



DIGITALNA FABRIKACIJA MONOCHROMATSKIH SLIKA OD NITI KORIŠĆENJEM INDUSTRIJSKOG ROBOTA

DIGITAL FABRICATION OF MONOCHROMATIC STRING ART USING INDUSTRIAL ROBOT

Stefan Tomić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ROBOTIKA U ARHITEKTURI

Kratak sadržaj – Ovaj rad se bavi algoritamskom problematikom jedne vrste kompjuterske umetnosti. To je iscrtavanje pomoću tankih niti, i dobijanje interesantne slike. Ovakva umetnost se naziva String art. String art predstavlja malo umetničko delo koje se kreira od konca, vune ili meke žice u boji. Posebna pažnja posvećena je algoritmu koji izbacuje pravila po kojima se povlače niti i iscrtava slika ili nečiji portret. Algoritam sadrži velik broj parametara, čija je promena potrebna za dobijanje kvalitetnog finalnog prikaza. Na većinu ovih parametara utiču rasporedi i nijanse piksela odabrane slike. Konačno, fabrikacija ovako dobijene slike pomoću industrijskog robota koji koncem za šivenje izvodi ovo umetničko delo, znatno olakšava i ubrzava naporan proces izvođenja koji dans umesto robota radi čovek.

Ključne reči: *Kompjuterska umetnost, fabrikacija pomoću konca, industrijski robot*

Abstract – This master thesis deals with algorithm problematic of computer designed art. Computer algorithm generates a drawing using thin threads and results in an interesting picture. This kind of computer generated art is called String art. String art represents a small master piece created by thread of wool or coloured soft wire. Special attention is given to the algorithm that expels the rules used for thread stretching and thus composing of a drawing or portrait. The algorithm contains a large number of parameters whose changes are necessary for obtaining a quality final display of picture. Most of these parameters are influenced by pixel scale and shades of the selected image. Finally, the fabrication of this kind of image with sewing thread by the use of industrial robot which performs this art work, greatly facilitates and accelerates the strenuous execution process that is being performed today by a man instead of a robot.

Keywords: *Computer designed art, fabrication with sewing thread, industrial computer*

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milutin Nikolić, docent.

1. UVOD

Koncept String arta se po prvi put javlja oko 1845. godine zahvaljujući engleskoj matematičarki Mari Everest Bole koja je koristila konac za prikazivanje krivih linija kako bi geometrijske oblike učinila razumljivijim svojim učenicima. Tako se ovaj interesantni način prikazivanja različitih oblika polako širi i prenosi se iz Engleske u Francusku gde ga matematičar i fizičar Pol de Casteljau dalje razvija i stvara novi princip pomoću koga Pijer Bezier, francuski inženjer, stvara čuvenu “bezier krivu”. String art se naglo popularizuje 60-ih godina kao vid zabave i umetničkog stvaralaštva.

Danas postoje razni algoritmi za dobijanje pravila u kojima se fotografija “prevodi” u niti, pa se ta pravila primenjuju od strane umetnika u procesu izrade umetničkog dela. Tako danas programiranje, inženjerstvo i automatizacija procesa izvođenja postaju jedan od značajnijih metoda za dobijanje velikog broja savremenih umetničkih dela. Samim tim programeri i inženjeri postaju umetnici i obrnuto. Cilj istraživanja je dizajniranje atraktivne slike na računaru ispletene tankim nitima i njen ubrzan proces fabrikacije.

2. METODOLOGIJA ISTRAŽIVANJA

Prva stavka koja je bila potrebna za pravljenje String arta jeste kreiranje VisualStudio komponente. Zatim su rezultati koje je ona dala prebačeni u RobotStudio, gde su iskorišćeni za programiranje robota i pravljenje simulacije celokupnog procesa fabrikovanja. Na kraju je ceo postupak iz simulacije izveden sa materijalima primenom industrijskog robota.

