「画像」の仕組み

教科書1章・2章

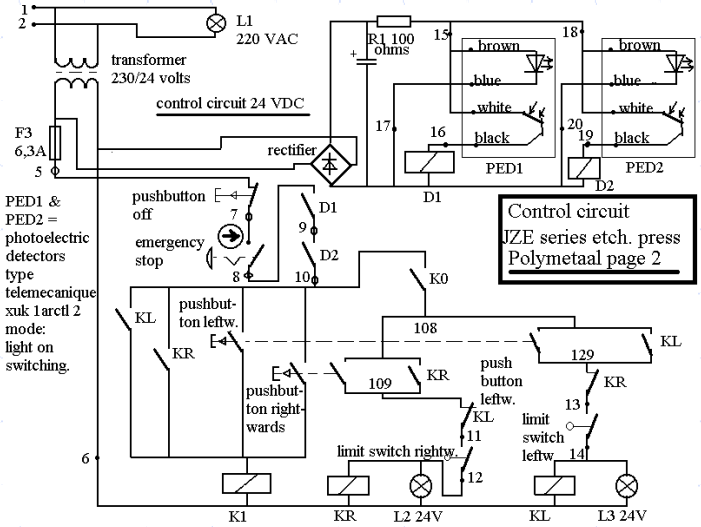
画像とは・・

写真

1. Introduction

Projective reconstruction
2D-images is a central p
Usually a finite set of feat
are characteristic for the
are identified in the differ
tion is computed. By the
is possible to use also 3D
tures in a reliable way. I
spondences between the
for the curves beforehand
gorithm. This is a difficul
the correspondence probl

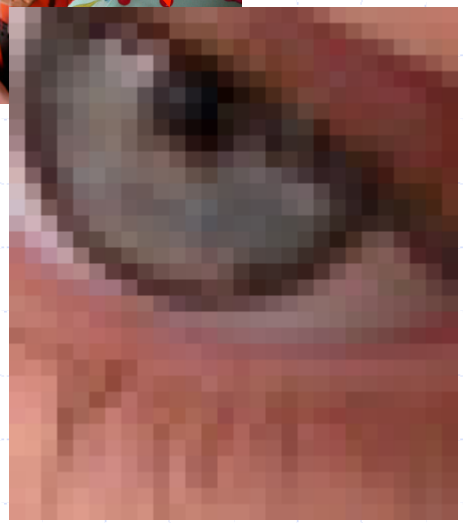
文書



図面

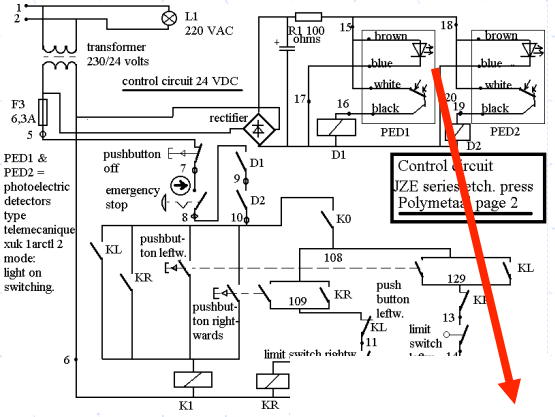
- 写真，図面等を電子化したもの

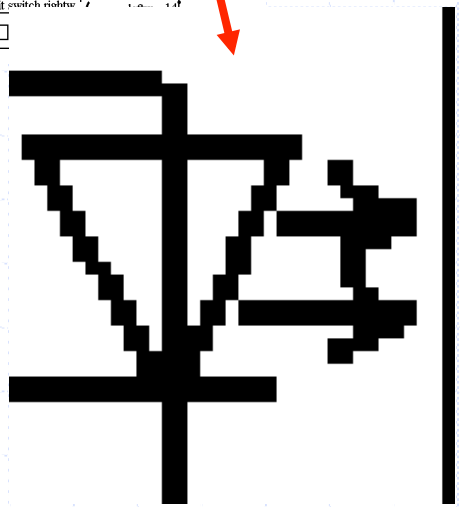
拡大していくと・・・



1. **Introduction**

Projecti e reconstruction
2D-image s is a central p
Usually a finite set of feat
are charac eristic for the
are identifie d in the differ
tion is comp uted. By the
is possible to use also 3D
tures in a reli ble way. I
spondences bet veen the
for the curves be forehand
gorithm. This is a difficul
the correspondenc probl

