Grafika képfeldolgozás

Tárolás:

Első módszer: Bittérképes/raszteres kép

felbontás: nem fizikai mennyiség de fizikai változást is okoz. felbontás: (részletgazdagság)

DPI(PPI) Dot/Inch Pixel/Inch felbontás mértékegysége 1 Inch=2,54cm Jelentés: 100DPI=100pixelt kell nyomtatni 2,54cm-nként.

Zh-n : X és Y felbontású kép ami 100DPI-s akkor helyesen megjelenítve papíron ? cm-nek kell lennie?

Gondolnánk: pixel 4zet alakú de NEM!

DAR (display aspect ratio) van egy kép amit ha nem torzítok el akkor ezzel jeleni meg pl tv-n a 16:9es

SAR(storage aspect ratio) X:Y felbontás amit tárolásnál alkalmazunk

PAR(pixel aspect ratio) kettő kombinációja -> kiszámolhatóak a pixelek oldal arányai a SAR és DAR ból ha SAR=DAR akkor a pixelek négyzet alakúak

Ezt az egészet majd meg kell jelenítenem egy eszközön aminek lehet h pixel aránya és a Dar is más.(monitor felbontása stb.) (DVD hátlap részlet 1.85:1 videoanyag 16:9es megjelenítéssel felül alul csík ha 16:9 akkor nincs csík) Régi 4:3 arány (sorozatok) mai 16:9

Ha a megjelenítőn más az arány akkor megjeleníthetem : felül alul csík-> letterboxing oldal csík-> pilarboxing

tárolásnál vagy felül alul csíkosan tárolunk vagy levágjuk 16:9es az anyag de 4:3as formátumban tárolom ekkor lehet úgy hogy nem fixen vágok le hanem a tartalomnak megfelelően vágok le: pan-and-scan

Függőlegesen: tilt-asd-scan

Miért úgy kezdődött a régi TV-k 4:3-al? miért az arány, hogy négyzetesből szélesvásznú felé tart?

Régen: katódsárcsöves TV-nél az elektronsugár soronként járja be és úgy frissít manapság már nem kell végigfutni a sorokon a sugárral ezért lehet lapos. DE miért szélesedik: Emberi látómező nem négyzetes hanem „szélesvásznú”

Egy bittérképes képen hogyan tudok vmilyen alakzatot megjeleníteni?

Ehhez az kell, hogy a bittérképes klaszteres képeken kívül milyen tárolási lehetőségeim vannak még?

* vektorgrafikus képek : képeket matematikai görbékkel leírható módon tároljuk(nincsenek pixelek) van egy végtelen pontosan leírható alakzatom és meg utdom nézni h xhez milyen y tart -> bármilyen felbontásban felrajzolgatom a képet.

Módszer: Bazier(Bezier?) görbék: legegyszerűbb: szakasz adott 2 pont és lineárisan végigmegyünk a szakaszon és aott rajzolunk ki pontot.

3 pont seetém ugyanazt követjük mint előbb végigmegyünk p0-p1en és p1-p2-n DE a két futópontot ois összekötjük és azon is végigfutunk és ezt rajzoljuk ki -> görbét kapunk.

4 pont esetén p0-p1 p1-p2 p2-p3 p3-p4 2-2 összekötő 2szakaszon végighaladó pontok szakasza és azokon végigfutó pontot rajzoljuk ki…

Nre is megy.

Általában 5 pontú Bazierrel le tudjuk írni a számunkra fontos görbéket.

Hasznos: fv-k ábrázolása és betűtípusaink(betűméret)-> nem kell minden betűjöz és betűmérethez külön bittérképes képet tárolni. monitorunk bittérképes + nyomtató is.(plotter az vektoros nyomtató)

Konvertálás: triviális mo: vektorgraf vonal-> bittérképes kitalálom a felbontást: x:Y csináljunk fekete fehér bittérképes képet ahol a vonalam metszi a pixelt legyen fekete a többit hagyja(bináris képzés).(lépcsős lesz)nem ezt csináljuk! Szürke skálás képet akarunk csinálni mert sokkal szebb lesz az eredmény! Megldás: mekkora hosszú a pixelt metsző szakaszunk ennek megfelelően skálázzuk a színt.(átló fekete). Módszer: lépcsős bináris képből simítunk úgy hívjuk hogy antialiasing. PL vektorgrafos cucca: font térkép (guglimappok) vekltorosan tárolva belenagyítok és újra számolja

Bittérképesből bittérképesbe képzés: átméretezés20\*20as->200\*200asra ->minőség vesztés nélkül megtehető midnen pixel 10\*10es lesz. Látjuk, hogy pusztán a nagyobb felbontású nem jelenti h jobb minőségű.

Mi van ha nem pont egésszámú töbszörösére méretezem át a képet?20\*20ról 25\*25re pixel határok nem esnek egybe! Mit csinálhatok? mennyire fedi a pixxel az eredetit-> szürke skálásat tudok csinálni belenagyítunk akkor látjuk, hogy sima fekete-fehéresből kiindulva szürkés lesz. egész szám szorosára növelve semmit nem ront el. Kis arányú átméretezés rossz.

