

SVEUČILIŠTE U ZAGREBU

GRAFIČKI FAKULTET

Dino Priselac

Programsko upravljanje i manipulacija portretom u
računarskoj grafici

Zagreb, 2016.

Ovaj rad izrađen je na Katedri za računarsku grafiku i multimedijske sustave pod vodstvom doc.dr.sc. Nikoline Stanić Loknar i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2015./2016.

Sadržaj

1. UVOD	1
2. OPĆI I SPECIFIČNI CILJEVI RADA	1
3. RAČUNARSKA GRAFIKA	2
3.1. ZAŠTITNA LINIJSKA GRAFIKA	2
3.1.1. Izrada portreta na vrijednosnicama linijskom grafikom.....	4
3.2. ZAŠTITNA RASTERSKA GRAFIKA.....	5
3.2.1. Romboidni rasterski element	7
3.2.2. Sinusoidni rasterski element	8
4. PROCESSING PROGRAMSKI JEZIK	9
5. EKSPERIMENTALNI DIO	11
5.1. IZRADA PORTRETA S LINIJSKIM ELEMENTIMA	12
5.1.1. Promjena vrijednosti sivih tonova ovisno o debljini linije.....	12
5.1.2. Promjena vrijednosti sivih tonova ovisno o duljini i debljini linije	14
5.1.3. Promjena vrijednosti sivih tonova ovisno o smjeru, duljini i debljini linije	17
5.2. IZRADA PORTRETA S KRUŽNIM OBLICIMA	19
5.2.1. Promjena vrijednosti sivih tonova ovisno o radijusu kružnih oblika.....	19
5.2.2. Promjena vrijednosti sivih tonova i boje ovisno o radijusu kružnih oblika	22
5.3. IZRADA PORTRETA S KVADRATIČASTIM OBLICIMA.....	24
6. REZULTATI I RASPRAVA	27
7. ZAKLJUČAK.....	31
8. LITERATURA.....	32
Sažetak.....	34
Summary	35

1. UVOD

Računalnom linijskom grafikom otvoreno je novo poglavlje za istraživanje i kreiranje reprodukcije portreta. Od pojave prvih papirnih novčanica portret kao najupečatljiviji dio posebno je privlačio pažnju. Izrada portreta izvodila se ručno, graviranjem. Današnja tehnologija nudi mnogobrojne mogućnosti obrade i reprodukcije portreta. Rasterske tehnike, opisane u teorijskom dijelu rada, prikazuju razne mogućnosti u izradi portreta. Kompjuterski programi nude razne alate za izradu i obradu fotografija, međutim algoritmi koji rade takve obrade nisu javno dostupni. Ne mogu se kontrolirati svi parametri koji utječu na promjene u strukturi fotografije. Ovaj rad prikazuje manipulaciju reprodukcije portreta s algoritmima stvorenima u programskom jeziku Processing. U eksperimentalnom dijelu rada posebno su kreirani potprogrami koji piksele zamjenjuju programiranim vektorskim oblicima u strogo kontroliranim uvjetima. Svi parametri koji utječu na izgled vektorskih oblika su poznati i definirani od autora. Primjeri su izvedeni od originalnog portreta na kojeg se djeluje programskim kodovima stvarajući zanimljiva rješenja. Dobiveni portreti se zatim implementiraju na dokumente i vrijednosnice kako bi se vidio njihov konačni rezultat i područje primjene istih.

2. OPĆI I SPECIFIČNI CILJEVI RADA

Cilj ovog rada je predstaviti nov način upravljanja reprodukcijom portreta preko programskih kodova kroz Processing programski jezik. Želja je stvoriti korisničko sučelje u kojem korisnik sam pomakom miša utječe na grafičke elemente koji grade portret. Mreža piksela iz originalne fotografije je zamijenjena mrežom vektorskih oblika. U radu se eksperimentira s osnovnim vektorskim oblicima linijom, kugom i kvadratom koji s udaljenošću miša od ishodišta mijenjaju svoju debljinu, boju i duljinu stvarajući originalna rješenja.

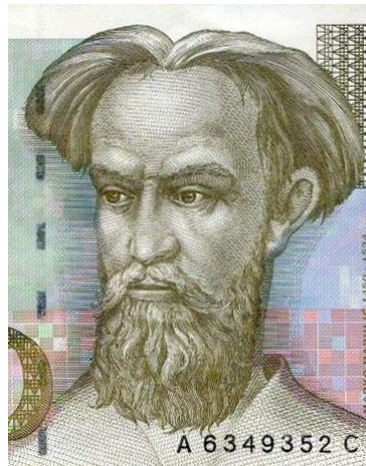
3. RAČUNARSKA GRAFIKA

Računarska grafika je dio informacijske tehnologije koji se bavi dobivanjem grafičkih prikaza uz pomoć računala. Prikazi se mogu generirati, obrađivati i pohranjivati u vektorskom zapisu (vektorska slika) ili rasterskom zapisu (rasterska slika ili bitmapa) te prikazati na zaslonu računala ili isprintati pomoću pisača. Primjenjuje se za vizualizaciju rezultata istraživanja u različitim područjima znanosti i tehnike, npr. u konstruiranju automobila i brodova, izradbi arhitektonskih projekata, industrijskom i grafičkom oblikovanju, izradbi specijalnih filmskih efekata, animiranih filmova, reklamnih prikaza te računalnih igara, a razvija se i kao zasebna grana likovne umjetnosti. [1.]

3.1. ZAŠTITNA LINIJSKA GRAFIKA

Zaštitna linijska grafika je jedna od najstarijih metoda zaštite dokumenata i vrijednosnica, poznata još od drevnih vremena. Prve novčanice u Europi počinju se izrađivati u 17. stoljeću na način da se crna tinta nanosila na portret koji je bio ugraviran u bakrenu ploču te se pod pritiskom prenosila na papir. Datum, nominalne vrijednosti, serijski broj i potpis uglavnom su se upisivali ručno što je bilo prilično nepraktično kod veće količine novčanica. U 18. i 19. stoljeću počinju se koristiti nove tehnike u izradi novčanica poput ubacivanja vlakana u strukturu papira, vodeni žig i gravirna ilustracija. Takva tehnika naziva se još i giljoše, a to je ime dobila prema Francuzu Guillocheu koji je 1770. godine napravio prvi stroj za graviranje sličnih uzoraka. Ta je tehnika prvo bila primjenjivana kao ukras na umjetninama, a nakon toga se počela primjenjivati i u zaštitne svrhe zbog i kompleksnosti linija i manjka tehnoloških mogućnosti da se linije kopiraju. [2.]

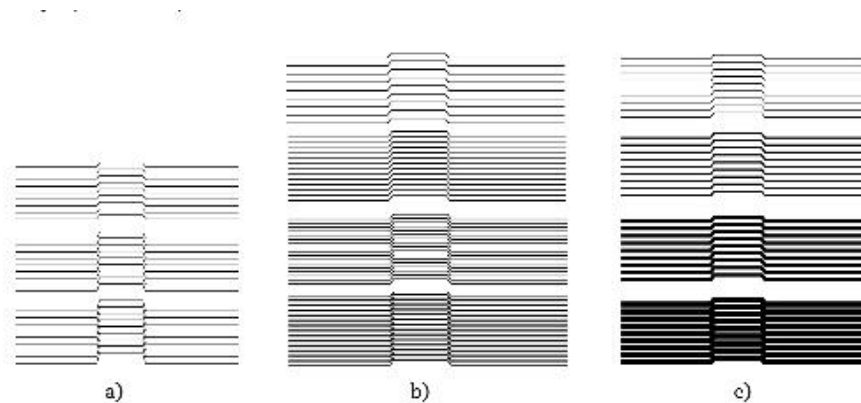
Giljoše se i danas zadržao kao efikasna metoda zaštite koja objedinjuje i vizualne i kao i zaštitne potrebe. Koristi se i u svrhu izrade motiva na novčanicama, najčešće portreta (slika 1).



Slika 1 - prikaz linijske grafike na novčanici

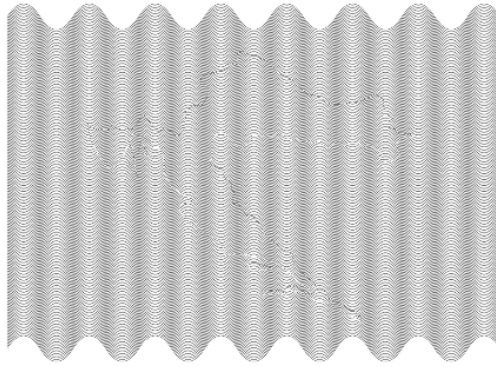
Autentičnost linijske grafike lako je dokazati. Jednostavnom provjerom linije dobivamo odgovor radi li se o originalu ili kopiji, a to je ujedno i najveća prednost linijske grafike na rasterskom.

Računalna linijska grafika može se stvoriti pomoću raznih grafičkih programskih jezika poput Postscripta. Može se izvesti korištenjem ravnih linija kojima se mijenja, nagib, linijatura i debljina linija (slika 2).



Slika 2 - promjena zacrnjenja povećanjem: a) nagiba, b) linijature, c) debljine linije

Također se mogu primijeniti i sinusoidne linije. Kao i kod ravnih linija i kod sinusoida povećanjem linijature, nagiba te debljine linije dobivaju se tamnija područja pokrivenosti. Parametri za linearizaciju sinusoidnim krivuljama su: linijatura, debljina, zacrnjenje, skok linije, amplituda i period. Slika 3 prikazuje primjer grafike izvedene sinusoidnim linijama. [3.]



Slika 3 - primjer sinusoidne linijske grafike

3.1.1. Izrada portreta na vrijednosnicama linijskom grafikom

Prva poštanska marka, kao sredstvo plaćanja poštarine, nastala je 1840. godine u Velikoj Britaniji. Nazvana je „Crni peni“ jer je vrijedila jedan peni i bila otisnuta crnom tintom. Prije uvođenja poštanske marke poštarinu je plaćao primatelj pisma. Neki su, želeći izbjeći plaćanje poštarine, na omotnicama pisali skrivene poruke. Primatelj pisma uzeo bi omotnicu, pročitao poruku te pismo samo vratio poštonoši. Plaćanjem poštarine prilikom slanja pisma doskočilo se ovom problemu. Prva poštanska marka na sebi je imala portret kraljice Viktorije izrađen linijskom grafikom (slika 4). [4.] [5.]