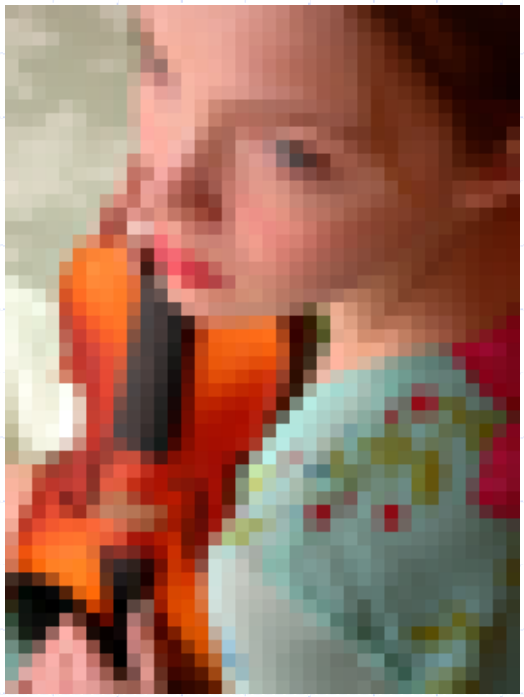




- 画像は点の集合で出来ている
  - 1つ1つの点を画素と呼ぶ

解像度とは？

解像度が高い

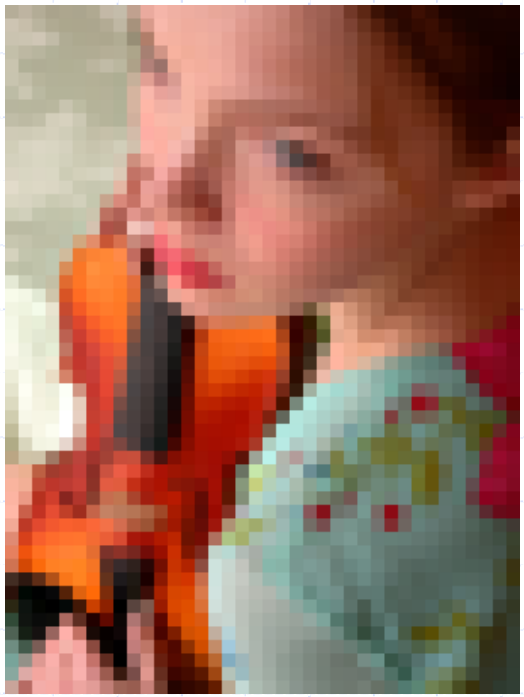


解像度が低い

- 解像度が高い＝綺麗な（高精細な）画像
  - それだけ，メモリ容量を多く必要とする

解像度の単位（印刷やディスプレイ表示）

1インチ＝100画素
100dpi



1インチ＝10画素
10dpi

1インチ＝25.4mm

- dpi (dot per inch) が良く使われる

画像の記憶容量

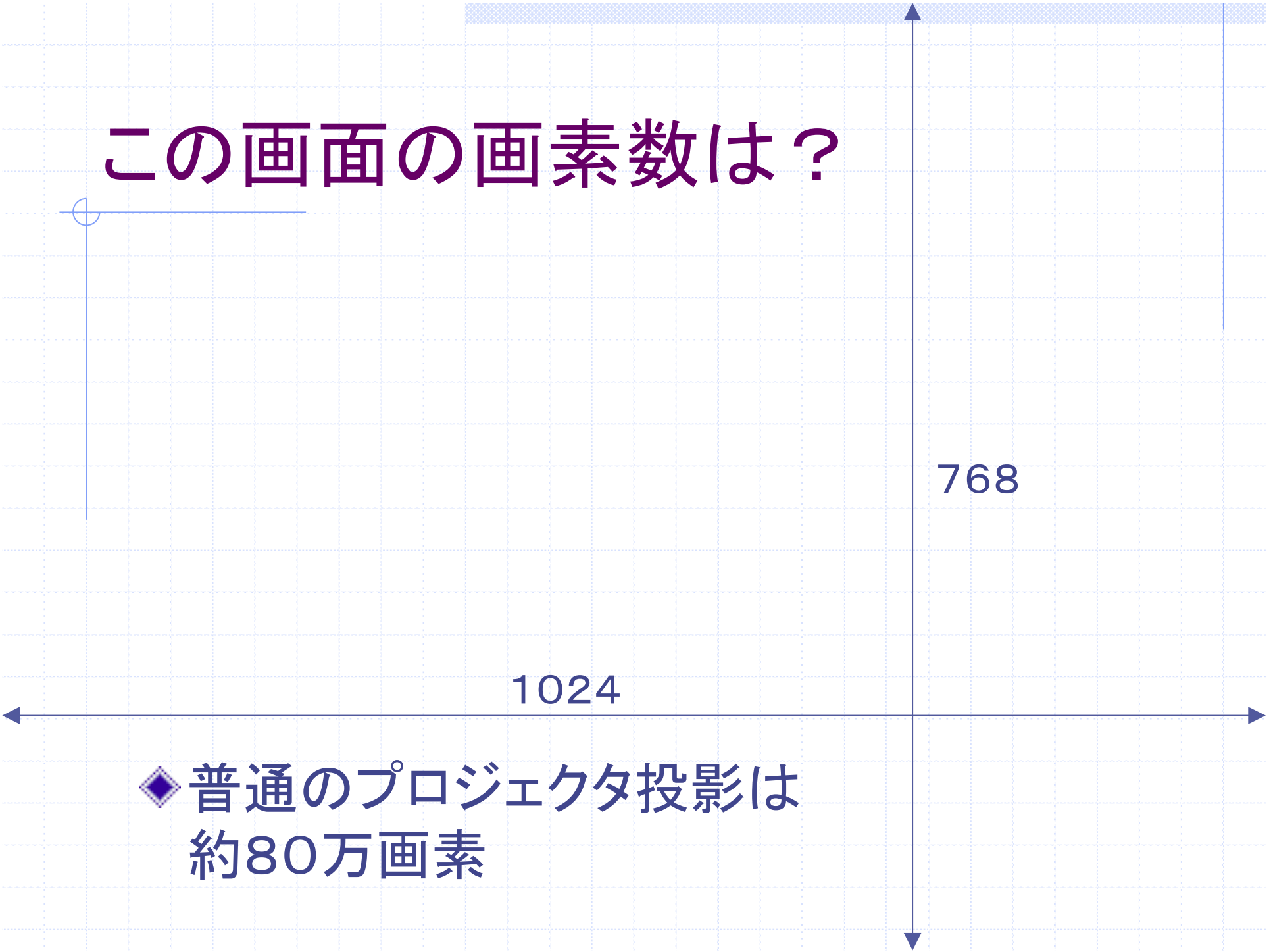
縦　　横　　色
1000 × 1000 × 3
= 3,000,000
（3MB）

