

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 3250

**PRIRODNA INTERAKCIJA KORISNIKA  
GRAFIČKIM OBJEKTIMA**

Marko Domlija

Zagreb, svibanj 2013.



# Sadržaj

|                                            |    |
|--------------------------------------------|----|
| Uvod.....                                  | 1  |
| 1. Uređaj Microsoft Kinect .....           | 2  |
| 1.1. Senzori Kinecta.....                  | 2  |
| 1.2. Praćenje kostura.....                 | 4  |
| 2. Interakcija korisnika i aplikacije..... | 7  |
| 2.1. Prirodna interakcija.....             | 7  |
| 2.2. Upravljanje gestama .....             | 8  |
| 2.3. Dinamične i statične geste .....      | 9  |
| 3. Aplikacija eIzlog.....                  | 10 |
| 3.1. Korištena tehnologija.....            | 10 |
| 3.2. Motivacija za izradu aplikacije ..... | 11 |
| 3.3. Izgled aplikacije.....                | 12 |
| 4. Analiza programskog produkta.....       | 18 |
| 4.1. Prirodna interakcija u praksi .....   | 18 |
| 4.2. Performanse aplikacije eIzlog.....    | 19 |
| Zaključak .....                            | 21 |
| Literatura.....                            | 22 |
| Sažetak .....                              | 23 |
| Abstract.....                              | 24 |
| Uputa za instalaciju.....                  | 25 |

# Uvod

U radu su razrađeni neki osnovni koncepti interakcije korisnika s računalom putem senzora Kinect. Prvo je ukratko opisan senzor Kinect i njegove mogućnosti, nakon čega slijedi kratak opis obrađene problematike.

Ukratko je objašnjen način na koji uređaj Kinect detektira pokrete korisnika uz osvrt na teorijsku podlogu problema. Razrađen je koncept prirodne interakcije korisnika s računalom i detaljnije je obrađen model upravljanja aplikacijom pomoću gesti. Opisana je aplikacija eIzlog koja je razvijena u sklopu rada i analizirane su njene karakteristike. Na kraju se nalazi kratki osvrt na sveukupni rezultat rada.

# 1. Uređaj Microsoft Kinect

Microsoft Kinect je uređaj koji se na tržištu pojavio 2010. godine kao kontroler za igraću konzolu Xbox 360. Sastoji se od nekoliko vizualnih senzora i mikrofona koji omogućavaju praćenje pokreta i govora jednog ili više korisnika te na taj način ostvaruje interakciju s konzolom bez potrebe fizičkog kontakta korisnika s bilo kakvom opremom [1].



Slika 1.1: Uređaj Kinect [2]

Inačica Kinecta za konzolu Xbox 360 prikazana je na slici (Slika 1.1).

Uređaj je vrlo brzo postigao uspjeh u svijetu konzola, te je Microsoft 2012. godine na tržište izašao s inačicom proizvoda za osobna računala zajedno s pripadajućim upravljačkim programima i opremom za razvoj aplikacija za Kinect (SDK, eng. software development kit).[1]

## 1.1. Senzori Kinecta

Uređaj Kinect sastoji se od tri vrste senzora[1]:

- video kamera visoke razlučivosti
- četiri mikrofona
- infracrveni senzor (uz infracrveni projektor)



Slika 1.2: Senzori uređaja Kinect [3]

Raspored navedenih senzora na uređaju Kinect prikazan je na slici (Slika 1.2). Kinect je smješten na motoriziranom stalku te je moguće podešavati nagib uređaja putem programske podrške.

RGB kamera omogućava prikaz slike visoke rezolucije i njene obrade u aplikacijama.

Mikrofoni omogućuju prepoznavanje govora korisnika i zadavanje glasovnih naredbi. Više ugrađenih mikrofona omogućuje kvalitetnije procesiranje zvuka (otklanjanje buke i lociranje izvora zvuka [1]).

Infracrveni senzor i projektor rade sinergično: projektor raspršuje infracrvene zrake po prostoru, a senzor detektira točke koje se odbijaju od predmeta koji se nalazi ispred Kinecta. Na osnovi međusobne udaljenosti tih točaka Kinect određuje udaljenost nekog predmeta od senzora. Što su infracrvene točke međusobno bliže jedna drugoj, predmet je bliže senzoru. Ova tehnologija koristi se i prilikom detekcije kostura korisnika ispred senzora što će biti objašnjeno kasnije.

## 1.2. Praćenje kostura

Osnovna mogućnost Kinecta koja ga čini prepoznatljivim jest upravljanje pokretima tijela što je ostvareno pomoću infracrvene tehnologije. Infracrveni senzor stvara dubinsku mapu prostora ispred senzora, te na taj način može razlikovati predmete od pozadine.

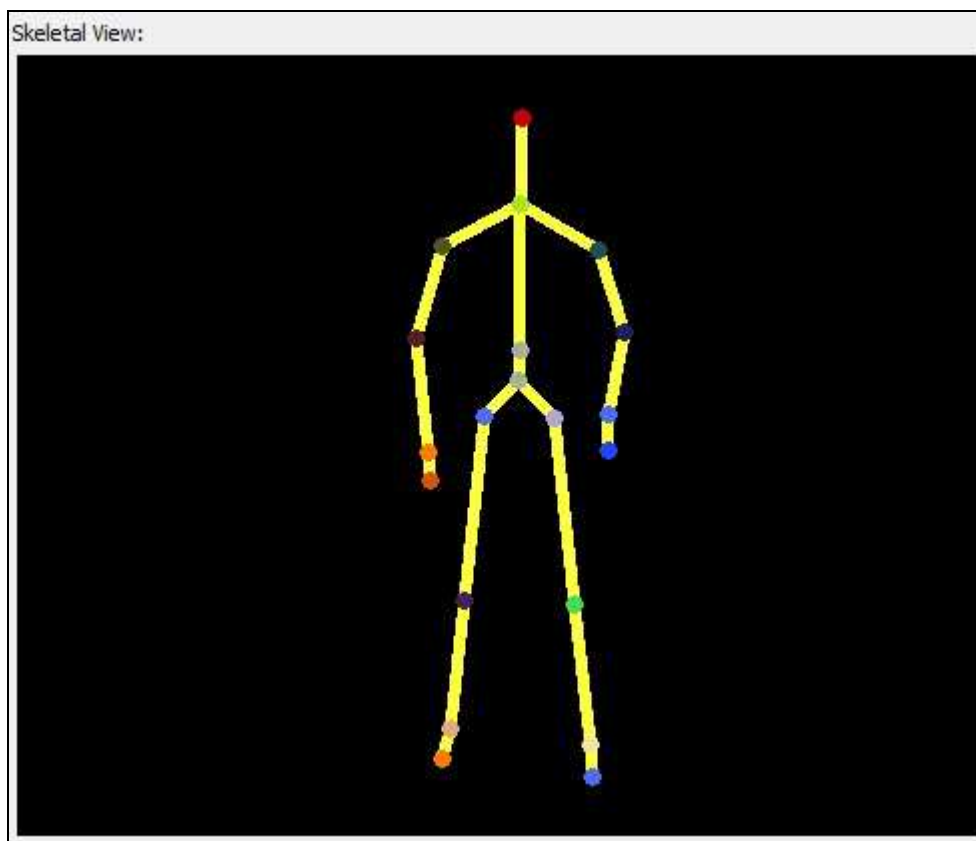


Slika 1.3: Dubinska mapa [4]

