




Multimediji

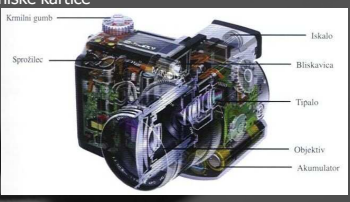
Slike

mag. Boštjan Vouk



Osnovni deli fotoaparata

- Pomembni deli digitalnega fotoaparata:
 - tipalo
 - objektiv
 - zaklop
 - zaslon LCD
 - pomnilniške kartice



boštjan.vouk@bsc.si



Senzor svetlobe

- Senzor je za video sisteme razvito vezje, ki je sposobno meriti svetlobo, ki pade nanj. V principu je to monokromatska naprava, sestavljena iz drobnih foto-diod, ki oddajajo pri osvetlitvi električni naboj.
- Senzor potrebuje barvne filtre, da lahko zazna različne barve.
- Barvni filtri so izdelani z direktnim nanašanjem barv na površino vezja.
- Senzorji zaznavanja barv se uporabljajo predvsem v digitalni video in foto tehniki ter povsod drugje, kje je potreba po potreba po strojnem vidu (tekstilni, avtomobilski industriji itn.)
- Najpogosteje uporabljeni tipali za zaznavanje svetlobe sta **CCD in CMOS**.

boštjan.vouk@bsc.si



Prenos motiva na tipalo

- Ko fotografiramo z digitalnim fotoaparatom se izbrani motiv po odprtju zaklopa prenese na svetlobno občutljivo tipalo CCD element ali CMOS. Mikro računalnik nato prenese sliko iz svetlobnega tipala, kjer ustvarijo električni signal točko za točko do pomnilnika in jo shrani v digitalni obliki.
- Ena izmed ključnih problematik digitalne fotografije je ločljivost slikovnega senzorja. Slikovni senzorji dražjih digitalnih kamer, ki se pojavljajo trenutno na tržišču, ponujajo že izjemno visoke ločljivosti, vendar je ta v primerjavi s človeškim očesom še vedno izjemno nizka.

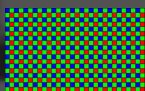
bostjan.vovk@isc.si

4



Bayer filter

- Trenutni CCD ali CMOS slikovni senzorji uporabljajo tako imenovano Bayer-jevo razporeditev slikovnih pik posameznih barv.
- Ker je občutljivost človeškega očesa približno 70 odstotna na zeleno barvo, tudi omenjen razpored barvnih točk teži organizaciji človeškega očesa.
- Tako so slikovne pike danih CCD ali CMOS senzorjev v razmerju 4:2:2 za zeleno, rdečo in modro barvo. Preprosteje povedano na slikovnem senzorju vsaka druga slikovna točka snema zeleno barvo, pri čemer se rdeči in modri senzorji izmenjujoče izmenjujejo. V takšnem klasičnem primeru imamo torej težavo z ločljivostjo predvsem zaradi tega, ker vsaka slikovna pika ne zmore zajemati polne barvne globine.



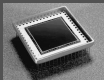
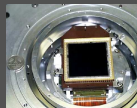
bostjan.vovk@isc.si

5



Tipalo CCD

- CCD (charge-coupled device)
 - Senzorski čip v fotoaparatu, ki pretvori svetlobo v digitalno informacijo. CCD čip ima 4 barvne senzorje - po enega za rdečo in modro ter dva za zeleno barvo.
 - film digitalnih fotoaparatom



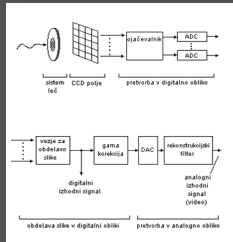
bostjan.vovk@isc.si

6



Tipalo CCD

- Svetlobno tipalo zazna svetlobo.
- Sestavljen je iz točk na njih pa so RGB barvni filtri.
- Svetloba, ki pade na CCD element, se v trenutku prenosa v naslednjo obdelavo.
- Ko fotoelement reagira, reakcijo spremeni v elektronski signal, ki podaja vizualne lastnosti posnete okolice na kraju in v času.
- Celice RGB zbirajo svetlobo, zato so slike ob veliki povečavi kockaste.
- Velikost je od 7 do 10 μm
- Vsebuje 5 ali 6 milijonov pikselov.
- Vsakemu pikslu pripada samo ena barva.

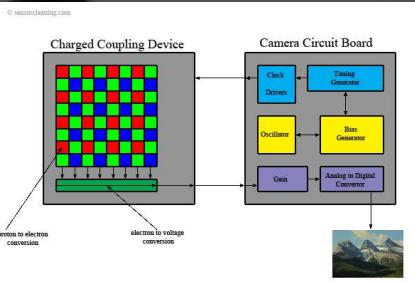


boštjan.vovak@bsc.si

7



Tipalo CCD



boštjan.vovak@bsc.si

8



Tipalo CCD - zanimivost

- Teksaško podjetje Dalsa Corporation je razvilo vrhunsko tipalo CCD, ki ima čez 111 milijonov pik. Tipalo meri približno 10 x 10 cm in zmora zajeti slike z ločljivostjo 10.560 x 10.560 pik, pri čemer je fizična velikost pik na tipalu komaj 9 mikrometrov. Žal bo slike s tovrstnim tipalom lahko izdelovalo le malo ljudi, saj je tipalo namenjeno pretežno za rabo v **astronomiji** za slikovni zajem oddaljenih vesoljskih teles in pojavov. Podjetje Dalsa ponuja tudi specializirana tipala CCD z visoko ločljivostjo, ki dosegajo od 6 do 33 milijonov pik, izdelujejo pa tudi tipala za hiter zajem fotografij s hitrostjo tja do 100 slik na sekundo.

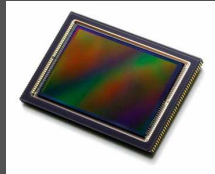
boštjan.vovak@bsc.si

9



Tipalo CMOS

- CMOS (complementary metal-oxide semiconductor)
 - čipi v fotoaparatu, ki deluje na principu fototranzistorjev

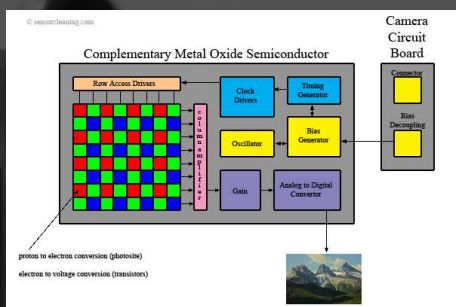


bostjan.vovk@bcs.si

10



CMOS



bostjan.vovk@bcs.si

11



Primerjava CMOS tipala s CCD

- bolj prilagodljiv
- tehnologija CMOS se je najprej pojavila pri digitalnih kamerah
- procesiranje slike direktno na čipu
- veliko dodatnih funkcij
- manjša poraba energije
- tipalo prebere točke za vsako barvo posebej
- vsebuje 6 milijonov pikslov – boljša slika (priključimo direktno na računalnik)
- zaradi večjega procesorja ob povečavi ne dobimo kockaste slike
- so podobni tehnologiji CCD s to prednostjo, da so grajeni z isto tehnologijo, kot jo uporabljajo pri izdelavi čipov
- pri tehnologiji CCD so tipala, ADP in ostala elektronika ločeni, medtem ko so pri tehnologiji CMOS ti elementi združeni
- CMOS ima manjšo porabo energije

bostjan.vovk@bcs.si

12



Primerjava tipal CCD in CMOS

CCD vs. CMOS Sensors		
	CCD	CMOS
cost	expensive to produce because of special manufacturing methods employed	inexpensive because CMOS wafers are used for many different types of semiconductors
power	consumes upto 100x more power than CMOS	low power consumption
noise	high quality, low noise images	susceptible to noise
maturity	produced for longer period; higher quality images, more pixels	less mature but equal in low and middle range resolutions to CCD
extended functionality	technically feasible; other chips are used	other circuitry easily incorporated on same chip
fill factor	high	low