「メガピクセル」≒100万画素

≒縦・横 各 1000画素

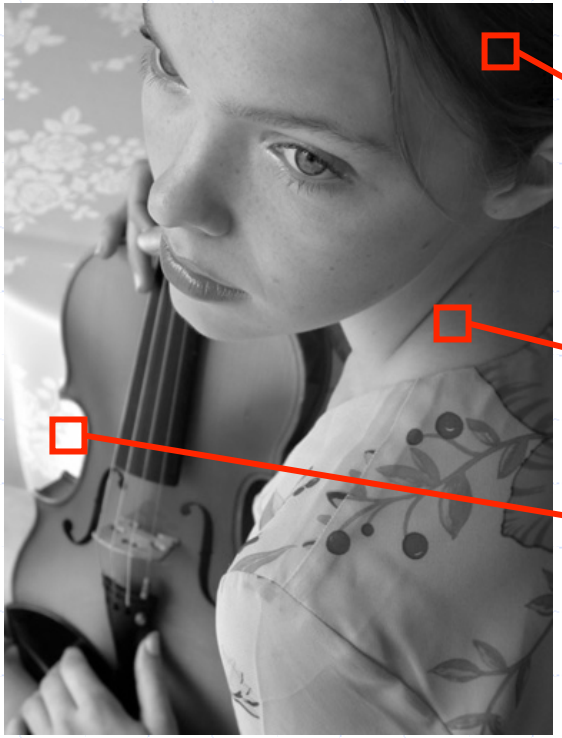
- 画像は「生」のままだと，大容量データ

この画面の画素数は？



- 普通のプロジェクタ投影は
  約80万画素

色・明るさの仕組み



- 1画素ごとの明るさを数値で表現
  - 明るいほど大きな数値（真っ黒は0）

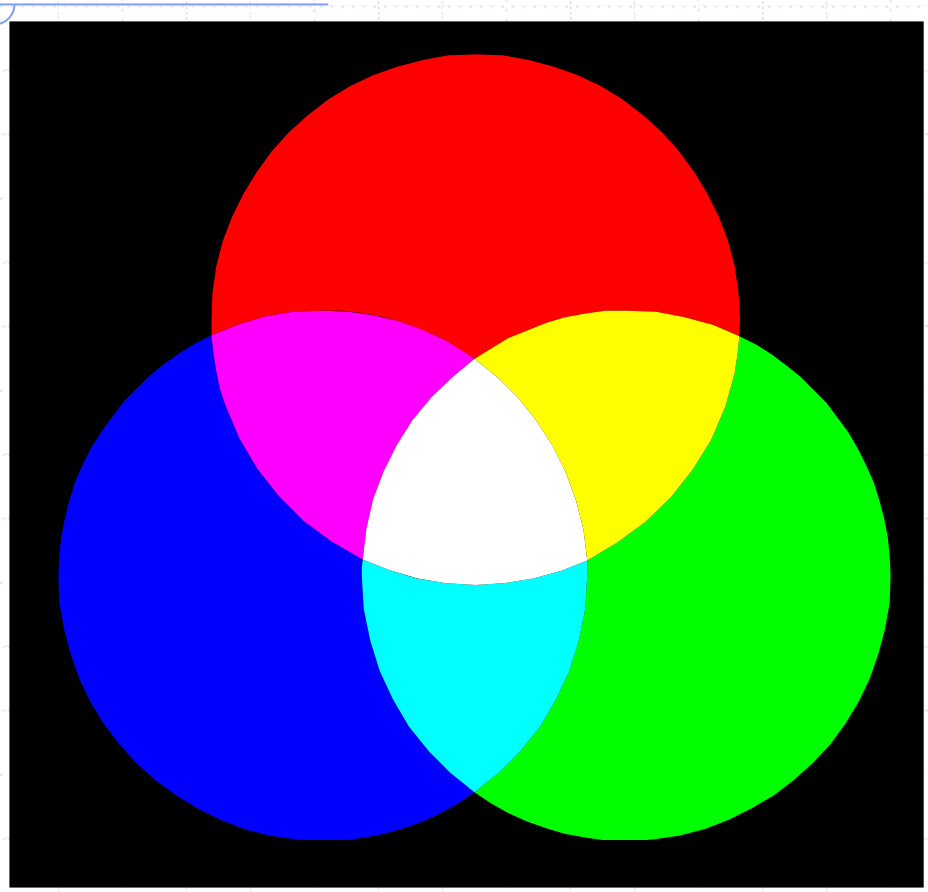
