

Tömörítés, kép ábrázolás

A tömörítés célja: hogy információt kisebb helyen lehessen tárolni (ill. gyorsabban lehessen kommunikációs csatornán átvinni)

A tömörítés lehet:

- **veszteségmentes**
nincs információ veszteség (pl. ZIP, BMP)
- **vesztességes**
az eredeti információhoz képest kevesebb (de a célnak megfelelő mennyiségű) információt tárolunk (pl: JPEG, MPEG)

A tömörítéssel együtt történhet

- **adatvédelem:** a tömörítéshez jó hatásfokú *hibajavító kódolást* használnak, amely u.n. folthibák esetén is lehetővé teszi a javítást.
- **titkosítás:** a tömörített file csak kulcs megadása után fejthető ki.

A tömörítés alapja:

- az eredeti információ redundáns

A tömörítés elve:

- a redundancia csökkentő eljárások alkalmazása

Vesztességmentes tömörítések

Run Length Encoding (RLE, Futam hossz kódolás)

- A fekete-fehér képet, ha sorokra bontjuk, sok egyforma képpont követi egymást. ehhez képest ritka a váltás.
- Az egyforma adatokból álló sorozat helyett a sorozat darabszámát és az elemet továbbítjuk.
- Legegyszerűbb esetben a tömörítés csak egy soron belül történik.

Pl. „A” betű képe esetén:

00000 1 10000000000	5w2b9w
0000 110 1100000000	4w2bw2b7w
000 11000 110000000	3w2b3w2b6w
00 11111111 1000000	2w9b5w
0 1100000000 110000	w2b7w2b4w
110000000000 11000	2b9w2b3w

Bonyolultabb algoritmus azt is kihasználja, hogy az egymás utáni sorok hasonlóak.

ZIP

Fő jellemzői

- Egy vagy több file *egyetlen file-ba becsomagolva*
- *A file-ok a ZIP file-ban szétválasztva vannak benne*, egyenként törölhetők, módosíthatók, vagy új file-ok tehetők be.
- A program az egyes file-okat *különböző módszerekkel tömöríti*
- A tömörítéssel együtt *titkosítás is történhet* (feltörhető)
- *A hiba hatása korlátozódik* arra az eredeti file-ra, melynek becsomagolt formájában a hiba előfordul.

Deflating

- A ZIP-ben használt egyik tömörítés, a Deflating (= gáz leeresztése)
- A Deflating a Huffman kódolást és az LZ77 eljárást kombinálja

Az LZ77 eljárás:

- ismétlődő részeket keres
- ezeket egy szótárban kigyűjti
- sorszámot rendelünk hozzájuk.
- A tömörített file két részből áll: a szótárból és egy sorszám listából

Pl. tömörítendő szöveg:

„Ha *buta vagy*, nem tudod, hogy *_buta vagy*, mert *buta vagy*.”

A szótár és szöveg:

\$1=” *buta vagy*”

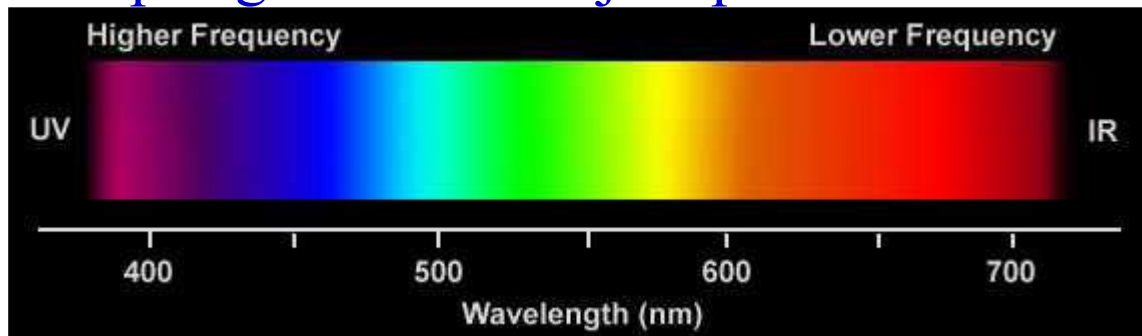
“Ha *\$1*, nem tudod, hogy *\$1*, mert *\$1*.”