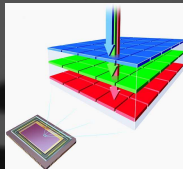
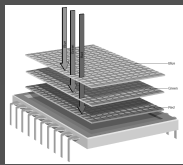
boštjan.vovak@isc.si

13



Foveon X3 tipalo

- Zaznavanje vseh treh primarnih barv: modre, zelene in rdeče na posameznem pikslu.
- trisojni filter
- Foveon X3 senzor **nima** trikrat več celic kot ostali senzorstvi.
- Foveon tako označuje svoje senzorstve, toda število celic je enako, večje je le število prebranih vzorcev.
- senzor s 3 Mpix označujejo kot 9 Mpix ločljivost
- Barva je določena na osnovi zbranih električnih nabojev na posameznem sloju filtra.
- ni potrebna interpolacija
- boljše, ostrejšje barve
- boljša kvaliteta slike



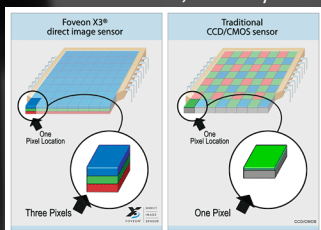
boštjan.vovak@isc.si

14



Primerjava

Foveon X3 s CCD/CMOS tipali



Za razliko od klasične CCD ali CMOS arhitekture, uporablja X3 senzor lastnost, da različne valovne dolžine prodirajo različno globoko v silicijev material. Najprej se v zgornjem sloju vpletajo vse tri valovne dolžine, za tem se najprej izgubi in odčita modra, nato zelena in na koncu, najgloblje, še rdeča. Ker so izgube precejšnje, morajo biti celice precej velike in zato je število celic na X3 senzorstvu precej manjše kot na senzorstvih enake velikosti a drugačne arhitekture.

boštjan.vovak@isc.si

15

 **Digitalizacija slik + in -**

V USTANOVI	IZVEN USTANOVE
<ul style="list-style-type: none"> • učenje z delovnimi izkušnjami in razvoj stroke znotraj ustanove • izgradnja sposobnosti produkcije • ohranitev nadzora nad vsemi vidiki dela • fleksibilnost pri definiranih zahtevah • varnost izvornega materiala 	<ul style="list-style-type: none"> • strokovnost in izobraževanje ponudnika storitev digitalizacije • postavitev cene na silko - cene se lahko pogajajo glede na količino dela • nižja cena delovne sile • cene zastarelosti tehnološke opreme nosi ponudnik storitev • omejeno tveganje • številčnost opcij in storitev
<ul style="list-style-type: none"> • velika investicija • ni postavljene cene na silko • vzpostavitev tehnične infrastrukture: prostor, oprema za digitalizacijo, računalniki • omejena sposobnost produkcije • ustanova nosi stroške zaradi zastarelosti tehnologije • vpliv na dejavnosti ustanove • ustanova plača bolj za opremo, vzdrževanje in osebje kot za produkt • potreba po izobraženem osebju, izobraževanja • tehnična podpora za opremo 	<ul style="list-style-type: none"> • ustanova nima funkcije digitalizacije • neizkušenost ponudnika na področju potreb in specifičnosti ustanove • kontrola kvalitete ni na kraju dela • slike morajo manipulirati delavci ustanove • jasno definiranje potreb • transport

boštjan.vovak@bsc.si 16

 **Standardni formati za zapis slike**

<ul style="list-style-type: none"> • Rasterski: <ul style="list-style-type: none"> – JPEG – PNG – TIFF – GIF – BMP – ... 	<ul style="list-style-type: none"> • Vektorski: <ul style="list-style-type: none"> – SVG – EMF – DWF – PDF – ...
--	---

boštjan.vovak@bsc.si 17

 **Rastrski format**

- **Sestava** - raster predstavlja neko sliko s pomočjo serije bitov, ki vsebujejo določene količine informacij (barvne globine) in jih pretvarjajo v pike (piksle), ki jih navadno vidimo na računalniškem monitorju, papirju ali kakšnem drugem mediju. Ogromne količine takšnih pikslov tako ustvarijo barvne pike, te pa nam prikažejo neko končno predstavo v obliki slike ali fotografije.
- **Kvaliteta oz. ločljivost** slike v rastrski obliki je torej odvisna od števila pik na inčo (DPI - Dots Per Inch). Večja kot je ločljivost oz. število pik na inčo, večja je kvaliteta barvnih prehodov med piksli. Takšen pristop posledično pomeni več maneverskega prostora za manipulacijo z nekim slikovnim gradivom za potrebe digitalnega ali klasičnega tiska. Za slikovno gradivo, kjer je zahtevana kar najvišja kvaliteta v sorazmerju z velikostjo dimenzij same fotografije, mora biti zagotovljena čimvečja ločljivost oz. DPI v izvorni datoteki.



boštjan.vovak@bsc.si 18



DPI/PPI

- Tiskalnik sestavlja sliko iz *okroglih pik*. Medtem ko so pike na monitorju kvadratne in poravnane druga ob drugi, so pike pri tisku okrogle in se pokrivajo ali odmikajo med seboj ter tako sestavljajo sliko. Večji kot je razmak med pikami, svetlejša je slika in obratno. **DPI** (dots per inch) označuje število pik oziroma razmak med njimi v enem palcu (1 palec = 2,54 cm), kot jih je zmožen določeni tiskalnik natisniti.

- PPI** (pixels per inch) označuje število oziroma gostoto pik na palec v digitalni sliki. 100 ppi pomeni 100 pik v dolžini enega palca, 50 ppi pomeni 50 pik v dolžini enega palca ... Pri pripravi digitalne slike za tisk se digitalne pike (piksli) spremenijo v tiskalne pike. 300 ppi je potrjeno število za optimalen tisk fotografskih slik. Težave pa nastajajo, ko želijo tiskarji (in drugi) tiskati fotografije s 300 dpi, čeprav dejansko mislijo



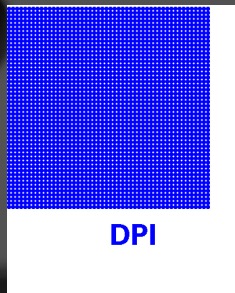
300 ppi. Načeloma je za optimalno kakovost tako priporočenih 300 pik na palec (300 ppi).

bostjan.vouk@bsc.si

19



DPI/PPI simulacija



bostjan.vouk@bsc.si

20



Vektorska grafika

- Alternativa rastrskemu zapisu, ki za izris slike namesto pikslov uporablja matematične formule, pa je vektorski zapis, ki je zelo razširjen in se ga uporablja predvsem v grafični in CAD/CAM industriji.
- Sestava in kvaliteta** - vektorsko sliko sestavljajo točke, ki so med seboj povezane z neko koordinatno potjo-črto-krivuljo ali daljicami. V računalniku je takšna slika shranjena v obliki geometričnih formul, katere s pomočjo posebnih grafičnih programov prevedemo v dejanske vektorje ali krivulje in jih nato po želji transformiramo (povečamo, vrtimo, raztegnemo itd...) brez poslabšanja ločljivosti slike, kot se to zgodi pri rastrskem zapisu. Izbrati in transformirati je mogoče tudi vsako posamezno komponento vektorsko zapisane slike, saj je vsaka komponenta v računalniškem pomnilniku definirana posebej. V teh pogledih vektorska grafika prekaša rastrsko.



