

Napisala: Laura Baronić

Osvrt na prvo predavanje „Važnost IKT u metodici nastave na Grafičkom fakultetu“

Tijekom prvog predavanja kolegija „Digitalni multimedij 1“ prof. dr. sc. Klaudio Pap govori o tome kako je informacijsko komunikacijska tehnologija bitna za obrazovanje na Grafičkom fakultetu. Na primjer, neka složena i zahtjevna gradiva puno je lakše objasniti primjenom nove tehnologije. Za početak, govori se o fontovima. Program na kojem se radi demonstracija zove se „Fontographer“. Definirano je da je font uređena nakupina kodnih pozicija, a na svakoj toj poziciji nalazi se nekakva slika (eng. „glyph“). U programu je otvorena tablica već postojećeg fonta. Dok se ona stavi u decimalni oblik, dobijemo klasičnu ascii kodnu tablicu. Ako se otvorи sustav jednog koordinatnog mjesta, možemo vidjeti „digitalni četverac“ (presjek polu-površina koje tvore beskonačni pravci u koordinatnom sustavu). U Fontographer-u se može i brzo provjeriti kako kreirani znakovi unutar istog fonta izgledaju jedan pored drugog. Nekad se dogodi da 2 znaka imaju veći ili manji međusobni razmak od ostalih koji narušava izgled fonta pa se onda kreira „kerning pair“ (par podrezivanja). To znači da se razmak između tih znakova posebno odredi (smanji ili poveća da odgovara ostatku) te se to primjenjuje samo na taj par, a takvih parova se može napraviti koliko je god potrebno. U ovom programu možemo i vlastiti rukopis pretvoriti u font vektorizacijom slike. Za to se preporuča da prije nego odaberemo najbolji izgled znaka koji ćemo prenijeti u program, prvo određeni znak rukom ispišemo više puta zato što u ljudskom rukopisu uvijek dolazi do nekakvih promjena.

Sljedeći software za demonstraciju je „PostScript program“. Zadajemo mu određene znakove te parametre (dpi, širina...) te dobivamo sliku koju odmah možemo otvoriti u npr. Photoshopu. Moguće je mijenjati razne stvari: visinu fonta, staviti znakove u spiralu te odrediti zakrivljenost spirale, boje...

Nakon toga fokusiramo se na vektorsku grafiku. Jedno od njenih najbitnijih svojstava je to da nije vezana uz rezoluciju, što znači da se kvaliteta slike ne mijenja nebitno koliko ju povećamo. Vezana je samo za moment ispisa kad nešto prikazujemo. Ono što je uobičajeno kod korištenja programa za vektorskiju grafiku je Bezierova krivulja čije se tangentne točke označuju plusom. Ona ima 4 točke, od koje svaka ima x i y

koordinatu. Vrlo je bitno znati zakonitosti ove krivulje za izradu fontova. Inače, ta se krivulja prvi puta počela primjenjivati za dizajn haube u Renault tvornici automobila. Danas se koristi čak i u vektorskoj grafici za web. Jedan od danas najstandardnijih jezika je svg (skalabilna vektorska grafika), njega poznaju svi poznati browseri. Ima vrlo slične naredbe kao i PostScript. Bitno je za napomenuti važnost „čistog“ PostScripta. Kad npr. kreaciju iz Illustratora spremimo u PostScript formatu, tada se generiraju naredbe koje sve što smo napravili (sve naše poteze) pretvori u PostScript jezik, a to je jezik koji ispisne tehnologije jedino razumiju. Zato je i bitno imati postscript drivere – oni rade konverziju iz svih ostalih jezika u jezik koji je ispisnim tehnologijama jedino poznat.

Ono što je najbitnije kod učenja je da ne brzamo. Najbolje se nešto nauči ako se krene od samih osnova, od jednostavnih stvari pa se polako ide na sve viši i viši nivo. Tako i ovdje – lakše je razumjeti sve ako znamo koja naredba što radi, koji red koda je dio čega. Tada lako možemo raditi i animacije u svg-u. U lekciji je prikazan primjer takve animacije (žuti trokutić klizi po „vijugavoj“ stazi) te je prikazano kako dio animacije učiniti nevidljivim i kako ubrzati kretanje objekta. Također, kod učenja studenata bitno je štedjeti na materijalu pa grafički fakultet nastoji prikazati lekcije bez ispisa, no to nekad zna biti komplikirano, npr. kad se radi o učenju rastriranja. Rastriranjem se može jednom bojom napraviti više nijansi te boje. Postoje dvije vrste rastriranja. Kad se radi o amplitudnom rastriranju, oblik i udaljenost između rasterskih elemenata su isti, a mijenja se veličina rasterskog elementa. Ako se radi o frekvencijskom rastriranju, tada isti ostaju veličina i oblik rasterskog elementa, a mijenja se udaljenost između njih.

Još jedna bitna stvar je važnost boje u grafičkoj tehnologiji, u svim oblicima. Poznato je da boja koju vidimo na ekranu kad dizajniramo npr. neki poster i boja koju vidimo kad dobijemo taj poster u fizičkom obliku nije ista. To je zato što ekran ima RGB sustav boja koji ne postoji ni u jednoj tehnologiji za ispis. Također, ako se bavimo ispisom, moramo znati kakav će utjecaj na završni produkt imati nekvalitetne ili transparentne boje kako bi mogli napraviti odgovarajuću pripremu. Sustav boja ovisi u kojem programskom jeziku radimo. Npr. u HTML-u je moguće koristiti samo RGB sustav. I Word poznaje samo RGB sustav. Važno je naučiti sva ta ograničenja kako ne bi došlo do neželjenih rezultata kod ispita. Standard za tiskanje te komunikaciju današnjice je PDF koji se od HTML-a razlikuje po tome što pozna i CMYK i HSB

color sustave, a poznaje i pojam stranice. Znači, u PDF-u postoje naredbe koje reguliraju lijevu, desnu, gornju, možda i središnju marginu... Postoji xml jezik koji omogućava kontrolu PDF-a.

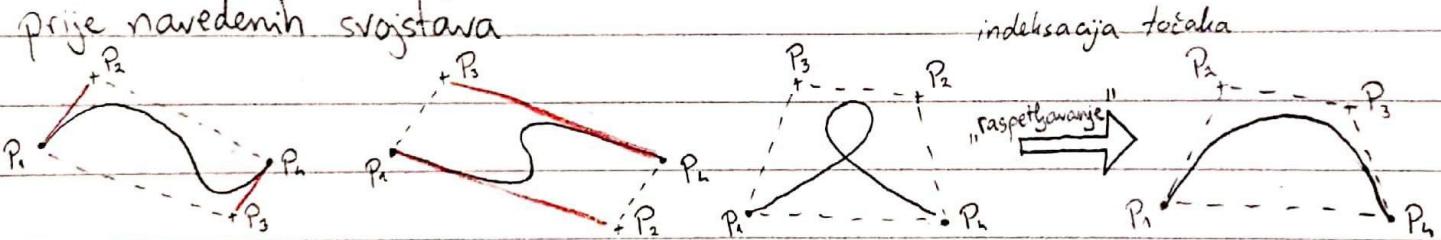
Smatram da je ovo predavanje vrlo korisno te imam volje početi učiti i stvari koje sam samo mislila da već razumijem, i to od samih osnova. HTML i JavaScript su me interesirali i u prošlosti, no sad ih imam i razloga i želje temeljno naučiti.