A sorszám listában *a sorszámok kódolására Huffman kódot alkalmaznak*, azaz annál rövidebb a sorszám kódja, minél gyakrabban fordul elő a szótár elem a file-ban.

A színlátásról

A látható fény hullámhossza ~ 400...700nm

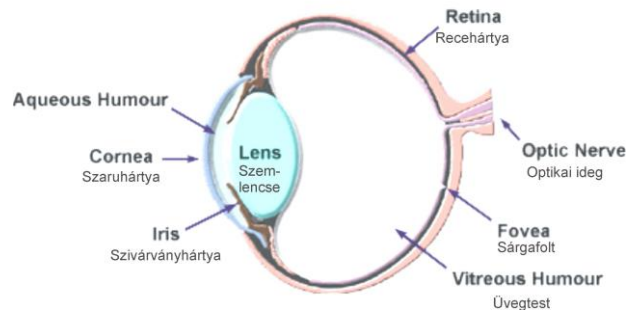
A nap sugárzásának teljes spektruma:



Az egyetlen hullámhosszúságú sugárzásból álló fény színét az adott hullámhossz feletti függőleges vonal színe mutatja.

Az emberi szem olyan színeket is lát, amely ezek között nem szerepel, ilyen a bíbor.

A szem

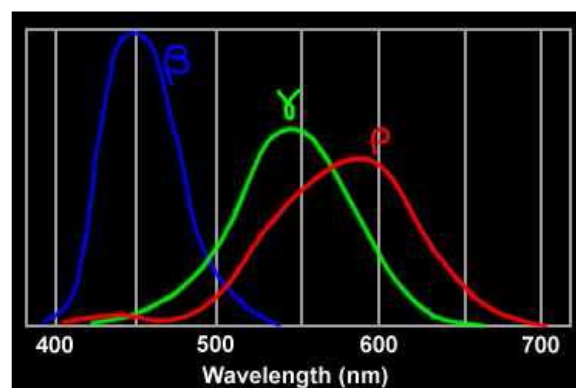


A retinában kétféle fényérzékelő sejt van.

Pálcikák: egyenletesen elosztva helyezkednek el. Elsősorban az *éjszakai és a periferiális látásban* van szerepük. Csak egyféle pálcika van, a pálcikák a *színlátásban nem vesznek részt*.

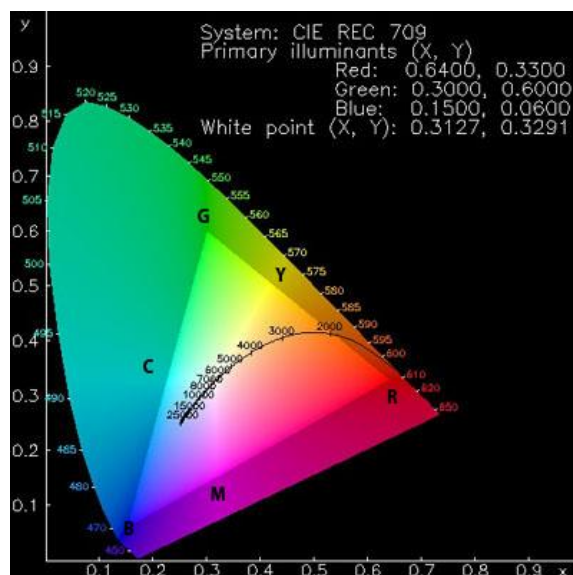
Csapok: háromféle csap van, melyek a **vörös (Red)**, **zöld (Green)** és **kék (Blue)** színre érzékenyek.

A csapok színérzékenysége:



- Az emberi szem a 3 féle csapját megfelelő hullámhossz eloszlású fénnel ingerelve tetszőleges tiszta színek megfelelő szint érzékel.
- Ezért a 3 alapszínből a látható színek *nagyobb részét* ki lehet keverni (additív színkeverés).
- A fényképezőgépek szenzora is ezt a 3 komponenst érzékeli, a színeket az érzékelő pontok feletti színszűrővel különböztetik meg
- A monitorok színvisszaadása az additív színkeverésen alapul

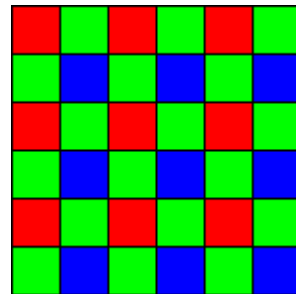
A CIE patkódiagram az összes látható színt mutatja. A háromszögön belül az RGB-vel kikeverhető színek vannak.