Dubinska mapa koju je snimio Kinect vidljiva je na slici (Slika 1.3). Kako bi odredio dubinu pojedinih dijelova slike Kinect koristi metodu strukturiranog svjetla u kombinaciji s dvije metode računalnog vida: određivanje dubine putem fokusa i putem stereo vida [5,6]. Detalji implementacije nisu poznati, no moguće je prezentirati generalnu ideju koja stoji u pozadini:

- Strukturirano svjetlo – tehnika se zasniva na projiciranju točkastog uzorka svjetlosti u prostor i određivanja dubinske komponente slike na osnovu deformacije očitano uzorka.
- Određivanje dubine putem fokusa – bazira se na opservaciji da će predmeti koji su udaljeniji od senzora biti mutniji od onih koji su bliže.
- Određivanje dubine putem stereo vida – promjenom točke gledišta na neku scenu mijenja se pozicija predmeta na slici. Što su predmeti bliži senzoru, pomak je izraženiji.

Nakon generiranja dubinske mape Kinect SDK njihovom obradom prepoznaje tijela osoba koje stoje ispred senzora i formira „kosture“ koji služe za praćenje pokreta korisnika.

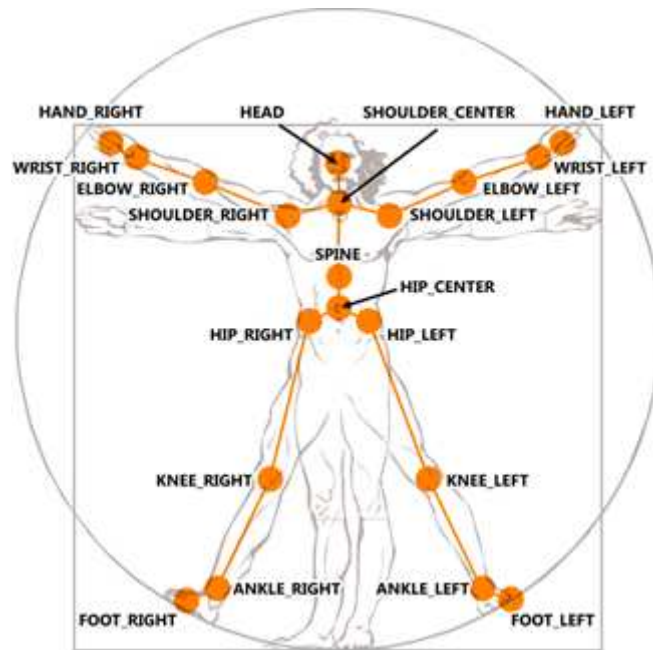


Slika 1.4: Prikaz kostura osobe [7]

Primjer programa koji iscrtava „kostur“ čovjeka prikazan je na slici (Slika 1.4). Iako je taj kostur samo aproksimacija dobivena obradom podataka senzora, efikasno se može koristiti za praćenje pokreta korisnika. Na taj način je moguće implementirati prepoznavanje gesti i poza, te ih iskoristiti u aplikacijama ili igrama umjesto uobičajenog ulaza koji bi dolazio s tipkovnice, miša ili kakve druge naprave koja zahtjeva fizički kontakt s korisnikom. U radu je prepoznavanje gesti implementirano spomenutom tehnikom.

Na slici su vidljivo istaknuti neki čvorovi. Ti čvorovi predstavljaju karakteristične točke kostura koje je Kinect prepoznao koristeći interne algoritme.





Slika 1.5: Karakteristične točke kostura [8]

Karakteristične točke kostura koje Kinect prepoznaje su naznačene na slici (**Error! Reference source not found.**). Ukupno ih je dvadeset. Svaka točka u sebi sadrži nekoliko podataka [9]:

- Poziciju u trodimenzionalnom prostoru – uređena trojka x, y i z koordinata
- Identifikator o kojoj točki se radi – lijevo koljeno, glava i sl.
- Stanje praćenja – procjena uređaja Kinect u kakvom stanju su podaci o toj točki, odnosno jesu li pouzdani („Tracked“), nepouzđani („Inferred“) ili nema podataka o točki („NotTracked“)

Kinect također vodi računa o pouzdanosti podataka koje generira. Ako iz određenih razloga postoji vjerojatnost da izračunate koordinate točke nisu točne ili uopće nisu registrirane, što može biti slučaj ako se dio tijela korisnika nalazi izvan vidnog polja uređaja.

Kratki opis ove metode dan je u članku [1]: „Praćenje skeletona je razvijeno nakon mnogo godina istraživanja i proučavanja ljudske anatomije. Program je u mogućnosti prepoznati karakteristične oblike pojedinih dijelova tijela poput šake, ruke ili noge, te njihove karakteristične pokrete. Ljudsko tijelo ima definirani broj zglobova i svaki od njih može izvesti specificirani set radnji za koje Kinect SDK izračunava poziciju i daje aplikaciji vrijednosti kako bi izvršila odgovarajuću akciju.“.

## 2. Interakcija korisnika i aplikacije

Sučelje aplikacije temeljno može biti oblikovano na dva osnovna načina: kao tekstualno sučelje i kao grafičko sučelje. S razvojem računalne grafike kod korisničkih aplikacija prevladala je uporaba grafičkih sučelja koja omogućuju prirodnu interakciju korisnika s aplikacijom kroz ustaljene elemente koje možemo naći na gotovo svim sučeljima: gumbi, klizači, tekstualni prozori, padajuće izbornike, alatne trake i sl. Iako je kroz povijest razvoja grafičkih sučelja vidljiv napredak u njihovoj kvaliteti i dizajnu, dosta korisničkih aplikacija uglavnom koristi periferiju koja postoji već dugo: tipkovnicu i miš. Izvjesna novina u tom pogledu, barem u domeni široke rasprostranjenosti, jest tehnologija ekrana osjetljivih na dodir koja uklanja posrednika između korisnika i elemenata sučelja. Korisnik dolazi u direktan kontakt sa sučeljem i samo sučelje postaje periferija. Na taj način je korisniku pružena mogućnost interakcije na prirodniji način uz smanjenu ili u potpunosti eliminiranu potrebu za dodatnim napravama. Od toga možemo otići još korak dalje i istražiti mogućnost potpunog izbacivanja bilo kakvih perifernih uređaja, uključujući i ekran. Korisnik pritom upravlja aplikacijom koristeći isključivo svoje tijelo i glas, a interakcija više nije ograničena karakteristikama perifernog uređaja, već sve kretnje i govor postaju potencijalne ulazne podatke za aplikaciju.