Osvojt na 2. predavanje:

Klaudio Pap - Bezierova krivulja

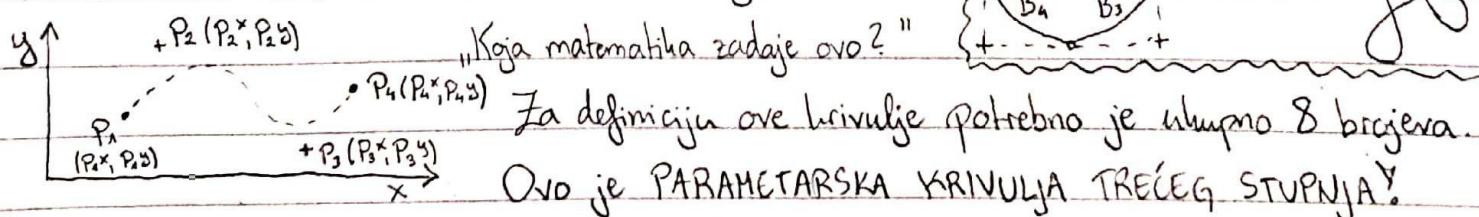
Osvojt napisala: Lauta Bartonić

Bezier krivulja je glavna krivulja današnje vektorске grafike. Već smo ju koristili kod izrade slovnih znakova u FontForge-u te u Illustratoru. U ovom predavanju profesor radi matematički izvod ove krivulje jer je ona glavna u svim vektorskim dizajnima današnjice. Vilo bitna karakteristika ove krivulje je ta da na temelju postavljanja 4 točke može se unaprijed predvidjeti njeno rasprostiranje, te samo s tim točkama ona ostvaruje svoju punu funkcionalnost. Točke se označuju s P_1, P_2, P_3 i P_4 , a između P_1 i P_2 te P_3 i P_4 postoji matematička veza. Zakonitost ove krivulje je da će se njen tijelo svjeh rasprostreti unutar konveksnog poligona kojeg ove 4 točke zatvaraju. Porodica u kojoj pripada Bezier krivulja naziva se „predvidljive krivulje/predictible curves”, upravo zbog prije navedenih svojstava.



Indeksacija se u software-ima za vektorski grafik ne koristi pa zato mi moramo unaprijed već znati tekrivulje. S ovom krivuljom se rade i dužine: $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4$ i kružnice: $B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow B_4$ zamjena "plusova".

MATEMATIČKI IZVOD BEZIER KRIVULJE:



To znači da se lako programira (parametar koji se upotrebljava za crtanje krivulje stavi se u FOR petlju i tako se dobivaju koordinate).

Prvo, definiramo krivulju u 1 dimenziji: [C=krivulja, t=parametar]

$$C(t) = [t^3 \ t^2 \ t \ 1] \times B \times \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

zbroj svakog redka je 0
zbroj je 1

-matrični zapis je skraćeni zapis

zbroj
je
0

zbroj
je
1

Zatim izvodimo ovu krivulju u 2 dimenzije:

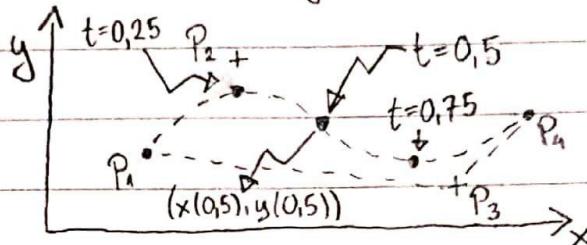
$$x(t) = (-t^3 + 3t^2 - 3t + 1) \cdot P_1^x + (3t^3 - 6t^2 + 3t + 0) \cdot P_2^x + (-3t^3 + 3t^2) \cdot P_3^x + t^3 \cdot P_4^x$$

$$y(t) = (-t^3 + 3t^2 - 3t + 1) \cdot P_1^y + (3t^3 - 6t^2 + 3t) \cdot P_2^y + (-3t^3 + 3t^2) \cdot P_3^y + t^3 \cdot P_4^y$$

Primjer: $t = \phi$
 $x(0) = P_1^x$ } P_1
 $y(0) = P_1^y$ } P_1

Primjer 2: $t = 1$
 $x(1) = P_4^x$ } P_4
 $y(1) = P_4^y$ } P_4

sve točke ove krivulje moraju imati $t \in [0, 1]$



Na elvami je potrebno puno „točica“ da bi prikazali željenu krivulju.

Pitanje je: Koliko je točica potrebno kako bi se cijela krivulja načrtala?

$\Leftrightarrow \Delta t = 0,1 : t \in [0, 1]$

$$t_0 = \phi \quad t_1 = t_0 + \Delta t = \phi + 0,1 = 0,1 \quad t_2 = 0,2 \quad t_3 = 0,3 \dots \quad t_{10} = 1, \phi$$

$\Delta t = 0,1 \Rightarrow 11$ t-ova / $\Delta t = 0,01 \Rightarrow 101$ t-ova / $\Delta t = 0,001 \Rightarrow 1001$ t-ova

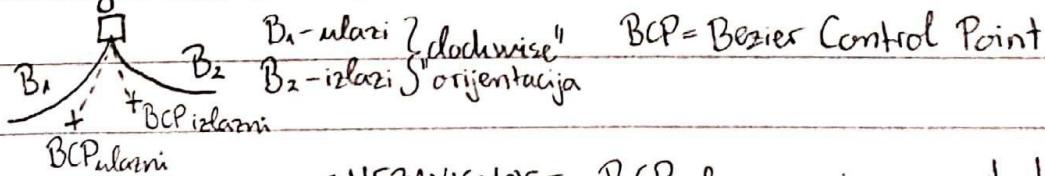
\hookrightarrow koliko se točaka može napraviti s određenim brojem t-ova

$$\text{BROJ TOČAKA} = \frac{1}{\Delta t} + 1$$

Δt je vrlo važna faza zato što se time definira gustoća točaka u Bezier krivulji.

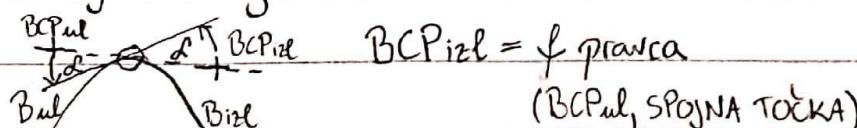
SPOJNE BEZIER TOČKE - Postoje 3 vrste SPOJNIH BEZIER TOČAKA:

1.) KUTNI SPOJ $\rightarrow \square$ (obično se označava kvadratičem)



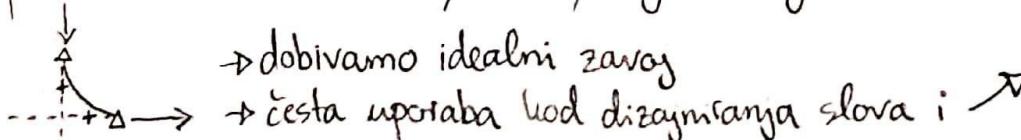
- NEZAVISNOST: BCPulazni nije u minkovoj funkcijskoj vezi s BCPizlaznim } BCPul ≠ BCPizl

2.) KRIVULJNI SPOJ $\rightarrow \circ$ (obično se označava hrušićem)

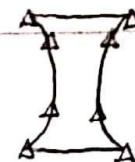


3.) TANGENTNI SPOJ $\rightarrow \Delta$ (obično se označava trokuticem)

problem: Kako idealno napraviti promjenu smjera?



\rightarrow česta uporaba kod dizajniranja slova i



Ovo predavanje je vrlo detaljno i odlično objašnjeno te sam vrlo zahvalna na tome. Shvatila sam način rada Bezier kurvula i bolje razumijem probleme koji su mi se javljali prilikom kreiranja fonta i nekih logo znakova u Illustratoru.

Osvrt napisala: Lanza Batomč

Osvrt na knjigu „PostScript“ Što je PostScript i što nam nudi?

Što je PostScript? - uvod

PostScript je programski jezik, vrlo bitan u računarškoj grafici.

Tj. znači grafički je jezik, i upotrebljava se za opis stranice te za pojednostavljenje i brzo prikazivanje grafike. Može se reći da je i programski i grafički jezik zato što ima sličnu strukturu komandi kao i Pascal, C ili Basic, no ipak je određen izuzetno godišnjim oblikovanjem grafike na stranicama za tiskah. Jedna od odličnih

karakteristika PostScripta je to da osoba koja ga želi naučiti, razumijeti, ili samo čitati o njemu, ne mora imati nikakvo predznanje u računalnom programiranju. Ali s druge strane, poželjna je komponenta silovost prema grafičkoj umjetnosti i tipografiji.

Ovaj jezik ima svog poseban grafički rječnik koji je stvoren zahvaljujući razvoju fotosloga, računarske tipografije, te računarske repro-fotografije. Sve navedeno je ono što PostScript čini savršenim alatom za sve one koji žele isprobavati nove stvari u grafici, slici, računarskoj tipografiji, bojama i rasterima. Dr. sc. Klaudio Pap i

dr. sc. Vilko Žilić napisali su ovu knjigu o PostScriptu u kojoj je svima omogućeno razumijeti ovaj jezik, i nije sve opisano samo telostom već je dano i mnogo primjera koji omogućuju lakoće snalaženje.