Slika 4 - prva poštanska marka

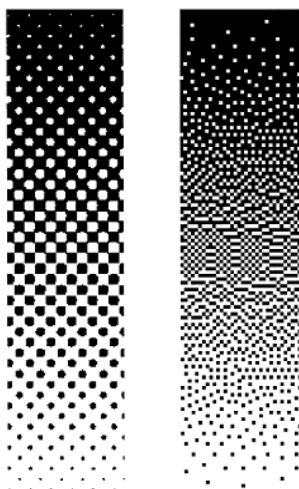
Portreti se vrlo često nalaze na novčanicama kao dio dizajna, a proces njihove implementacije na novčanice poprilično je zahtjevan posao. Portret mora biti prepoznatljiv i mora vjerodostojno prikazati osobu koju predstavlja. Nekoć su se portreti crtali ručno, direktnim graviranjem u čeliku ili bakru i na taj način ih se vrlo teško moglo krivotvoriti. Slika 5 prikazuje ručno gravirani portret na njemačkoj novčanici s početka 20. stoljeća, koji je napravljen pomoću linijske grafike.



Slika 5 - njemačka novčanica napravljena linijskom grafikom

3.2. ZAŠTITNA RASTERSKA GRAFIKA

Zaštitna rasterska grafika nastala je otkrićem rastera. U počecima su se kao rasteri koristili jednostavni oblici točke, krugovi, trokuti. Na taj način rasterska grafika predstavlja veliki stupanj zaštite pojedinog dokumenta. Rasteri se mogu podijeliti na amplitudalno modulirane rastere (AM raster) i frekvencijski modulirane rastere (FM raster) (slika 6). Kako bismo dobili individualizirani rasterski oblik, potrebno je osmisliti matematičku funkciju te prilagoditi oblik, kut i linijaturu za svaku CMYK separaciju. Tisak takvih rastera se uglavnom izvodi pomoću spotnih boja koji predstavljaju još jednu metodu zaštite, a uz njih se koriste i niske linijature kako bi se dobio pravilni i oštri oblik rasterskog elementa. [6.]



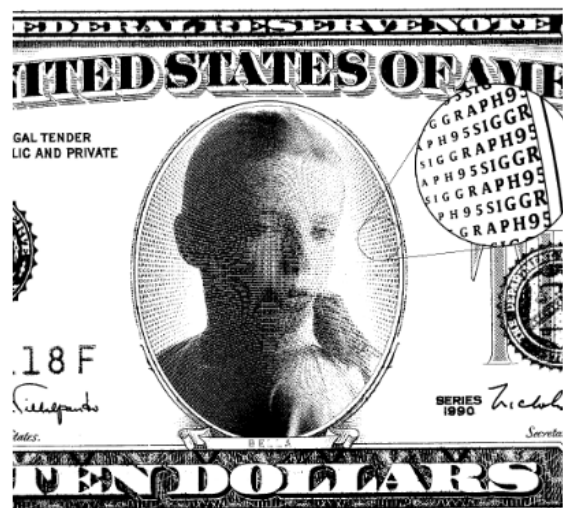
Slika 6 - AM raster (lijevo), FM raster (desno)

Otkrićem novih zaštitnih elemenata povećala se mogućnost stvaranja kvalitetnih i jedinstvenih dizajnerskih rješenja. Izmjena piksela moguća je jednostavnim programskim rješenjima koja u potpunosti kontroliraju elemente zaštitne grafike. Takvi programi su načinjeni na taj način da se svakom pikselu pristupa pojedinačno unutar slike. Stohastičke metode omogućuju promjenu tipa, linijature, kuta, boje i oblika rastera. [7.]

Viktor Ostromoukhov i Roger D. Hersch razvili su 1995. godine raster koji se umjesto točkica sastojao od motiva riba i ptica (slika 7), također su klasične rasterske elemente zamijenili tipografskim elementima (slika 8). Takav individualizirani raster može se koristiti u umjetničke svrhe ili u svrhu zaštitu umjetničkog djela. S obzirom na to da se slika koju je potrebno reproducirati i raster stvaraju neovisno, sloboda dizajniranja je gotovo neograničena. [8.]



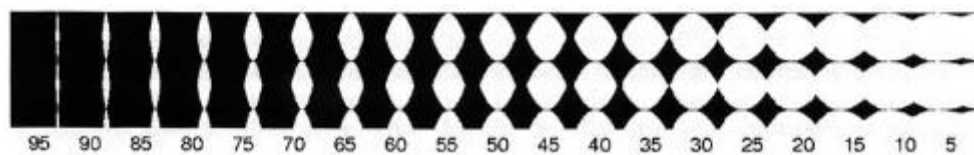
Slika 7 - primjer individualiziranog umjetničkog rastera



Slika 8 - primjer rastera s tipografskim elementima

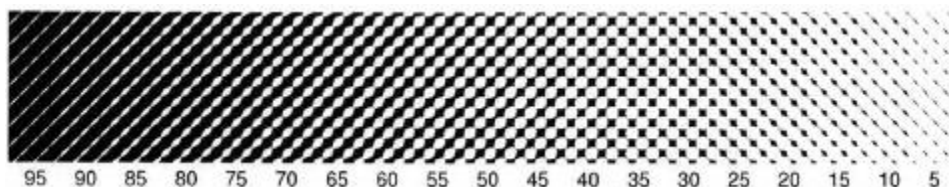
3.2.1. Romboidni rasterski element

Pojavom računala i brojnih grafičkih programskih jezika moguće je stvoriti razne individualizirane rasterske oblike. Romboidni rasterski element jedan je od primjera zaštite pomoću rasterskih elemenata. Glavno obilježje mu je promjena oblika prilikom pokrivanja neke površine. Ovaj element za minimalnu pokrivenost površine od 5% do 15% poprima oblik tanke horizontalne linije pri kutu rastriranja od 0°. Pri pokrivenosti površine između 20% i 75% ovaj rasterski element poprima kružni oblik, a pri maksimalnoj pokrivenosti površine između 80% i 95% postaje vertikalna linija (slika 9).



Slika 9 - prikaz pokrivenosti površine romboidnim rasterom od 5% do 95%

Ovim rasterom se postiže osjećaj promjene kuta tijekom procesa pokrivanja površine, što također pruža još jedan način zaštite. (slika 4.)



Slika 10 - prikaz romboidnog rastera pri kutu rastriranja od 45°

Romboidni raster je zadan matematičkom jednačbom:

$$Z = Abs(\sqrt{Abs(y)} - \sqrt{Abs(y + \sin(2x) + y)}),$$

a u Postscriptu ta jednačba glasi:

```
/SoftRomb {exch 2 mul sin exch dup 3 1 roll add abs sqrt exch abs sqrt exch  
sub abs} def [9.]
```

3.2.2. Sinusoidni rasterski element

Sinusoidni rasterski element se može koristiti kao zaštita u dokumentima i na novčanicama, ali se također može koristiti i u dizajnu. Sinusoidni rasterski elementi pronašli su svoju primjenu na našim novčanicama. Mogu se primjenjivati u vektorskoj grafici, piksel grafici te u tipografiji.

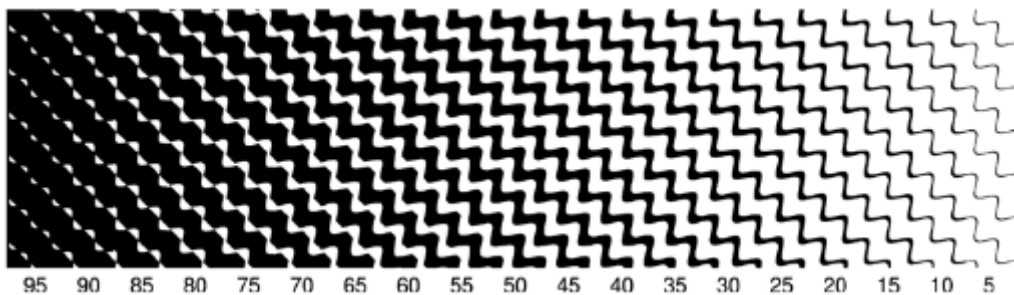
Matematički su zadani jednadžbom:

$$RS = 0.5[1 - \sin(x\pi)],$$

a prevedena u Postscript jezik ta jednadžba glasi:

```
/rSinus {neg exch 180 mul sin 2 div add 3 div abs 1 exch su} bind def.
```

Slika 11 prikazuje kontinuirani prijelaz pokrivenosti površine rasterom od 95% do 5% s linijaturom od 5 lpi (linija po inču) i kutom rastriranja od 45°.



Slika 11 - pokrivenost površine sinusoidnog rastera od 5% do 95%

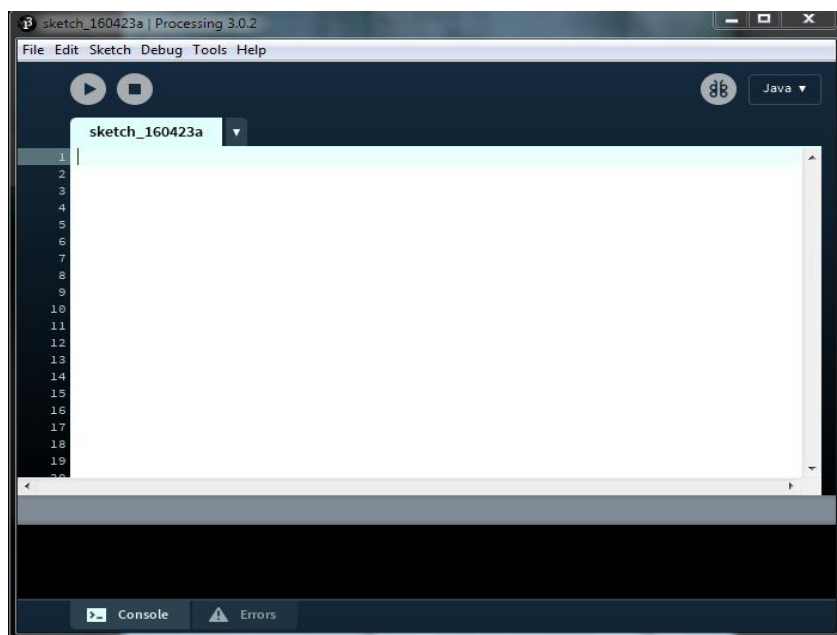
Promjenom vrijednosti varijabli u navedenoj jednadžbi moguće je proizvesti razne sinusoidne oblike. Kada se koristi više sličnih rasterskih oblika, moguće je dobiti kompleksnije rješenje. Cilj ovakvih rastera je postizanje nemogućnosti kopiranja. Takva rješenja kopiranjem gube svoju finoću pa kopirana slika u boji sadrži točke koje su nastale u CMYK sustavu otiskivanja. Također, nije moguće kopirati elemente bez saznanja o njihovim Postscript linijaturama, kutevima i individualiziranim parametrima iz njihovih matematičkih relacija. Uvođenjem stohastičkih brojeva i generiranjem slučajnih sekvenci postižu se rezultati koje je nemoguće reproducirati bez znanja svih veličina i parametara koje sačinjavaju jednadžbe. [10.]