### 2.1. Prirodna interakcija

U razvoju sučelja aplikacija postoji težnja za pojednostavljenjem komunikacije između korisnika i aplikacije. Grafička sučelja pružaju intuitivniji način interakcije od tekstualnih i ostavljaju mnogo slobodnog prostora za razvoj tehnologija (kao što je su to ekrani osjetljivi na dodir) koje će tu interakciju učiniti zanimljivijom i jednostavnijom, a samim time i „prirodnijom“.

Kada govorimo o prirodnoj interakciji javlja se potreba za izvjesnom definicijom što taj pojam zapravo predstavlja. Budući da se radi o području koje je kao koncept još u razvoju, ne možemo ga vezati isključivo uz određeni skup tehnologija, metodologija ili paradigmi. Stoga možemo reći da pod pojmom prirodne interakcije korisnika i aplikacije podrazumijevamo sve tehnologije i metode koje na neki način uklanjaju barijere između korisnika i aplikacije, odnosno čine korištenje aplikacije intuitivnijim, oslanjajući se pritom više na korisnika a

manje na popratne uređaje. Jedan od načina na koji to možemo ostvariti je da prekinemo bilo kakav fizički kontakt između korisnika i računala. Tada jedino ograničenje u interakciji postaje sam korisnik, odnosno njegove motoričke i govorne sposobnosti (što je u krajnjoj liniji neizbježno). Mogućnosti postaju teoretski neograničene.

Neki od elemenata koji mogu biti od značaja prilikom ostvarivanja prirodnije interakcije su:

- Prepoznavanje govora
- Praćenje pogleda
- Tjelesne geste
- Prepoznavanje izraza lica

Područje od interesa vrlo brzo možemo proširiti do nepraktičnih razmjera budući da se radi o temi koja je, kako je ranije navedeno, predmet intenzivnog istraživanja. Iz tog razloga u daljnjim razmatranjima ograničit ćemo se na tjelesne pokrete korisnika kao primarni medij ostvarivanja prirodne interakcije.

## **2.2. Upravljanje gestama**

Za ostvarivanje bilo kakve interakcije s aplikacijom potreban je nekakav okidač, odnosno u grubom smislu signal od strane korisnika koji je moguće prepoznati. Općenito govoreći, tjelesnom gestom možemo smatrati bilo što: od mahanja rukom, držanja za glavu do raznih tjelesnih ekspresija, što statičnih, što pokretnih. U toj domeni izazov postaje osmisliti i implementirati prepoznavanje onih gesti koje će omogućiti što intuitivniji odnos korisnika i elemenata na sučelju. Budući da je svatko individua za sebe, svatko ima vlastitu percepciju sučelja, procjenu koje interakcije bi bile moguće s ponuđenim elementima, koje su komponente uopće interaktivne i ideju kako postići željeni rezultat. Iako zadiranjem u ovo područje potencijalno povećavamo mogućnosti i kvalitetu interakcije, također nailazimo na nove probleme. Kako korisniku koncizno dati do znanja što se od njega očekuje i koje su sve interakcije moguće sa elementima na sučelju? Koji su elementi pasivni a koji aktivni? Na koji način možemo korisniku omogućiti neku funkcionalnost koja nebi bila moguća putem drugih perifernih uređaja?

U okviru ovog rada gestom smatramo bilo kakav tjelesni pokret ili statičnu pozu korisnika. Pritom geste koje se prepoznaju kao određeni položaj tijela zovemo statičnima, a geste koje karakterizira gibanje tijela korisnika za vrijeme trajanja interakcije nazivamo dinamičnim

gestama. U projektu koji je uključen u rad obrađeno je nekoliko primjera dinamičnih i jedna statična gesta.

## **2.3. Dinamične i statične geste**

Ako gestu promatramo kao okidač neke radnje u programu, kao što je na primjer otvaranje padajućeg izbornika ili pomicanje klizača na tekstualnom okviru, možemo ih podijeliti u dvije osnovne grupe ovisno o tome izazivaju li samo trenutnu akciju u programu (kao što je zatvaranje prozora) ili se odvijaju za vrijeme trajanja interakcije (kao što je pomicanje klizača na sučelju pokretom ruke). Prvu grupu nazivamo statičnim gestama, a drugu dinamičnim gestama.

Prepoznavanje statičnih gesti je lakše implementirati u aplikaciji, no pružaju manje mogućnosti od dinamičnih. S druge strane, dinamičke geste su zahtjevnije glede performansi. Ako promatramo podatke koje dobivamo iz uređaja Kinect u svrhu prepoznavanja gesti, radi se o kontinuiranom toku podataka koji dolazi brzinom od trideset „slika“ u sekundi, gdje svaka slika predstavlja strukturirani blok podataka koji sadrži pozicije karakterističnih točaka kostura koje je Kinect prepoznao. Ovisno o veličini baze podataka koja sadrži poznate geste duljina obrade može varirati, iako je u principu zanemariva s obzirom na današnje snage procesora.

### **3. Aplikacija elzlog**

Aplikacija elzlog demonstrira koncepte obrađene u ovom radu. Izrađena je u okruženju Microsoft Visual Studio 2012 koristeći programski jezik C#. Dodatni alati koji su korišteni prilikom izrade aplikacije su: Microsoft Expression Blend + Sketch Flow for Visual Studio 2012, Kinect for Windows SDK 1.7, Kinect for Windows Developer Toolkit i Helix 3D Toolkit.

Aplikacija je zamišljena kao interaktivni računalni katalog proizvoda. Korisniku omogućava odabir između nekoliko ponuđenih kategorija proizvoda te ih prikazuje na ekranu: njihov izgled na nekoliko različitih slika, kratak opis i trodimenzionalni prikaz.

#### **3.1. Korištena tehnologija**

Aplikacija je napisana u jeziku C# i XAML (Extensible Application Markup Language) i namijenjena je pokretanju na Microsoft WPF (Windows Presentation Foundation) platformi uz .NET Framework 4.5. Kod aplikacije sastoji se od dijelova pisanih u jezicima C# i XAML. XAML je derivat jezika XML koji koristi platforma WPF za kreiranje i inicijalizaciju elemenata svake WPF aplikacije.

