

Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.

# **Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta**

## **Tehnička dokumentacija Verzija 2.0**

**Studentski tim:** Adnan Abdagić  
Nikola Martinec  
Petar Mrazović  
Ana Nekić  
Marko Pilipović  
Mario Volarević

**Nastavnik:** Prof. dr. sc. Željka Mihajlović

Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.

## Sadržaj

1. Opis razvijenog proizvoda	3
2. Korišteni alati	3
2.1 Kinect	3
2.2 Microsoft XNA 4.0	3
3. Tehničke značajke	4
3.1 Iscrtavanje scene	4
3.2 Detekcija šaka	4
3.3 Geste	6
3.4 Skaliranje	6
3.5 Rotiranje	6
3.6 Ostvarivanje sučelja	7
3.7 Animacije	7
3.8 Interakcija sa sučeljem	7
3.9 Rastavljanje modela	8
4. Algoritam ekspanzije	9
4.1 Implementacija	10
5. Upute za korištenje	13
6. Zaključak i budući razvoj	13
7. Literatura	13

Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.

# Tehnička dokumentacija

## 1. Opis razvijenog proizvoda

Rezultat projekta je program koji predstavlja prototip interaktivnog priručnika za održavanje. Implementirana je interakcija s Microsoft modelom uz korištenje prirodnog sučelja (engl. *Natural User Interface*), tj. kontrola je ostvarena gestama ruku i tijela, a kao uređaj za detekciju je korišten Microsoft Kinect.

## 2. Korišteni alati

### 2.1 Kinect

Kinect predstavlja ulazni uređaj za očitavanje pokreta, namijenjen za Windows PC računala i Xbox igraću konzolu. Interakcija se obavlja pomoću integrirane kamere, kojom se korisnicima omogućuje kontroliranje interakcije pomoću tzv. prirodnog korisničkog sučelja, odnosno gesti, te pomoću govornih naredbi.

Kinect je ostvaren na software tehnologiji razvijenoj u Rare-u, podružnici Microsoft Game Studio tvrtke. Tehnologija kamere bazira se na tehnologiji razvijenoj u izraelskoj tvrtki PrimeSense. Navedena tehnologija može interpretirati specifične geste i time omogućuje potpuno prirodno upravljanje koristeći isključivo projector, kameru i specifičan mikročip za detekciju kretanja objekata u trodimenzionalnom prostoru. Light Coding naziv je za korišteni trodimenzionalni sustav skeniranja, koji koristi 3D rekonstrukciju temeljenu na slikama (eng. image).

Senzor koji koristi uređaj Microsoft Kinect predstavlja vodoravnu traku spojenu s malom bazom s automatiziranim pivotom, koja je pozicionirana iznad ili ispod video prikaza. Kinect posjeduje RGB kameru, detector dubine i tzv. multimikrofon, koji omogućuju cjelokupnu detekciju pokreta tijela (eng. 3D motion capture), prepoznavanje lica i glasovnih naredbi. Multimikrofon senzora Kinecta omogućava akustičku lokalizaciju izvora i potiskivanje okolnih glasovnih smetnji (eng. noise). Senzor detekcije dubine sastoji se od infracrvenog laserskog projektora kombiniranog s monokromatskim CMOS sensorom, koji omogućava neometani rad u proizvoljnim uvjetima osvjetljenja. Raspon dubine senzora je u potpunosti prilagodljiv, s obzirom na korisnikove fizičke uvjete prilikom korištenja, na koje prvenstveno utječu fizički prisutne prepreke

### 2.2 Microsoft XNA 4.0

Prilikom implementacije simulacije korišteni su Microsoft Visual Studio 2010 i biblioteka Microsoft XNA 4.0. Korišten je programski jezik C#.

Microsoft Visual Studio 2010 predstavlja integriranu razvojnu okolinu (eng. *integrated development environment* - IDE). Koristi se za razvijanje konzolnih aplikacija, kao i aplikacija s korisničkim sučeljem, web stranica i aplikacija, te web servisa. Razvijene aplikacije moguće je koristiti na platformama koje podržava Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework i Microsoft Silverlight.

Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.

Microsoft XNA je naziv za biblioteku koja sadrži skup alata za razvoj grafičkih aplikacija i video igara. Obuhvaća razvojne alate Xbox Development Kit i XNA Game Studio. Korištena varijanta XNA 4.0 izdana je u ožujku 2010. godine. Ona, između ostaloga, pruža podršku za platformu Windows Phone 7, korištenje podesivih efekata i integraciju s razvojnom okolinom Microsoft Visual Studio 2010.

### 3. Tehničke značajke

Iako su Kinect SDK i XNA Studio u zajedničkoj kombinaciji moćan alat i pružaju mnoštvo mogućnosti do kojih je lako doći na jednostavan način, ipak je za kompleksnije zadatke potrebno ručno implementirati neke stvari i riješiti mnoge probleme, a ovdje su navedeni neki od većih s kojima smo se susreli.