PostScript-ov koordinatni sustav zasnovan je u mjerim veličinama „točka“ sa X/Y koordinatama, a početak se nalazi u donjem lijevom dijelu stranice. Standardna veličina točke određena je preko inča, koji ima 72 točke od kojih je svaka 0,353 mm. Primjerim ovaj knjizi imaju mrežu 100×100 točaka ($35,3 \times 35,3$ mm) zbog koristenja decimalne podjele. Za određivanje grafičkog oblika slova ili slike koristi se komandni jezik kojim se odredi „staza“ (put), „operator“ i „operandi“ (parametri). Operatori („komande“) pišu se kucenjem i verzalnim slovima na engleskom jeziku (npr.: image, fill...) ili kraticama.

Većina komandi se sastoji od više engleskih riječi ili katica pa se one pišu zajedno, kao jedna tipografska riječ (npr.: setlinewidth). Parametri su obično numeričke veličine i pišu se ispred komandi, a određuju kako će komanda djelovati. U pisanju se koriste i razmak i tabulator i kod za novi redak kao razdijelna znaka između parametara i komandi. Moguće je pisati niz komandi u istom redak. Ako želimo napraviti komentar, koristimo znak „%“ te on završava prijelazom u novi redak (cr).

Programiranje grafike - 1. POGлавље

Kada započinjemo s izradom PostScript stranice, najčešće se prvo koristi moreto komanda. Ispred nje se definiraju koordinate točke koja će u početku nove grafike \Rightarrow npr. 270 80 moreto. Ta točka se ne prihvati, no od tog mesta će početi npr. prva dužina. Nakon što definisamo određeni broj točaka (koje želimo da budu spajene), linije koje ih spajaju nisu vidljive zato što još nemaju ni jednu dimenziju, postaje samo točke s horizontalnim i vertikalnim koordinatama. Linije prihvatuju komandom strokje kojom zadajemo parametre debeline linije, boje i stilova. Ne učimimo li to, automatski će se iscrati linija debeline jedne točke, crne boje. Komandom lineto povlačimo dužinu od posljednje određene točke do nove, uz pomoć parametra koji definiraju koordinate te nove točke (npr. 70,156 lineto).

Mnogo je komandi i nećih sad sve spominjati, no željela bih ponoviti značaj Bezierove krivulje u vektorskoj grafici. Prijetho kad u nos.

Illustratoru kreiramo neki lik ili logo bez zadržanih linija, pogotovo prilikom kreiranja slovnih znakova, i tome je sve osnova Bezier krivulja. Ona je polinom trećeg stupnja te je definisana s četiri točke: početna točka, zavisna točka, i dvije tangentne točke koje uvjetuju izgled krivulje. U PostScriptu je prilikom definiranja točaka Bezierove krivulje (prilikom uporabe naredbe curve) potrebno zadati samo koordinate druge, treće i zavisne točke jer je prva uvijek ona koja je prethodnom potom krajnja. Takođe, moguća je definicija boja i u CMYK, i u RGB, i u HSB sustavu.

Mnogo je komandi, mogućnosti... No nije previše komplikirano za shvatiti pogotovo kad imamo znanje osnovne/više razine engleskog jezika.

Programiranje tipografije - 2. POGLAVLJE

Pojam tipografije je naširoko poznat, pogotovo u grafičkom dizajnu. To je i vještina i umjetnost upravljanja slovnim znakovima → njihovog oblikovanja, izrade, poslagivanja... Zahvaljujući PostScript-u, možemo stvoriti font koji pivo možemo samo zamisliti u glavi. Ova knjiga opisuje sve što je potrebno znati kako bi tako nešto lako ostvarili.

Kad oblikujemo slovne znakove, to radimo unutar „četverca“. To je pravokutni unutar kojeg kreiramo izgled slovnog znaka. Tijekom pisanja teksta (npr. u Wordu), kad definiramo visinu slova, zapravo upravljamo visinom četverca, a ne visinom slike slova. Donji dio četverca je zapravo pismorna linija, a donji bijevi vrh četverca predstavlja mlinu tačku slovnog znaka. Pošto najviše koristimo latničko pismo, bitno je znati da je ono četverolinijsko. Sadrži ovo: descender, krov verzala, krov kuranta i pismornu liniju. Visina slike verzalnih slova je obično 70% četverca, a kurantnih 50%. Što se tiče spuštanja descendera u kurantnim slovima, to je od 20 do 30%.

Inače, dizajneri su dodali još i dvije linije za krov nekih kurantnih slova (b, t...), liniju alicenta (š, ē) i liniju spajanja kurantnih slova koja se koristi kod nekih rukopisnih fontova. Pravilni odabir visine slova je vrlo važna stvar zato što o tome ovise kolika će biti čitljivost.

I zbog toga treba pozorno promotriti kakav nam je odnos visine slova i udaljenost s koje ih promatramo. Pažnju treba obratiti i na širim cesta te razmaku između riječi, a svim time se može lako upravljati zahvaljujući PostScriptu. I opet sponinjem koristeći Bezierove staze, naiime, slova su već dugo vremena u računarskoj grafici određena putanjom crvica. Najčešće se slovi znakovi prikazuju kao ispunjen prostor sreden crvicom, no mogu biti prikazani i samo linijama

koje se nalaze na Bezier putanjama, ali one moraju imati zadatu duljinu. Opet, mnogo je komandi i mogućnosti upravljanja slovnim znakovima, previše da ih sad nabrajam. Možemo čak i oblikovati putanje teksta tako da izgleda kao neki slovni znak → npr. $\frac{1}{\pi} \sin^2(\theta)$ (D)

Programiranje piksel grafike - 3. POGлавље

(u knjizi je prikazana slika sastavljena od sivih piksela). Piksel je najmanja jedinica koja čini sliku. Svi pikseli nema istu nijansu boje (sivu u primjeru), ali boja koju ima je jednolična. Vrijednost te sivoće određena je jednim brojem u jednom bajtu (od bijelog do crnog ima 255 tonova sive). Termin „siva skala“ koristi se u digitalnoj reprofotografiji kako bi se mogao razlikovati od tonskog prijelaza u tradicionalnoj fotografskoj tehnici. Određivanje slike se postiže mizom brojeva koji tvore cijelu površinu slike. Informacije o broju piksela u reštu, broju stupaca, veličini otisnutog piksela i polotrajanje slike na stranici omogućuju prikaz slike ekranu ili pisaču. Sve je sadržano u tim brojkama. Velo bitna stvar koju sam naučila iz ove lekcije je da su pikseli zapravo definirani kao paralelogrami, a ne kvadratice, te zato imaju mogućnost da poprime zahtevene deformisane oblike. Sljedeće, prva karakteristika digitalne slike je broj razina sive skale. PostScript je u mogućnosti interpretirati slike sa 2, 4, 16, 256 i 4096 stepenica sivih tonova. Današnja reprodukcija fotografije najčešće je sa 8 bitnim razinama sivog. Čudno da može razlikovati 50 stepenica sive skale, a zato je dovoljno 6 bita ($2^6 = 64$). Prvi standard je standard od 8 bita i na boju (CMYK, RGB). U standardnim programima za piksel grafiku (primjer: Photoshop), piksel je uvijek kvadratične dimenzije na ekranu. Zadajemo veličinu veličina implicitno, brojem piksela po nekoj dužinskoj mjerenoj jedinici, tj. preko rezolucije. Ako želimo dobiti dobre rezultate prilikom ispisa slike, preporučuje se da ima 300 piksela po inču (to je recimo standard). Piksel grafiku i sama često koristim jer se bavim fotografijom, no priznayem, puno je stvari koje još misam naučila.



OSVRT NA PREDAVANJE:

Dimenzija piksela, dimenzija slike i resempliranje

Riječ piksel može dolazi od picture element (to je skraćenica). Inače, standardni oblik tog slikovnog elementa je kvadratič, ali nije uvijek tako. Može se promijeniti i u npr. romb... no, zbog velike rasirenosti u današnjim konstruktorima slike, smatramo da piksel uvijek ima formu kvadratika. Mnogi ljudi današnje smatraju da piksel grafika i rasterška grafika označavaju isti pojam, no to nije točno. U rasterškoj grafici, raster ma oblik sinonime, nema samo uredno postloženih kvadratika (█) kao u piksel grafici.

Za primjer stvaranja piksela vrlo je dobro koristiti Photoshop.