4. PROCESSING PROGRAMSKI JEZIK

Processing je open source programski jezik i integrirano razvojno okruženje (IDE) koji su osmislili Ben Fry i Casey Reas 2001. godine. Napravljen za elektroničku umjetnost, umjetnost novih medija i vizualni dizajn s ciljem pružanja jednostavnog pristupa programiranju vizualnim umjetnicima. Temelji se na programskom jeziku Java, ali koristi jednostavniju sintaksu i grafički model programiranja. Programi napravljeni u Processingu zovu se „skice“ (sketches), iz tog razloga Processing se može shvatiti kao okruženje za brzo stvaranje digitalnih skica. [11.] [12.]

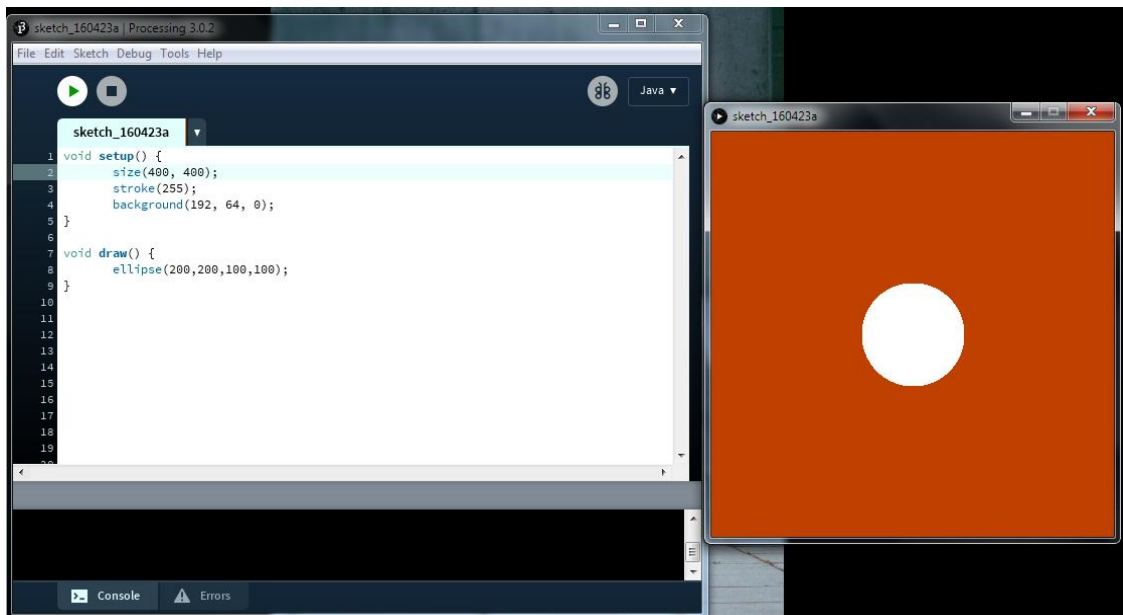
Processing povezuje koncepte softvera s principima vizualnog oblikovanja, pokreta i interakcije. U njemu se integriraju programski jezik, razvojno okruženje i metodologije učenja u jedinstven sustav. Osmišljen je kako bi se njime mogli generirati i modificirati slikovni zapisi, također pazi na ravnotežu između jasnoće prikaza i naprednih mogućnosti. Zahvaljujući tome, početnici mogu napisati kratke funkcionalne programe nakon samo nekoliko minuta praćenja uputa, dok oni iskusniji korisnici mogu kreirati vlastite datoteke i funkcije. Sustav olakšava učenje brojnim tehnikama računarske grafike i interakcije uključujući vektorsku i rastersku grafiku, obradu slike, modele boja, pokrete miša i korištenje tipkovnice, mrežnu komunikaciju i objektno-orijentirano programiranje. [13.] [14.]

Processing pruža vrlo jednostavno razvojno okruženje te se prilikom pokretanja otvara jednostavan prozor (slika 12).



Slika 12 - osnovni Processing prozor

Programski kod se unosi u središnji bijeli prozor. Iznad se nalazi alatna traka koja sadrži gumbe koji služe za pokretanje i zaustavljanje programa ili za spremanje, učitavanje i izvoz programa. Informacije i obavijesti o greškama u programu se nalaze na dnu u konzoli. Kada se napiše i pokrene program, ako u njemu nema nikakvih greški, pojavljuje se još jedan prozor (korisničko sučelje) u kojem se pokreće program. (slika 13)

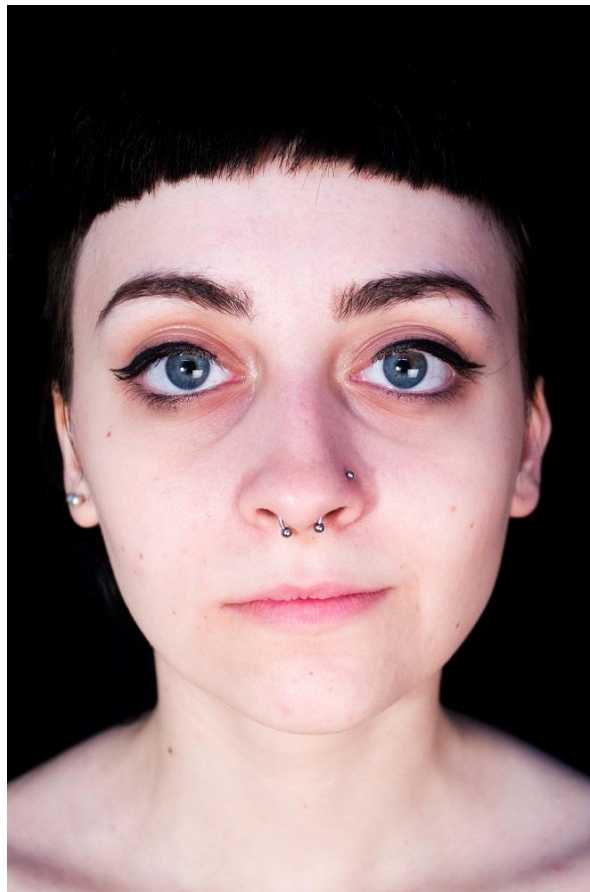


Slika 13 - pisanje i pokretanje programa u Processingu

Veličina korisničkog sučelja definira se naredbom "*size()*" unutar koje se piše dva parametra, a to su širina i visina prozora. Processing je postavljen u mod Java, što se vidi u gornjem desnom kutu glavnog prozora. Budući da je u njegovo programsko okruženje uključeno više programskih jezika, postoji mogućnost promjene moda kao što su JavaScript, Python, Android. [15.] [16.]

5. EKSPERIMENTALNI DIO

U eksperimentalnom dijelu ovoga rada korišten je program Processing. U eksperimentu se koristi portret (slika 14) kod kojeg klasični pikseli mijenjaju svoj oblik u krugove, linije ili kvadrate. Pomakom miša ovisno o udaljenosti od ishodišta mijenja se njihova duljina, debljina i boja. Ishodište slike nalazi se u gornjem lijevom uglu prozora u kojem se slika pojavljuje. Slika je smanjena na veličinu 64x97 piksela, a prozor u kojem se otvara je veličine 500x800 piksela, tako se dobije bolji prikaz piksela. Prikazane su po tri slike za svaki primjer. Prva slika u svakom primjeru prikazuje rezultat koji se dobiva kada je strelica miša u blizini ishodišta, druga kada je na sredini prozora, a treća prikazuje rezultat koji se dobiva kada se strelica miša nalazi u donjem desnom kutu, odnosno u krajnjoj točki od ishodišta.



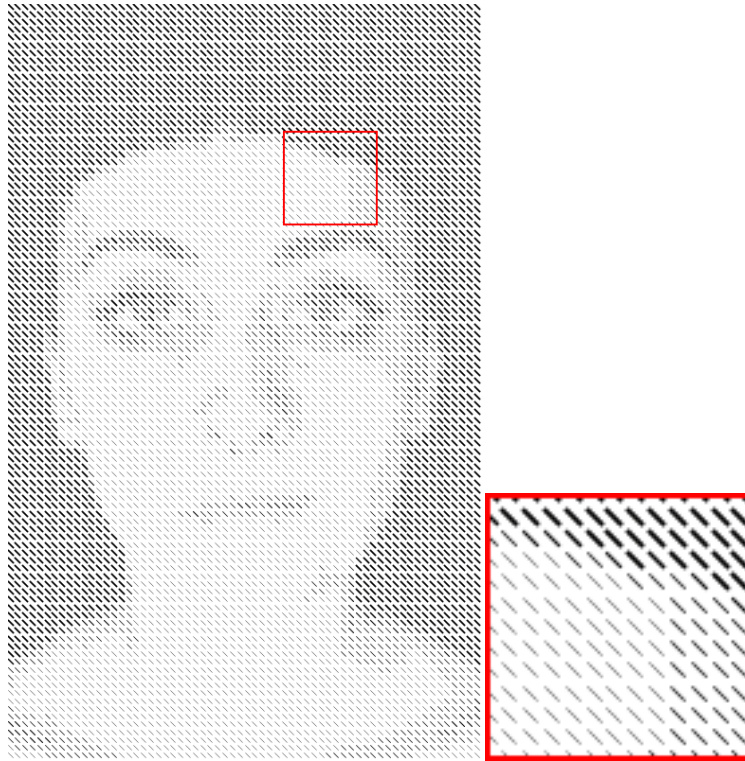
Slika 14 - portret korišten u eksperimentalnom dijelu