2.1. VisualStudio komponenta

Microsoft Visual Studio je integrisano programsko okruženje, kreirano od strane kompanije Majkrosoft. Visual Studio se koristi za programiranje računarskih igara, programa, veb-sajtova, veb-servisa i veb-aplikacija na Microsoft Windowsu. Visual Studio podržava 36 različitih programskih jezika. Kodiranje ovog problema rađeno je u programskoj jeziku C#, uz pomoć veze sa Rhinoceros-om i Grasshopperom.

Ova biblioteka u sebi ima mogućnost da kreira nove plugin-ove, specijalizovane za Grasshopper, i tako “obogati” funkcije i mogućnosti ovog sve više popularnog plugin-a.

Grasshopper funkcioniše na osnovu proceduralnog vizualnog programiranja i sadrži vizuelne nodove. Nodovi mogu imati svoje ulazne i izlazne parametere. Ceo algoritam za pravljenje Strings arta kodiran u

VisualStudios se zapravo prikazuje kao jedan nod u Grasshopperu sa mnoštvom ulaznih i izlaznih parametara. Ovi parametri su potrebni za odabir slike i modifikaciju željenih rezultata stringova. Takođe VisualStudio ima svoje biblioteke povezane sa Rhinoceros-om i Grasshopper-om te se u njemu mogu kreirati objekti poput tačke, linije, polilinije, krive, kruga... Ova mogućnost olakšava nam "prevođenje" pikselizovane slike na ekranu u vektorske, matematičke dimenzije na kojima upravo i radi Rhinoceros3D. Grasshopper nam daje potrebne informacije koje se kanije prebacuju u RobotStudio, program koji služi za programiranje robota što je neophodno za fabrikaciju slike.

2.2. Ulazni parametri komponente

VisualStudio komponenta mora imati neke ulazne informacije da bi dala željena pravila. Te informacije su većinom slajderi koji mogu biti promenljivi, i u zavisnosti od različitosti fotografije, davati bolji ili lošiji kvalitet dizajna niti. Sam algoritam sadrži 8 ulaznih i 5 izlaznih parametara, slika 1.



Slika 1. Prikaz komponente sa parametrima

1. Prvi parametar koji uzimamo iz Rhina jeste centar kruga. U Rhinoseros3D stavimo tačku bilo gde na radnoj površini, i vežemo je za komponentu.
2. Sledeća stvar koju definišemo je poluprečnik kruga koji zadajemo nekim brojem u slajderu. Ovo je potrebno za kasniju fabrikaciju slike, da bi se tačno znale dimenzije kruga koji treba napraviti.
3. Treći parametar je broj pinova koji se kreira po krugu i koji čine početne i završne krajeve linija koje se is crtavaju.
4. Pošto niti is crtavamo od jednog do drugog pina, 100 stringova bi davalo 100 pozicija tačaka spremnih da se implementiraju u RobotStudio. Međutim, ovo bi sve bilo dobro kada bi robotska ruka imala olovku, i samo is crtavala linije od jednog pina do drugog. U uvodnom delu je pomenuto da ovakvu umetnost radi čovek ručno, tako što obmotava konac oko pinova (koji mogu biti metalni završetci, klinovi ili ekseri...). Kada bi fabrikaciju vršio robot, javio bi se problem obmotavanja konca oko pina. Komponenta je kodirana tako da ima mogućnost pomeranja jednog pina u 4 pina, jednako udaljena od

početnog, tako da su dva pina udaljena po poluprečniku kruga koji spaja centar kruga i taj pin, a dva po tangenti kruga na isti poluprečnik.

5. Sledeći ulazni parametar je slika koju kod treba da obrati. Umesto slajdera sa brojevima ulazni parametar prima tekst, i samo je potrebno u panel upisati tačnu putanju u računaru na kojoj se željena slika nalazi.

6. Parametrom Niti zadajemo koliki će biti tačan broj linija za is crtavanje. Deo algoritma koji zaključuje gde treba povući pravu liniju koja sadrži par pinova napravljen je tako što se između svakog mogućeg para pinova is crta linija. Tako da se faktički dobije pun obojen krug. Tako povučene zgusnute linije se zatim boduju prema svetlini piksela koje sadrže. Tako da će linije sa najviše bodova biti one koje sadrže najtamnije piksele, a sa najmanje one koje sadrže najsvetlije piksele. Ovaj problem je olakšan ubacivanjem crno belih fotografija, kako bi se dobile samo grayscale nijanse piksela.