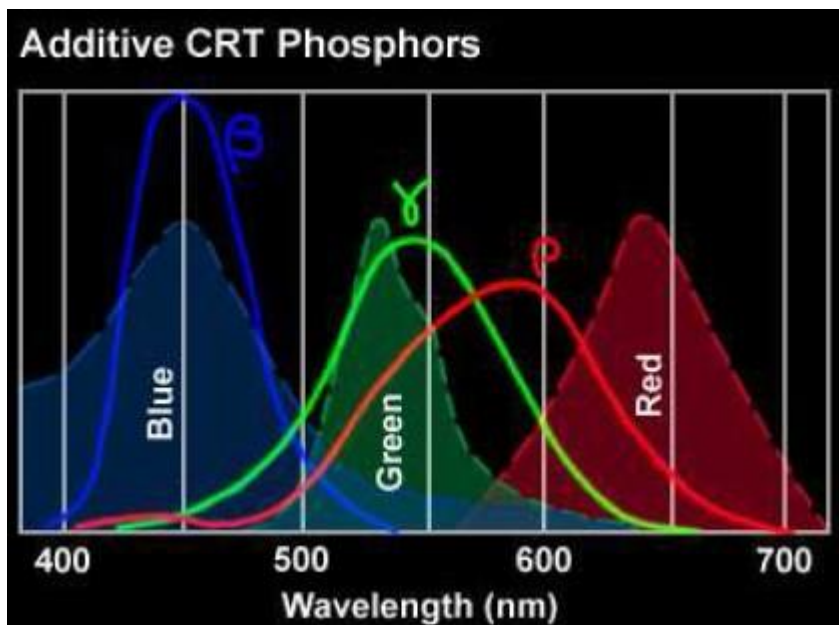
Fényképezőgép szenzor



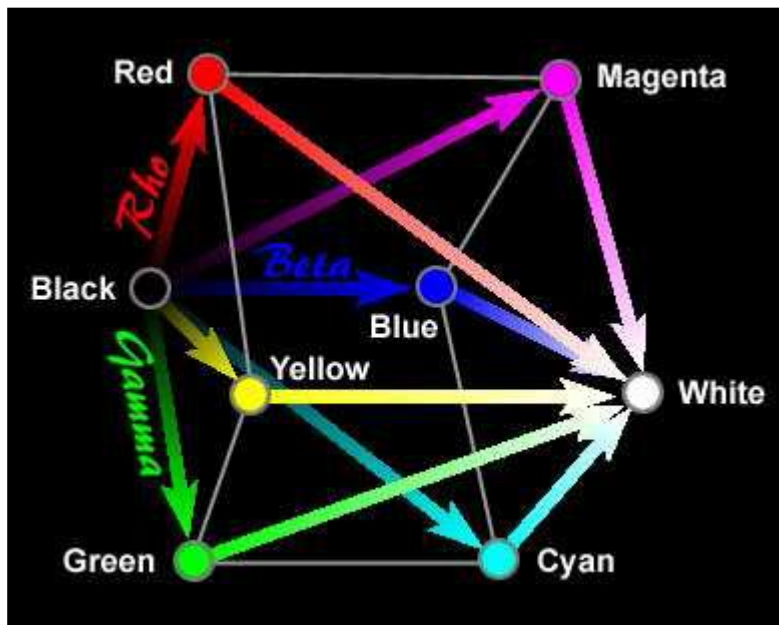
Színszűrők a
pixelek felett



A színes képernyőn a kép RGB
komponensekből tevődik össze:



A szín a komponensek arányától függ. Az úgynevezett szín kocka most az elsődleges és másodlagos színeket mutatja a telített színtől a fehérig változóan:



Látszik, hogy például a sárga szín a piros és zöld összegeként állítható elő, és minél több kéket keverünk hozzá, annál fehéresebb (precízebben: csökken a szín telítettsége).