5.1. IZRADA PORTRETA S LINIJSKIM ELEMENTIMA

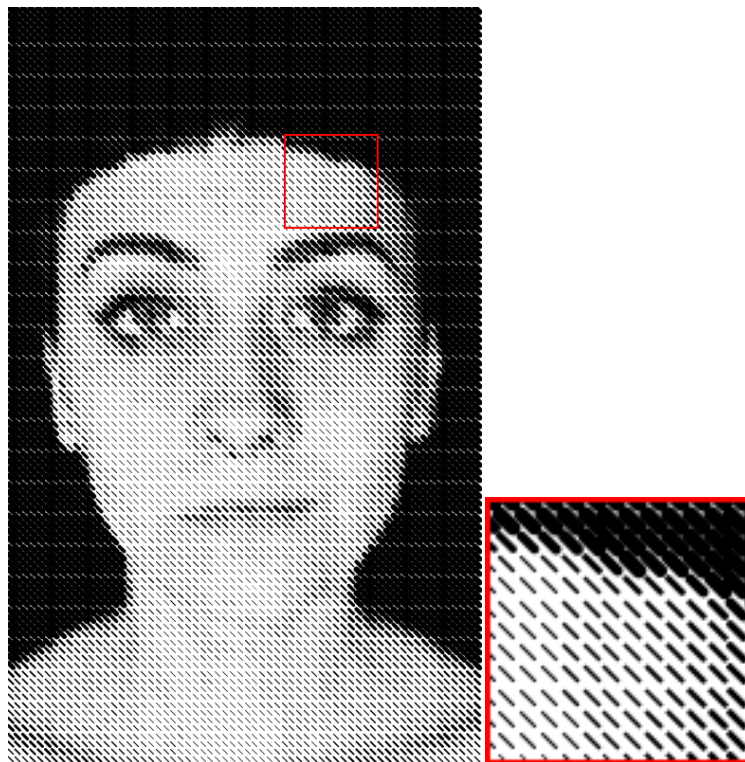
5.1.1. Promjena vrijednosti sivih tonova ovisno o debljini linije

```
case 1:  
float w1 = map(greyscale, 0,255, 10,0.1);  
stroke(0);  
strokeWeight(w1 * pomakX);  
line(posX, posY, posX+5, posY+5);  
break;
```

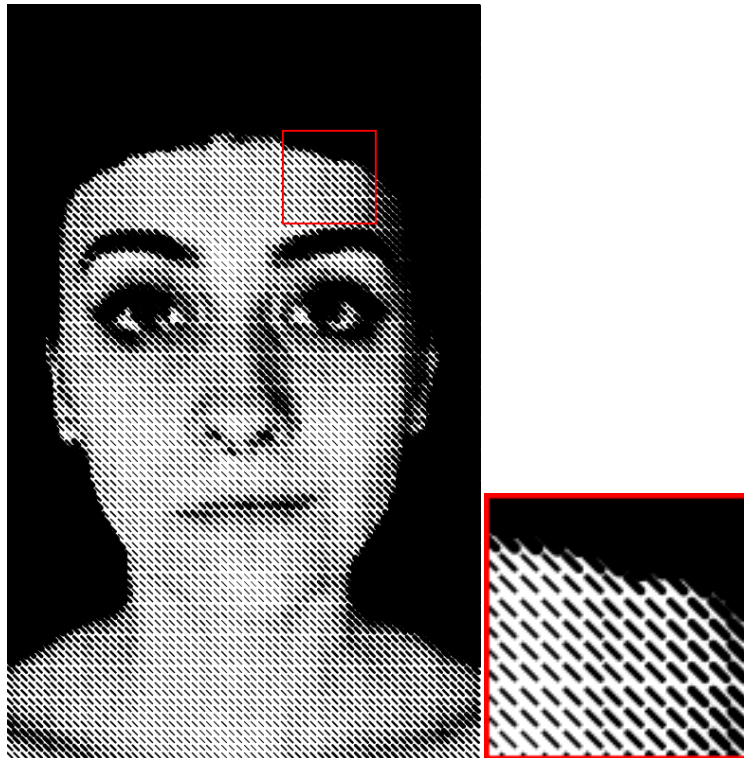
U prvom slučaju klasične kvadratne piksele zamjenjuju linije. Raspon sivih tonova inače se nalazi u intervalu od 0 do 255, ali zbog bolje reprezentativnosti rezultata taj se interval raspona sivih tonova prebacuje u interval od 10 do 0.1 pomoću naredbe 'map'. Boja linija je crna, a to je definirano naredbom 'stroke(0)'. Debljina linija ('strokeWeight') dobivena je umnoškom varijable 'w1' koja predstavlja iznos greyscale vrijednosti i varijable 'pomakX' koja označava pomak strelice miša po osi X. Linija se crta pomoću naredbe 'line' i 4 parametra od kojih prva dva označavaju početnu točku linije, a posljednja dva označuju završnu točku te linije. Varijable 'posX' i 'posY' označuju položaj određenog piksela u koordinatnom sustavu. Što je strelica miša udaljenija od ishodišta po osi X, to je debljina linije veća. Na slici 15 strelica miša je najbliža ishodištu pa su linije stoga najtanje, kako se udaljuje od ishodišta po osi X, one postaju sve deblje, a samim time se slika doima tamnijom (slika 16). Na slici 17 strelica miša se nalazi pri samom desnom rubu slike što znači da je najudaljenija od ishodišta po osi X pa su zato linije najdeblje i slika je najtamnija.



Slika 15 - Primjer 1: položaj miša na ishodištu i označeni detalj



Slika 16 - Primjer 1: položaj miša u sredini prozora i označeni detalj



Slika 17 - Primjer 1: položaj miša u krajnjoj točki i označeni detalj

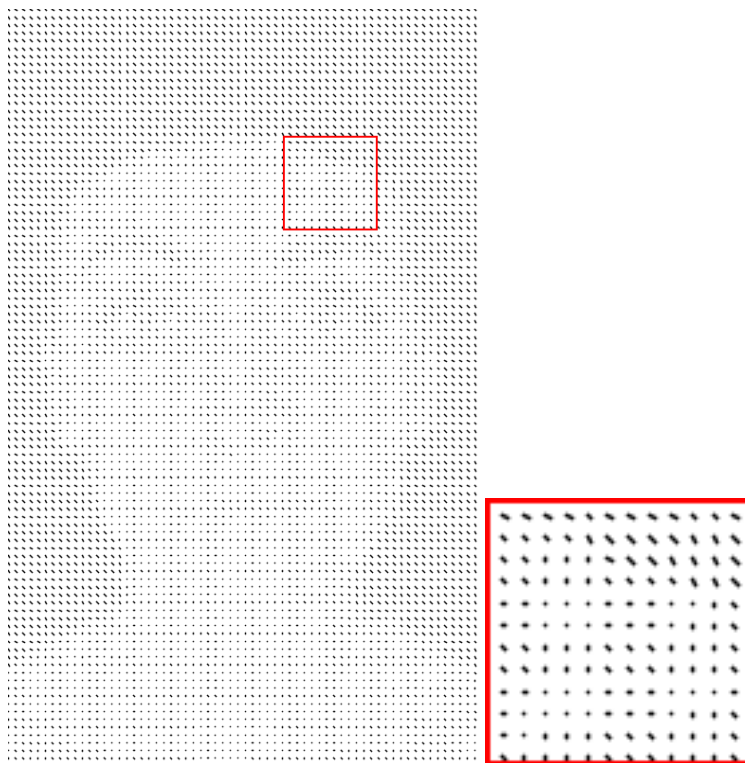
5.1.2. Promjena vrijednosti sivih tonova ovisno o duljini i debljini linije

```
case 2:  
float l2 = map(greyscale, 0,255, 10,0.1);  
l2 = l2 * pomakX;  
stroke(0);  
strokeWeight(10 * pomakY);  
line(posX, posY, posX+l2, posY+l2);  
break;
```

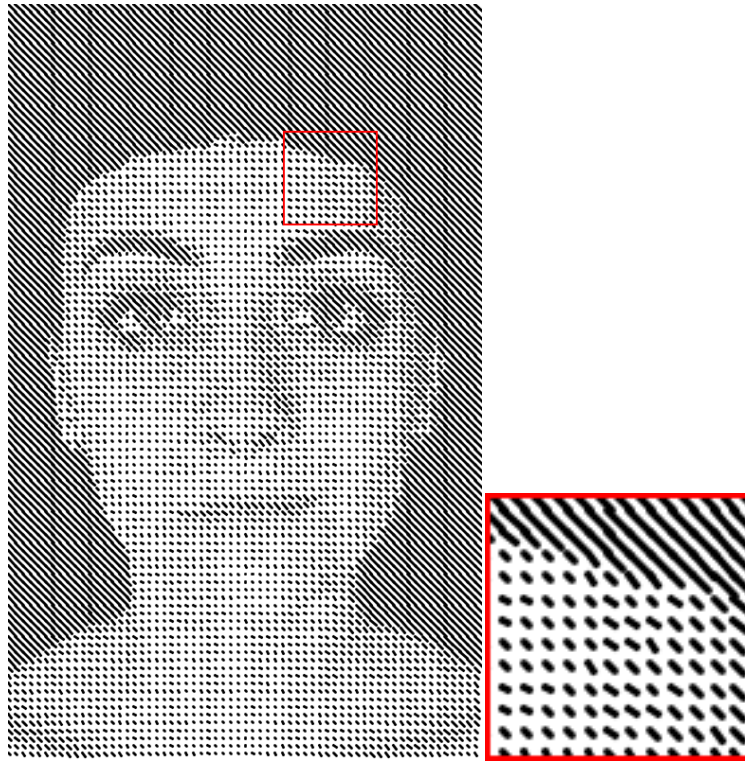
U drugom primjeru mijenjaju se i duljina i debljina linije. Pomicanjem strelice miša po osi X mijenja se duljina linija, a po osi Y mijenja se debljina linija. I u ovom primjeru se koristi nareda 'map' kako bi vrijednosti sivih tonova poprimile iznose između 10 i 0.1. Varijabla 'l2' je definirana umnoškom vrijednosti sivih tonova i pomaka miša u smjeru osi X te prikazuje duljinu linije. Širina linije definira na taj način da je za svaki pomak miša u smjeru osi Y ta debljina 10 puta veća no ona prijašnja. Početna točka linije je definirana pomoću varijabli 'posX' i 'posY', a završna se definira kao udaljenost u iznosu vrijednosti varijable 'l2' od početne točke.