Naime, ako bi fotografije ostale u boji, funkcija za očitavanje piksela bi imala mnogo veći opseg boja kao što su na primer RGB ili CMYK. Logika ovog dela algoritma je vrlo jednostavna, najcrnji pikseli dobijaju najviše bodova, a najbelji najmanje bodova.

Kada imamo određene vrednosti svakog piksela i metodu za definisanje para pinova linije, postavlja se pitanje kako dobiti redosled pinova do kojih prvo treba doći? Ovo se dobija uz pomoć metode koja određuje svaki sledeći pin do kog treba doći. Ova metoda u sebi sadrži maksimalnu vrednost izbodovanih linija u prethodnoj metodi koja je davala vrednost svakom pikselu. Ona se poziva pri određenom paru pinova. Ova metoda mora imati proveru da li su neke linije već is crtane, da ne bi dolazilo do bespotrebnog crtanja više puna po istom paru pinova. Pravila po kojima se bira redosled pinova za is crtavanje linija je takav da prva linija kreće od Pina 0, i u svakoj sledećoj iteraciji bira sledeći pin, tako da linija nije već nacrtana, ima maksimalnu izbodovanu vrednost i polazi iz tekućeg pina.

7. Slika koja bi trebalo da se formira od is crtanih stringova gotovo nikada ne ispada dobro, i ni blizu ne liči na lik sa slike. Naime, može se javiti problem "blizine". Par pinova koji se spajaju u jednu nit može biti preblizu, gotovo jedan pored drugog, tako da bi u ovom slučaju jednu nit činili na primer pinovi 67 i 68. Ovako se dobija zgušnjavanje niti na nekim mestima i nedostatak istih na drugim. Ovaj problem je rešen uvođenjem parametra koji je nazvan Razmak_niti. Ovaj parametar je u stvari zabrana da dođe do problema "blizine". On služi za povećanje dužine stringova. Ako se on stavi npr. na 15, to znači da 15 pinova mora biti preskočeno kao počenta i krajnja tačka jednog stringa. Ovo znači da minimalna dužina jednog stringa, mora podeliti krug tako da sa jedne njegove strane ostane minimum 15 pinova, a sa druge ostatak (Pinovi - 15). Ovaj parametar u stvari daje minimalnu distancu dva pina između kojih se is crtava linija.

8. Pomenuto je već da algoritam funkcioniše tako što prvo obradi sve piksele slike, zatim is crta sve moguće stringove između svih mogućih kombinacija pinova, a zatim ih izboduje. Ovo znači da ako bismo imali beskonačno mnogo pinova, dobio bi se crn ispunjen krug. Ako imamo recimo 100 pinova, i sve moguće

kombinacije stringova između nekih dva pina od tih 100, dobili bismo preklapanja linija na mestima gde se seku koja su tamnija.

Parametar koji ima mogućnost izbeljivanja ovih tamnih delova linija, uveo bi promene u postojećem algoritmu. Parametar u stvari posvetljava piksele najviše izbodovanih linija, i smanjuje im skalu bodovanja, tako da one dobijaju niže bodove pri sledećim prolaženjima kroz petlju u kojoj se iscrtavaju. Kada se povećava parametar Izbledi, povećava se i mogućnost da se na središnjem delu iscrtavaju niti, a smanjuje mogućnost iscrtavanja blizu već iscrtavane niti. Ovako se dobija efekat žgušnjavanja niti ka praznim mestima koja su na fotografiji imala svetlije piksele, a proređenja po mestima koja su mala tamnije piksele.

2.3. Izlazni parametri komponente

1. Prvi izlazni parametar se svodi na informacije o broju pinova koje bi trebale da se prebace u sintaksu RAPID-a, programskog jezika implementiranog u RobotStudio. Naime RAPID prima listu konstruktora u obliku teksta, koji sadrže informacije o tačnoj poziciji svakog od 4 virtuelna pina koji nastaju translacijom glavnog pina na kružnici. RAPID kasnije prevodi svaki od konstruktora u tačne pozicije u RobotStudiju.