bostjan.vouk@bsc.si

21



Primerjava

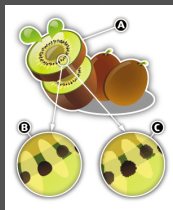
- **Bitna grafika** je pojem, povezan z računalniško sliko, ki je sestavljena iz množice obarvanih točk (ang. pixel), razporejenih v mreži z določeno gostoto. Pravimo ji tudi **rastrska grafika**. Najpreprostejša slika je črno-bela ali enobarvna, kjer imamo opraviti z 1-bitno grafiko, saj so tukaj točke le črne (oz. kakšne druge barve) ali bele (oz. kakšne druge barve) barve. V praksi pa se seveda srečujemo tudi z barvnimi slikami, pri katerih govorimo o barvni globini. Tako je slika, v kateri imamo 256 barv 8-bitna (2^8), slika s 65536 barvami 16-bitna, pri 24-bitnih slikah pa imamo na voljo kar 16,7 milijona barv oz. barvnih odtentkov.
- **Vektorska grafika**, včasih ji pravimo tudi objektna ali predmetna, se od rastrske grafike bistveno razlikuje. Namesto z množico obarvanih točk imamo tukaj opraviti s predmeti ali objekti. Predmeti oz. krivulje, iz katerih so sestavljeni, so predstavljeni s pomočjo geometričnih formul oz. matematičnih izrazov. Morda se nam na prvi pogled takšen opis zazdi težje razumljiv, vendar pa je v določenih okoliščinah delo z vektorsko grafiko precej elegantnejše v primerjavi z rastrsko grafiko.

boštjan.vovak@bsc.si

22



Vektorski/rastrski format



Slika zgoraj prikazuje glavno razliko med vektorsko in rastrsko grafiko ter učinek ob povečavi posamezne slike: (A) originalna vektorska slika, (B) 8x povečana vektorska slika, (C) 8x povečana rastrska slika. **Rezultat:** rastrska slika močno izgubi kvaliteto pri povečevanju, medtem ko lahko vektorsko sliko povečujemo brez izgube kvalitete.

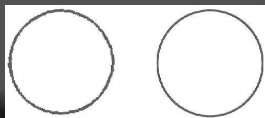
boštjan.vovak@bsc.si

23



Vektorski/rastrski format

- Primer daljice
 - V programu za delo z rastrsko grafiko bo daljica opisana z množico točk, ki bodo razporejene med obema krajiščema. Zapis v vektorski grafiki je bolj enostaven, saj si mora program zapomniti le nekaj osnovnih podatkov: koordinati obeh krajišč in debelino črte.
- Primer kroga



Prikaz kroga v bitni obliki (potrebovali smo 5.184 točk) in v vektorski obliki.

boštjan.vovak@bsc.si


24



TIFF (Tagged Image File Format)

- Postopek za stiskanje: LZW, RLE, JPEG.
 - LZW (Lempel-Ziv-Welch), RLE (Run length Encoding)
- Barvne globine: 1, sivinske slike do 8 in barvne do 24 bitov.
- Format TIFF je vsekakor eden tistih formatov, ki jih lahko brez pomislekov izmenjujemo med računalniki, ki so osnovani celo na različnih platformah.
- Poudarek je na podpori namiznemu založništvu, kjer se uporablja za skeniranje slik in za računalniško generirane slike.
- Njegova prednost je robustna struktura, saj standard TIFF vključuje številne vrste bitnih slik, tako glede barvne globine, kot tudi glede algoritmov za stiskanje in mnogih drugih lastnosti, ki jih preostali rastrski formati sploh ne poznajo.
- Pri datotekah TIFF poznamo celo vrsto algoritmov za stiskanje podatkov o točkah v sliki (našteti zgoraj). Od vseh pa je najpogostejši LZW, ki povzroča tudi najmanj težav. Poleg tega pa moramo vedeti, da so lahko podatki v datotekah TIFF zapisani tudi v nestisnjeni obliki.

bostjan.vovak@bsc.si 25



BMP (Windows BitMap)

- Postopek za stiskanje: RLE.
- Barvne globine: 1, 8, 24 in 32.
- Sliko shranjujemo v originalnem formatu takrat, ko še ni končana in jo želimo še spreminjati.
- Podatki so shranjeni kot slikovna pika za slikovno piko → zapis je zelo neučinkovit in datoteka je tako daljša kot bi lahko bila.
- Kodiranje: Informacijo o točkah zapišemo z dvema zlogoma, pri čemer prvi pomeni število ponovitev, drugi pa indeks v barvni tabeli. Vrednost 0 v prvem zlogu predstavlja ubežno zaporedje, drugi zlog pa pomeni: 0 za konec vrstice, 1 za konec slike, 2 za odmik naslednje točke (vodoravno ali navpično), 3 za rrr absolutni način (vrednost pomeni število točk, ki sledijo).
- Primeren je za slike, ki jih nameravamo uporabiti kot tapeto za ozadje. Pomanjkljivost pa je v tem, da je stiskanje podatkov možno le pri 4 in 8-bitnih zapisih, torej 24-bitnega zapisa ni možno stisniti.


bostjan.vovak@bsc.si 26



GIF (Graphics Interchange Format)

- Postopek za stiskanje: LZW.
- Barvne globine: indeksirane barvne palete do 256 barv.
- Podpira
 - animacije
 - omogoča shranjevanje več slik v eni sami datoteki
 - prepletanje
 - postopno izboljševanje slike, medtem ko se slika še nalaga, torej sprva motna slika se s časom izostri
 - transparentnost
 - eno izmed barv v barvni paleti določimo kot prozorno
- Omogoča učinkovito kodiranje slik visoke ločljivosti in kakovosti. Kodiranje temelji na uporabi vgrajenega LZW algoritma, ki za zmanjšanje količine podatkov izkorišča dejstvo, da se določena zaporedja ponavljajo, pri čemer ne izgubljamo podatkov. Torej najprej poišče slikovne pike z enako barvo in jih nato indeksira v barvno paleto slike, za kar ima na voljo največ 8 bitov oz. max 256 barv.
- Uporabljamo ga za: izrezke, risalne sličice, logotipe, animacije, skice ...


bostjan.vovak@bsc.si 27



JPEG (Joint Photographic Experts Group)

- Postopek za stiskanje: JPEG.
- Barvne globine: 1, 4, 8, 24 in 32.
- Omogoča prepletanje.
- Postopek JPEG zmora stisniti rastrske slike na zelo majhno velikost. Standardni postopek omogoča razmerja približno 10:1, čeprav ni nič presenetljivega, če uspemo rastrsko sliko stisniti celo v razmerju 100:1. Ključ za tako uspešnost pa je v tem, da pri stiskanju izgubimo del podatkov oz. jih program preze in zanemari. Stisnjena slika torej ni povsem enaka prvotni, vendar je izguba običajno tako majhna, da je s prostim očesom ne bomo zaznali.
- Sam algoritem stisne podatke o sliki tako, da natančno ohrani svetlost vsake slikovne pike, barve več sosednjih slikovnih pik pa združi in jim priredi enak povprečni barvni odtenek.


bostjan.vovk@bsc.si 28



PNG (Portable Network Graphics)

- Postopek za stiskanje: DEFLATE.
- Barvne globine: od 1 do 48.
- Omogoča prepletanje, transparentnost, stiskanje slik in korekcijo game (omogoča nadzor svetlosti slike oz. popravek barvnih vrednosti, ki jih prikaže monitor).
- Pred stiskanjem sliko s pomočjo filtriranja pripravimo na optimalno stiskanje. Pri vsaki slikovni vrstici izbran precejalni postopek na podlagi barv prejšnjih točk, napove barvo tekoče točke. Napovedana barva se odšteje od dejanske barve tekoče točke. Na ta način je precejeno slikovno vrstico pogosto mogoče stisniti bolj kot neobdelano slikovno vrstico.
- Postopek stiskanja podatkov DEFLATE je brez izgubni. Podatki, ki jih želimo stisniti, se razdelijo na bloke, kjer se vsak blok s pomočjo algoritma LZ77 (išče ponavljajoča se zaporedja, ki jih zapiše v krajši obliki) pretvori v niz, ki ga potem kodiramo s pomočjo Huffmanovega kodiranja.
- To je eno najbolj učinkovitih kodiranj, saj datoteko s formatom PNG običajno stisnemo za **5 do 25 odstotkov** učinkoviteje kot pri formatu GIF.


bostjan.vovk@bsc.si 29






Primerjava


	+	-
GIF	<ul style="list-style-type: none"> • Transparentna • Prepletanje • Animacije • Kompresija LZW 	<ul style="list-style-type: none"> • 256 barv • Ločljivost
JPEG	<ul style="list-style-type: none"> • Barvna paleta • Razširjenost • Velikost • Prepletanje 	<ul style="list-style-type: none"> • Stiskanje • Transparentna
PNG	<ul style="list-style-type: none"> • Odprt standard • Kompresija ZLIB • Transparentna • Velikost 	<ul style="list-style-type: none"> • Podpora