Na slici 18 strelica miša se nalazi u blizini ishodišta pa su linije u tom slučaju najkraće i najtanje. Pomicanjem miša i udaljavanjem strelice od ishodišta linije postaju dulje i

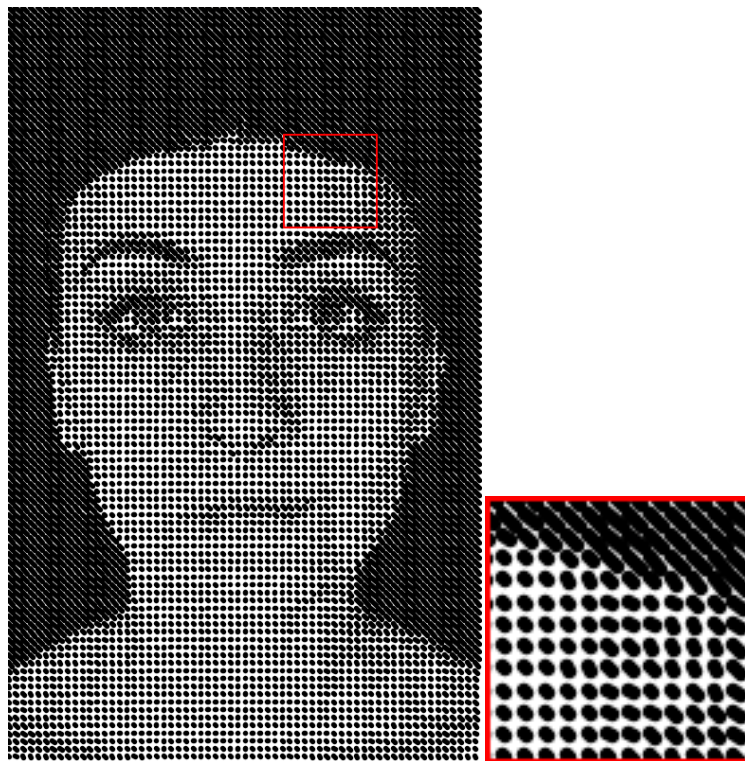
deblje nego u početku (slika 19), a kada se strelica miša pomakne u donji desni kut prozora, gdje je ona ujedno najudaljenija od ishodišta linije postaju najdeblje i najdulje, a samim time i cijela slika postaje najtamnija (slika 20).



Slika 18 - Primjer 2: položaj miša na ishodištu i označeni detalj



Slika 19 - Primjer 2: položaj miša u sredini prozora i označeni detalj



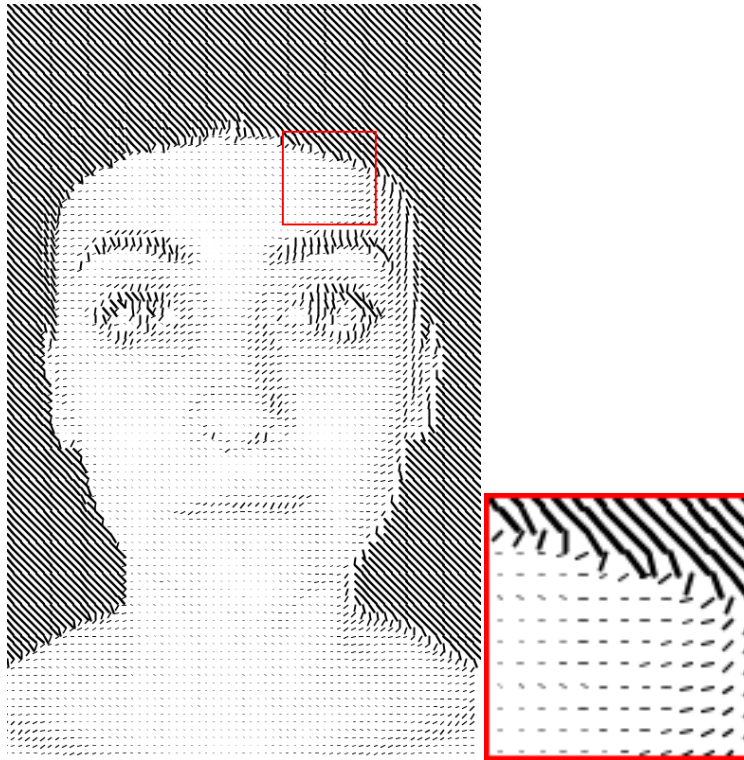
Slika 20 - Primjer 2: položaj miša u krajnjoj točki i označeni detalj

5.1.3. Promjena vrijednosti sivih tonova ovisno o smjeru, duljini i debljini linije

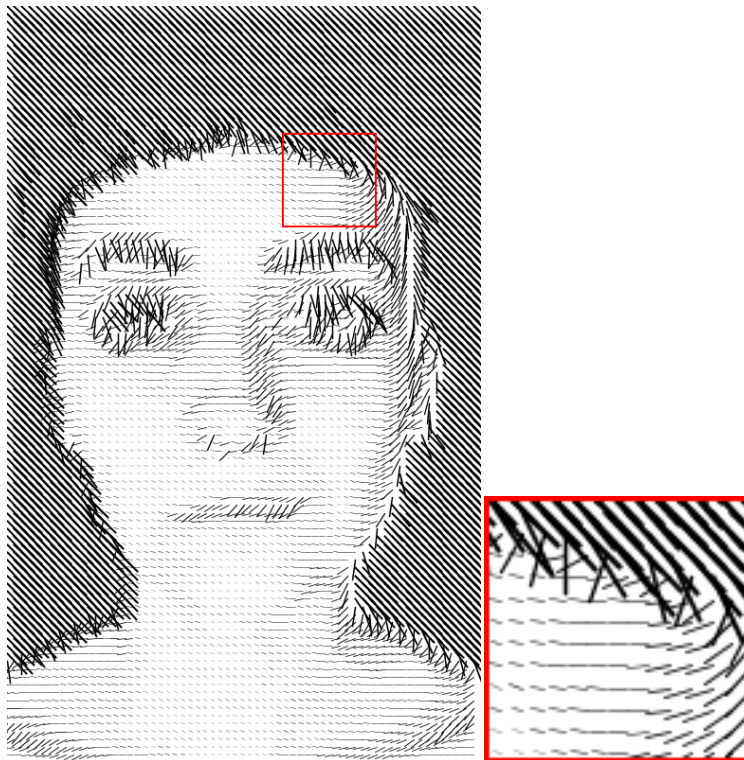
```
case 3:
stroke(0);
float w3 = map(greyscale, 0,255, 10,0);
strokeWeight(w3 * pomakX + 0.1);
float l3 = map(greyscale, 0,255, 25,0);
l3 = l3 * pomakY;
pushMatrix();
translate(posX, posY);
rotate(greyscale/255.0 * PI);
line(0, 0, 0+l3, 0+l3);
popMatrix();
break;
```

U trećem primjeru se uz promjenu duljine i debljine linije mijenja njihov smjer. Linije su crne boje. Naredba `'strokeWeight'` definira debljinu linije koja je određena umnoškom varijable `'w3'` i vrijednosti pomaka miša po osi X, a na taj umnožak se još dodaje i vrijednost 0.1. Duljina linije se definira pomoću varijable `'l3'` na način da se vrijednost sivih tonova množi s varijablom `'pomakY'` koja predstavlja vrijednosti pomaka miša uzduž osi Y. U ovom primjeru se pojavljuju naredbe `'pushMatrix()'` i `'pushPop()'`. Naredba `'pushMatrix()'` sprema trenutni položaj ishodišta koordinatnog sustava u memoriju (`'stack'`), a naredba `'popMatrix()'` vraća taj položaj na početnu vrijednost. Promjena položaja ishodišta provodi se pomoću naredbe `'translate'`. Smjer linije određuje se pomoću naredbe `'rotate'`. Linije se iscrtavaju naredbom `'line'`, početna točka linije je u ishodištu koordinatnog sustava, a završna točka se nalazi udaljeno za vrijednost varijable `'l3'` po osi X i osi Y.

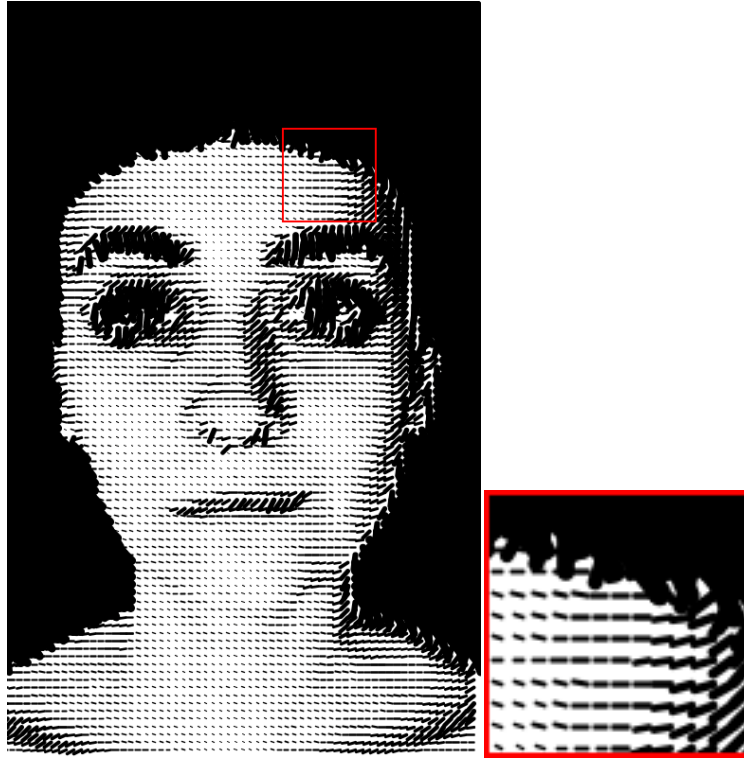
Na slici 21 strelica miša se nalazi u blizini ishodišta pa su linije najtanje i najkraće. Na slici 22 strelica miša je pomaknuta u donji lijevi kut pa su se linije produljile, ali debljina im je ostala gotovo ista, a na slici 23 strelica miša je pomaknuta u desni donji kut pa se linijama osim duljina povećala i debljina i slika djeluje najtamnije.