#### 3.1 Iscrtavanje scene

Tijekom razvoje je bilo potrebno povezati 3D grafički pokretač i Kinect SDK. Nije bilo moguće koristiti Kinect SDK s klasičnim, jednostavnijim WPF sučeljem jer je bila potrebna podrška za određene značajke koje WPF ne poržava, kao što su učitavanje animacije i 3D modela. Korištenjem XNA umjesto WPF-a smo dobili na performansama i mogućnostima, ali je zato bilo potrebno ručno implementirati neke stvari, kao što je npr. konverzija boje iz slike s Kinect kamere, za što u WPF-u postoje već gotove funkcije. S obzirom da podaci koji se dobivaju dolaze u BGRA (blue, green, red, alpha boje) formatu, potrebno ih je transformirati u RGBA format koji koristi XNA: Za to je napisan kratki program za sjenčanje da se ne bi izgubile performanse, a to je odlučeno zato jer je u prethodnim verzijama slična operacija na dubinskoj slici bila izvođena na glavnom procesoru i strašno je opterećivala program ovisno o rezoluciji slike.

Naposljetku, nakon što je dohvaćena slika s kamere, učitani model, osvježene pozicije te izvršene ostale zadatke u koracima osvježavanja varijabli scene potrebno je iscrtati sliku. Sam postupak iscrtavanja je jednostavan, ali je problematično naći ispravan redoslijed iscrtavanja pojedinih dijelova i postavljanja stanja načina rada grafičkog adaptera. Problemi koji se događaju s neispravnim postavkama su nevidljivi ili poluprozirni modeli ili grafičko sučelje.

#### 3.2 Detekcija šaka

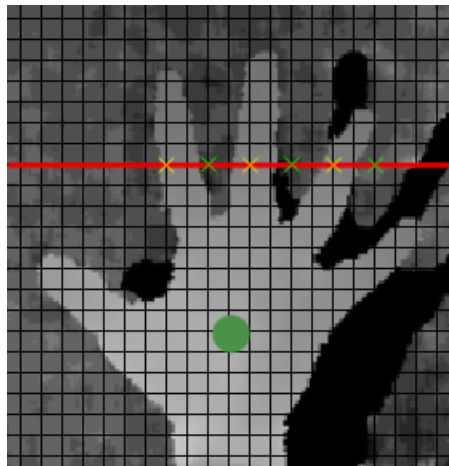
Za manipuliranje objektima koriste se ruke, odnosno šake. Time se javlja problem kako razlikovati namjeru korisnika da manipulira objektom u sceni od običnog pomicanja ruku tijekom rada s aplikacijom. Kako bi to riješili, objektom se manipulira samo kada su obje šake otvorene. Kinect nema ugrađenu mogućnost detekcije otvorene ili zatvorene šake, stoga smo to morali implementirati.

Zbog jednostavnije implementacije, otvorenom šakom smatra se otvoren dlan s razmaknutim prstima. Ideja je detektirati otvorenu šaku na temelju praznina između prstiju. Postavlja se pitanje što je praznina, a što prst, odnosno kako znati razliku. Kada bi koristili sliku u boji s Kinecta to pitanje bi predstavljalo veliki problem. Na sreću, koristimo dubinsku sliku koja za svaki slikovni element daje i podatak o dubini u prostoru. Štoviše, u kombinaciji s kosturom, kojeg Kinect stvara, za svaki slikovni element dubinske slike znamo i da li on pripada korisniku ili je dio okoline.

Trenutni položaj šake u prostoru određuje se iz pozicije šake na kostoru. Uzima se kvadratni dio slike oko šake kako bi za obradu dobili manju sliku samo sa šakom. Kroz sliku se prolazi na dva načina – horizontalno i vertikalno. Jedno je analogno drugom pa će u nastavku biti objašnjeno horizontalno

Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.

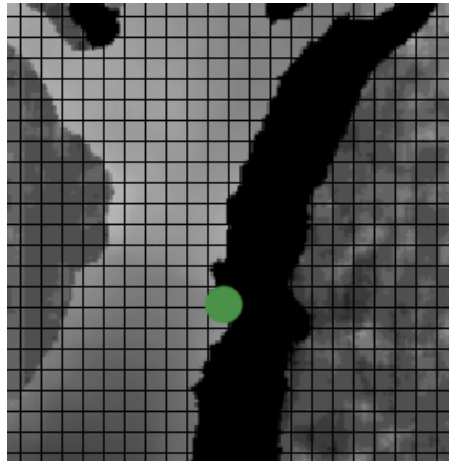
skeniranje. Kroz sliku se prolazi u linijama od gornjeg lijevog kuta prema donjem. Za svaki slikovni element ispituje se dubina. Referentna dubina je dubina točke kostura s pozicijom šake. Ako je dubina ispitivanog slikovnog elementa unutar određenih granica može se pretpostaviti da je on upravo i dio šake. U suprotnom je ispitivani element dio pozadine. Ako horizontalna linija koju trenutno pregledavamo siječe prste detektiraju se mnogi prijelazi prst – praznina. Zbrajaju se svi prijelazi za sve linije te se na temelju broja prijelaza zaključuje stanje šake. Za zatvorenu šaku broj prijelaza je malen ili nula. Vertikalno skeniranje se dodaje kako bi pokrili različite položaje šake. Na sljedećoj slici prikazano je horizontalno skeniranje. Crvenom bojom označena je linija koju promatramo. Zbog bržeg skeniranja horizontalni korak nije jedan slikovni element već više njih. Žuti križići bilježe prvu detekciju šake, a zeleni prvu detekciju okoline. Broj tih prijelaza za cijelu sliku određuje stanje šake.