階調表現

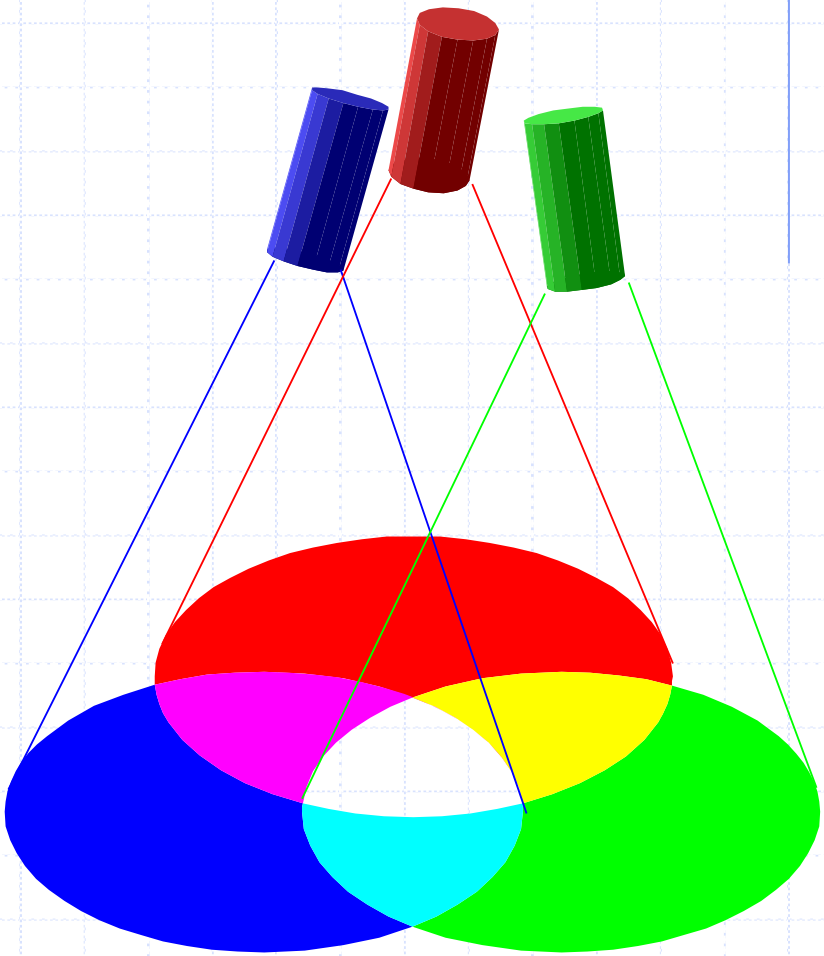



	2階調 $2=2^1$	4階調 $4=2^2$	16階調 $16=2^4$	256階調 $256=2^8$
2進数	0 1	00 01 10 11	0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111	00000000~ 11111111