Slika 21 - Primjer 3: položaj miša na ishodištu i označeni detalj



Slika 22 - Primjer 3: položaj miša u sredini prozora i označeni detalj



Slika 23 - Primjer 3: položaj miša u krajnjoj točki i označeni detalj

5.2. IZRADA PORTRETA S KRUŽNIM OBLICIMA

5.2.1. Promjena vrijednosti sivih tonova ovisno o radijusu kružnih oblika

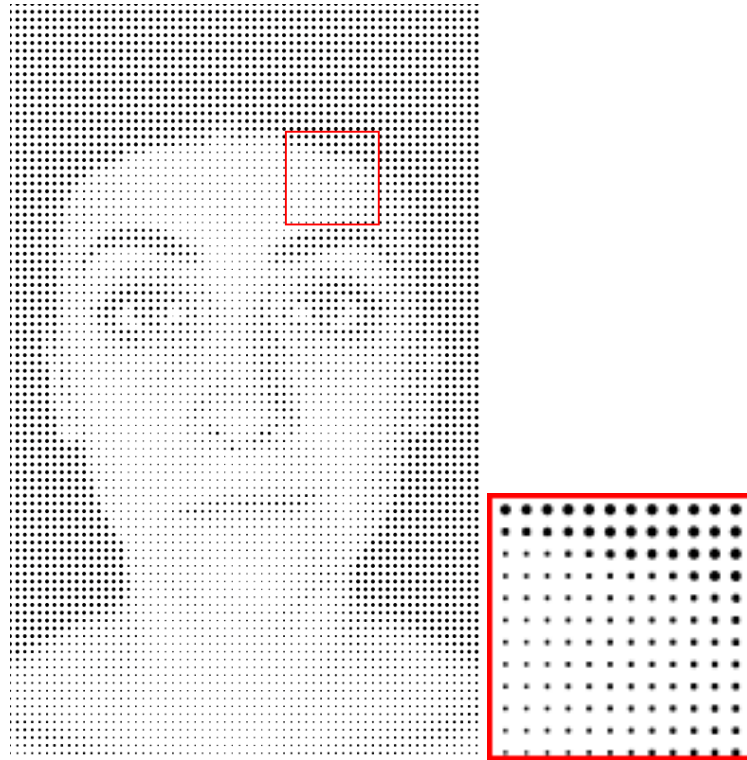
```

case 4:
fill(0);
noStroke();
float r4 = 1.5 * sqrt(tileWidth*tileWidth*(1-greyscale/255.0));
r4 = r4 * pomakX ;
ellipse(posX, posY, r4, r4);
break;

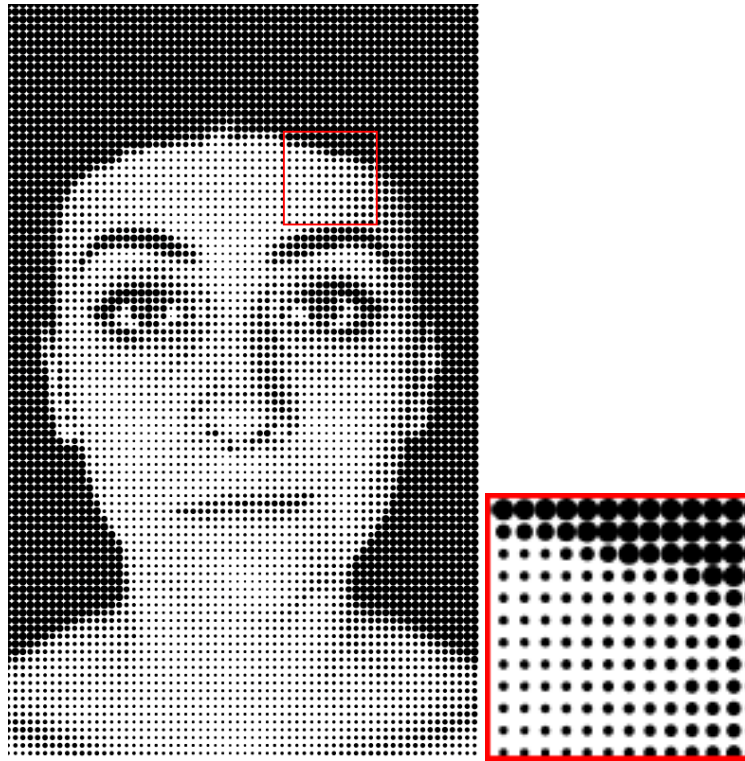
```

U četvrtom primjeru klasične piksele zamjenjuju kružni oblici. Pomoću naredbe 'fill' definira se boja tih oblika, u ovome slučaju crna. Varijabla 'r4' predstavlja radijus kružnog oblika. Kružni oblik zadaje se naredbom 'ellipse' pomoću 4 parametra od kojih prva dva predstavljaju točku središta kružnog oblika, treći predstavlja radijus kružnog oblika po osi X, a četvrta varijabla predstavlja radijus kružnog oblika po osi Y.

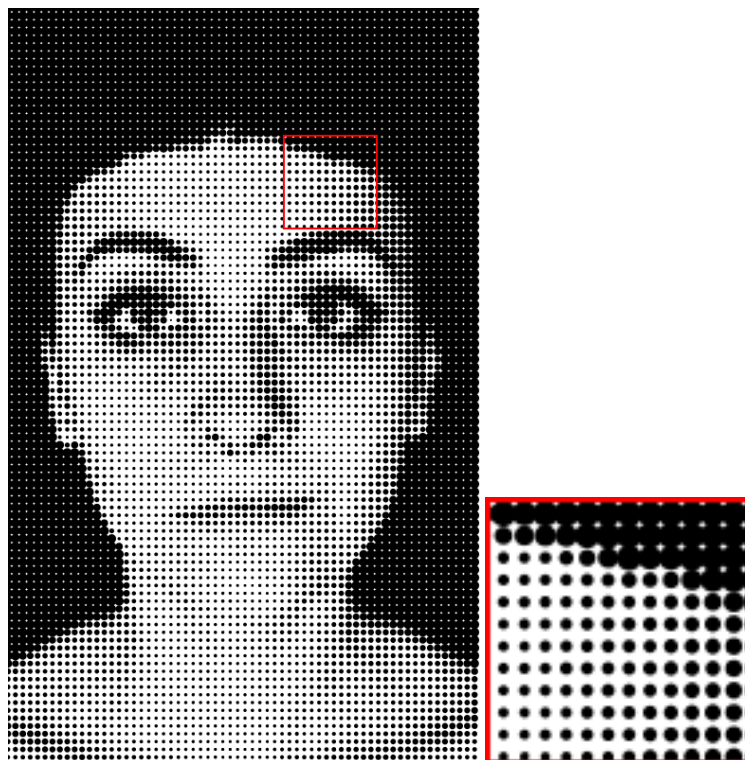
U ovome slučaju dok je strelica miša u blizini ishodišta, pikseli imaju najmanji radijus pa je slika najsvjetlija (slika 24). Pomicanjem po osi X, radijus se povećava (slika 25), a kada strelica miša dođe do desnog ruba slike, radijus poprima maksimalnu vrijednost pa je i sama slika najtamnija (slika 26).



Slika 24 - Primjer 4: položaj miša na ishodištu i označeni detalj



Slika 25 - Primjer 4: položaj miša u sredini prozora i označeni detalj



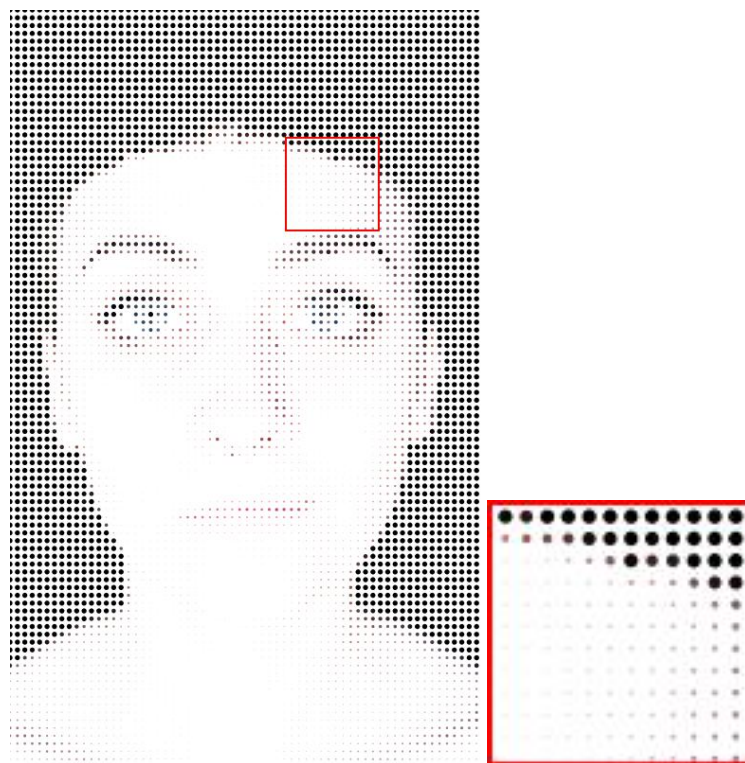
Slika 26 - Primjer 4: položaj miša u krajnjoj točki i označeni detalj