**Slika 1 Detekcija**

Kada su pozicije šaka stalne te se one samo otvaraju ili zatvaraju detekcija radi dobro. Problem se javlja kod pomicanja šaka. Kinect kreira kostur na temelju dubinske mape. Pomicanjem šaka dubinska mapa se mijenja te se stvaraju nove točke kostura. Te točke često znaju biti i sasvim krivo pozicionirane. U tom slučaju algoritam za detekciju šake će raditi s dijelom dubinske slike na kojem se uopće se nalazi šaka! Time dolazi do krive detekcije stanja šake. Na sljedećoj slici zelenim krugom označen je položaj šake kostura. Iako je šaka otvorena detektirana je zatvorena šaka zbog krive slike koju koristi algoritam detekcije. Kako bi se takvi događaji u što većoj mjeri izbjegli uvedena je stabilizacija detekcije stanja šake.

Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.



Slika 2 Pogrešna detekcija

Stabilizacija radi na principu pamćenja prijašnjih stanja šake te otežavanju promjene stanja. Kako bi došlo do promjene stanja šake detekcija šake mora vratiti istu vrijednost za više uzastopnih poziva nad različitim dubinskim slikama. Time se kratkotrajne pogreške pozicije šake u kosturu mogu izbjeći, iako ne u potpunosti. Pokazalo se da se teže detektira otvorena šaka te su stoga parametri stabilizacije podešeni da je lakše prijeći iz stanja zatvorene šake u otvorenu nego obrnuto. Negativna posljedica toga je da kada se sustav nalazi u stanju zatvorene šake te dođe do pogrešne detekcije otvorene šake trebat će više vremena da se stanje šake ponovno vrati u zatvoreno.

### 3.3 Geste

Pored upravljanja 3D modelom predefiniranim animacijama i explode algoritmima željeli smo dodati u naš sustav i ručno upravljanje uređajem. Stoga smo implementirali geste za rotiranje i skaliranje uređaja. Da bi mogli manipulirati uređajem uveli smo novu klasu koja u svakom trenutku prati položaj i zatvorenost šaka, te smjerove u kojima se šake gibaju.

### 3.4 Skaliranje

Skaliranje se pokreće kada korisnik približava/udaljava zatvorene šake. Sustav pamti trenutnu skalu modela pa kada se detektira približavanje ili udaljavanje zatvorenih šaka skala uređaja se umanjuje ili povećava za fiksni inkrement svake vremenske jedinice izvođenja. Fiksni inkrement je odabran radi izbjegavanja naglih skokova prilikom nepreciznosti praćenja šaka. Skala je ograničena na intervale između 50% i 200% veličine uređaja.

### 3.5 Rotiranje

Rotiranje se pokreće kada korisnik ima jednu šaku otvorenu, a drugu koja je zatvorena pomiče u prostoru. Podizanje/spuštanje šake mjenja uređaju dubinu (engl. *pitch*), a pomicanje šake lijevo/desno mjenja uređaju smjer (engl. *yaw*). Sustav pamti trenutnu orijentaciju pomoću kvaterniona. Promjene dubine/smjera su fiksnog iznosa, da bi se izbjegli nagli skokovi prilikom nepreciznosti praćenja šaka.

Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.

### 3.6 Ostvarivanje sučelja

Budući da programsko rješenje nudi čitav niz različitih funkcionalnosti koje olakšavaju proces održavanja uređaja Micro Soot, javlja se potreba za kreiranjem praktičnog grafičkog korisničkog sučelja kao dodatak interakciji gestama. Prevelikim brojem gesta narušava se praktičnost i upotrebljivost ovakvog interaktivnog virtualnog priručnika. Grafičko korisničko sučelje čine tipke za upravljanje animiranim koracima u redovnom servisu uređaja. Tipke su implementirane korištenjem jednostavnih 2D tekstura (eng. *sprite*), a odabiru se klikom miša ili prekrivanjem šakom. Za sam odabir tipke bitna je pozicija kursora, tj. šake korisnika. Za svaku tipku definirana je njena 2D pozicija na ekranu aplikacije. Ukoliko se kursor ili šaka korisnika nađe unutar definiranog područja tipke, tekstura koja prekriva tipku mijenja izgled čime se korisniku daje do znanja da drži ruku na pravom mjestu.