- 高階調ほどメモリ容量を要する
  - 256階調は，2階調の画像8枚分
  - 普通は多くて，256階調まで

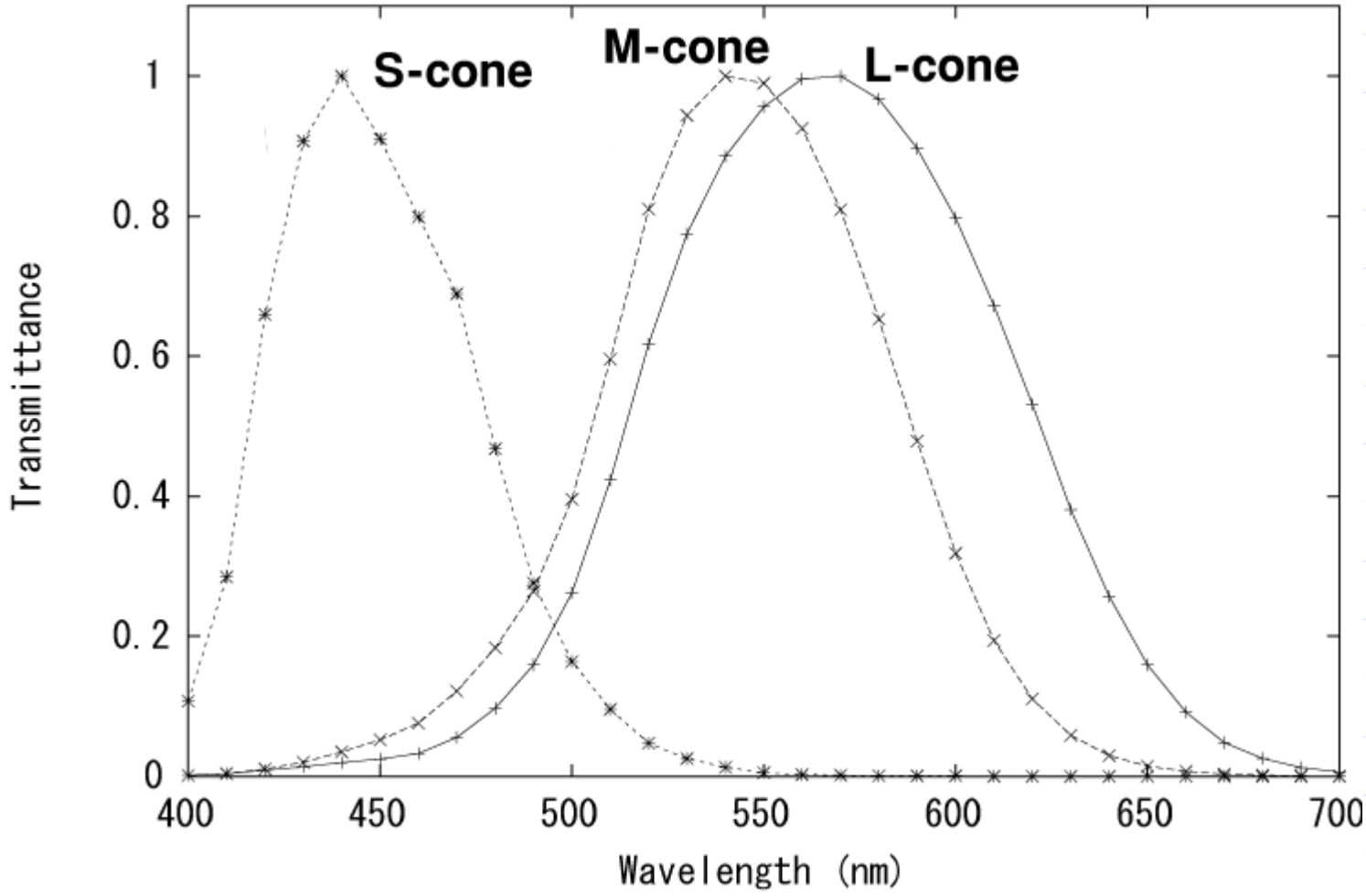
三原色について





- どんな色でも，赤・青・緑の混合で表される

錐体の分光感度特性



人間は3種類の視細胞を持ち，それぞれ感度分布が異なる

カラー画像

カラー

赤

緑

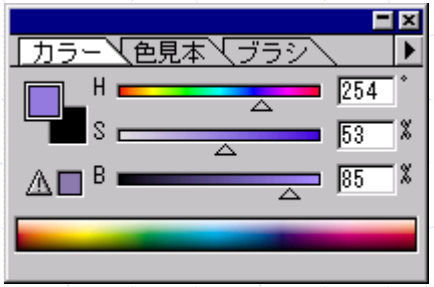
青

- カラー画像は，赤・緑・青の画像の合成

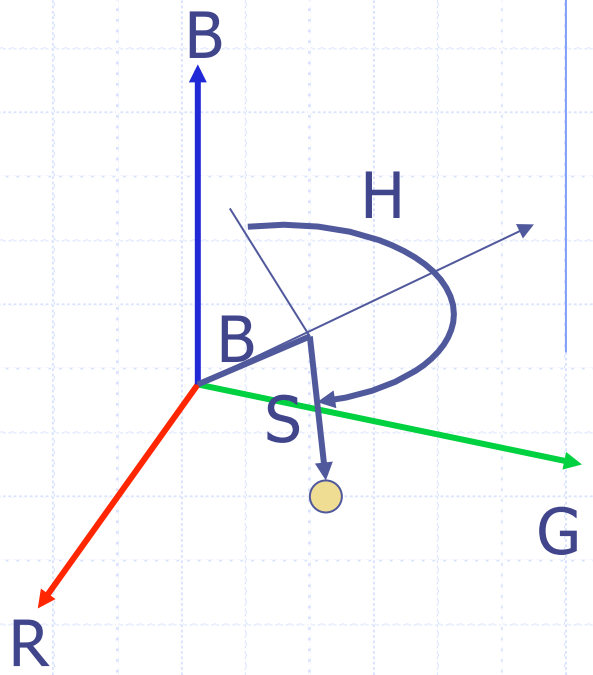
色の表し方

PhotoShop の例

RGB（三原色）表現



HSB表現
H: 色相（色合い）
S: 彩度（色の濃さ）
B: 明度（明るさ）



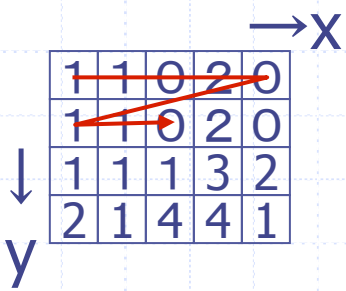
- 3次元空間の1点をどのようにして指定するか？
  - RGB と HSB の間は，座標変換で表現可能