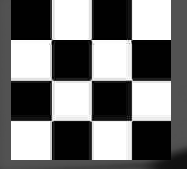
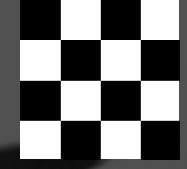
bostjan.vovk@bsc.si 30

 **Primerjava 1**

GIF	JPEG	PNG
73kB	32kB	302kB
		

29.11.2007 bozjan.vouk@bsc.si 31

 **Primerjava 2**

GIF	JPEG	PNG
1kB	1,1kB	192B
		

bozjan.vouk@bsc.si 32

 **Primerjava 3**

GIF	JPEG	PNG
11kB	6,8kB	5,8kB
		

bozjan.vouk@bsc.si 33



Ločljivost rastrskih slik

- Na spodnji animaciji je razvidno:
 - Slike so sestavljene iz mozaika barvnih ali črno-belih kvadratkov (pik), ki so postavljene ena poleg druge.
 - Če je pik dovolj in so dovolj majhne, človeško oko ne zazna posamezne pike, marveč jih sestavi v sliko.



boštjan.vovk@iit.si 34



Barvni prostori

- Barvne prostore uporabljamo za matematično predstavitev barvne palete.
- Obstaja več vrst barvnih prostorov (RGB, CMY, HSI, CIE Lab ter CIE Luv), od katerih sta najpogostejša RGB ter HSI.
- Izpeljanke CIE XYZ barvnega prostora so sledeče: CIE LUV, CIEUW in CIELAB
- Barvni prostori: RGB, CMY, XYZ, xyY, I1I2I3, UVW, LSLM, L*a*b*, L*u*v*, LHC, LHS, HSV, HSV Polar, HSI, HSI Polar, YUV, YIQ, YCbCr

boštjan.vovk@iit.si 35



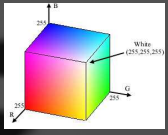
RGB

- **Barvni prostor RGB (aditivno mešanje)**
- Aditivno mešanje je mešanje barvnih valenc Rdeče, Zelene in Modre svetlobe. Z mešanjem polikromatskih barvnih dražljajev v enakih razmerjih nastane na belem zaslonu v zatemnjenem prostoru vtis bele svetlobe. Na zaslonu nastanejo novi barvni dražljaji v očesu pa nove barvne valence. Pogoji je, da vsi primarni barvni dražljaji istočasno osvetlujejo isto mesto zaslona in se iz njega hkrati odbijejo v oko.
- Primarni barvni dražljaji so Rd, Ze in Md (RGB).
- Sekundarni barvni dražljaji pa so Cy, Mg, Y in črna (CMYK).
- **sRGB** (standard RGB)

boštjan.vovk@iit.si 36

RGB

- na ortogonalnih oseh so tri barvne komponente (primarne barve ali značilke): R, G, B
- vsaka barvna komponenta ima območje od 0 do 255
- $\{p\} \{n\}^r = n^r$
- 16,78 Mio. (primer nakup monitorja) monitorji RGB
- vse tri komponente enake 0 → črna barva
- vse tri komponente enake 255 → bela barva
- uporablja se pri računalniški grafiki
- zasloni s katodno cevjo
- Barve, ki jih uporablja računalniški zaslon lahko opišemo z RGB barvnim prostorom, kjer vsaka od osnovnih barv predstavlja os, na kateri lahko posamezna barva doseže vrednost med 0 in 255 - torej 256 barvnih odtenkov vsake barve. Skupno število barv zaobjetih v takem prostoru je $256 \times 256 \times 256 = 16,78$ mil. barv.



boštjan.vovak@bcs.si 37

RGB

RGB barvni prostor

Prednosti	Slabosti
<ul style="list-style-type: none"> + dobimo direktno iz kamere + pretvorba ni potrebna + hitre operacije + široka uporaba v trenutnih sistemih + dobri rezultati, kadar so objekti dobro ločeni od okolice + za merjenje razlik med barvami ga lahko pretvorimo v R'G'B', vendar s tem izgubimo na hitrosti + veliko obstoječih - že razvitih algoritmov 	<ul style="list-style-type: none"> - pogosto potrebujemo dodatne informacije - zaznavna neuniformnost - potrebno veliko računskih operacij za spremembo svetlosti ali odtenka - veliko število podatkov (24 bitov) - občutljivost na sence - občutljivost na neenakomerno osvetlitev

boštjan.vovak@bcs.si 38

Pretvorba RGB v HEX

- Barva v RGB je oblike npr.: **233 205 15**, hex dobimo tako da **delimo** vsako številko s **16**.
- $$\left. \begin{array}{l} 233 \text{ MOD } 16 = 9 \\ 14 \text{ MOD } 16 = 14 \end{array} \right\} \begin{array}{l} \rightarrow 9 \\ \rightarrow E \end{array} \right\} \Rightarrow \text{E9}$$
- $233_{(10)} = E9_{(16)}$
- Ko vsa števila pretvorimo dobimo rešitev: **E9CD0F**

boštjan.vovak@bcs.si 39



Barvni prostor HSL

- posnemajo človekovo zaznavanje in interpretiranje barv
- ločena svetlost in kromatičnost barve
- Imajo naslednje komponente prostora:
 - H (ang. Hue): barvni odtenek nam pove, katera je njena zaznana barva; območje: $0 \leq H \leq 360$
 - S (ang. Saturation): nasičenost nam pove, kako prosojna oziroma nasičena je barva. Nenasičena barva = bleda barva, nasičena = živa barva; območje: $0 \leq S \leq 1$
 - L (V = Value, I = Intensity, L = Lightness) je komponenta svetlosti, prostori se ločijo po tem, kako je definirana. Pri vseh definicijah pa velja, da ima svetlejša barva tudi večjo vrednost komponente

boštjan.vovk@frc.si

43



Barvni prostor HSL

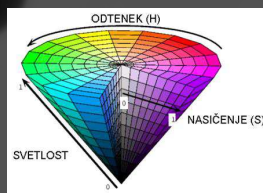
- Barvni ton (Hue - H)
 
- Nasičenost (Saturation - S)
 
- Svetlost (Lightness - L, Brightness - B)
 

boštjan.vovk@frc.si

44



Barvni prostor HSL



boštjan.vovk@frc.si

45



HSV, HSI

- **HSV (hue, saturation, value)**, also known as HSB (hue, saturation, brightness) is often used by artists because it is often more natural to think about a color in terms of hue and saturation than in terms of additive or subtractive color components. HSV is a transformation of an RGB color space, and its components and colorimetry are relative to the RGB color space from which it was derived.
- **HSI (hue, saturation, intensity)**. The Hue component describes the color itself in the form of an angle between $[0,360]$ degrees. 0 degree means red, 120 means green, 240 means blue. 60 degrees is yellow, 300 degrees is magenta. The Saturation component signals how much the color is polluted with white color. The range of the S component is $[0,1]$. The Intensity range is between $[0,1]$ and 0 means black, 1 means white.

boštjan.vouk@frc.si

46



HSI, HSV in HSL

Barvni prostori HSI, HSV, HLS

Prednosti	Slabosti
<ul style="list-style-type: none"> + po pretvorbi imamo manj računanja + zelo interaktiven + hitra sprememba barve ali svetlosti + razširjenost v sistemih strojnega vida + manjše število podatkov (ponavadi samo 16 bitov) + obstajajo korekcijski mehanizmi za računanje razdalj med barvami, vendar s tem izgubimo na hitrosti 	<ul style="list-style-type: none"> – potrebna pretvorba – zaznavna neuniformnost – slabost pri desaturiranih barvah (blizu bele ali črne) – singularnost odtenka na točki 0/255

