *Goblin XNA: A Platform for 3D AR and VR Research and Education*, http://monet.cs.columbia.edu/projects/goblin/, svibanj 2012.

35. Željka Mihajlović, Interaktivna računalna grafika, http://www.zemris.fer.hr/predmeti/irg/, svibanj 2012.

36. Paul Mason, Image Processing and the ARToolkit, http://comptuicomputervision.blogspot.com/, svibanj 2012.

Nadopunjena stvarnost objektima računalne grafike

Sažetak

Nadopunjena stvarnost je relativno novo područje koje otvara put razvitku mnogim tehnologijama koje su još u prošlom stoljeću bile dio znanstvene fantastike. Kombinacija nadopunjavanja stvarnog svijeta virtualnim, uz pomoć boljeg prepoznavanja uzoraka i sve moćnijih i mobilnijih uređaja utječe na razvoj znanosti na nevjerojatan način. Osim prolaska kroz trenutno popularne aplikacije za izradu programa s nadopunjenom stvarnosti, detaljnije je opisan primjer napravljen pomoću alata ARToolKit, koristeći OpenGL, specifikaciju za rad s računalnom grafikom, te programski jezik C++. Posebna pažnja posvećena je mogućnostima i ograničenjima ARToolKita, a i općenito nadopunjene stvarnosti. U rad je također uključen kratki pregled povijesti razvoja nadopunjene stvarnosti, popis područja u kojima se ona već koristi i neka područja u kojima je tek otkrivena korist ove tehnologije.

Ključne riječi: nadopunjena stvarnost, računalna grafika, ARToolKit, OpenGL

Augmented reality with the use of computer graphics

Abstract

Augmented reality is a relatively new field that opens the path to developing other technologies that were considered science fiction during the last century. The combination of augmenting the real world with the virtual, with the aid of improved pattern recognition and an increase in gadget power and mobility impacts science in an incredible way. Apart from the short overview of the currently popular applications used for creating programs with augmented reality, an example has been created and described with the help of the ARToolKit program, the OpenGL specification for computer graphics and the programming language C++. Special attention was given to the potential and limitations of ARToolKit, as well as augmented reality itself. A short review of the history of augmented reality, a list of areas where it is already used and a list of areas where it was just recently introduced have also been included.

Keywords: Augmented reality, computer graphics, ARToolKit, OpenGL