Microsoft Expression Blend je besplatna aplikacija koja je korištena za animaciju elemenata sučelja u WPF aplikacijama.

Microsoft Kinect for Windows SDK službeni je paket alata i biblioteka namijenjenih razvoju aplikacija koje koriste uređaj Kinect. Kinect for Windows Developer Toolkit skup je biblioteka koje olakšavaju osnovne operacije nad uređajem Kinect, kao što je inicijalizacija uređaja, odabir aktivnog uređaja u slučaju da je na računalo spojeno više njih, emulacija miša i sl.

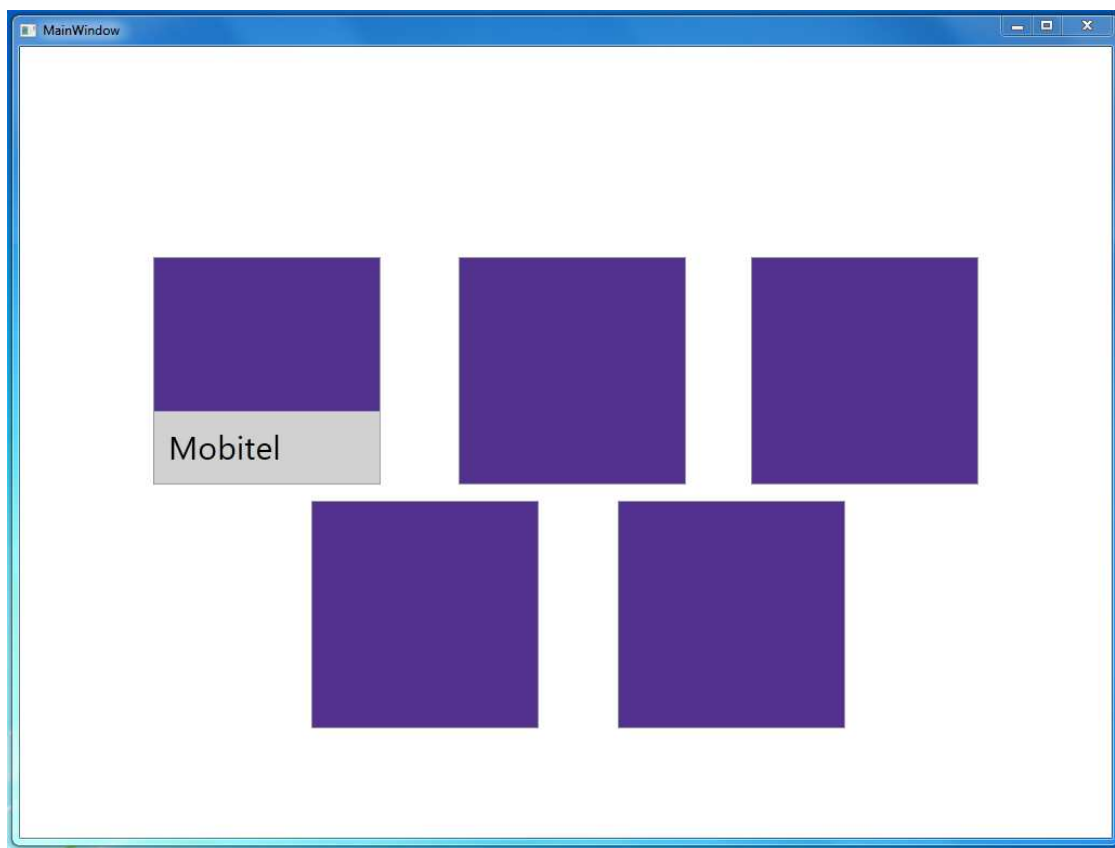
Helix 3D Toolkit je biblioteka otvorenog koda koja je korištena samo za učitavanje trodimenzionalnih modela u aplikaciju i njihov prikaz.

## 3.2. Motivacija za izradu aplikacije

Kako što je već ranije navedeno prirodna interakcija čovjeka s računalom relativno je svjež pojam na području računalne industrije i kao takav nije veoma striktno definiran, te kao takav pruža gotovo neograničene mogućnosti inovacije. Tehnologija koja podupire takav pristup interakciji još je uvijek u razvoju. S obzirom na okolnosti izazov prilikom pristupa ovoj temi bio je osmisliti relativno jednostavnu aplikaciju koja bi mogla poslužiti kao demonstracija nekih koncepata prirodne interakcije korisnika s računalom koji bi mogli biti relevantni za daljnji razvoj i nadograđivanje. Bilo je potrebno istražiti potencijal ovakvog pristupa upravljanju aplikacijom te istaknuti neke njegove prednosti i nedostatke. Imajući u vidu navedene potrebe razvijena je aplikacija „eIzlog“ koja implementira koncepte prirodne interakcije čovjeka s računalom na način koji ima potencijala za primjenu u praksi.

Aplikacija je sama po sebi prilično jednostavna, budući da se radi o interaktivnom katalogu proizvoda, no poslužila je kao istraživački primjer na području ostvarivanja interakcije korisnika sa računalom na prirodniji način koji ne uključuje nikakva pomagala osim vlastitog tijela korisnika.

### 3.3. Izgled aplikacije



Slika 3.1 : Glavni prozor aplikacije

Glavni prozor aplikacije eIzlog prikazan je na slici (Slika 3.1 : Glavni prozor aplikacije). Prozor se sastoji od pet gumba koji predstavljaju različite kategorije proizvoda koje korisnik može odabrati. Klikom miša, odnosno u slučaju korištenja uređaja Kinect, gestom pritiska ruke na određenom gumbu korisniku se otvara slijedeći prozor koji prikazuje dostupne proizvode te kategorije. Broj gumba na zaslonu je proizvoljan i sa slike je vidljivo da nisu svi gumbi funkcionalni budući da je naglasak stavljen da demonstraciju mogućnosti uređaja Kinect za ostvarivanje prirodne interakcije s grafičkim objektima, a ne na samu funkcionalnost kataloga.



Slika 3.2: Prikaz kataloga

Izgled kataloga prikazan je na slici (Slika 3.2: Prikaz kataloga). Ovaj prozor predstavlja glavni dio aplikacije, odnosno pregledavanje samog kataloga. Na slici su prikazana samo četiri elementa ilustracije radi, a veličina stvarnog kataloga može biti proizvoljna. Kada broj proizvoda u prozoru postane prevelik kako bi bio prikazan na ekranu, na prozoru se pojavljuje horizontalni klizač (kako je uobičajeno u aplikacijama na korisničkim računalima) koji omogućava korisniku pregledavanje više elemenata odjednom.