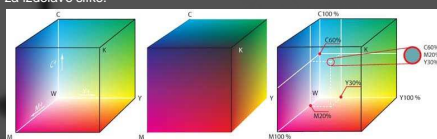
boštjan.vouk@frc.si

47



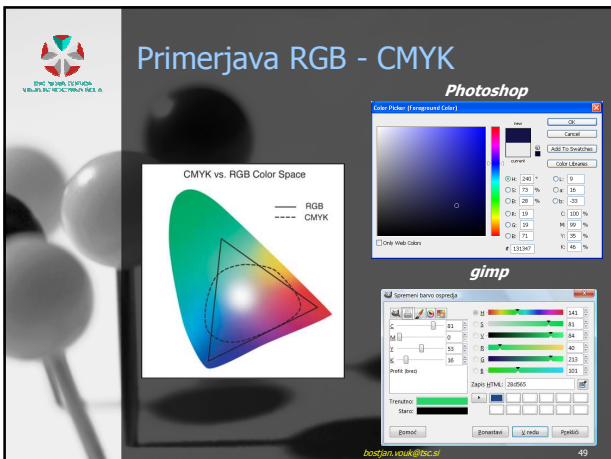
Barvni prostor CMYK

- **CMYK barvni model** je subtraktivni barvni model oziroma način v katerem so cian, magenta, rumena in črna barva sestavljene tako, da z njihovim mešanjem nastanejo druge barve.
- Ta barvni način se uporablja, ko želimo sliko natisniti, saj tiskalniki in večina tiskarskih strojev uporablja CMYK barve za izdelavo slike.

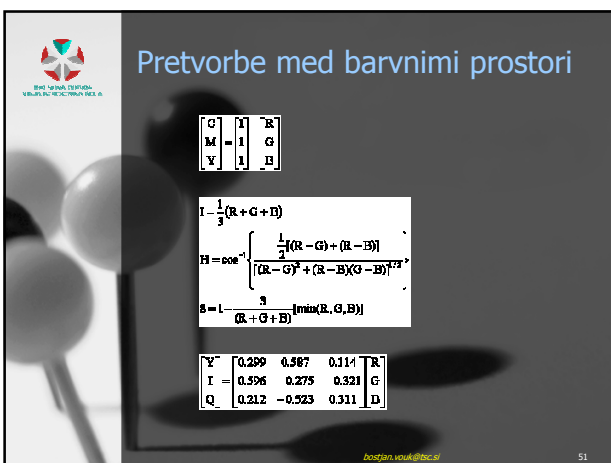



boštjan.vouk@frc.si

48









Predstavitev (računska) digitalne slike

- koordinate piksla so cela števila
- izmerjena vrednost piksla je celo število ali niz celih števil
- Velikost spomina, ki zavzema vrednost piksla določa globino slike:
 - 1 bit/piksel: črno-bela slika
 - 4 biti/piksel: 16 nians sive
 - 8 bitov/piksel: 256 nians sive
 - 3 x 8 bitov/piksel za RGB: $256^3 = 16777216$ barv

boštjan.vovk@frc.si 52



Barvni prostori in spomin

- RGB, HSB, YUV
 - za vsako komponento po 8 bitov: en piksel zavzema tri bajte
- CMYK
 - za vsako komponento po 8 bitov: en piksel zavzema štiri bajte

boštjan.vovk@frc.si 53



Histogram intenzitete

- Histogram v obsegu $[x_{\min}, x_{\max}]$ želimo "razvleči" na obseg $[y_{\min}, y_{\max}]$ (tipično $[0, 255]$)



- linearno "razvlečenje" histograma

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} y_{\max}$$
- Tisti pikslji ki so imeli vrednost (nians sive) x , bodo imeli po transformaciji vrednost y

boštjan.vovk@frc.si 54



Histogram intenzitete

Slika slabega kontrasta



Slika dobrega kontrasta



boštjan.vovk@fbc.si 55




Geometrijske transformacije

- skaliranje slike
 - primer: iz slike 512x512 pikslov želimo narediti sliko 2048x2048 pikslov
 - najenostavnejša rešitev: replikacija pikslov
 - Dobimo sliko sestavljeno iz kvadratkov!

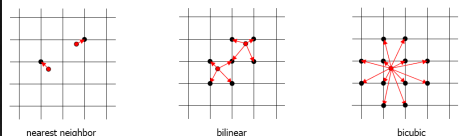


boštjan.vovk@fbc.si 56



Skaliranje s pomočjo interpolacije

- Poznamo tri tipične algoritme
 - nearest neighbor
 - vrednost novega piksela se računa kot vrednost najbližjega preslikanega piksela iz originala
 - bilinear
 - preslikani piksel vpliva na 4 najbližje piksle, odvisno od oddaljenosti
 - bicubic
 - preslikani piksel vpliva na 12 najbližjih pikslov, odvisno od oddaljenosti



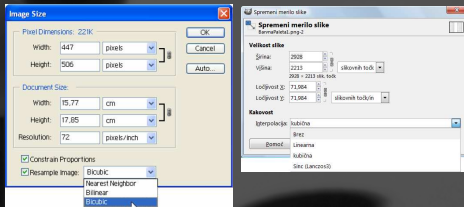
boštjan.vovk@fbc.si 57



Algoritmi interpolacije

Photoshop

Gimp



Dober algoritem za manjšanje je Lanczos3

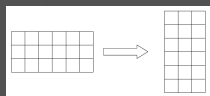
boštjan.vovk@frc.si

58

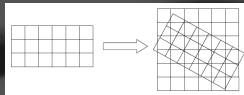


Geometrijske transformacije

- Rotacija Simulacija
 - rotacija za 90°, 180° ali 270°:
 - manipulacija stolpcev in vrstic matrike



- rotacija za poljuben θ
 - potrebna je interpolacija



boštjan.vovk@frc.si

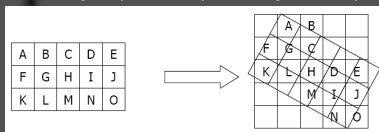
59



Rotacija

- Direktna transformacija Simulacija
 - koordinate rotiranega piksla
- za tako dobljene preslikane piksele izvajamo interpolacijo

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$



- nekateri deli originalne slike so lahko "odsekani"
- nekateri deli rezultirajoče slike lahko ostanejo prazni

boštjan.vovk@frc.si

60

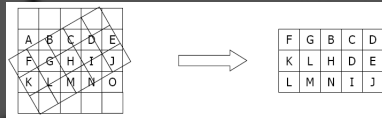


Rotacija

- Inverzna transformacija **Simulacija**
 - za vsaki piksel rezultirajoče slike izračunamo inverzno rotacijo na originalno sliko

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

- naredimo interpolacijo nad originalno sliko



- izračun se izvaja nad originalno sliko
- ne potrebujemo dodatnega spomina

boštjan.vovk@frc.si

61



Kompresija podatkov

- Prihranki
 - nekompresirana slika → JPEG
 - velikost kompresirane slike je nekajkrat manjša
 - nekompresirani CD audio → MP3
 - velikost kompresirane datoteke je tudi do 10 x manjša
 - nekompresirani PAL video → DivX
 - velikost kompresiranega videa je manjša za ~ 230 krat
- Zakaj ne bi delali vedno s kompresiranimi podatki?
 - Obdelava kompresiranih podatkov bi bila prezapletena.
 - Neposredno pred obdelavo se podatki dekomprimirajo.