Ovde zapravo vidimo da ulazni parametar Pinovi utiče na izlazni parametar Pozicije_za_RS, tako što zadati broj pinova u slajderu množi sa 4, i dobija broj Pozicija (program se pravi tako da se orijentacija vrha robota ne menja, pa su samo pozicije uzete u obzir) koji kreiraju isti broj konstruktora potrebnih RAPIDU.

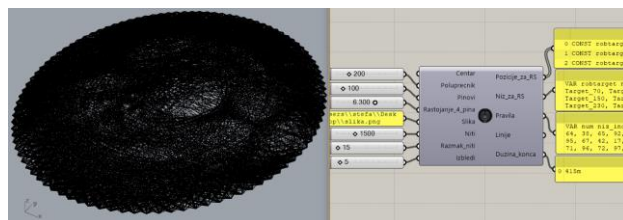
2. Drugi izlazni parametar definiše niz koji sadrži sve prethodno definisane pozicije. Njegova funkcija je da se olakša pristup istim, čime se dobija kompaktniji i čitljiviji kod u RAPIDu. Njegova definicija zavisi samo od broja pozicija, koja direktno zavisi od postavljenog broja pinova.

3. Treći izlazni parametar je suštinski najbitniji za ceo algoritam. On zapravo daje informacije od kog do kog pina treba povući liniju da bi slika mogla da se fabrikuje u realnosti. Postupak dobijanja redosleda pinova do kojih treba doći (ovih pravila) objašnjen je u delu gde je opisan šesti ulazni parametar Niti.

Poslednja tri ulazna parametra, zajedno sa brojem pinova, utiču na konstruisanje ovih pravila. Sve slike se razlikuju, stoga se može reći da je i parametar Slika takođe utiče na Pravila.

4. Parametar Linije je neophodan samo u vizualnom efektu, kako bi mogli da sagledamo kako će ispasti finalni rad, da ne bi vršili fabrikciju na slepo. Naravno, ovaj parametar iscrta isti broj linija koliko smo na početku zadali niti. Naravno da ovaj parametar zavisi od broja unetih stringova. Ako smo uneli 1500 stringova, prikazaće se 1500 linija u Rhino-u, slika 2.

5. I konačno, dolazimo do poslednjeg izlaznog parametra, Duzina_konca, vrlo značajnog za sam proces fabrikcije. Ovaj parametar izbacuje dužinu potrebnog konca. Vrlo je značajan zbog toga što možemo planirati potrebnu količinu konca na osnovu svih odabranih parametara. On radi tako što računa dužinu svake linije, koju već ima iscrtanu u vektorskom obliku, zatim sabira sve dužine i tako dobija celokupnu. Pored broja stringova takođe na dužinu konca utiče i poluprečnik kruga.

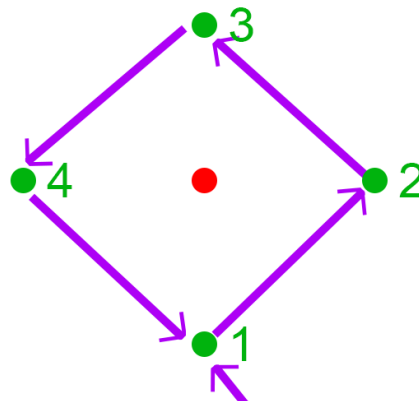


Slika 2. Konačan izgled slike sa parametrima iz komponente

3. FABRIKACIJA

3.1. Simulacija fabrikcije

Da bismo bili sigurni da robot može da izvrši fabrikciju, najbolje je prvo da se fabrikcija izvede u virtualnom svetu. Simulacija u RobotStudiju će pomoći da se tačno vidi da li će proces fabrikcije proći kako treba, i kako će tačno on izgledati. Pomenuto je da izlazni parametar izbacuje listu konstruktora svih pozicija. U ovom slučaju, prema logici koja je korišćena, množi broj zadatih pozicija sa 4, slika 3. Tako da za 100 pinova, izbacuje se 400 pozicija. Ove pozicije sortirane su od prve (Target_10) do poslednje (Target_4000) u nizu po kružnici.