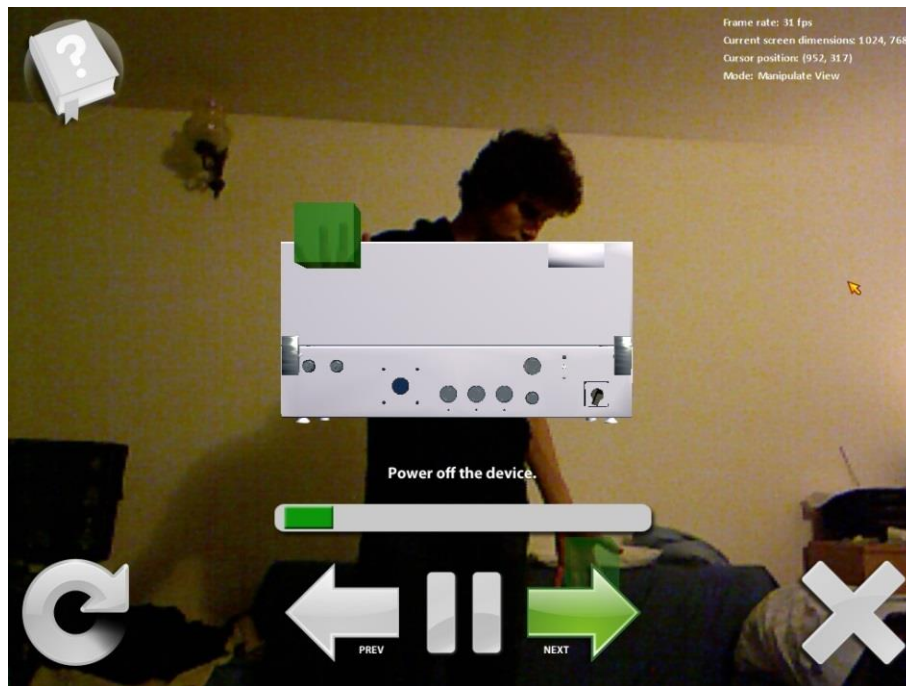
### 3.7 Animacije

Da bi korisniku bio omogućen ilustrativan uvid u servisne korake u redovitom održavanju uređaja, svaki je animiran u Autodesk 3DS Max-u, a zatim učitani u XNA program. Na taj način korisnik može pregledavati servisne korake koristeći navigacijske tipke na dnu ekrana. Sedam animacija kreiranih u 3DS Max-u izvezene su u odvojenim FBX datotekama. Ove datoteke sadrže samo animacije i vrlo malo geometrije. Mreže modela svedene su na trokutiće kako bi se smanjila veličina datoteke. Ovakve „gole“ datoteke učitane su u XNA program gdje se izdvajaju informacije o transformacijama, te zatim primjenjuju na prethodno učitani neanimirani kompletni model uređaja (referentni model).

### 3.8 Interakcija sa sučeljem

Osim korištenja sučelja pomoću miša i kursora, moguća je i kontrola gumba na sučelju pomoću ruku. Naime, praćenjem položaja ruku je moguće odrediti kada se one nađu unutar kvadrata koji predstavlja gumb (na sličan način kao što radi i detekcija položaja kursora miša). Na temelju toga gumb se oboji da pruži vizualni podražaj (implementacija podražaja je iznimno bitna jer je kontrola pomoću Kinecta pomalo nepouzdana i inače ne bi bili sigurni što se s programom događa i je li se preklapanje stvarno dogodilo), a s obzirom da ne postoji pouzdana akcija kao što je klik miša, kao sredstvo potvrđivanja je izabran protok vremena, čime se izbjegavaju slučajne aktivacije gumba, a opet se pruža pouzdana tehnika aktivacije. Također i uz samu aktivaciju gumba je vezan vizualni podražaj, koji je prikazan kao blijeđenje kocke koja predstavlja položaj ruke, kao što je opisano u Microsoftovim smjernicama za ljudski zasnovana sučelja.

Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.