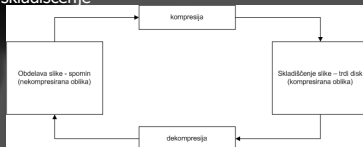
boštjan.vovk@frc.si

62

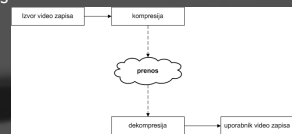


Kompresija podatkov

- Tok podatkov
 - skladiščenje




- prenos



boštjan.vovk@frc.si

63



Kompresija podatkov

- Tipi kompresije
 - kompresija brez izgube podatkov (lossless)
 - izguba podatkov zaradi kompresije/dekompresije ni sprejemljiva
 - tekstovni podatki
 - pomembni tekstovni dokumenti
 - numerički podatki
 - podatki o finančnih transakcijah

bostjan.vovk@bsc.si 64



Kompresija podatkov

- kompresija z izgubo podatkov (lossy)
 - ko je izguba podatkov zaradi kompresije/dekompresije sprejemljiva do določene mere
 - statične slike – če dekomprimirana slika dovolj dobro odraža originalno
 - avdio posnetki – če dekomprimirani posnetek dovolj dobro odraža originalni
 - video posnetki – če dekomprimirani posnetek dovolj dobro odraža originalni
 - sprejemljivost izgube ocenjuje uporabnik

bostjan.vovk@bsc.si 65



Lossless kompresija

- Klasični algoritmi za lossless kompresijo
 - Run-length encoding
 - LZW (Lempel-Ziv-Welch) algoritem
 - GIF format
 - Huffmanovi znaki

bostjan.vovk@bsc.si 66



Primer 2

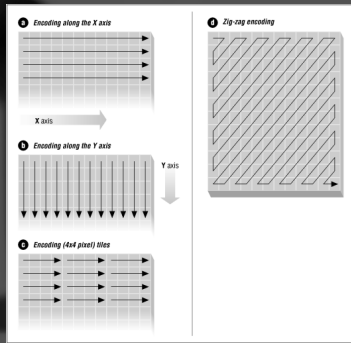
- Primer:
 - AAAAAAbbbXXXXt
- S pomočjo RLE lahko kompresiramo v 4, 2byte pakete
- **Rešitev:** 6A3b5X1t
- Nov paket je generiran vedno ko se niz zaporednih enakih vrednosti konča ali ko število znakov v nizu preseže maksimum.

boštjan.vovk@isc.si

70



Možnosti Run-length encoding



boštjan.vovk@isc.si

71




Stiskanje po postopku LZW

- v primerjavi z algoritmom RLE mnogo bolj natančen
- sloni na prepoznavanju določenih vzorcev v sliki in ne na prepoznavanju posameznih točk
- prebere vrednost vseh točk v sliki in izdela tabelo s kodami, ki ustrezajo ponavljajočim se prepoznavnim vzorcem točk
- tako stisnjena bitna slika zavzame 1/4 ali 1/3 prvotne velikosti (odvisno od vrste slike)
- slike z izrazitimi vzorci, večjimi deli enakomerne barve je moč stisniti celo na 1/10 prvotne velikosti
- pomanjkljivosti LZW algoritma se pokažejo predvsem pri skeniranju fotografij ali pri slikah, ki nimajo izrazitih vzorcev

boštjan.vovk@isc.si

72




Lempel-Ziv-Welch

- predstavnik družine algoritmov zasnovanih na slovarju
- pred kompresijo in dekompresije moramo poznati abecedo
- Primer
 - abeceda: {a,e,g,m,x}
 - tekst za kompresijo: megaxmegaxmegaxmeexmee
 - Začetni del

indeks	znak
1	a
2	e
3	g
4	m
5	x

boštjan.vovk@frc.si 73




Lempel-Ziv-Welch

- Začnemo od začetka teksta
- Pregledamo tekoči znak
 - Ali obstaja v slovarju?
 - Obstaja z indeksom 4
 - Tekočemu znaku dodamo naslednjega
 - **me**
 - Ali obstaja sekvenca v slovarju?
 - ne obstaja
 - Dodamo na konec slovarja
 - na izhod napišemo indeks tistega znaka katerega smo nazadnje našli v slovarju – to je indeks 4 za simbol m
 - Naslednji znak (i) postane tekoči

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me

vhod: m e g a x m e g a x m e g a x m e e x m e e
 izhod: 4

boštjan.vovk@frc.si 74



Lempel-Ziv-Welch

- Pregledujemo tekoči znak
 - Ali obstaja e v slovarju?
 - obstaja z indeksom 2
 - tekočemu simbolu dodamo naslednjega – eg
 - Ali obstaja sekvenca eg v slovarju?
 - ne obstaja
 - Dodamo jo na konec slovarja
 - na izhod napišemo indeks tistega znaka katerega smo nazadnje našli v slovarju – to je indeks 2 za znak e
 - naslednji simbol (g) postane tekoči

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me
7	eg

vhod: m e g a x m e g a x m e g a x m e e x m e e
 izhod: 2 4

boštjan.vovk@frc.si 75



LZW – kompresija

- Pregledujemo tekoči znak
 - Ali obstaja g v slovarju? –obstaja, z indeksom 3
 - tekočemu simbolu dodamo naslednjega – ga
 - Ali obstaja sekvenca ga v slovarju?
 - Ne obstaja

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me
7	eg
8	ga

- Dodamo jo na konec slovarja
 - na izhod napišemo indeks tistega znaka katerega smo nazadnje našli v slovarju – to je indeks 3 za znak g
 - naslednji simbol (a) postane tekoči

```
vhod| m e g a x m e g a x m e g a x m e e x m e e
izhod| 4 2 3
```



LZW – kompresija

- Pregledujemo tekoči znak
 - Ali obstaja a v slovarju? –obstaja, z indeksom 1
 - tekočemu simbolu dodamo naslednjega –ax
 - Ali obstaja sekvenca ax v slovarju?
 - Ne obstaja

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me
7	eg
8	ga
9	ax

- Dodamo jo na konec slovarja
 - na izhod napišemo indeks tistega znaka katerega smo nazadnje našli v slovarju – to je indeks 1 za znak a
 - naslednji simbol (x) postane tekoči

```
vhod| m e g a x m e g a x m e g a x m e e x m e e
izhod| 4 2 3 1
```




LZW – kompresija

- Pregledujemo tekoči znak
 - Ali obstaja x v slovarju? –obstaja, z indeksom 5
 - tekočemu simbolu dodamo naslednjega –xm
 - Ali obstaja sekvenca xm v slovarju?
 - Ne obstaja

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me
7	eg
8	ga
9	ax
10	xm

- Dodamo jo na konec slovarja
 - na izhod napišemo indeks tistega znaka katerega smo nazadnje našli v slovarju – to je indeks 5 za znak x
 - naslednji simbol (m) postane tekoči

```
vhod| m e g a x m e g a x m e g a x m e e x m e e
izhod| 4 2 3 1 5
```


 LZW – kompresija

- Pregledujemo tekoči znak
 - Ali obstaja m v slovarju? –obstaja, z indeksom 4
 - tekočemu simbolu dodamo naslednjega –me
 - Ali obstaja sekvenca me v slovarju?
 - obstaja, z indeksom 4
 - tekočemu simbolu dodamo naslednjega – meg
 - Ali obstaja sekvenca meg v slovarju?
 - Ne obstaja
 - Dodamo jo na konec slovarja
 - Na izhod napišemo indeks tiste sekvence znakov, ki smo jo nazadnje našli v slovarju – to je indeks 6 za sekvenco me
 - naslednji simbol (g) postane tekoči

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me
7	eg
8	ga
9	ax
10	xm
11	meg

Vhod: m e g a x m e g a x m e g a x m e e x m e e
 Izhod: 4 2 3 1 5 6 ← prihranek

boštjan.vouk@bsc.si 79


 LZW – kompresija

- Pregledujemo tekoči znak
 - Ali obstaja g v slovarju? –obstaja, z indeksom 3
 - tekočemu simbolu dodamo naslednjega - ga
 - Ali obstaja sekvenca ga v slovarju? –obstaja, z indeksom 8
 - Tekoči sekvenci dodamo naslednji simbol – gax
 - Ali obstaja sekvenca gax v slovarju?
 - Ne obstaja
 - Dodamo jo na konec slovarja
 - Na izhod napišemo indeks tiste sekvence znakov, ki smo jo nazadnje našli v slovarju – to je indeks 8 za sekvenco ga
 - naslednji simbol (x) postane tekoči

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me
7	eg
8	ga
9	ax
10	xm
11	meg
12	gax

Vhod: m e g a x m e g a x m e g a x m e e x m e e
 Izhod: 4 2 3 1 5 6 8

boštjan.vouk@bsc.si 80

 LZW – kompresija

- Končni rezultat

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me
7	eg
8	ga
9	ax
10	xm
11	meg
12	gax
13	xme
14	ega
15	axm
16	mex
17	ex
18	xmee