Slika 3. Logika kretanja robotske ruke oko jednog pina (obmotavanje pina koncem)

Pošto robot treba da napravi krug oko pina, formula po kojoj se računa indeks prve tačke prilikom obilaska oko pina je: $\text{indeks} = \text{niz_indeksa}\{i\} * 4 + 1$

Na primer, ako bi prvi indeks u nizu dobijenih pravila bio $i = 17$, dobilo bi se: $17 * 4 + 1 = 69$. Ovo znači da bi prva pozicija do koje robot treba da dođe bila 69. pozicija u nizu a to je Target_690, i nalazi se u unutrašnjosti kružnice.

Dobili smo željenu sliku sa parametrima, programirali njenu putanju i kretanje robota u RAPIDu, sada je samo potrebno postaviti je na tablu i postolje.

Tabla na kojoj treba centrirati krug prečnika 40cm je dimenzija 50x50cm.

Kada se pozicije targeta kopiraju u RobotStudio, potrebno je izvršiti kalibraciju table (proveru da li robot može da dohvati krajnje ivice table i sve pinove). U RobotStudiju možemo pomerati targete sa tablom sve dok robot ne bude mogao da dohvati sve targete.

Tako ćemo biti sigurni da pri fizičkom izvođenju neće doći do problema.

3.2. Alat

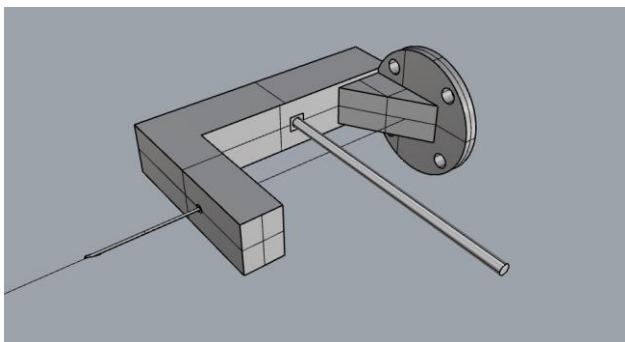
Gotovo najvažnija stvar pri izvođenju rada jeste alat kojim će robot da "plete".

Bez alata, celokupni dosadašnji rad bi bio uzaludan. Oni se pričvršćuju na krajnji deo robotske ruke sa 4 zavrtnja. Zaključeno je da nam je potrebna tabla od 50x50cm, i da se u nju centrira krug, tačnije 100 određenih pinova.

Oni su u RobotStudiju pozicionirani tako da bi alat robota mogao da dohvati svaki od ovih pinova bez problema. Ovi pinovi bi pri realnoj izvedbi trebali biti što tanji, i imati određenu visinu iznad ravni table. Pošto je odabrani prečnik kruga 40cm, kada bi se kreirali tanki pinovi po kružnici, razmak između njih bi bio oko 12mm.

Iz ovoga je izveden zaključak da bi se na završni deo alata mogao da bude prilično tanak poput igle, čak tanji od samih pinova za koje je najrealnije uzeti najtanje eksere. Alat je štampan na 3D štamšaču, a na sam vrh alata dodata je tanka igla, slika 4.

Kroz nju je provučen tanak crni konac, koji je materijal za kreiranje slike.