Otvor se novi dokument i sami zadajemo koliko piksela želimo u širini te koliko u dužini slike. Sliku također možemo stvoriti i digitalizacijom (pomoću scannera, digitalnog fotoaparata) te je tu vrlo bitno znati odrediti veličinu piksela slike jer će to utjecati na kasniju prezentaciju te slike na određenim tehnologijama. Ono što je tu također vrlo bitna stvar je udaljenost gledanja. Time se može optimizirati sivoća, broj piksela, limljature tiska itd.

Znači, pikseli se mogu stvoriti ili unijetno ili skeniranjem / digitalnom fotografijom. A što se dimenzija piksela tiče - one se zadaju isključivo indirektnim načinom što je važnije sa poglavom rezolucija. Rezolucija slike je gustoća piksela (kvadratika) po jedinicnoj mjeri. Tu se najčešće upotrebljavaju inch-i (2,54 cm), pa se zato i rezolucija najčešće izražava kao broj piksela po inchu (pixel per inch → latice: ppi).

$$10 \text{ ppi} \Rightarrow a = \frac{1''}{10} = \frac{25,4 \text{ mm}}{10} = 2,54 \text{ mm}$$

→ Ovo znači da stranica jednog piksela iznosi 2,54 mm ako je rezolucija 10 piksela po inchu.

$$600 \text{ ppi} \Rightarrow a = \frac{1''}{600} = \frac{25,4 \text{ mm}}{600} = 0,04233 \text{ mm} = 42,3 \mu\text{m} (\text{mikrona})$$

Rezolucijom se određuje i veličina piksela

ZADATAK:

$$300 \text{ ppi} \Rightarrow a = \frac{1''}{300} = \frac{25,4 \text{ mm}}{300} = 0,08467 \text{ mm} = 84,67 \mu\text{m}$$

$$150 \text{ ppi} \Rightarrow a = \frac{1''}{150} = \frac{25,4 \text{ mm}}{150} = 0,16933 \text{ mm} = 169,33 \mu\text{m}$$

Upravljati pikselima neke slike u Photoshopu možemo pomoću opcije Image Size. Mijenjanjem rezolucije mijenjamo dimenzije piksela, a ako isključimo opciju Resample Image, mijenjamći rezoluciju ne mijenjamo veličinu piksela već broj bajtova (piksela).

Dimenzije slike moguće je zadavati u inchima, pikselima ili centimetrima.
(Napomena: Rezolucija video kartica na ekranima je obično 80-100 ppi.)

U radu s pikselima moramo biti oprezni zato što ih ne možemo vrati u original prebitno stanje nakon što smo ih već dodavali i pomakivali.

Zato je kod profesionalnog rada bitno da spremimo original slike u najvećoj mogućoj rezoluciji (tako dobivamo puno rezorata iz originalne scene koja je prešla proces digitalizacije), a kasnije resamplingom možemo smanjiti sliku ako nam je namjera koristiti ju za web ili nešto slično.

Broj piksela mijenjamo ovisno o tome za što nam je slika potrebna.
Bitno je za zapamtiti → broj piksela ne određuje dimenziju slike, već to je sam veličina piksela koja izvire iz početne rezolucije.

Otvaramanjem već digitalizirane slike u Photoshopu vidimo da ona ima svoje zadane dimenzije koje se samo pročitaju u Image Size-u. Ako shinemo sliku s neke web stranice koja prilikom izrade stranice nije bila namijenjena povećavanju, ona će najverovatnije imati rezoluciju oko 80-100 ppi.

Kod resamplinga slike, 3 bitne opcije su: Scale Styles, Constrain Proportions i Resample Image. Ako isključimo Constrain Proportions, sve

vezu između određenih varijabli slike nestam. Znači, što god da promijenimo, samo ta varijabla će se promijeniti (obično se tako dobiva deformacija slike). Ako ishlijucimo Resample Image i radimo na veći broj piksela; konstruktar slike će umjetno stvoriti nove piksele, a ako radimo na manji broj piksela; izbacivat će se iz već postojecog.

Ovisno o tome kako želimo reproducirati sliku na uredaj određene rezolucije (i u određenoj veličini), možemo optimalno koristiti broj piksela.

Osrt napisala: Lauta Batonić

OSVRT NA PREDAVANJE: Kodiranje sivoće piksela.

U ovom predavanju odgovara se na pitanje: „Kako se definira površina koju zatvaraju stranice kvadrata piksela?“. Ljudi već mnogo vremena rade kodiranje binarnim snstavom, a tako se kodira i sivoća piksela. Primjer \rightarrow Ako uzmem 1 bit, njegova moguća stanja su ili 1 ili 0, što znači dvije moguće kombinacije sa samo dvije moguće sivoće. Standardno je da te dvije mogućnosti budu bijeli i crni piksel, tj. 0% zacrnjeni piksel i 100% zacrnjeni piksel (2 sive razine). Moguće je imati i dvije razne s npr 17%; 60% zacrnjenja, no standard su ekstremi. Ako za kodiranje sivoće odredimo 2 bita, tada imamo 4 sive razine (mogućnost bitova su 00, 01, 10 i 11). U tom slučaju imamo ove razine: $\square \quad \square \quad \blacksquare \quad \blacksquare$. U slučaju s 3 bita imamo $8 (2^3)$ sivih razina. I tako možemo sve dalje: sa 6 bita dobivamo $64 (2^6)$ sivih razina, s 8 dobivamo $256 (2^8)$ razina...

Da bi naše (ljudsko) oko bilo prevareno tako da vidi što manje tih sivih gradacija, potrebno je 8-7 bitova.

Kad radimo u Photoshopu, alatom Image adjustments \rightarrow Posterize možemo ručno odrediti koliko sivih razina želimo, tj. koliko „fina“ gradaciju želimo dobiti. Photoshop je namijesten na 8 bitova zato što se tako ljudskom oku pruža najoptimalnija ugoda.

U Photoshop je uvezena slika još s prešlog predavanja te nam služi da na njoj odredimo broj sivih razina. Posterizacijom odredujemo da ona ima 255 sivih razina (8 bitova).

Slijedeća uvezena slika veličine je 2×2 piksela. Ima 4 sive razine (2 bita). Pomoću nje se demonstrira kako čitati postotak zacrnjenja određenog piksela. To se radi pomoću pipete (alata u

Photoshopu). Kada nismo prijedemo preko željenog piksela, sa strane nam nudi podatak o tome koliko postoje taj piksel zacijen. Čitanja za ovu sliku su: 100% 80% 80% 50%. Pomoću Sample Size

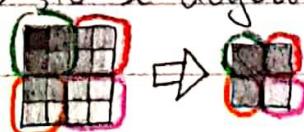


mogućemo odrediti koliko će piksela digitalna pipeta odjednom čitati. Npr. Kad na ovoj slici čitanje prebacimo na veličinu 3×3 , rezultat čitanja je $77\% = (100\% + 80\% + 80\% + 50\%) : 4$.

Zatim se ponovno demonstracija odvija na prvoj slici. Ta slika se previše detaljna za veličinu Point Sample te moramo uzeti mnogo veću veličinu pipete. Zadovoljavajuća čitanja dobivamo veličinom 101 by 101 Average. Naravno, za veličinu digitalne pipete ne postoji određeno pravilo, nju sami biramo prema željenoj vrsti čitanja.

SIVOĆA NA POJMU RESEMPLIRANJA

U Photoshopu je otvorena slika veličine 4×4 piksela. Ta slika je zatim resemplirana (Image Size) na 2×2 piksela. Polraj te slike otvorimo i izvornu (veličine 4×4) te uzmemos digitalnu pipetu na veličini Point Sample. Algoritam koristen u resempliranju je Bicubic. U prvom redu 4×4 slike razine su od 100% do 53%, a u 2×2 slici od 90% do 50%. Evo što se dogodilo (boje označavaju koji pikseli su se pretvorili u boji):



postotak ovih prijašnjih piksela dobiva se postotak novog piksela (prosjek prijašnjih 4 iz 4×4 slike daje postotak jednog iz 2×2 slike).

Napomena: Paziti kako resempliramo, koji algoritam koristimo. Ako mijenjamo nešto, ne dobivamo isti rezultat. Takođe, moramo paziti i na razlike u redoslijedu na kojima radimo. Puno je različitih video

kartica i zbog toga ne vidimo svi na svojem ekranu isti rezultat. Za grafičare je vrlo bitno znanje o radu s potencijama jer se to u grafičkom dizaynu jako puno koristi. Npr. ako imamo 16 bitova, moramo znati da tada imamo mogućih $2^{16} = 65.536$ sivih razina. Prva razina je nulta, a zadnja 65535. (ukupno ih je 65536).