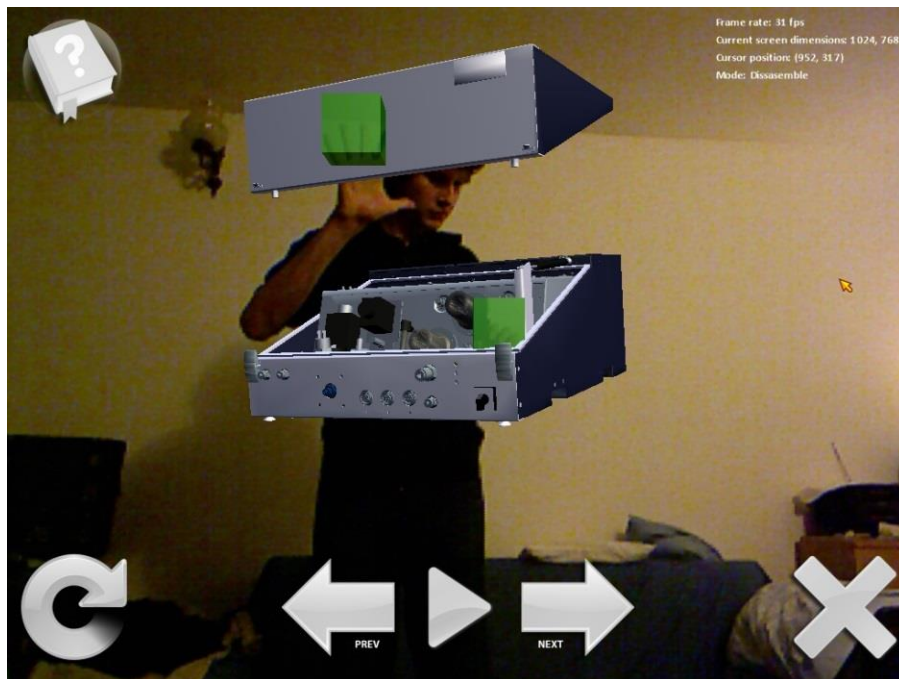
Slika 3 Aktivacija gumba rukom

### 3.9 Rastavljanje modela

Uz korištenje sučelja i prethodno navedenih gesti u programu je također moguće koristiti i funkcionalnost rastavljanja modela koja je također zasnovana na principima prirodnog sučelja (*NUI*). Koristeći akcije koje su intuitivne i analogne onima iz stvarnog svijeta, kao što su posezanje sa stvarima, hvatanje i premještanje, na isti način je moguće i upravljati dijelovima prikazanog *Microsoft* modela. Funkcionalnost je implementirana korištenjem ugrađene podrške za detekciju sudara u *XNA Frameworku*, a sudari se detektiraju između kocki koje predstavljaju šake i dijelova *Microsoft* modela. Kao primitivi za koliziju su odabrane sfere (engl. *Bounding Sphere*) jer je bila potrebna detekcija neovisna o rotaciji modela, a nažalost, *XNA* ne podržava kocke orijentirane kao objekt (engl. *Object Oriented Bounding Boxes*) koje bi pružile bolju detekciju. No iako postoji ugrađena podrška, sama implementacija nije toliko jednostavna jer *XNA* ne zna automatski translirati i skalirati sfere pa je potrebno ručno podešavanje.



Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.



Slika 4 Rastavljanje modela

#### 4. Algoritam ekspanzije

U suvremenoj računalnoj grafici većinom se vizualiziraju složeni 3D modeli. Navedenim je modelima između ostalog moguće predstaviti mehaničke sklopove, elektroničke uređaje, te arhitektonska okruženja. Zajednička karakteristika im je sastavljenost od većeg broja dijelova. Za vizualizaciju unutarnje strukture dotičnih dijelova, potrebno je ostvariti određeni način tzv. „eksplozije“ dijelova, kako bi se svaki uklopljeni dio mogao vizualizirati, tj. istaknuti promatraču. Pri tome nužno je paziti na međusobne odnose dijelova u cjelokupnoj strukturi modela.

Ograničenja eksplozije određuju smjerove prema kojima se rastavljaju dijelovi modela. Pomak pojedinih dijelova potrebno je bazirati na sjedećim ograničenjima:

Ograničenje blokiranja – prilikom eksplozije dijelova potrebno je osigurati da ne dolazi do preklapanja pojedinih dijelova, tj. pomak mora osigurati potpunu vidljivost svakog dijela modela.

Kompaktnosti – prilikom pomaka osigurati što manju relativnu udaljenost pozicije pomaknutog dijela od njegove prvobitne pozicije.

Kanonski smjerovi eksplozije – mnogi objekti posjeduju kanonski koordinirajući okvir, definiran primjerice simetrijom, orijentacijom stvarnog svijeta, te ostalim ograničenjima iz područja domene primjene. Potrebno je ograničiti broj mogućih smjerova ekspanzije, kako bi se promatraču olakšala interpretacija premještaja svakog pojedinog dijela prikaza „eksplozije“ u originalni prikaz (prije „eksplozije dijelova“).

Implementirana verzija dijagrama eksplozije je klasičan primjer, prilagođen konkretno za model AVL uređaja. Kao takav teži korisnika istovremeno izložiti svim dijelovima uređaja. Problem prikaza

Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.

odgovarajućih dijagrama eksplozije je više problem korištenosti u struci, te se u prvom redu treba filtrirati najbolja moguća konvencija dekompozicije određenog modela. S obzirom na to, različiti dijagrami eksplozije mogu imati različite algoritme u odnosu na područje i model gdje se koriste.

#### 4.1 Implementacija

Algoritam ekspanzije implementiran je pomoću Visual Studia 2010 i biblioteke Microsoft XNA 4.0. Microsoft XNA predstavlja skup alata s upravljanim „runtime“ okruženjem, koji olakšavaju razvoj video igara i ostalih programa iz područja računalne grafike.

Trodimenzionalni objekt je prikazan klasom Model. Klasa Model sastoji se od brojnih mreža (eng. *ModelMesh*). Pomoću navedenih mreža moguće je pristupiti dijelovima modela, te pri tome mijenjati njihovu poziciju, odnosno animirati ih.

