

OSVRT

Kodiranje sivoće piksela

U današnjem predavanju gospodin Klaudio nam je govorio kako se površina kvadrata sa stranicama a ispunjava u rasterskoj grafici. Na prvim vježbama smo radili kako se kodiraju kodne pozicije određenih znakova unutar fonta, upravo tako se kodiraju i ispune tog kvadratića. Radimo u binarnom sustavu. Za prvi primjer ako primjenjujemo kodiranje sivoće piksela 1 bit imat ćemo samo dvije varijable: 0% zacrnjenja te 100%zacrnjenja. Primijenimo li 2 bita imati ćemo 4 varijable od koje će 1 biti 100% zacrnjenja dok je druga 0%zacrnjenja. Preostale 2 varijable se moraju nalaziti unutar prve dvije varijable, tj. između 100% i 0% zacrnjenja. Uzmemo li 3 bita imat ćemo mogućih 8 varijabli od kojih kao i u prethodnom primjeru 1 varijabla mora biti 0% dok je druga 100%, sve preostale varijable moraju se nalaziti između 9% i 100% zacrnjenja. Varijable dobivamo tako što broj bitova ide na broj 2. Npr. imamo li 5 bitova imati ćemo 2^5 mogućih varijabli. Nakon objašnjavanja na papiru gospodin Klaudio nam je predstavio to i u Photoshopu. Napravimo novi *file* te unesemo visinu i širinu novog *file-a*, stisnemo **ok** te stavimo gradaciju na taj *file* odlično ćemo vidjeti gradaciju koju nam je gospodin Klaudio objašnjavao. Kako bismo vidjeli u koliko bitnom sustavu smo radili to, jednostavno pogledamo broj u zagradi iznad *file-a* kojeg smo napravili. Nova opcija koju možemo koristiti za naš *file* jest opcija **posterize** do koje dođemo klikom na image>adjustment>posterize, a ona nam daje opciju da upišemo koliko razina želimo da naš file ima. Stavimo li broj levela na 2, imati ćemo gradijent crne i bijele boje. Stavimo li broj levela na 10 imati ćemo 10 nijansi sive. Što veći level stavimo, slika će nam biti jasnija a gradacija ljepša. Kao što je gospodin Klaudio rekao poznato je da ljudsko oko može raspoznati 150 nijansi sive, stoga nam je dovoljno kodirati s 8 bita jer nam je to dovoljno da nam slika bude ugodna za gledati te da ne vidimo više tu gradaciju tj. stepenice gradacije kao što smo vidjeli u npr. 4 bitnom kodiranju. Kako bismo vidjeli koliko je posto zacrnjen određeni piksel sve što trebamo napraviti jest otvoriti alat sa desne strane, nakon toga uzmemo digitalnu pipetu sa lijeve strane te jednostavno pređemo tom pipetom preko piksela kojem želimo odrediti zacrnjenje. Kada smo stavili pipetu na taj piksel jednostavno iščitamo postotak koji se nalazi kraj slova K koje se nalazi na alatu sa desne strane. No to je ovisno o *sample size-u* kojeg uzmemo. Uzmemo li da bude 3x3 Average, zacrnjenje će nam uvijek biti 77% upravo zato što zbroj sva 4 piksela podijeljen

sa 4 nam daje broj 77. to se događa upravo zato što su svi veći od površine slike koja iznosi 2×2 u ovom slučaju. U slučaju slike koja sadrži više od 101×101 piksel što je najveće u *sample size optionu* pipeta će drugačije reagirati. Uzmemo li npr. 3×3 average i krenemo li prelaziti po slici vidit ćemo da će se svaki mali pomak promijeniti K. No uzmemo li na primjer 101×101 razlike u pomacima će biti znatno manje te ćemo lakše moći odrediti prosječno zacrnjenje na nekom području. Nakon toga primjera gospodin Klaudio nam je otvorio 2 iste slike veličine 4×4 piksela. Jedno sliku je reducirao na 2×2 piksela sa algoritmom *Bicubic best for smooth gradients* te ih je usporedio. Jako jasno vidimo i logički možemo zaključiti što se točno dogodilo na slici. Prelaskom preko piksela slike koja je bila veličine 4×4 vidit ćemo da svaki piksel ima određeno zacrnjenje. Na drugoj slici vidimo kako npr. lijevi gornji piksel nema isto zacrnjenje kao i lijevi gornji piksel na 4×4 slici. To se dogodilo jer se reduciranjem slike promijenilo zacrnjenje. To znači da će novo zacrnjenje gornjeg lijevog piksela iznositi prosjek 2×2 gornja lijeva piksela na 4×4 slici. Ista situacija i sa ostala 3 piksela na 2×2 slici. Uzima se prosjek sa 4×4 slike od određenih piksela. Napravimo li istu stvar ali promijenimo algoritam promijenit će se zacrnjenje. Za kraj kao što sam već objasnio na početku, gospodin Klaudio nam objašnjava koliko razina imamo sa npr. u ovom slučaju 16bitnim kodiranjem. Broj dobijemo tako što stavimo 2^{16} . Rezultat nam ispadne 65 536 od čega će nam nulta razina biti siva boja a ostalih 25 535 razine sive boje. Osim toga radimo li sa bojama u RGB sustavu te svaki kanal kodiramo sa 8 bita dobit ćemo sve ukupno 2^{24} . Od toga bi mogli napraviti čak 16 mega boja. Mega boja nam iznosi 1024×1024 . Mega nam znači k^2 , giga je k^3 , a tera k^4 , penta k^5 itd... Kao osobno mišljenje predavanje mi se jako svidjelo, gospodin Klaudio nam je lijepo predočio kodiranje sivoće piksela te sa sigurnošću mogu reći da sam sve shvatio. Vrlo zanimljivo predavanje.