OSVRT NAPISALA: Lauta Batonić

b = bit
B = bajt

OSVRT NA PREDAVANJE:

Kapacitet i histogram slike

Kapacitet slike je zapravo veličina slike u memoriji. Česti nazivi su još i „težina“ slike, workload. To opterećenje slike potiče iz broja bita po jednom pikselu.

Značenje ovoga je: što je veći broj piksela kojih imamo, to će slika biti teža za prenos, rad u memoriji ili procesorski rad.

IZRACUN TEŽINE Slike

Pervi primjer: imamo sliku 4×4 (16 piksela od kojih je svaki određene snoviće)



$$\begin{aligned} &\rightarrow \text{svaki piksel ima 8 bita} \rightarrow \\ &8 \text{ bit} = 1 \text{ B} \quad \left\{ \begin{array}{l} 1 p = 8 \text{ bit} = 1 \text{ B} \\ \hline \end{array} \right. \\ &4 \times 4 = 16 p = \underline{\underline{16 \text{ B}}} \end{aligned}$$

Dруги примјер: имамо sliku 400×600 p

$$\begin{aligned} 400 \times 600 [p] &= 240000 p = 240000 \text{ B} \\ &= (240000 : 1024) \text{ kB} \\ &= 234,4 \text{ kB} \quad \left\{ \begin{array}{l} 8 \text{ bita} \\ \text{po} \\ \text{pikselu} \\ (\text{grey}) \end{array} \right. \end{aligned}$$

Treći primjer: 1 bit na slika $\rightarrow \left\{ \begin{array}{l} 1 p = 1 \text{ bit} \end{array} \right.$

$$\begin{aligned} 400 \times 600 [p] &= (240000 : 8) \text{ B} = 30000 \text{ B} \\ (30000 : 1024) \text{ kB} &= \underline{\underline{29,3 \text{ kB}}} \end{aligned}$$

Cetvrti primjer:

$$\begin{aligned} &\text{RGB} \\ &8b + 8b + 8b = 24b \quad \left\{ \begin{array}{l} 3 manala \end{array} \right. \end{aligned}$$

$$400 \times 600 [p]$$

$$234,4 \text{ kB} \cdot 3 = \underline{\underline{703,2 \text{ kB}}}$$

Peti primjer:

$$\begin{aligned} &\text{CMYK} \\ &8b + 8b + 8b + 8b = 32 \text{ bit} \end{aligned}$$

\rightarrow mora biti 4 puta veća od početne (drugi primjer):

$$234,4 \text{ kB} \cdot 4 = \underline{\underline{937,6 \text{ kB}}}$$

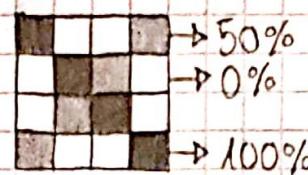
u Photoshop-u \Rightarrow graf koji prikazuje distribuciju sivočih piksela

Histogram slike = normalizirana funkcija distribucije sivih
razina slike

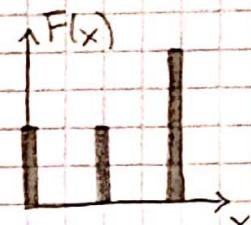
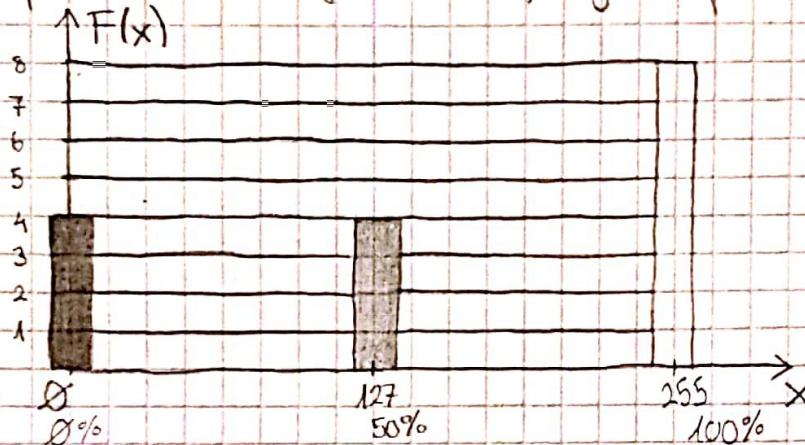
\Rightarrow Funkcija distribucije sivih razina:

- 1 - kanalna slika

- sivoča kodirana sa 8 vita

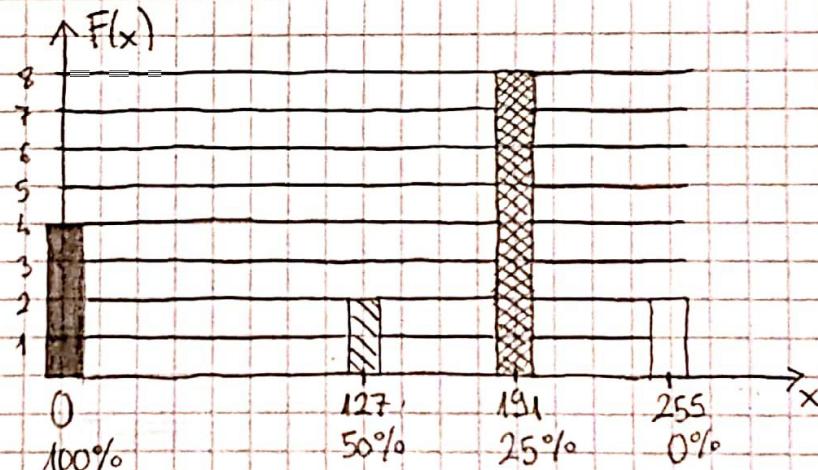
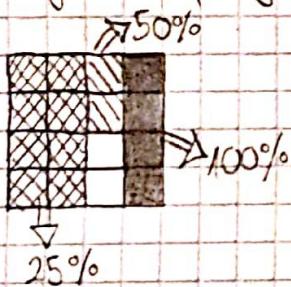


\hookrightarrow parametar x : definira sivoču koju 1 piksel može imati



\rightarrow ako se gleda na elemenat $\rightarrow 0\%$ je mali, a 100% je svijetlo (PS)

Još jedan primjer:

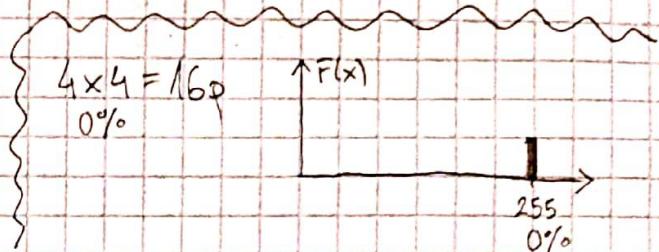
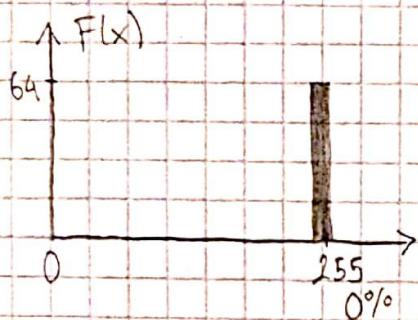


$$\sum_{x=0}^{255} F(x) = \text{BROJ PIKSELA NA SLICI}$$

\hookrightarrow za ovaj primjer: $4 \times 4 = 16 = 4 + 2 + 8 + 2$

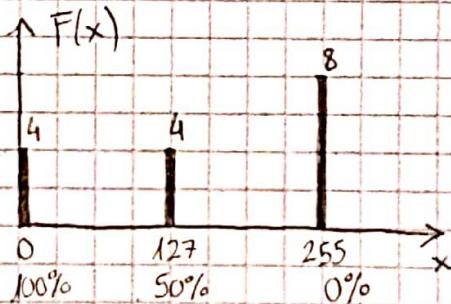
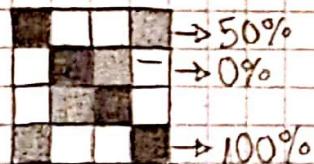
HISTOGRAM SLIKE