Moguće interakcije u ovom prozoru su pritisak gumba jednostavnim pokretom dlana (kao da korisnik stvarno pokušava pritisnuti gumb rukom) i navigacija kroz listu putem klizača na način da korisnik stiskom šake „uhvati“ bilo koji dio prozora i pomicanjem ruke povlači njegov sadržaj lijevo ili desno. Također je moguće uhvatiti prozor, te nakon zamaha rukom otvoriti šaku i „baciti“ sadržaj prozora u određenu stranu. Sadržaj će se tada nastaviti gibati i polako usporavati dok u potpunosti ne stane. Gibanje je moguće prekinuti i prije nego samo od sebe prestane ponovnim stiskom šake.

Podrška za ove geste je implementirana u paketu Microsoft Kinect Developer Toolkit. Preko cijelog glavnog prozora je rastegnuta kontrola „KinectRegion“ omogućava praćenje dlana



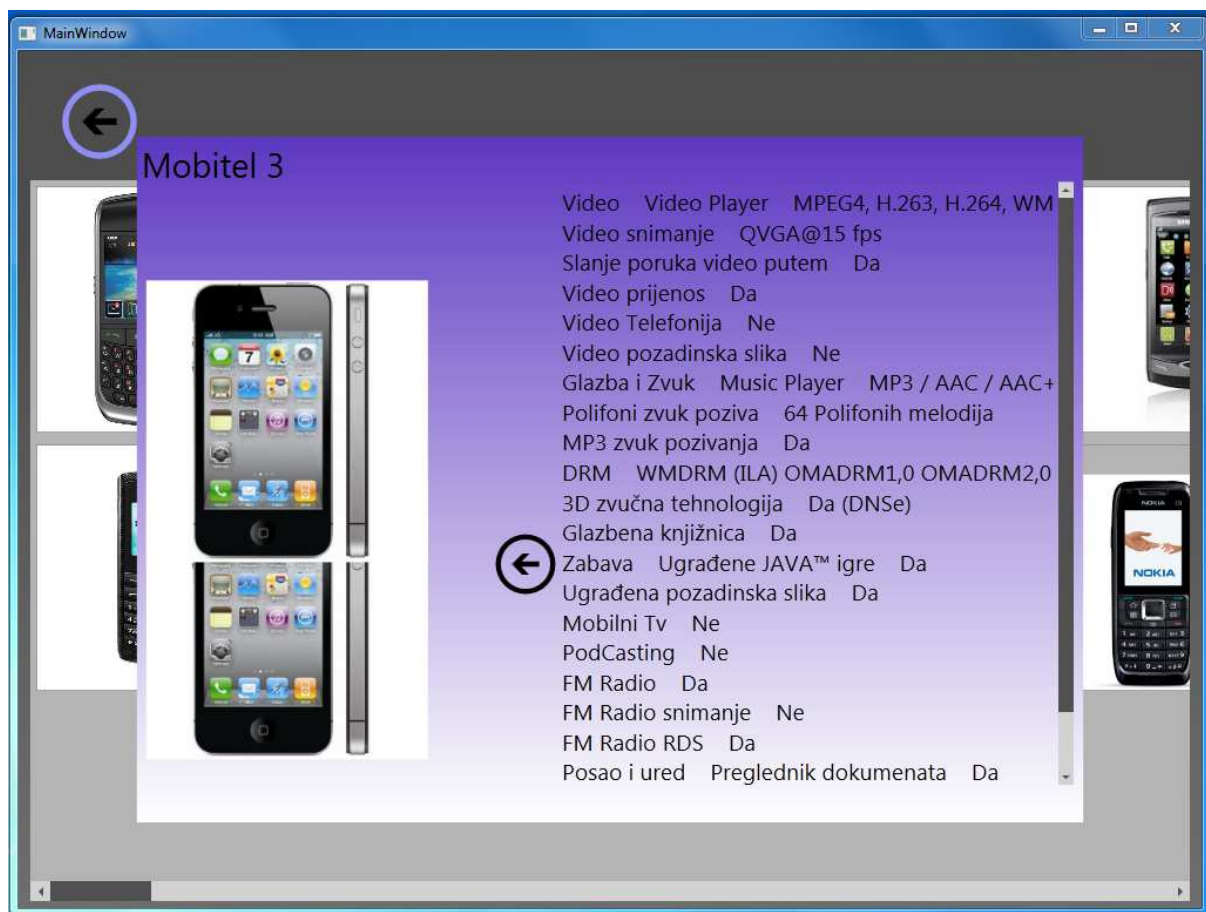
jedne ruke korisnika i iscrtavanje pratećeg kursora na ekranu. Pritom su svi elementi sučelja zasebni objekti koji implementiraju vlastite metode za obavljanje svojih specifičnih funkcija. Svaki element koji je „zainteresiran“ za signalizaciju koja dolazi od senzora prijavljuje klasi KinectRegion svoju metodu koja će biti korištena za obradu događaja koji se odnosi na pokrete dlana korisnika. Moguće je registrirati metode koje će biti pozvane kada npr. kursor prijeđe preko određenog elementa sučelja, kada ga korisnik pritišće dlanom, kada korisnik stisne šaku i sl. Svaki gumb sučelja pripada nekoj klasi koja je izvedena iz klase KinectButtonBase koja prilikom svoje inicijalizacije poziva slijedeću metodu:

```
private void InitializeKinectButtonBase()
{
    KinectRegion.AddHandPointerPressHandler(this,
this.OnHandPointerPress);
    KinectRegion.AddHandPointerGotCaptureHandler(this,
this.OnHandPointerCaptured);
    KinectRegion.AddHandPointerPressReleaseHandler(this,
this.OnHandPointerPressRelease);
    KinectRegion.AddHandPointerLostCaptureHandler(this,
this.OnHandPointerLostCapture);
    KinectRegion.AddHandPointerEnterHandler(this,
this.OnHandPointerEnter);
    KinectRegion.AddHandPointerLeaveHandler(this,
this.OnHandPointerLeave);

    KinectRegion.SetIsPressTarget(this, true);
}
```

Na taj način svaki gumb prijavljuje klasi KinectRegion metode koje obrađuju događaje kada kursor uđe u područje gumba ili kada korisnik dlanom pritišće gumb. Slično je i s klasom KinectScrollView koja u sebi može sadržavati element koji je veći od nje same i prikazivati ga djelomično na način da se dio koji je prikazan regulira pomoću klizača.