Vhod: m e g a x m e g a x m e g a x m e e x m e e
 Izhod: 4 2 3 1 5 6 8 10 7 9 6 2 13 2

boštjan.vouk@bsc.si 81

LZW – dekompresija

- Začnemo na vhodu (z desne strani)
- Pregledujemo tekoče število: 4
 - Indeks 4 odgovarja znaku m
 - Napišemo ga na izhod
 - Pregledujemo naslednje število: 2
 - Združimo znaka, ki odgovarjata tekočem in naslednjemu številu – me
 - Če sekvenca me ne obstaja v slovarju jo dodamo
 - Naslednji znak postane tekoči

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me

Vhod: 4 2 3 1 5 6 8 10 7 9 6 2 13 2
Izhod: m

LZW – dekompresija

- Pregledujemo tekoče število: 2
 - indeks 2 odgovarja znaku e
 - Napišemo ga na izhod
 - Pregledujemo naslednje število: 3
 - Združimo znaka, ki odgovarjata tekočem in naslednjemu številu – eg
 - Če sekvenca eg ne obstaja v slovarju jo dodamo
 - Naslednji znak postane tekoči

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me
7	eg

Vhod: 4 2 3 1 5 6 8 10 7 9 6 2 13 2
Izhod: m e

LZW – dekompresija

- Končni rezultat

Slovar je enak kot pri kompresiji

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me
7	eg
8	ga
9	ax
10	xm
11	meg
12	gax
13	xme
14	ega
15	axm
16	mee
17	ex
18	xmee

Vhod: 4 2 3 1 5 6 8 10 7 9 6 2 13 2
Izhod: m e g a x m e g a x m e g a x m e e x m e e



LZW

- Vsebina slovarja odslikava realno vsebino teksta
 - Nahajajo se samo tisti pari, trojke, četvorke, itd., ki se zares pojavljajo v tekstu
- Za kodiranje števil v obsegu 1-18 (število sekvenc v slovarju) potrebujemo 5 bitov
 - $2^4 = 16$
 - $2^5 = 32$

indeks	znak
1	a
2	e
3	g
4	m
5	x
6	me
7	eg
8	ga
9	ax
10	xm
11	meg
12	gax
13	xme
14	ega
15	gam
16	mee
17	ex
18	xmee

boštjan.vovk@bsc.si

85



LZW

- Simulacija

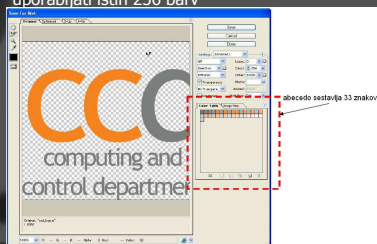
boštjan.vovk@bsc.si

86




LZW – primeri

- GIF format
 - piksli imajo lahko barvo iz danega skupka barv (= abeceda)
 - max. število barv je 256; vedno ne moremo uporabljati istih 256 barv



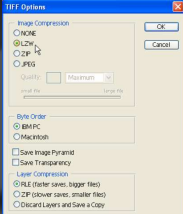

boštjan.vovk@bsc.si

87




LZW – primeri

- TIFF format
 - RGB ali CMYK model
 - Lahko se izbere LZW kompresija


boštjan.vovak@bsc.si 88



LZW – primeri (Photoshop)

- GIF format
 - uporaba barv iz končne palete ni vezana samo za GIF format

Photoshop



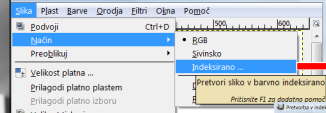
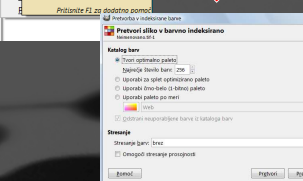

boštjan.vovak@bsc.si 89




LZW – primeri (gimp)

- GIF format
 - uporaba barv iz končne palete ni vezana samo za GIF format

gimp


boštjan.vovak@bsc.si 90



Izračuni - slika

- Podano imamo sliko formata A4 v CMYK barvnem prostoru z ločljivostjo 300 dpi
 - dimenzija A4: 8,27in x 11,69in = 96,68in²
 - Skupno število pikslov: 96,68in² x 300²= 8701200
 - 4 bajti/piksel (CMYK)
 - skupna velikost slike
 - 8701200 x 4 = 34804800 ≈33.2 MB
- Opomba:** Pretvorba: 1in = 2.54cm


boštjan.vovk@bsc.si 91



JPEG

- Format definira štiri načine kompresije
 - sequential
 - Ta način največkrat razumemo pod "JPEG kompresija"
 - kompresija z izgubami
 - progressive
 - Podoben načinu sequential
 - Format, ki je prilagojen velikim slikam
 - lossless
 - Možnost kompresije brez izgub (redka uporaba)
 - hierarchical
 - slika se kodira v več resolucijah, tako da uporabnik gleda slike v nizki resoluciji brez dekompresije visokoresolucijskih slik

boštjan.vovk@bsc.si 92



JPEG kompresija

- transformacija barv v svetlostni/barvnostni prostor
- predstiskanje (downsample) barvnih komponent z združevanjem pikslov v skupine točk
- nanašanje diskretne kosinusne transformacije na te skupine točk in s tem odstranjevanje odvečnih podatkov
- kvantizacija DCT blokov (skupine točk), z izluščevanjem barv na katere človeško oko ni toliko občutljivo
- kodiranje preostalih podatkov s Huffmanovim algoritmom

boštjan.vovk@bsc.si 93

Sequential encoding

dekompozicija na bloke 8 x 8 pikselov

diskretna kosinova transformacija

kvantizacija

Huffmanovo kodiranje

tabele kvantizacije

boštjan.vovk@bcs.si 94

DCT

- blok pikselov 8x8 predstavimo kot diskretno funkcijo dveh spremenljivk podatkov v 64 točk

$$F(u, v) = \frac{\Lambda(u)\Lambda(v)}{4} \sum_{i=0}^7 \sum_{j=0}^7 \cos \frac{(2i+1)u\pi}{16} \cdot \cos \frac{(2j+1)v\pi}{16} \cdot f(i, j)$$

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \xi = 0 \\ 1 & \text{drugače} \end{cases}$$

boštjan.vovk@bcs.si 95

DCT

- Za izračun vsakega DCT koeficienta (na položajih i in j) formula sešteje prispevke za vsakega od štirinšestdesetih točk (8x8) na položajih i in j. Z vrednostni točk, kjer je f(i,j) med 0 in 255, dobi enačba AC (high-order) koeficiente med -1023 in 1023, te pa predstavimo z 11-bitnim predznačenim integerjem. DC (low-order) koeficienti pa bi bili v vrednostih med 0 in 2040, ta nepredznačen 11-bitni integer bi potreboval različno strojno in programsko opremo, če bi hoteli z njim računati.
- Zato se vsaki točki odšteje vrednost 128. To ne vpliva na AC koeficiente, DC koeficienti pa padejo v isti razpon, tako da jih lahko predstavimo z 11-bitnim predznačenim integerjem.

boštjan.vovk@bcs.si 96



Kvantizacija

- Občutljivost HVS-ja za vsako osnovno funkcijo DCT-ja se lahko določi na podlagi poskusov, s temi pa lahko določimo možno kvantizacijsko matriko. Taki poskusi so pripeljali do dveh kvantizacijskih matrik- eno za svetlostno, drugo za barvnostno razlikovanje. Te matrike niso del JPEG standarda, vseeno pa jih večina JPEG aplikacij uporablja.
- Vrednosti v matrikah predstavljajo kvantizacijske intervale. Vsak DCT koeficient delimo z vrednostjo iz matrike. Zaradi enostavnejšega računanja, se magnitudo vsakega koeficienta poveča za polovico kvantizacijskega intervala pred deljenjem. Rezultat se zaokroži navzdol na celoštevilsko vrednost. Za rekonstrukcija je potem potrebno le zmnožiti kvantizacijski koeficient s kvantizacijskim intervalom.