Primjer 1: slika ima $8 \times 8 = 64$ piksela i svi su bijeli (0%)



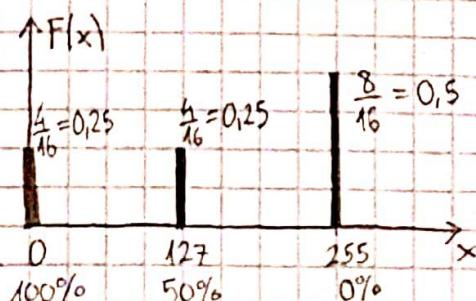
MANA FUNKCIJE DISTRIBUCIJE: y-os previsje raste kad se poveća
broj piksela slike.
↳ potrebna je NORMALIZACIJA

$$\{ f(x) = \frac{F(x)}{\sum_{x=0}^{255} F(x)} \}$$



$$\sum_{x=0}^{255} F(x) = 16$$

↓ NORMALIZACIJA

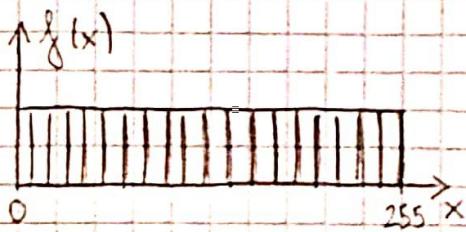


$$0,25 + 0,25 + 0,5 = \boxed{1} \quad \left(\frac{16}{16} \right)$$

Primjer 2:



imamo jednaki broj piksela svake sive razine

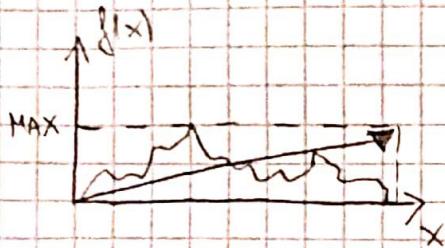


$$\sum_{x=0}^{255} f(x) = 1$$

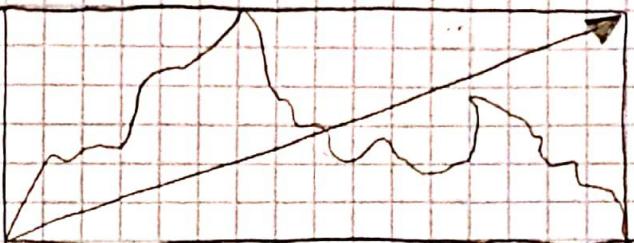
$$255 \cdot v = 1$$

$$\boxed{\frac{1}{255}}$$

MAKSIMIZACIJA:



MAX



U Photoshop-u, histogram slike se otvara {control L}

→ RELATIVNO gledamo distribuciju sivoće na slici

↳ samo vidimo koliko ima određenih razine sivoće u odnosu na druge sivoće

- REDISTRIBUCIJA SIVOĆA → oprezno da se ne izgube određene razine sivoće

OSVRT NAPISALA: Lauta Batonić

OSVRT NA PREDAVANJE: Digitalni video

Digitalni video je, po definiciji, serija digitalnih slika koje se izmjenjuju u nekom vremenskom intervalu, a njegovi podaci spremaju se na DVD/CD medije, memorijске kartice, diskove. Ti podaci se priklom prikazivanja dekodiraju (inace su kodirani određenim procesima). Prije današnjeg digitalnog prijenosa, u svijetu su postojala 3 standarda analognog televizijskog prijenosa:

J1 Amerika

- 1) PAL (Phase Alternating Line) - u Europi, dijelu Afrike, J Aziji, Australiji
 - 2) SECAM (Sequential colour with memory) - Rusija, srednja Afrika, Francuska
 - 3) NTSC (National Television System Committee) - Sj Amerika, Japan, Filipini
- ↳ PAL i SECAM - dosta slični, na strujnoj mreži od 50Hz, slika od 625 horizontalnih linija, izmjena 25 fps
↳ NTSC - strujna mreža od 50Hz, 525 linija, 30 sličica u sekundi/fps

Danas postoji digitalne inačice PAL i NTSC standarda. Oni imaju istu horizontalnu rezoluciju (h), a različitu vertikalnu (v).

• PAL DV: $720 \text{ h} \times 576 \text{ v}$

• NTSC DV: $720 \text{ h} \times 480 \text{ v}$

$$h:v = 4:3$$

Ovalne dimenzije se nazivaju SDTV (Standard Definition TV).

Poslije toga nastaje HDTV (High Definition TV). Rezolucija mu je ili 1280×720 ili 1920×1080 (Full HD). Do ovih brojih se došlo tako da su se dimenzije prethodnjeg NTSC formata pomnožile s 1,5: $480 \times 1,5 = 720$ } $h:v = 16:9$ $720 \times 1,5 = 1280$ v widescreen

Prije 10-ak godina, pojavio se i UHD (Ultra High Definition).

On može biti 4K (4096×3072 za 4:3; 3840×2160 za 16:9) ili 8K (8192×6144 za 4:3; 7680×4320 za 16:9).

Kad se govori o rezoluciji, bitan pojam je omjer stranica slike (Aspect ratio). To je omjer širine i visine video slike. Prvi standard omjera stranica pohrane slike uspostavljen je početkom 20. stoljeća, za vrijeme nijemog filma. Baziran je na fotografiskom 35 mm filmu omjera 4:3 (1,33:1). Svi filmovi u himma su bili projicirani u tom formatu, a koristio se i kada se pojavila televizija. Nakon toga, kino industrija je morala nekako vratiti gledatelje u kina, pa su osmisili nove standarde nazvane Widescreen (široki ekran):

cinerama - 2,59:1, academy ratio - 1,37:1,

cinemascope - 2,35:1, vista vision - 1,85:1,

MGM - 2,76:1, Panavision - 2,20:1..

Danas popularan omjer je 16:9 (1,78:1). On se pojavio 80-ih, kao kompromis prihvađanja raznih formata širokutnih filmova na televiziji. Također, to je zagrano geometrijska sredina popularnih omjera 4:3 i 2,35:1. Traže lijepe se dobiju sa strane približnom prihvađanja 1,33 na 16:9 - to se naziva pillarbox, a kod 2,35 na 16:9 - to je letterbox. Svih toga je da nema rezanja slike ili bespotrebnog praznog prostora oko slike.

Slijedeća važna karakteristika kod video prikaza je izmjena broja sličica u sekundi (frame rate):

- 10-12 fps → ljudsko oko vidi kontinuirani pokret
 - 24 fps → filmski standard
 - 25 fps → PAL standard
 - 29,97 (30) fps → NTSC standard (veći fps za posebne namjene)
- VIDEO SA 5fps: ne vidi se kontinuirani pokret, "preskače"
- VIDEO SA 10fps: pokret je kontinuiran, no kretanje kula nije realno
- VIDEO SA 29,97fps: glatki, realan pokret kula

Iduća bitna karakteristika su načini prikaza slike (frameova):

- isprepleten (Interlaced) - 480i (prvo neparni, onda parni redovi)

- progresivan (Progressive) - 720p (slika se prenosi u cijelosti)

(HD - 720p, 1080i, 1080p)

Važna stvar je i veličina video materijala. Kolikor učinimo podataka zaista sadrži 1 video? PRIMJER: $640 \times 480 \text{ px} = 307200 \text{ px}$

$$24 \text{ b}/8 = 3 \text{ B} = 1 \text{ RGB piksel} \leftarrow \text{RGB} \rightarrow 24 \text{ bit (8 b po kanalu)}$$

$$\text{TEŽINA CIJELE SLIKE: } 3 \times 307200 = 921600 \text{ B} = 900 \text{ KB} \rightarrow \text{za 1 frame}$$

$$30 \text{ fps} \rightarrow 30 \times 921600 = 27648000 \text{ B} = 27000 \text{ KB} \approx 26,5 \text{ MB} \rightarrow \text{za 1 s video}$$

→ Zbog ovoga se radi kompresija podataka (smanjuje se veličina podataka).

Optimizacija veličine video datoteke temelji se na rezoluciji, broju sličica po sekundi te jatini kompresije. CODEC (CODE/DECODE) je algoritam prema kojem se sirovi podaci kodiraju da bi se smanjila ukupna težina datoteke.