5.2.2. Promjena vrijednosti sivih tonova i boje ovisno o radijusu kružnih oblika

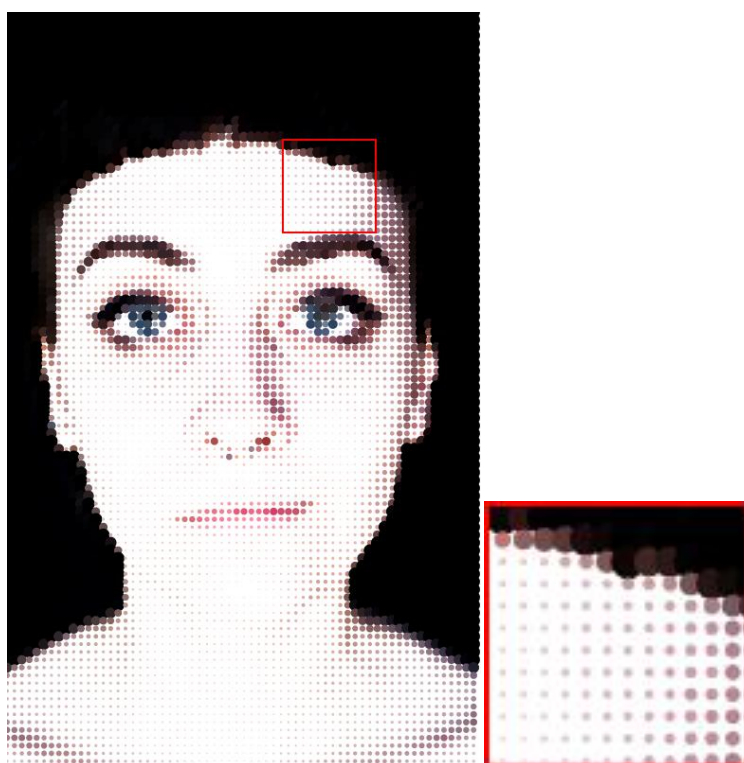
```
case 5:  
float r5 = map(greyscale, 0,255, 20,0);  
noStroke();  
fill(c);  
ellipse(posX, posY, r5 * pomakX, r5 * pomakX);  
break;
```

U petom primjeru kružnim oblici imaju i boju. Varijabla 'r5' predstavlja vrijednosti sivih tonova premapiranih iz intervala od 0 do 255 u interval od 20 do 0. Naredbom 'fill' definira se boja pojedinih oblika, a u ovom slučaju je ona zadana varijablom 'c'. Naredbom 'ellipse' iscrtava se kružni oblik koji ima središte u točki 'posX', 'posY', s radijusom koji je i po osi X i po osi Y definiran kao umnožak varijable 'r5' i varijable 'pomakX' tako da se pomicanjem miša po osi X taj radijus povećava.

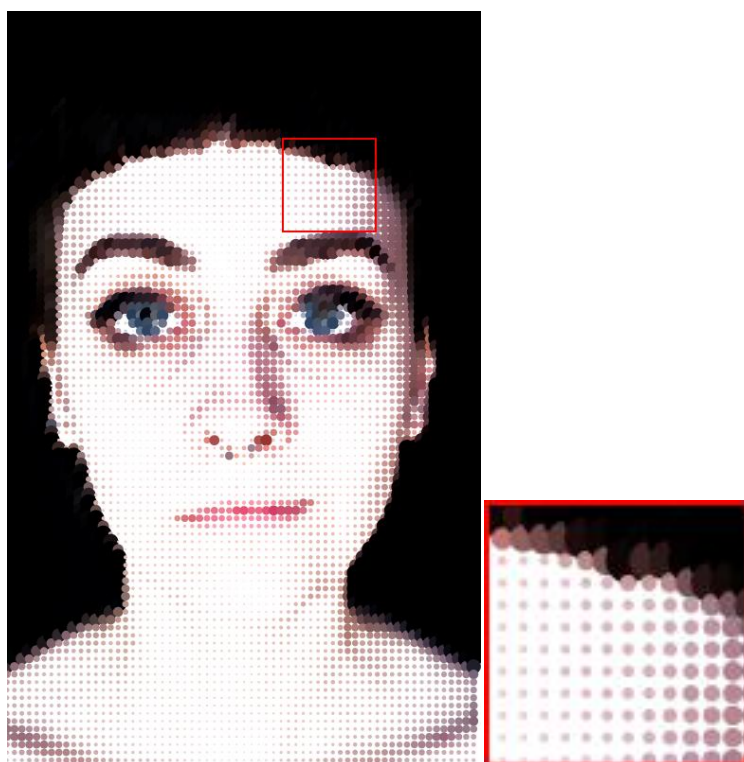
U ishodištu kružni elementi imaju najmanji radijus pa je slika najsvjetlija (slika 27). Pomicanjem strelice miša po osi X dolazi do povećanja radijusa elemenata (slika 28). Radijus postiže maksimalnu vrijednost kad je strelica miša na desnom rubu slike pa i sama slika postaje najtamnija (slika 29).



Slika 27 - Primjer 5: položaj miša na ishodištu i označeni detalj



Slika 28 - Primjer 5: položaj miša u sredini prozora i označeni detalj



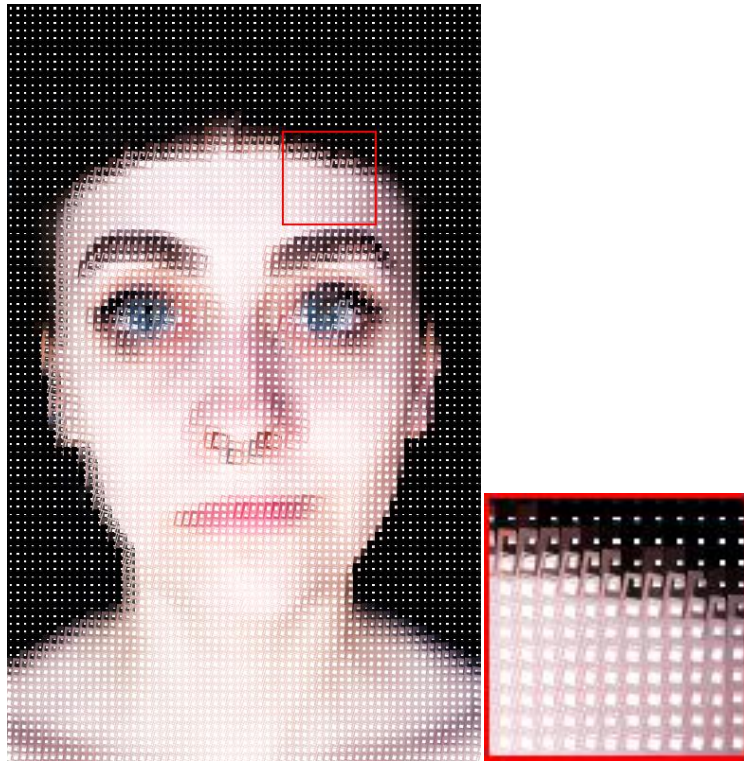
Slika 29 - Primjer 5: položaj miša u krajnjoj točki i označeni detalj

5.3. IZRADA PORTRETA S KVADRATIČASTIM OBLICIMA

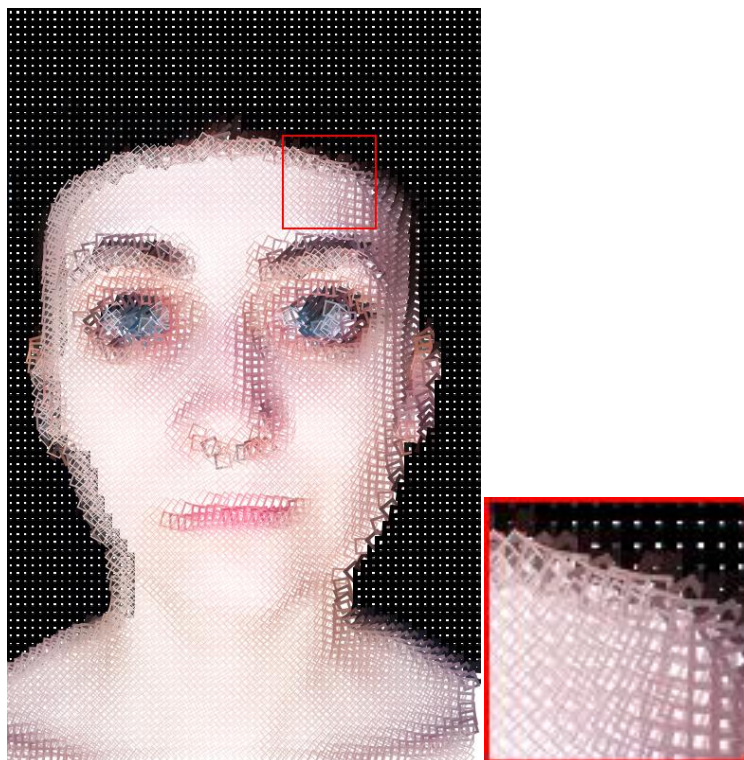
```
case 6:
stroke(c);
float w6 = map(greyscale, 0,255, 5,0.1);
strokeWeight(w6);
fill(255,255* pomakY);
pushMatrix();
translate(posX, posY);
rotate(greyscale/255.0 * PI* pomakX);
rect(0,0,15,15);
popMatrix();
break;
```

U šestom primjeru klasične piksele zamjenjuju kvadrati. Boja obruba definirana je naredbom 'stroke' s parametrom 'c'. Debljina obruba kvadrata je definirana varijablom 'w6' koja se definira kao vrijednost sivih tonova u intervalu od 5 do 0.1. Naredbom 'fill' definira se boja unutar obruba kvadrata. Zadana je pomoću dva parametra, prvi predstavlja boju, u ovome slučaju bijelu jer ima maksimalnu vrijednost koja iznosi 255, a drugi parametar predstavlja alfa kanal odnosno opacitet te boje koji je pomnožen varijablom 'pomakY', što znači da se opacitet mijenja pomakom miša po osi Y. Kao i u primjeru 6.1.3., ovdje također imamo naredbe 'pushMatrix()' i 'pushPop()'. Kvadrati se rotiraju pomoću naredbe 'rotate'.