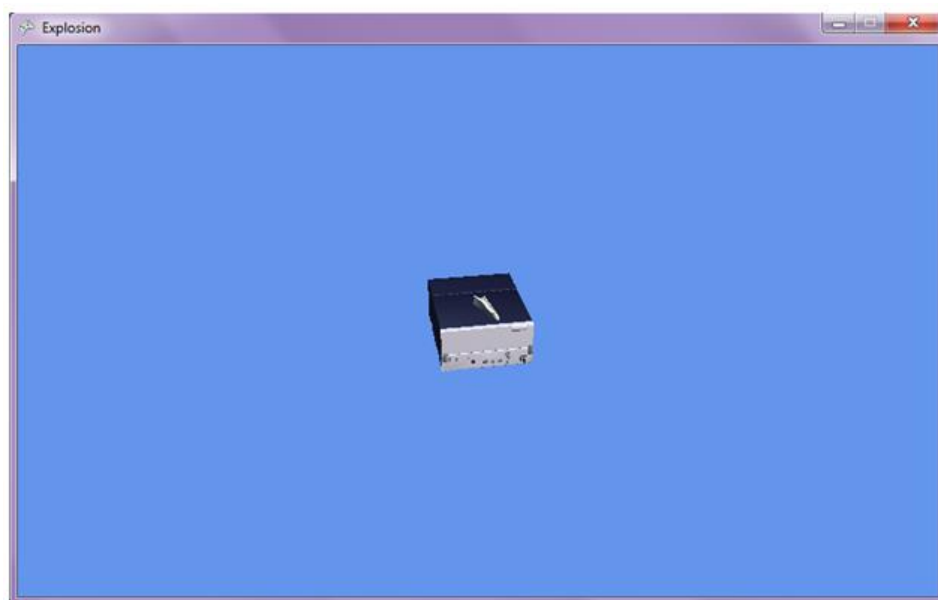
Pomak pojedinog dijela, odnosno mreže modela, ostvaren je pomoću funkcije `Matrix.CreateTranslation`. Navedena funkcija ostvaruje pomak koji se definira pomoću relativne udaljenosti u smjeru x, y i z osi.

Prilikom ostvarivanja eksplozije nastojalo se paziti na razmještaj dijelova u originalnog strukturi. Analizom redoslijeda zapisa mreža modela uočena je određena pravilnost. Naime, povezani dijelovi navedeni su slijedno u zapisu mreža modela. Ovo svojstvo je iskorišteno u implementaciji eksplozije dijelova.

Također, prilikom ostvarivanja eksplozije bilo je nužno podijeliti model u dva podskupa. Podjela je izvršena s obzirom na poziciju pojedinih dijelova. Dvije skupine predstavljaju dijelove koji se nalaze na gornjem, odnosno donjem dijelu modela.

Implementirana je i animacija, odnosno interakcija s korisnikom. Naime, „eksplozijom“ upravlja korisnik pritiskom odgovarajuće tipke.

Prilikom pokretanja programa prikazuje se originalni model.



Slika 5 Prikaz originalnog modela

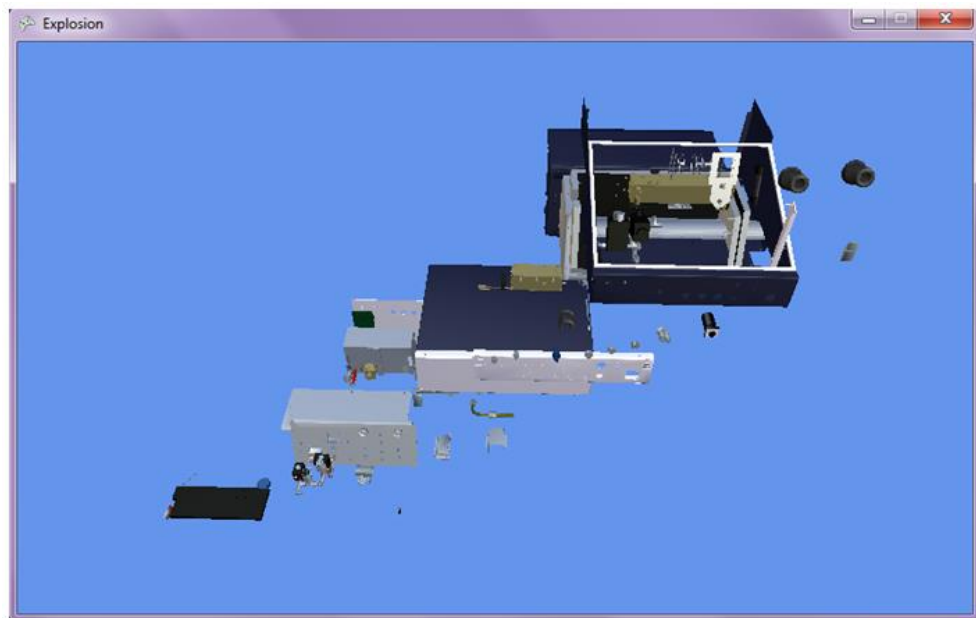
Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.