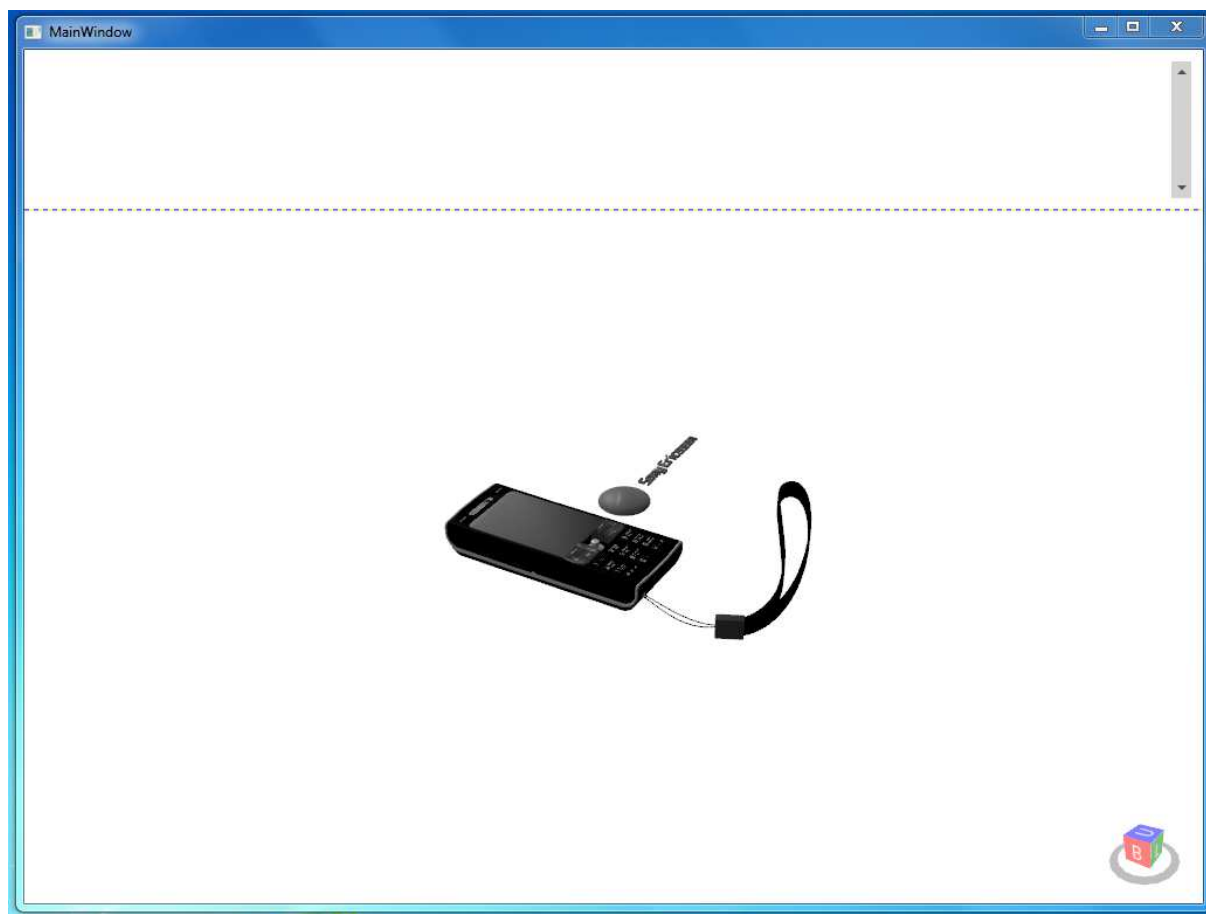
Klasa KinectRegion zadužena je za procesuiranje podataka koji dolaze iz senzora Kinect i pozivanje prijavljenih metoda kada nastupe uvijeti za koje je pojedina metoda prijavljena. Pritom koristi podatke o kosturu koje generira Kinect.



Slika 3.3 : Prikaz podataka o mobitelu

Pregled podataka o proizvodu prikazan je na slici (Slika 3.3 : Prikaz podataka o mobitelu). U ovom prozoru je moguće pregledavati tekstualne podatke o mobitelu na način da korisnik pozicionira kursor bio gdje na tekstualnom okviru, stisne šaku i pokretima ruke prema gore ili prema dolje pomiče tekst. Također je moguće otvoriti prozor s trodimenzionalnim prikazom objekta na način da korisnik pozicionira kursor na sliku proizvoda, ispruži ruku ispred sebe, stisne šaku kao da hvata sliku i povuče je prema sebi.

Gesta otvaranja trodimenzionalnog prikaza objekta implementirana je koristeći neke metode koje su ugrađene u Microsoft Kinect Developer Toolkit. Moguće je za svaki element sučelja definirati metode koje se pozivaju kada kursor uđe u područje koje pripada objektu, kada se po njemu pomiče i sl. U slučaju otvaranja trodimenzionalnog prikaza



Slika 3.4 : Trodimenzionalni prikaz objekta

Mogućnosti interakcije s trodimenzionalnim objektima prikazan je na slici (Slika 3.4 : Trodimenzionalni prikaz objekta). U ovom prozoru korisnik može pregledavati trodimenzionalni objekt, odnosno proizvod koji je odabrao. Objekt je moguće rotirati na sličan način kao što je na prethodnom ekranu bilo ostvareno pregledavanje sadržaja pomicanjem ruke. Korisnik stiskom šake može „uhvatiti“ model i rotirati ga slobodnim pomicanjem ruke u prostoru.

Objekt je također moguće približiti i udaljiti, odnosno uvećati i umanjiti njegov prikaz. To se ostvaruje koristeći obje ruke. Korisnik pritom mora ispružiti obje ruke ispred sebe i međusobno udaljavati ili skupljati dlanove kao da „pliva“ prema objektu ili od objekta. Ako korisnik dlanove odmiče jedan od drugog objekt se približava, a ako ih skuplja objekt se udaljava.

Spomenute interakcije ostvarene su koristeći mogućnosti prepoznavanja kostura koje su objašnjene u prvom poglavlju. Kinect for Windows SDK omogućava programeru pristup strukturiranim podacima koji predstavljaju kostur koji je odredio Kinect. U aplikaciji postoji metoda koja se poziva svaki put kada je spreman novi skup podataka o kosturu, odnosno nova

„slika“ ili „frame“ (Kinect te podatke šalje brzinom od trideset slika u sekundi). Svaka „slika“ sadrži podatke o pozicijama svih čvorova kostura kako je objašnjeno u potpoglavlju 1.2: Praćenje kostura.