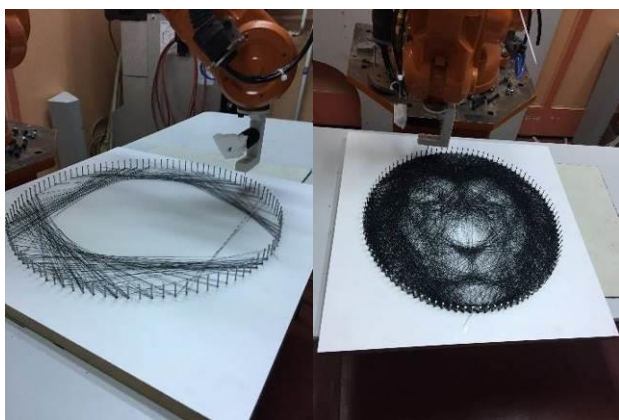


Slika 4. Dizajn alata

4. REZULTATI

Pri navedenim parametrima, na brzini robota od 6%, s tim da je brzina u kodu povećana na 2000, robot je celokupnu sliku "ispleo" za nepuna 2 sata.

Komponenta u Grasshopper-u je izračunala da je potrebno 415 metara dužine konca, što je manje od polovine kupljenog. Tako se sa jednim koncem od 1000 metara dužine mogu izvesti dve ovakve slike, slika 5.



Slika 5. Postupak izrade i konačan rezultat

5. ZAKLJUČAK

Nakon završetka izvođenja ovog projekta, može se izvesti zaključak da je mnogo efikasnije ovakve String art portrete izvoditi industrijskim robotom. Ovo pokazuju i dobijeni rezultati. Kada je robot pušten u rad, trebalo mu je nepuna dva sata da izvede sliku lava. Čoveku bi za ovaj postupak trebalo znatno više. Međutim, kada bi se radile kompleksnije stvari i sa robotom, morala bi se smanjiti brzina, te ona ne bi imala toliku ulogu u poboljšanju fabrikacije u poređenju robota i čoveka.

Čak i da brzina nema znatnu ulogu u poboljšanju, preciznost sigurno ima. Kada bi ovakvu stvar izvodio čovek, on bi gotovo sigurno pogrešio kod čitanja pravila ili samog obmotavanja pinova. U jednom trenutku ljudska koncentracija popusti, naročito ako ima velik broj niti, i ako izvođenja traju jako dugo. Robot se zato nikada neće umoriti, on i služi da duge i dosadne procese rada nesmetano, istim tempom istera do kraja.

Treba pomenuti da se pre same fabrikacije gubi prilično vremena na nameštanje robota za početak rada, te da je ovaj deo posla ponekad mukotrpan. Međutim, bez obzira na ovaj deo posla, izvođenje pokazuje da je u ovom slučaju mnogo bolje zameniti ljudsko biće mašinom.

LITERATURA

- [1] Predrag Janičić, *Računarska grafika*, Matematički Fakultet, Beograd, 2014.
- [2] RobotStudio, *Product specification*, ABB Robotics, 2013.
- [3] Introduction to RAPI, *Operating manual*, RobotWare 5.0, 2007.
- [4] <https://imi.pmf.kg.ac.rs/dokuwiki/doku.php?id=vektorska-i-rasterska-grafika>
- [5] <https://www.arhitektura.rs/rubrike/umetnost-i-dizajn/dizajn/409-kompjuterska-umetnost-novijeg-datuma>
- [6] <https://hackaday.com/2016/07/28/computer-designed-portraits-knit-by-hand/>
- [7] <http://sites.psu.edu/algotecton/2015/04/21/string-sculpture-tool-vernelle-noel/>
- [8] <http://sites.psu.edu/algotecton/tag/robots/>
- [9] <http://23mag.me/robotika-automatizacija-arhitektura-1/>
- [10] <http://23mag.me/robotika-i-automatizacija-u-arhitekturi-mogucnost-ili-potreba-ii/>

Kratka biografija:



Stefan Tomić je rođen 13.01.1994. godine u Šapcu. Osnovnu školu „Laza K. Lazarević“ u Šapcu završio je 2009. godine. Gimnaziju „Šabačka gimnazija“ u Šapcu završio je 2013. godine. Iste godine upisao se na Fakultet tehničkih nauka, osek Arhitektura i urbanizam. Osnovne studije je završio 2017. godine, nakon čega upisuje master akademske studije na Fakultetu tehničkih nauka, smer Digitalne tehnike, dizajn i produkcija u arhitekturi i urbanizmu. Položio je sve ispite propisane planom i programom.