Prikaz je moguće približavati pritiskom na tipku X, te udaljavati pritiskom na tipku Z. Također, moguće je rotirati model pomoću strelica.



**Slika 6 Model je „približen“ pritiskom na tipku X**

Pritiskom na tipku R dolazi do eksplozije modela. Dijelovi modela se odvajaju (tj. eksplodiraju) u smjeru pozitivnog i negativnog pomaka relativne vrijednosti pozicije x i y osi. Pri tome se dijelovi koji pripadaju gornjem podskupu pomiču u pozitivnom smjeru, a dijelovi koji pripadaju donjem podskupu pomiču se u negativnom smjeru.

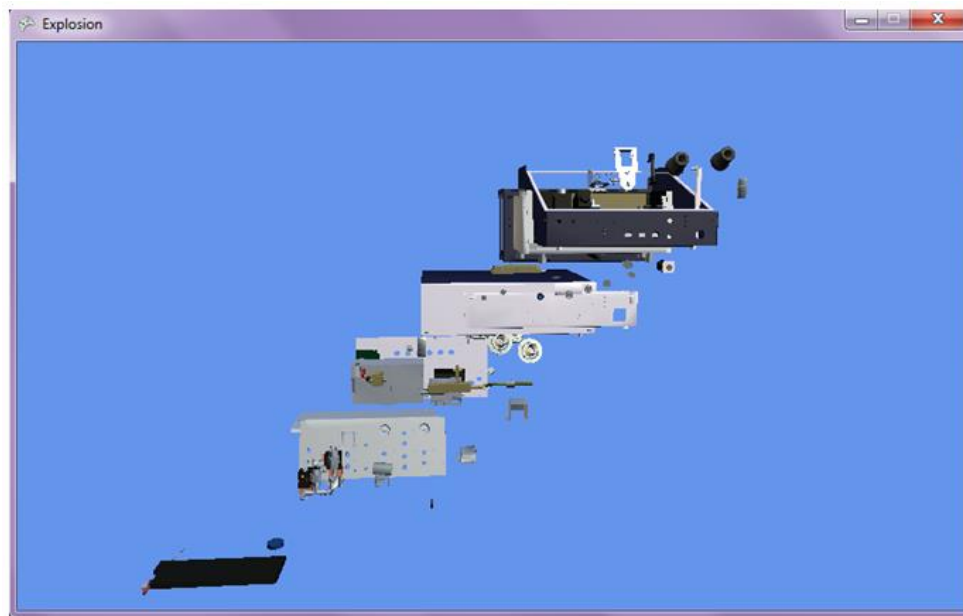


**Slika 7 Eksplozija dijelova**

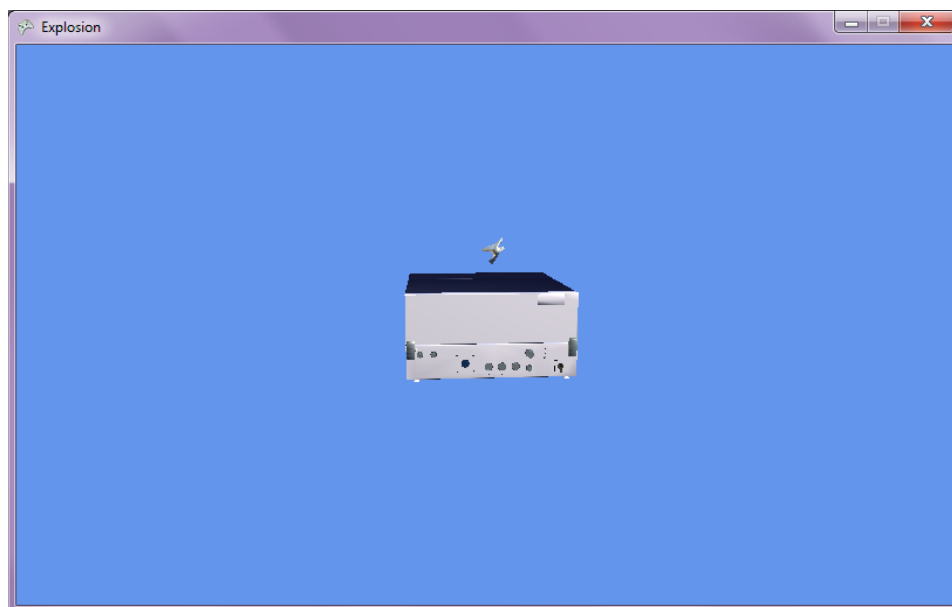
Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.

Svakim pritiskom na tipku R relativna udaljenost između dijelova se povećava za jednoliki faktor i oni postaju udaljeniji. Maksimalna udaljenost nije ograničena, te korisnik može algoritam eksplozije provoditi sve dok dijelovi ne napuste ravninu prikaza i postanu nevidljivi.