Hogyan tároljuk a pixelek színét?

szinterek: kérdés hogyan tudjuk felbontani vhogy a szineket. Szemben: színérzékelő receptorokból 3 féle van

érzékenység~hullámhossz 3 görbe az egyik csúcsa Rnél másik G (inkább sárgás) B-nél a harmadik

Monokromatikus fény esetén: belenézet és az y tengelyen látjuk melyiket milyen mértékben ingerli.(valóságban bonyolultabb) -> a 3 komponensnek a jelei jutnak el az agyunkig. Nem látunk monokromatikus fényt általában mert ritkán találkozunk vele.

* kérdés: egy színt akkor többféleképpen ki tudunk keverni? igen ki lehet-> szemünk nem tud megkülönböztetni egy csomó színt (kimérhetően különbözik de mi azonosnak látjuk)

Vannak emberek akiknek 4 színérzékelő receptora van ezért a mások által ugyanolyannak látott színt különbözőnek érzékeli.(genetikai mutáció és a nőkben gyakoribb)

Első színerünk: RGB színtér ezek keverésébőől állítjuk elp a színt.

általában 8bit/komponens felbontásban tároljuk el a színeket. (előjel nélküli egész) -> 1pixel tárolására: 3\*8bit = 24bit vannak területek ahol nem elegendp ez a felbontás. legegyszerűbb formátum: bmp lényegében rgb picelek felsorolása.(elején x\*y mint 2d grafikánál progból)

Néha alkalmazzák 16 biten : 5:6:5ös eloszlásban.

palettán alapuló színtér: ahelyett h RGB komponenseket adnánk meg mint a palettán ki van keverve néhány szín és azokból tudok keverni. pl GIF

pl lista 1. Szín 2. 3. … n. RGB-ben meg vannak adva (színpaletta)

képet úgy tárolom, hogy 1-1 pixelnél nem RGB-t mondok meg hanem pixelenként megmondom h 1-es színű 2-es 5-ös stb.

YUV(YCrCb) színtér Y-> fényesség kompunens UV->szín (UV->00 akkor szürke skála fekete-fehér tvrőlátállásnál hozták létre) (8bit/komp) az MPEG ezzel a színtérrel történik. Miben különbözik még? (Biológiai háttré: csapok pálcikák stb.. receptorok elhelyeztkedése sem azonos-> középen főleg színérzékelő receptorok szélen meg kb nincs -> a periférián nem látunk szineket de nagyon kis kül-t meglátunk fényességben. de általában is a fényérzékelők sűrűbbek mint a színérzékelők)

Sokkal jobb lenne nagy felbontásban fényességet és kicsiben szineket tárolni.

4\*2-es cucc

1.minden pixelnek táruljuk az y,u,v-t 1->4:4:4

2. módszer: minden pixelnek van Y-ja de csak kettessével van 1 UV 4:2:2

-> pixelenkén ugyanaz az UV(szín komponensek de a színhez hozzá tartozik az Yhoz) nayjából aznons színűek

3. változat midnenkinek Y-ja de 4essével van 1 közös UV komponens : 4:2:0

1-es: y-> 8\*8 bit UV-> 8\*16 bit

2-es 8\*8 bit Y 4\*16 UV [bit/8 pixel]

3-as 8\*8 2\*16

Palettásnál: pl 8 színű őaletta minden pixelhez 3 bit kell (8\*3)+[paletta 8\*24]

RGB: 8\*24

* tömöríts nélkül színteret megváltoztatva kevesebb helyet foglal. kis számú sínpaletta-> színpalettás a legjobb embereknek megfelelőre a legjobb a YUV

Nyomtatóknál CYMK színteret használnak Cián Sárga Magenta Fekete(Cyan Yellow Magenta K mint Black(történetileg alakult így eredetileg Key ből ott volt jukas a lap ahol nyomtatni kellett)) feketét úgy állít elő, hogy minden színt odanyom(antialiasing)

fizikai megvalósítás: [[R][G][B]] LCD-ken fordított sorrendben. sibpixelekbőláll -> profi antialiasing a szubpixeleket is változtatja -> 3\*os felbontásban tudja az antialiasingot tolni

Mozgó képek: egy vmivel több ennél a mozgókép FPS(frame per sec)

Videó/ TV adás 25->európai(PAL), 30(NTSC), 24film(amikor nem digitálisan rögzítik)

Nem egész Framek mennek végig hanem félképek (1. lépés páros sor 2. páratlanok) interlaced

párja maikor midnen sort tárolják. progresive.

Baj ? FPS-es a fim de a megjelenítő nem egész számú töbszöröse?

úgy csinálnak filmet hogy a képkockákat felbontják 2framekre 1 frame kicsúszik..

megoldás felgyorsítják

de ez nagy probláma, nincs szabvány fps-re felvételre…

VÉGE mégse

\*interlaced ma a sugárzás: 1080i-s (i-> interlaced)

azt csinálják, hogy az egyik frame első felét és a másik frame második felét küldik el.-> egyik framen itt a labda másikon már előrébb-> csíkozva jelenik meg.

deinterlacing -> 2 félképből csinák progressivet mert olyan monitoraink vannak.

VÉGE