Interakcija uvećavanja i umanjivanja objekta ostvarena je prateći položaj obje šake korisnika. Prilikom računanja udaljenosti različitih točaka uzeta je relativna mjerna jedinica, odnosno udaljenost od dna vrata pa do vrha glave, kako bi bili postignuti bolji rezultati s kosturima različitih proporcija. Sve udaljenosti s kojima barata aplikacija izražene su u tim jedinicama. Ako korisnik ispruži obje ruke ispred sebe za određeni broj jedinica (3.5 u ovom slučaju), aplikacije to prepozna, aktivira takozvani „zoom“ i nastavlja pratiti pozicije dlanova korisnika. Objekt približavamo ako se dlanovi odmiču jedan od drugog, a ako se primiču udaljavamo ga. Izvorni kod koji obavlja ove operacije prikazan je u nastavku:

```
private void SkeletonInput(Skeleton s)
{
    double zoomDistance = 3.5f, zoomFactor = 1.4;

    // Odredimo pozicije svih čvorova koji nas interesiraju
    SkeletonPoint leftHand = s.Joints[JointType.HandLeft].Position;
    SkeletonPoint rightHand = s.Joints[JointType.HandRight].Position;
    SkeletonPoint head = s.Joints[JointType.Head].Position;
    SkeletonPoint neck = s.Joints[JointType.ShoulderCenter].Position;
    // Relativna "jedinica", udaljenost od dna vrata do vrha glave
    unit = Dist(neck, head);

    // Ako se oba dlana nalaze za više od 3.5 jedinica ispred baze
    // kostura (kao baza je uzet vrat), pokrećemo zoom
    if ((ZUnitDist(neck, leftHand) > zoomDistance) &&
        (ZUnitDist(neck, rightHand) > zoomDistance) &&
        (zoom == false))
    {
        zoom = true;
        lastDistance = Dist(leftHand, rightHand);
    }

    // Ovdje zumiramo objekt
    if (zoom)
    {
        // Ako korisnik privuče bilo koju ruku, zumiranje
        // se prekida
        if ((ZUnitDist(neck, leftHand) < zoomDistance) ||
            (ZUnitDist(neck, rightHand) < zoomDistance))
            zoom = false;

        double newDistance = Dist(leftHand, rightHand);
        double delta = newDistance - lastDistance;
        // Objekt zumiramo ovisno o promjeni udaljenosti
        // dlanova od prošlog poziva funkcije
        view.CameraController.AddZoomForce(-delta / zoomFactor);
        lastDistance = newDistance;
    }
}
```

## 4. Analiza programskog produkta

Razvijena aplikacija eIzlog relativno je jednostavna glede funkcionalnosti koju pruža: pregledavanje interaktivnog kataloga koji sadrži slike, kratak opis i trodimenzionalni model svakog proizvoda. Naglasak prilikom izrade aplikacije bio je na istraživanju različitih mogućnosti primjene uređaja Microsoft Kinect za ostvarivanje što prirodnije i intuitivnije interakcije između korisnika i elemenata sučelja. U ovom poglavlju biti će istaknuti izazovi, prednosti i nedostaci ovakvog pristupa upravljanju aplikacijom iz perspektive programera i krajnjeg korisnika.

### 4.1. Prirodna interakcija u praksi

Prirodna interakcija kao pojam zvuči interesantno, međutim prilikom njegovog razmatranja moramo u obzir uzeti vrlo važnu činjenicu, a to je da svaki pojedinac ima svoju definiciju riječi „prirodno“. Što je za jednu osobu prirodno, za drugu je neprirodno. Cilj primjene navedenih koncepata je postići što jednostavnije sučelje za korisnika koje od njega zahtjeva minimalno vrijeme prilagodbe i omogućiti mu da se fokusira na rezultat koji želi postići, a ne na logiku razvojnog tima aplikacije kojom su prožeti svi elementi sučelja. Međutim, problem ovakvog pristupa je što još uvijek ne postoji jasna ideja što bi zapravo bilo prirodno, odnosno kako programer može korisniku dati do znanja što se od njega očekuje, koji su elementi sučelja interaktivni a koji nisu. Ako sučelje gradimo na način da pokušavamo elemente učiniti sličnijima klasičnim elementima koji se koriste na sučeljima modernih aplikacija kao što su to gumbi, padajući izbornici, tekstualni okviri i sl., ograničavamo potencijal novih tehnologija jer cijelo sučelje stavljamo u već postojeći okvir koji karakteriziraju klikovi mišem i unos teksta na tipkovnici. Dobivena promjena nebi bila gotovo nikakva. Jedini rezultat biti će promjena ulaznog uređaja: istu stvar koju je korisnik do nedavno radio mišem sada radi vlastitom rukom. Razlika je dakako samo kozmetička. Ako pak s druge strane krenemo s osmišljavanjem potpuno novih elemenata koji zahtijevaju bitno različiti način interakcije rezultati će se ponovo kositi s osnovnom idejom prirodne interakcije, a to je intuitivnost. U konačnici moramo pronaći balans između oponašanja „starih“ elemenata i uvođenja nove funkcionalnosti kako bi postigli željeni rezultat i sučelje učinili atraktivnijim i lakšim za korištenje.

S druge pak strane, čak i „zlatna sredina“ sa sobom nosi izvjesne probleme od kojih je glavni ograničenje senzora. Uređaj Kinect koji je korišten prilikom razvoja aplikacije namijenjen je konzoli Xbox 360 i kao takav nije predviđen za preciznu detekciju pokreta i analizu izraza lica ili položaja prstiju ruke. S te strane ograničava mogućnosti gestikulacije na grublje pokrete tijela i eventualnu detekciju jeli dlan otvoren ili skupljen u šaku. Inačica uređaja za osobna računala postoji, no nije u raširenoj primjeni.

S razvojem tehnologije možemo očekivati poboljšanje u preciznosti senzora, a samim time i veće mogućnosti primjene takvih tehnologija u razvoju aplikacija. Zasad je glavni adut ovakvog pristupa ipak vizualna atraktivnost.

## 4.2. Performanse aplikacije elzlog

Aplikacije je testirana na računalu HP 530 sa slijedećom konfiguracijom (navedeni su samo osnovni parametri):

Tabela 4.1 : Konfiguracija računala

|                           |                                                                    |
|---------------------------|--------------------------------------------------------------------|
| <b>Procesor</b>           | Intel Celeron Processor 530 (1.73-GHz, 533-MHz FSB, 1-MB L2 cache) |
| <b>Radna memorija</b>     | 1536-MB DDR2 SDRAM                                                 |
| <b>Operacijski sustav</b> | Micfosoft Windows 7                                                |
| <b>Grafička kartica</b>   | Intel Graphics Media Accelerator 950                               |

U nastavku je prezentirana kratka analiza iskustvene kvalitete aplikacije:

- Sama inicijalizacija senzora utječe na performanse aplikacije i produljuje vrijeme njenog pokretanja za oko tri sekunde, što je relativno puno s obzirom da se inače pokreće gotovo trenutno. Produljenje vremena pokretanja iznosi oko 150%.
- Korištenje senzora Kinect za interakciju sa sučeljem također narušava performanse aplikacije. Interakcije s gumbima i klizačima su implementirane u Microsoft Kinect Developer Toolkit-u usporavaju rad aplikacije u odnosu na situaciju u kojoj se koristi samo miš i tipkovnica. Cursor uređaja Kinect na ekranu često trza, što je primjetna razlika u odnosu na korištenje miša za pomicanje kursora. Pomicanje prozora s klizačem također je zamjetno sporije.