Skupljanje dijelova provodi se pritiskom na tipku T. Svakim se pritiskom tipke T dijelovi sve više približavaju originalnim pozicijama, tj. smanjuje se udaljenost između dijelova za jednoliki faktor. Skupljanje je ograničeno originalnom pozicijom modela, tj. skupljanje završava kada se model ponovo nađe u složenom stanju.



**Slika 8 Skupljanje dijelova**



**Slika 9 Skupljanje je ograničeno originalnom pozicijom modela**

Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.

## 5. Upute za korištenje

Modelom Microsoft moguće je upravljati gestama pomoću uređaja Kinect. Nakon pokretanja programa korisniku se prikazuje model Microsoft. Navedeni model korisnik može rastavljati pomoću gesti, odnosno pokretima ruku.

Također, implementiran je i algoritam eksplozije, koji model rastavlja i ponovo skuplja pritiskom na odgovarajuće tipke.

## 6. Zaključak i budući razvoj

Ostvarenu implementaciju odobrili su predstavnici AVL-a. Također, predložili su manje izmjene u vidu usporavanja animacije. Također dogovoren je budući razvoj naprednijeg algoritma ekspanzije, kao temelj buduće suradnje Fakulteta elektrotehnike i računarstva i tvrtke AVL.

Budući da smo imali malo vremena za izraditi projekt, a i od tog vremena je većina provedena na upoznavanju sa novim tehnologijama (*Kinect*, *XNA*), ostalo je nekoliko dodatnih mogućnosti za dodati našem sustavu. Najbitnija za geste su:

- mogućnost promjene nagiba (engl. *roll*) uređaja prilikom rotiranja, za ovu gestu bi se koristile dvije zatvorene šake koje bi vršile kružno gibanje
- rotiranje izvesti u odnosu na koordinatni sustav kamere, trenutno se vrši na koordinatnom sustavu uređaja
- dodati mogućnost rastavljanja uređaja dio po dio, tako da prividno šakom grabimo dio uređaja pa ga pomičemo u prostoru

Razvoj same implementacije dijagrama eksplozije, kao i razvoj nekog konkretnijeg algoritma je moguća tema daljnjeg razvoja u budućnosti. Pri tome bi se moglo osvrnuti na generaliziranje primjene algoritma na različite modele uređaja, ili čak modela nekih predmeta. Još jedna moguća točka poboljšanja je strukturiranje prikaza – korisniku prikazivati samo relevantne podatke za pojedine dijelove koje ga zanimaju u globalnoj slici modela. Specifična problematika mogla bi se proširiti i složenim metodama dijagrama eksplozije u okolini proširene stvarnosti. Kao takva predstavlja takođe jednu prepreku, jer imamo jednu novu dimenziju o kojoj se treba voditi računa, te koja bi se mogla iskoristiti da krajnjem korisniku pruži dodatne informacije.

## 7. Literatura

- [1] W.Li, M. Agrawala, B. Curless, D. Salesin - Automated Generation of Interactive 3D Exploded View Diagrams, University of Washington, University of California, Adobe Systems, SIGGraph 2008.
- [2] D.Kalkofen, M.Tatzgern, D. Smalstieg - Explosion Diagrams in Augmented Reality, Gratz University of Technology, 2009.
- [3] M. Hincapie, A. Caponio, H. Rios and E.G. Mendivil, "An introduction to Augmented Reality with applications in aeronautical maintenance," Transparent Optical Networks (ICTON), 2011 13th International Conference on , vol., no., pp.1-4, 26-30 June 2011.

Ostvarivanje prirodne interakcije između virtualnog i stvarnog svijeta pomoću Microsoft Kinecta	Verzija: 2.0
Tehnička dokumentacija	Datum 21.12.2012.

- [4] S. Bernd and L. De Blandine, "An augmented reality system for training and assistance to maintenance in the industrial context," Journal of Winter School of Computer Graphics, vol. 11, pp. 101-110, 2003.
- [5] N. Zenati, N. Zerhouni and K. Achour, "Assistance to maintenance in industrial process using an augmented reality system," Industrial Technology, 2004. IEEE ICIT '04. 2004 IEEE International Conference on , vol.2, no., pp. 848- 852 Vol. 2, 8-10 Dec. 2004.
- [6] T. Leyvand, C. Meekhof, Yi-Chen Wei, Jian Sun and Baining Guo, "Kinect Identity: Technology and Experience," Computer , vol.44, no.4, pp.94-96, April 2011.
- [7] M. Tang, "Recognizing Hand Gestures with Microsoft's Kinect," Stanford University, Department of Electrical Engineering, Technical Report, March 16, 2011.
- [8] A. Drake, "Kinect Hand Recognition and Tracking," Washington University in St. Louis, Kinect Hand Recognition and Tracking, Project Report, May 1, 2012.
- [9] F.T. Cerezo, "3D Hand and Finger Recognition using Kinect," University of Granada, Project Report, 2011.
- [10] Human Interface Guidelines for Kinect for Windows, Vers. 1.5.0., 2012.