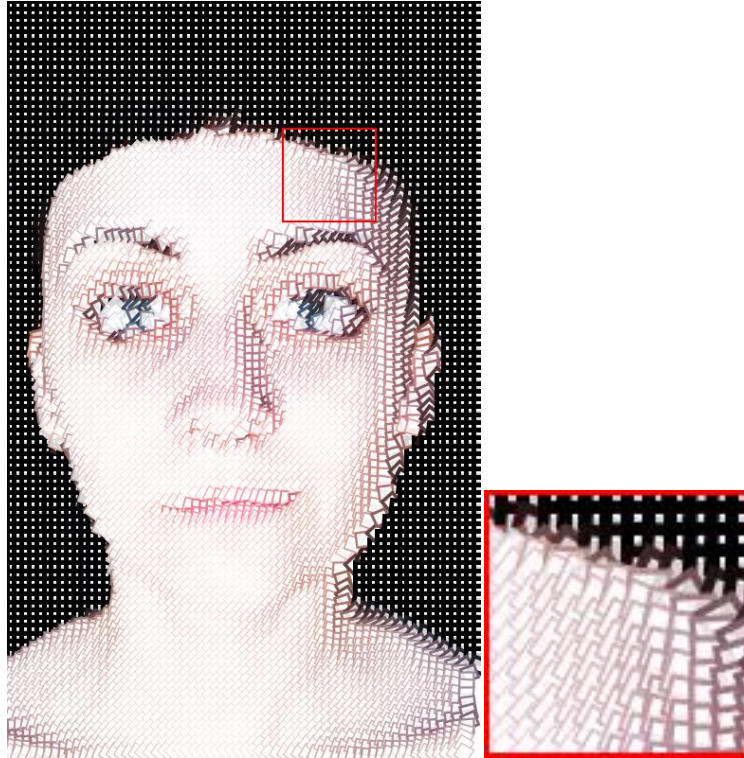
U zadnjem primjeru pomakom miša mijenja se kut kvadratnih elemenata i njihov opacitet. Na slici 30 kvadrati su minimalno rotirani, pomicanjem strelice miša po osi X oni se sve više rotiraju (slika 31), a na slici 32 su maksimalno rotirani.



Slika 30 - Primjer 6: položaj miša na ishodištu i označeni detalj



Slika 31 - Primjer 6: položaj miša u sredini prozora i označeni detalj



Slika 32 - Primjer 6: položaj miša u krajnjoj točki i označeni detalj

6. REZULTATI I RASPRAVA

Izvedeni su primjeri na kojima su implementirane programske reprodukcije portreta. Prikazani su eksperimenti upotrebe takvih portreta na osobnim dokumentima i na vrijednosnicama. Za osobne dokumente uzeti su uzorci hrvatske vozačke dozvole i osobne iskaznice dok su za vrijednosnice uzeti primjeri od 100 hrvatskih kuna, 5 australskih dolara te poštanska marka.



Slika 33 - implementacija eksperimentalno dobivenog portreta na vozačkoj dozvoli

Portret izabran za apliciranje na vozačkoj dozvoli vjerodostojno prikazuje lice osobe što je i osnovni uvjet osobnih dokumenata. Iz eksperimentalnog dijela rada izabrana je slika 29. koja zadovoljava uvjet prepoznatljivosti osobe dok programski definirani pikseli u portretu osiguravaju dodatnu zaštitu. Na ovakve dokumente mogu biti aplicirani portreti u boji što olakšava identifikaciju posebno boju očiju što je kod ovog portreta jako naglašeno. Isto tako mogu biti aplicirani i crno bijeli portret kao što je prikazano na osobnoj iskaznici.



Slika 34 - implementacija eksperimentalno dobivenog portreta na osobnoj iskaznici

Za osobnu iskaznicu izabran je portret (slika 26) koji također vjerodostojno prikazuje osobu, a programski definirani pikseli osiguravaju dodatnu zaštitu. Najčešće krivotvorenje osobnih dokumenata se izrađuje apliciranjem tuđe fotografije na ukradene osobne dokumente. Ovakav pristup programske kontrole portreta osigurava jedinstvenost i originalnost uz vrlo laku provjeru autentičnosti. Vizualno bez dodatnih pomagala je uočljiva mreža vektorskih oblika koja čini strukturu portreta što olakšava provjeru originalnosti. Isto tako istovremeno je vrlo jasno prepoznatljiva reprodukcija originalnog portreta. Slike na kojima se strelica miša nalazi u ishodištu nemaju dovoljno detalja da bi se mogle koristiti kao portreti na dokumentima i vrijednosnicama pa se preporučuje koristiti one koje se dobivaju kad se strelica miša nalazi oko sredine prozora.

Na vrijednosnicama mogu biti aplicirani portreti kod kojih je namjerno naglašena apstraktnija reprodukcija. Portret prikazan na novčanici od 100 kn (slika 35) izgleda apstraktno te kao takav ne bi se mogao iskoristiti na osobnim dokumentima gdje je portret važan kako bi se osoba mogla identificirati. Na novčanicama ovakav izbor portreta pruža još jedan nivo zaštite. Njegove zamršene linije osiguravaju originalnost i teško su ponovljive bez poznavanja programskih parametara.



Slika 35 - implementacija eksperimentalno dobivenog portreta na novčanici od 100 kn

Kao i u prethodnom primjeru i na novčanici australskog dolara (slika 36) bi se mogao iskoristiti apstraktni portret kao metoda zaštite novčanice. Osim zaštitnih elemenata ovakve reprodukcije portreta pružaju i estetski zanimljiva rješenja. Svojom strukturom portreti plijene pažnju korisnika.



Slika 36 - implementacija eksperimentalno dobivenog portreta na novčanici od 5 australskih dolara

Primjena ovakve reprodukcije portreta posebno dolazi do izražaja na poštanskim markama (slika 37). Portret ne mora nužno biti zaštita te vrijednosnice već jedan od načina umjetničkog izražavanja.



Slika 37 - implementacija eksperimentalno dobivenog portreta na poštanskoj marci

7. ZAKLJUČAK

U ovom radu prikazana je realizacija reprodukcije portreta preko programskih kodova kroz Processing programski jezik. Osmišljeni su programi kojima se pikseli iz originalne fotografije zamjenjuju s vektorskim oblicima kao što su linija, krug i kvadrat. Iskorištena je i mogućnost interakcije sa korisnikom na način da pomak miša utječe na izgled mreže vektorskih oblika a samim time i na izgled reprodukcije. U programskom kodu je uvjetovan raspon promjene parametara debljine, boje i duljine vektorskih oblika. Pomakom miša reprodukcija prelazi iz vjerne originalu u apstraktnu. Primjena reprodukcija vjernih originalu je na osobnim dokumentima gdje je svrha slike identificiranje osobe, a promjena oblika piksela služi kao zaštitni element. Apstraktne reprodukcije svoju primjenu nalaze na vrijednosnicama gdje u prvi plan dolazi originalnost i jedinstvenost rješenja što osiguravaju programski oblici piksela i njihova zamršena struktura.

Iz izvedenih rješenja može se zaključiti da vektorski oblici kod kojih se miš nalazi na ishodištu odnosno gdje nema pomaka daju uglavnom presvijetle rezultate te kao takvi ne mogu vjerodostojno prikazati portret osobe (slika 15, slika 18, slika 21, slika 24, slika 27). Udaljavanjem miša od ishodišta prema središtu lice osobe postaje prepoznatljivo pa takve slike mogu poslužiti kao portreti na dokumentima i vrijednosnicama (slika 16, slika 25, slika 26, slika 29). Također, približavanjem krajnjoj točki u prozoru slika postaje pretamna i ne može se koristiti kao portret na dokumentima i vrijednosnicama (slika 17, slika 20) Najrazvedenija rješenja nastaju rotacijom kvadratnih oblika (slika 31.) te promjenom smjera, duljine i debljine linije (slika 22).

8. LITERATURA

- [1.] <http://www.enciklopedija.hr/Natuknica.aspx?ID=68673>, 25.3.2016.
- [2.] <http://www.tourbywatches.com/collection/guilloch%C3%A9/>, 23.4.2016.
- [3.] B. Ampova: Diplomski rad: Programiranje linijske grafike na novčanicama, 2012.
- [4.] <http://www.posta.hr/marko-markica-jeste-li-znali>, 27.4.2016.
- [5.] <http://postalmuseum.si.edu/queen's/worldsfirststamps.html>, 27.4.2016.
- [6.] <http://www.britannica.com/topic/raster-graphics>, 23.4.2016.
- [7.] M. Rudolf, N. Stanić Loknar, M. Turčić: Pixel manipulation in security graphics, 11th international design conference Design 2010., Dubrovnik, 2010.
- [8.] V. Ostromoukhov, Roger D. Hersch: Artistic Screening, Computer Graphics Proceedings, Annual Conference Series, 1995.
- [9.] J. Žiljak-Vujić, N. Stanić Loknar, M. Rudolf, T. Koren, I. Žiljak: New mutation screen element "Soft Rhomb", International journal, Advanced engineering, 2009.
- [10.] T. Koren, V. Žiljak, M. Rudolf, N. Stanić-Loknar, Aleksandra Bernašek: Mathematical models of the sinusoidal screen family, Acta Graphica 22, 2011
- [11.] <https://processing.org/overview/>, 23.4.2016.
- [12.] C. Reas, B. Fry: Processing: A programming for visual designers and artists, MIT, 2007.
- [13.] I. Greenberg: Processing: Creative coding and computational art, 2007.

[14.] K. Terzidis: Algorithms for Visual Design Using the Processing Language, 2009.

[15.] <https://processing.org/>, 23.4.2016.

[16.] H. Bohnacker, B. Groß, J. Laub: Generative design: Visualize, program and create with Processing, New York, 2012.

Programsko upravljanje i manipulacija portretom u računarskoj grafici

Dino Priselac

Sažetak:

Prije pojave računala portreti na vrijednosnicama su se gravirali ručno. To su bili jedinstveni i vrlo teško ponovljivi linijski crteži. Danas se izrađuju računalnom grafikom. U izradi portreta u ovom radu korišten je programski jezik Processing. Napisani su programski kodovi koji pikselima iz originalnog portreta (fotografije) mijenjaju oblik u krugove, linije ili kvadrate. Korisničko sučelje omogućuje korisniku programa da pomakom miša ovisno o udaljenosti od ishodišta mijenja parametre oblika: duljinu, debljinu i boju. Na taj način reprodukcija portreta pomakom miša prelazi iz realne u apstraktnu. Ovakva programska rješenja reprodukcije portreta nalaze svoju primjenu na osobnim dokumentima ili vrijednosnicama kao što je papirnati novac.

Ključne riječi: linijski portret, računarska grafika, osobni dokumenti

Program management and manipulation of portrait in computer graphics

Dino Priselac

Summary:

Before the appearance of computer, portraits on securities were engraved by hand. Those were unique and hardly repeatable linear artworks. Today they are made by computer graphics. For creating portraits in this work, the programming language named Processing was used. Program codes for changing pixel on the original portrait (photography) to circles, lines or squares were written. User interface allows the user, depending on distance from origination, to change shape parameters like length, weight and color by moving the mouse. That way the reproduction of portrait by moving the mouse exceeds from real to abstract. These program solutions for portrait reproduction are applicated on the personal documents or securities like banknotes.

Keywords: linear portrait, computer graphics, personal documents