- Od svih elemenata trodimenzionalni prikaz objekta najviše utječe na performanse. Čak i kada se nad njim ne vrši nikakva interakcija prozor HelixViewport3D (element koji je korišten za prikaz trodimenzionalnih modela) bitno usporava rad aplikacije do te mjere da opći dojam koji pruža ta funkcionalnost postaje relativno loš.
- Korištenje .NET i WPF okruženja također utječu na performanse jer dodaju još jedan posredni sloj između procesora i aplikacije, a to je virtualizacija na CLR virtualnom stroju (Common Language Runtime).

Korištenje senzora Kinect generalno pogoršava performanse aplikacije, iako ne toliko drastično kao trodimenzionalni prikaz. Helix 3D Toolkit izgrađen je na bazi standardnih WPF klasa i samim time ne uzima mnogo dodatnog procesorskog vremena u odnosu na standardne klase, što govori u prilog tome da platforma WPF nije pogodna za trodimenzionalni prikaz objekata.

U konačnici interakcija s trodimenzionalnim objektima ima veći utjecaj na performanse aplikacije od korištenja Kinecta, iako ni zamjena klasične periferije s Kinectom nije zanemariva. Samo prepoznavanje gesti preko podataka o kosturu korisnika nije zahtjevno jer se radi o malom skupu podataka, točnije o kolekciji od šesto čvorova po sekundi za svakog korisnika. S obzirom na današnje procesorske snage i ograničenje Kinecta koji može istovremeno analizirati samo dva korisnika, dok za ostala četiri može samo odrediti poziciju osobe u prostoru, procesuiranje podataka ne zahtjeva puno vremena i primjenjivo je u aplikacijama koje rade u stvarnom vremenu.

## Zaključak

U radu je istražen i razrađen koncept prirodne interakcije korisnika s grafičkim objektima. Dana je jednostavna definicija prirodne interakcije kao i opis što ona sve uključuje. Opisana je jednostavna aplikacija eIzlog koja sadrži elemente prirodne interakcije s korisnikom kroz upravljanje grafičkim objektima putem tjelesnih gesti, kao i motivacija za njenu implementaciju.

Dat je kratak pregled mogućnosti uređaja Kinect koji je korišten u radu kao i osnovna teoretska podloga analize pokreta tijela koja je implementirana na njemu. Opisane su neke interne strukture podataka korištenih alata koje su bitne za obrađenu temu.

Na kraju je analizirana rezultirajuća programska potpora. Navedene su neke prednosti i nedostaci dobivenog produkta i samog pristupa problemu, mogućnosti za daljnju nadogradnju i dat je kratak osvrt na sveukupni dojam aplikacije i implementirane tehnologije.



# Literatura

- [1] Bronzin, Tomislav; Curić, Ivan: „Pregled mogućnosti razvoja Kinect aplikacija za osobna računala“, pdf izdanje, Zagreb, 2012.
- [2] <http://upload.wikimedia.org/wikipedia/commons/6/67/Xbox-360-Kinect-Standalone.png>, 22.05.2013.
- [3] <http://files.channel9.msdn.com/wlwimages/9c00b398b405423b99d19efa016fae96/image%5B7%5D-1.png>, 22.05.2013.
- [4] <http://www.brekel.com/wp-content/uploads/2010/12/kinect-3D-scanner-capture-infrared.jpg>, s interneta, 23.05.2013.
- [5] MacCormick, John: How does Kinect work?, s interneta <http://users.dickinson.edu/~jmac/selected-talks/kinect.pdf>, 22.05.2013.
- [6] [http://www.zemris.fer.hr/predmeti/ra/predavanja/2\\_scan.pdf](http://www.zemris.fer.hr/predmeti/ra/predavanja/2_scan.pdf), s interneta, 22.05.2013.
- [7] <http://blog.cse-net.co.uk/wp-content/uploads/2011/07/KinectSkel.jpg>, s interneta, 25.05.2013.
- [8] <http://i.msdn.microsoft.com/dynimg/IC534688.png>, s interneta, 13.06.2013.
- [9] [http://msdn.microsoft.com/en-us/library/microsoft.kinect.joint\\_members.aspx](http://msdn.microsoft.com/en-us/library/microsoft.kinect.joint_members.aspx), s interneta, 13.06.2013.

# Sažetak

## Prirodna interakcija korisnika grafičkim objektima

U ovom radu je obrađena tema prirodne interakcije korisnika s elementima grafičkog sučelja. Definiran je pojam prirodne interakcije. Navedene su mogućnosti uređaja Microsoft Kinect za detekciju pokreta korisnika kao i primjena tih mogućnosti u ostvarivanju prirodne interakcije korisnika s računalom.

Opisana je aplikacija eIzlog koja demonstrira koncepte prirodne interakcije u primjeni, analizirani su dobiveni rezultati i performanse aplikacije uz kratak osvrt na ukupni dojam.

**Ključne riječi:** kinect, prirodna interakcija, grafički objekti

# **Abstract**

## **Natural user interaction on graphical objects**

In this paper the topic of natural user interaction with the elements of the graphical user interface is analyzed. The concept of natural interaction is defined. The capabilities for motion detection of Microsoft Kinect as well as a number of applications of those capabilities in user-computer interaction are described.

The program elzlog that demonstrates the concepts of natural interaction is presented, an analysis of the achieved results and program performance is given along with a brief overview of the overall impression.

**Keywords:** kinect, natural interaction, graphical objects

## Uputa za instalaciju

Na priloženom CD-u se nalazi mapa „Instalacija“. Sadržaj tog direktorija potrebno je kopirati na proizvoljno mjesto na tvrdom disku i pokrenuti „KinectRuntime-v1.7-Setup.exe“ kako bi se instalirali programski paketi za rad sa Microsoft Kinectom. Također je potrebno preuzeti i instalirati Microsoft .NET Framework 4.5 ako već nije instaliran na računalu (<http://www.microsoft.com/en-us/download/details.aspx?id=30653>).

Nakon toga potrebno je pokrenuti „eIzlog“ dvostrukim klikom na ikonu. Program je namijenjen pokretanju na operacijskom sustavu Microsoft Windows 7 i Microsoft Windows 8.