Podaci koji su svišni (redundantni), nevažni (ko ih ne primjećuje) uklanjanju se i sačinjavaju. Viste CODEC standarda su: MPEG-4 Part 2 /DivX (za .avi), MPEG-4 Part 10 /AVC /H.264 (za .mp4, .m4v...), VP8; VP9 (za .webm), THEORA...

Pojam Bit rate označava učinak podataka video-datoteke pa 1 sekundu video, govori algoritmu koliko podataka smije smanjiti za željenu kvalitetu slike. Mjerna jedinica mu je bps (Kbps, Mbps). Veći bitrate → manja kompresija. Smjernice za određivanje bit rate-a: do 10 Mbps za HD video od 720p, od 15 do 25 Mbps za Full HD 1920×1080 , od 15 do 100 Mbps za UHD 4K video.

PRIMJER - VIDEO (H.264 - HD 720p, 29,97fps, 6sek):

$$0,19 \text{ Mbps} \rightarrow 140 \text{ KB} \mid 10 \text{ Mbps} \rightarrow 7,7 \text{ MB} \mid 5 \text{ Mbps} \rightarrow 3,8 \text{ MB}$$

loša kvaliteta, malo detalja

više detalja, bolja kvaliteta

dovoljna kvaliteta ✓ → smanjen bit rate, manja veličina

ZADATAK	ORIGINAL	OBRADENI VIDEO
format datoteke	mp4	flv
veličina datoteke	16,9 MB	926 KB
trajanje	11s 19 ms	10s 10ms
CODEC	AVC	VP6
bit rate	13761 kb/s	96 kb/s
rezolucija	1920×1080	1026×578
frame rate	30,01	12

OSVRT NAPISALA: Lauta Bartomić

OSVRT NA PREDAVANJE: Boja i zvuk u video kompresiji

Prijenos podataka / Bitrate nekomprimiranog video signala u rasponu je od 270 Mb/s za SDTV (Standard Definition TV), 1,5 Gb/s za HDTV te 3 ili više Gb/s za Ultra High Definition TV. Ta količina podataka previše je velika za efikasnu pohranu ili transmisiju, pa zato dolazi do kompresije (sažimanja) podataka pomoću perceptualnog kodiranja. Na taj način smanjujemo količinu podataka. Perceptualno kodiranje uzima u obzir karakteristike ljudskih oči i ušiju kako bi se saznala ograničenja naše percepcije te se prema tome makhnu podaci koji su nama (ljudima) neprimjetni. Te operacije se izvode tijekom prijenosa video signala ili prilikom samog snimanja kamerama, fotoaparatima, mobilnim uređajima... Njima je kapacitet pohrane malen te moraju izvoditi neku vrstu kompresije kako bi spremili podatke. Tijekom prijenosa video i zvučnog signala (streaminga), moramo paziti i na količinu podataka koja se šalje kako bi primatelji signala bio u mogućnosti pratiti signal bez ometanja i prevelikih gubitaka.

Kod sažimanja podataka direktno prilikom snimanja, govorimo o kodiranju izvora (Source Coding) zato što se sačima analogni signal koji dolazi na senzore uređaja za snimanje. Kod post-procesiranja dolazi do prilagostavanja različitim medijima. Znači, ovdje se razlikuju 2 vrste kodiranja: odmah na izvoru i kasnije u post-procesiranju. Nakon kodiranja izvora, količina podataka će biti svedena na 1-15 Mb (to ovisi o algoritmu kompresije koju koristimo).

2 su načina reduciranja podataka: reduciranje svištm (redundantnih) i nevažnih (irrelevantnih podataka). Svišni podaci su oni koji se više puta ponavljaju, a nevažni su oni koje ljudsko oko ne primjećuje ali nedostaju. Oni mogu smanjiti količinu podataka i preko 100 puta. Svišni podaci se mogu lako i bez gubitaka izračunati korištenjem nekog matematičkog algoritma prilikom kodiranja (dijelovi koda se zamjenjuju kraćim kodovima koji ih matematički opisuju → KOMPRESIJA BEZ GUBITAKA / lossless). Ljudsko oko zbog prirodne adaptacije sadrži više receptora za svjetlost nego boju pa preciznost boje u video signalu može biti reducirana → redukcija količine podataka o boji. Također, ljudsko oko ne percipira fine strukture u slici → KOMPRESIJA S GUBITCIMA / lossy.

REDUKCIJA PODATAKA ZA BOJU:

→ RGB → luminantna i krominantna komponenta

→ YCbCr (YUV)

→ $Y = (0,3 \cdot R) + (0,59 \cdot G) + (0,11 \cdot B)$

$$Cb = 0,56 \cdot (B - Y)$$

$$Cr = 0,71 \cdot (R - Y)$$

VIDEO SIGNAL = LUMA (svjetlost - crno bijelo) + CHROMA (boja).

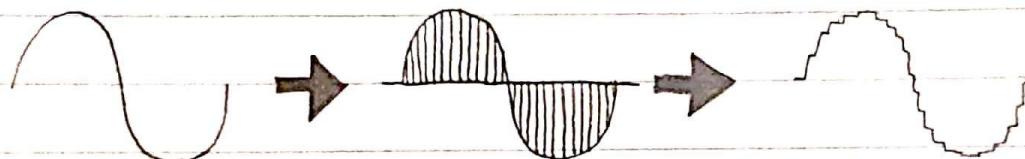
Odnos frekvencija mrezovanja za luminantnu i dvije krominantne komponente: 4:4:4, 4:2:2, 4:2:0.

DIGITALNI AUDIO SIGNAL:

Dinamički raspon ljudskog uha je otprilike 140 decibela, a raspon frekvencija mu je 20 - 20 000 Hz. Valovi koji nastaju titranjem izvora frekvencije veće od 20 000 Hz nazivaju se → ultrazvuk, a frekvencije manje od 20 Hz → infrazvuk. Audio signal koji želimo transmitirati digitalnim putem mora imati karakteristike koje poštuju raspone čujnosti ljudskog uha.

Dio uha koji je zadužen za prevođenje zvučne energije u zvuk naziva se pužnica.

Termi koji se koriste prilikom definiranja audio signala:
Analogni signal se sruša različitim usrednjajima te se filterima ograničava njegov raspon prije digitalizacije. Taj proces se naziva sampliranje ili uzorkovanje signala. Tako analogni signal postaje mjerljiv. SAMPLING RATE / BRZINA UZORKOVANJA mjeri se kao broj snimljenih audio uzoraka u munitar i sekunde.
Najmanji broj uzoraka po sekundi (za audio signal) je 8 kHz.



continuous analog signal sampling process discrete digital signal

SAMPLING DEPTH / PRECIZNOST UZORKA mjeri se u bitovima po uzorku. Određuje koliko mogućih razina amplitude može postići zvučni signal. Preporučeno 24-bitni zvuk za kvalitetne audio snimke.

$$\begin{aligned} \text{BIT RATE} &= \text{SAMPLE RATE} \cdot \text{SAMPLE DEPTH} \\ &= \text{uzorak/sec} \cdot \text{bit/uzorak} = \text{bit/sec} \end{aligned}$$

→ Najčešće koristi samplig rate:

- 8 kHz - telefonija (da je manje → izgubila bi se kvaliteta zvuka)
- 11,1 / 48 kHz - TV / CD (Nyquist-Shannonov teorem)
- 96 / 192 kHz - blu-ray
- > 300 kHz

OSVRT NA PREDAVANJE:

Logička organizacija web sjedišta (MODELI LOGIČKE ORGANIZACIJE WEB SJEDIŠTA)

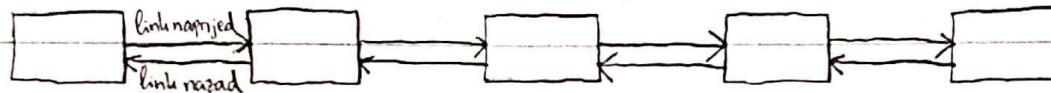
U ovom predavanju je cilj shvatiti osnovne postulate izgradnje web sjedišta. A su osnovne logičke organizacijske forme:

a) LINEARNA LOGIČKA ORGANIZACIJA

Pojam linearnosti u području organizacije označava mogućnost da se dokumenti unutar web sjedišta čitaju kao knjiga/tiskani medij.