A 24 bites színmélység esetén egy képpont színét és fényességét 3x8 biten adjuk meg, ahol a 3 byte az RGB komponensek intenzitása.

A PC-ben a maximális intenzitású alapszínek 24 bites kódja:

Kék	FF	00	00
Zöld	00	FF	00
Piros	00	00	FF

Ebből tehető össze a többi szín, például:

Cián	FF	FF	00
Magenta	FF	00	FF
Sárga	00	FF	FF
Fehér	FF	FF	FF
Szürke	80	80	80

Képfeldolgozás

Raszter grafikus képfeldolgozás

- a képet raszterpontokra osztjuk
- az egyes pontok jellemzőit (világosság, szín) adjuk meg

A legismertebb tömörítetlen raszter grafikus ábrázolási forma a *Windows Bitmap* (fájlnév.BMP).

A fájlban tárolt információk:

- Információs fejlécek (pl. szélesség, magasság képpontokban megadva, színmélység bitben)
- Színpaletta (a képben szereplő színek kódjai)
- Bittérkép adatok (maga a tényleges kép, pixelenként a színkódokkal)

Például egy 16 x 6 pixeles kétszínű kép és bitmapja binárisan és hexadecimálisan:

00000 1 10000000000	06,00
0000 110 1100000000	0d,80
000 11000 110000000	18,c0
00 11111111 11000000	3f,e0
0 1100000000 110000	60,30
110000000000 11000	c0,18

A bitmap képek nagyon nagyok lehetnek. Egy 800 x 600 pontos 24 bites bitmap mérete:
 $800 \times 600 \times 3 \text{ byte} = 1\,440\,000 \text{ byte}$.

A bitmap kép mérete tömörítéssel csak kismértékben csökkenthető. (van RLE tömörítést alkalmazó BMP formátum).

Képtömörítés, JPEG

Csak speciális tulajdonságú képek pl. vonalas ábra bitmapja tömöríthető jól ZIP-pel. Nem ilyen jó a helyzet fénykép tömörítésnél. Ezért fényképek tömörítésére más elven működő eljárást dolgoztak ki. Ilyen a JPEG.

- A JPEG *veszteséges tömörítés*, azaz nem állítható vissza pontosan az eredeti file, de a képminőség romlása nagyon kicsi lehet.
- A JPEG nagy előnye, hogy *különböző minőségű tömörítést tesz lehetővé*, és ha a minőségből engedünk, a tömörítés egyre nagyobb lehet.



Tömörítetlen 423 x 408 x 24bit BMP: 519030 byte
byte



JPG 10-es tömörítés: 58908 byte



JPG 90-es tömörítés: 8492

A képtömörítésnél *a szem tulajdonságait veszik figyelembe*

Olyan információkat hagynak ki a képből, amelyeket a szemünk nehezen vesz észre.

Ilyen tulajdonságok pl.:

- *a szem fényesség felbontása sokkal jobb, mint szín felbontása.* Ezért szétválasztják a szín és fényesség információt és a *színinformáció felbontását csökkentik* (az eredetnél kevesebb biten ábrázolják)
- a szem ugyan nagyon érzékeny a nagyobb területek fényesség változására, de *nem észleli pontosan a gyorsan változó kisebb területek fényességét.* Ezért *utóbbiakat kevésbé pontosan ábrázolják.*
- A fentiek megvalósításához viszonylag bonyolult matematikai apparátus szükséges, ezt nem tárgyaljuk. (Közben az egyszerűbb RLE és Huffman kódlást is alkalmazzák.)
- A tömörített kép közvetlenül nem ábrázolható, ezért a megjelenítés előtt vissza kell alakítani.

Mozgó kép tömörítés

Elve:

- A mozgóképek „első” kockáját tömörítik.
- A további kockáknál kihasználják, hogy azok *az előző kockához hasonlítanak*, és csak *a különbséget tárolják*, tömörítve.
- Bizonyos számú kocka után, vagy ha vágás van a filmben, újra a teljes képtartalmat tárolják tömörítve.