boštjan.vovk@iis.si

97



Kvantizacija

- Večji kot je kvantizacijski koeficient, več podatkov se izgubi. Ker je oko manj občutljivo na barvne spremembe, so koeficienti v barvnosti tabeli večji. V tem koraku se kontrolira kvaliteto JPEG kompresije. Z nastavljanjem **q faktorja** povečujemo ali zmanjšujemo vrednosti kvantizacijskih matrik. Kvantizacijska tabela, ki je bila uporabljena pri kompresiji slike, se zapiše skupaj s sliko, tako da se jo lahko ponovno uporabi pri dekompresiji.
- Kvantizacija je tisti del JPEG algoritma, pri katerem se izgubi največ informacij.

boštjan.vovk@iis.si

98



Q faktor

Izbor kvalitete komprimirane slike predstavlja izbor ene v nizu vnaprej definiranih Q matrik

boštjan.vovk@iis.si

99



Huffmanovo kodiranje

- Ko gredo DCT koeficienti skozi kvantizacijski postopek je verjetnost, da bodo ničelne vrednosti v zgornjem levem kotu majhna. Zaradi te lastnosti skenira vrednosti blokov v cik-cak obliki. Tako se dobi večjo podobnost skupaj zapisanih vrednosti, kar omogoča učinkovitejše stiskanje.



- Na zgornji sliki je prikazano cik-cak skeniranje bloka. Zaporedje teh vrednosti se nato kodira s Huffmanovim kodiranjem. Huffmanovo kodiranje je brez izgubno, stisne pa ponavadi za **10 do 15 procentov**.


boštjan.vovk@fbc.si 100



Dodatek

- Algoritmi, ki se uporabljajo pri interpolaciji pri spreminjanju velikosti slike v programskem orodju Gimp
 - Nearest-neighbor
 - Linear
 - Bicubic
 - Lanczos


boštjan.vovk@fbc.si 101



Gimp interpolation algorithms

- GIMP offers three interpolation methods, which differ in quality and speed. In general, the better the quality, the more time the interpolation takes. The methods are:
 - None (sometimes called "Nearest Neighbor"): The color of each pixel is copied from its closest neighboring pixel in the original image. This often results in aliasing (the "stair-step" effect) and a coarse image, but it is the fastest method.
 - Linear (sometimes called "Bilinear"): The color of each pixel is computed as the average color of the four closest pixels in the original image. This gives a satisfactory result for most images and is a good compromise between speed and quality. This method is relatively fast, but still provides fairly good results.
 - Cubic (sometimes called "Bicubic"): The color of each pixel is computed as the average color of the eight closest pixels in the original image. This usually gives the best result, but it naturally takes more time. The method that produces the best results, but also the slowest method.
 - Sinc (Lanczos 3): New with GIMP-2.4, this method gives less blur in important resizing.

boštjan.vovk@fbc.si 102




Lanczos

- Lanczos resampling is a multivariate interpolation method used to make digital images larger or smaller by resampling them. That is to say, the final values are a weighted sum of the original values (based on relative position to the original image) where the weighting is given by the Lanczos weighted sinc function.
- Lanczos uses a windowed product of sinc functions as a convolution kernel for image resampling. In one dimension, its formula is given by:

$$L(x) = \begin{cases} \frac{2 \sin(\pi x) \sin(\frac{3}{2}\pi x)}{\pi^2 x^2} & -2 < x < 2, x \neq 0 \\ 1 & x = 0 \\ 0 & \text{otherwise} \end{cases}$$
- The filter is named after Cornelius Lanczos, because he showed how to use Fourier series and Chebyshev polynomials for various problems where it was not used before.


basim.vouk@bcu.cz 103



Bicubic interpolation

- In numerical analysis, a branch of mathematics, bicubic interpolation is one of the most common interpolation methods in two dimensions. With this method, the value $f(x, y)$ of a function f at a point (x, y) is computed as a weighted average of the nearest sixteen samples in a rectangular grid (a 4×4 array). Here, two cubic interpolation polynomials, one for each plane direction, are used.
- Bicubic interpolation results in an interpolating function which is continuous, has continuous first partial derivatives, and has continuous cross derivatives everywhere.

basim.vouk@bcu.cz 104



Bicubic interpolation

Formula

- The bicubic interpolation is calculated as follows:
- $$a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{21}x^2y + a_{12}xy^2 + a_{22}x^2y^2 + a_{30}x^3 + a_{03}y^3 + a_{31}x^3y + a_{13}xy^3 + a_{32}x^3y^2 + a_{23}x^2y^3 + a_{33}x^3y^3$$
- Or, in more compact form:

$$\sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$
- Suppose the function values f and the derivatives f_x, f_y and f_{xy} are known at the four corners $(0,0), (1,0), (0,1),$ and $(1,1)$ of the unit square.

basim.vouk@bcu.cz 105



Bicubic interpolation

- The procedure used to calculate the coefficients a_{ij} depends on the interpolated data source properties. If we are interpolating a function with known derivatives, the common approach is to use the height at the four vertices, together with three derivatives at each vertex. The first derivatives h'_x and h'_y express the slope of the surface in the x and y directions, while the second (cross) derivative h''_{xy} represents the slope in both x and y. In terms of solving the coefficients of the formula, these values can be expressed by differentiating the x and y vectors independently, and then consecutively. For each of the vertices of the grid cell, the local coordinates (at (0,0), (1,0), (0,1) and (1,1)) can be input into these equations to generate the 16 equations to solve.

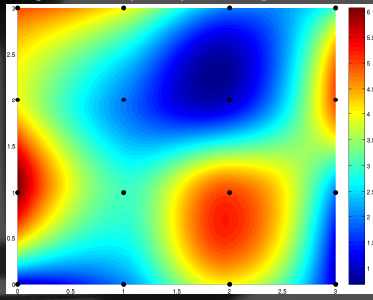
bozjan.vouk@bnc.si

106



Bicubic interpolation

- Bicubic interpolation on the square $[0,3] \times [0,3]$ consisting of 9 unit squares patched together.



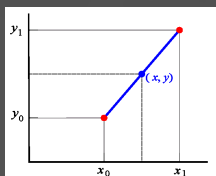
bozjan.vouk@bnc.si

107




Linear interpolation

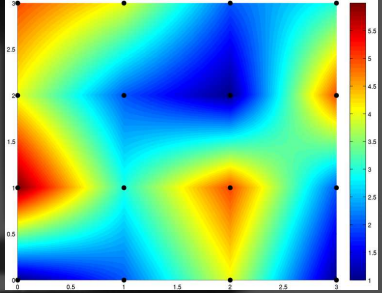
- If the two known points are given by the coordinates (x_0, y_0) and (x_1, y_1) , the linear interpolant is the straight line between these points. For a value x in the interval (x_0, x_1) , the value y along the straight line is given from the equation
$$\frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0}$$
- which can be derived geometrically from the figure on the right.
- Solving this equation for y , which is the unknown value at x , gives
$$y = y_0 + (x - x_0) \frac{y_1 - y_0}{x_1 - x_0}$$
- which is the formula for linear interpolation in the interval (x_0, x_1)




bozjan.vouk@bnc.si

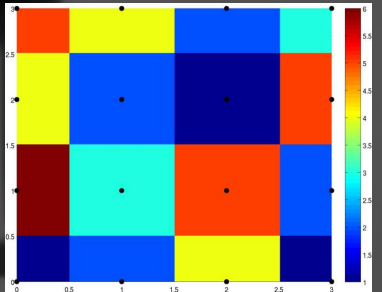
108

 **Bilinear interpolation**




boostjan.vouk@hsc.nl 109

 **Nearest-neighbor interpolation**



boostjan.vouk@hsc.nl 110

 **Algorithm comparison**

- Linear gives you a mid-quality interpolation. Linear interpolation has less quality trade off for speed than Nearest Neighbour interpolation algorithm does. Nearest Neighbour is the fastest way to interpolate, but it has the lowest quality of the interpolation algorithms. The Cubic interpolation algorithm produces the highest quality of the three interpolation types supported by GIMP. However it is considerably slower than the linear interpolation algorithm.

boostjan.vouk@hsc.nl 111