To je poželjno kada se neki proces opisuje korak po koraku.

a) OSNOVNA LINEARNA ORGANIZACIJA (pravdušnik □ = HTML stranica)

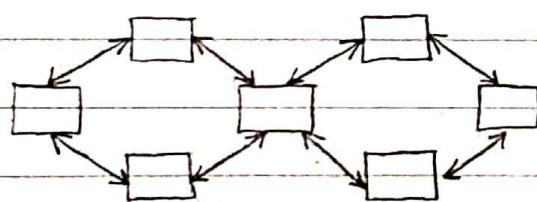


- korisnik prima informaciju željenim redoslijedom

- PREDNOSTI: preload/precash

NAZAD [MOTOR] AUTO DIJELOVI [stranica 5 od 10] NAPRIJED [lojnes]

b) LINEARNA LOGIČKA ORGANIZACIJA S ALTERNATIVAMA



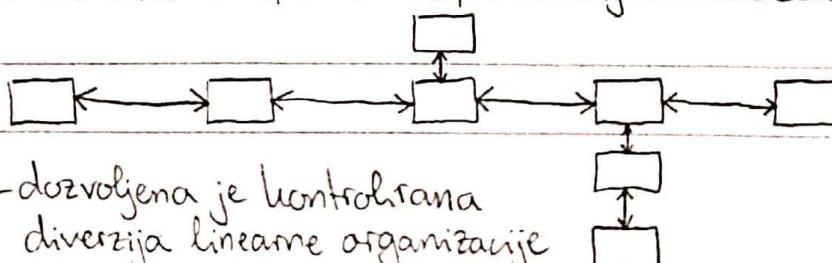
- nema baze podataka u pozadini
- simulacija interaktivnosti
- moguće vrtenje statistike
- otežan preload

c) LINEARNA LOGIČKA ORGANIZACIJA S OPCIJAMA (sa skokom unaprijed)



- moguće kontrolirano preskativanje stranica tijekom pregledavanja

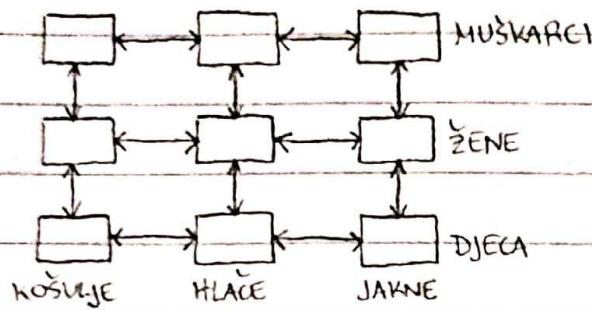
d) LINEARNA LOGIČKA ORGANIZACIJA S IZLETIMA



- dozvoljena je kontrolirana diverzija lineare organizacije

2) MREŽNA LOGIČKA ORGANIZACIJA

Ovakvu organizaciju često koriste web trgovine, na primer:



- prezentiraju se horizontalni i vertikalni odnosi između različitih objekata
- moguće je dodati i razine pretrage (npr. za cijene)

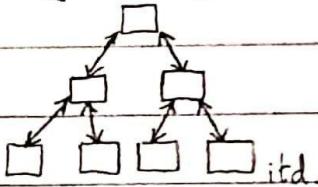
3) STABlena (Hijerarhijska) Logička Organizacija

Ovo je jedna od najčešćih organizacija na web-u zato što se može lako modificirati te je lako dodati/skriti informacije.

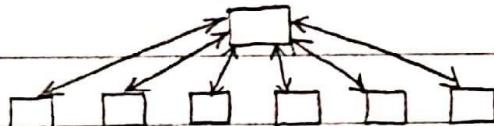
a) USKO STABLO (binarno stablo): $2^0=1$

- mali izbor na stranici, puno „klikanja” $2^1=2$

- dubina je najvažnija, pogodno je za $2^2=4$ itd.
npr. traženje posla



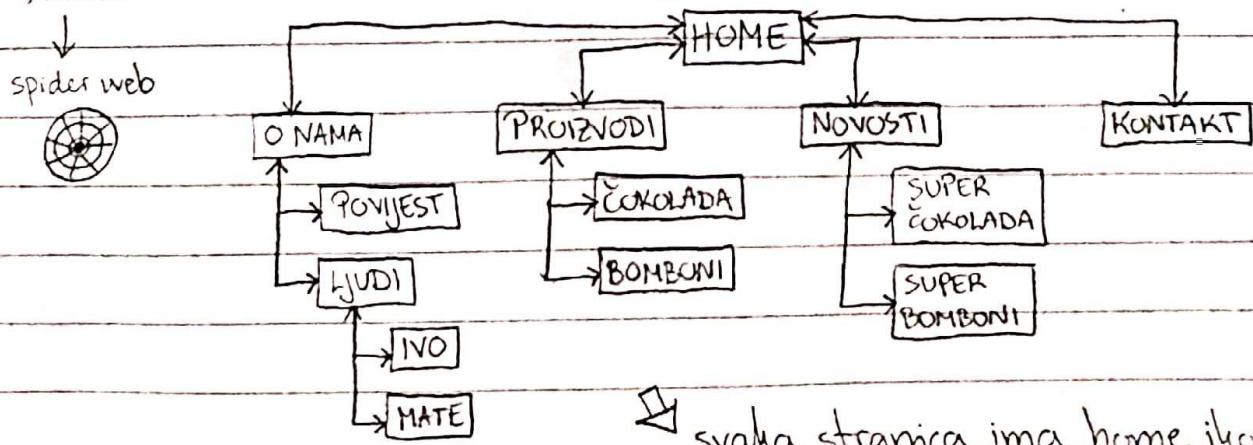
b) ŠIROKO STABLO



- preferira se širina; mnogo je „klikanja”

- brzo se dolazi do željene informacije, no lako se izgubiti u puno linkova na 1 stranici
- primjer za koristenje: u autosalonu

4) WEB LOGIČKA ORGANIZACIJA



svaka stranica ima home ikonu (⌂)

⇒ dobro je povezati i „IVO“ te „MATE“ sa „ONAMA“ (direktno)

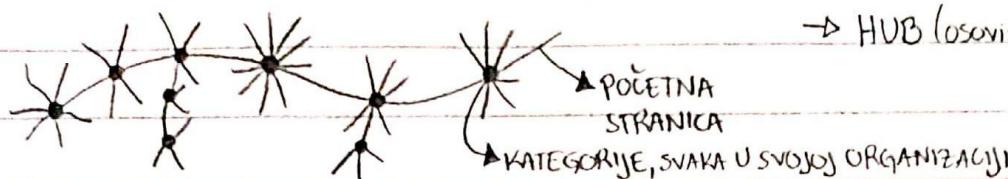
koja direktno vraća na početnu stranicu

- sve stranice su povezane sa svima
- ukupan broj stranica = n
- ukupan broj linkova = $n \cdot (n-1)$

• FULL MESH (POTPUNA MREŽA / ZAHVAT)

Postoje još neki posebni načini pristupa organiziranju web sadržaja:

1) MIKSANA FORMA (MIKSANA HIJERARHIJA)

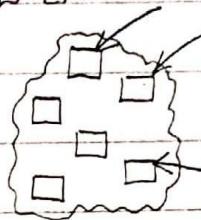


→ HUB (osovina) AND SPOKE (špica) FORMA

2) KATEGORIZACIJA WEB SJEDIŠTA PO BROJU ULAZA

a) POREZNA, POLUPOREZNA ORGANIZACIJA

- zapamtiti se gdje stajemo s pretraživanjem te snadjan nastavimo bez problema



- PREDNOSTI: korisnik ima potpunu kontrolu te može direktnoći na većinu URL-ova i staviti bookmark

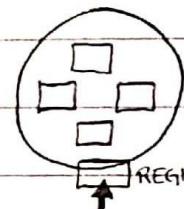
- MANE: manja mogućnost promjene dubokih stranica, manja orijentacija te manja mogućnost reklamiranja

b) ČVRSTA ORGANIZACIJA

- najviša je crna sa samo jednim ulazom:

- PREDNOSTI: laka modifikacija i održavanje

web sjedišta, prisiljava korisnika da uđe kroz poznate točke (dobro za reklamiranje i dnevno informiranje)



- MANE: korisniku se oduzima kontrola, smanjuje se mogućnost pretrage raznim vanjskim pretraživačima

IZRAŽAJNOST ↑



• LINEARNE

• WEB

• MIKSANA FORMA

• STABLO

PREDVIDLJIVOST →

MOGUĆA KONFUFIZIJA