画像をプログラムで扱うには

- 輝度は普通，8bit で扱う．正の値のみ．
  - unsigned char 型を用いる．
  - カラー画像は，赤・青・緑の3つが必要．
  - 画素は縦・横に並んでいるので，2次元配列．
- C言語での例
  - 幅が640画素，高さが480画素のカラー画像
    unsigned char image[480][640][3];

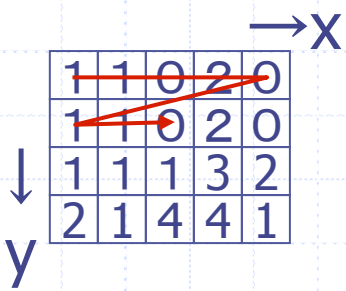
```
image[y][x][c] = 100;
```

画像の配列での扱い(1)



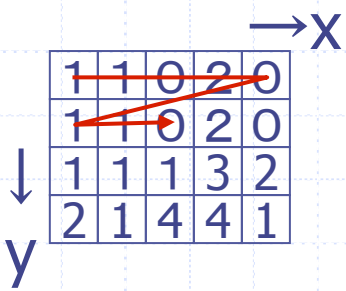
- 画像は，普通，左上の角を原点とする
  - 画素値は，文字が書かれる順に格納される．
    つまり，右横の値が次に来る．
  - よってCの配列では，image[y][x] の順となる．
    右上の例では，
    unsigned char image[4][5]; となる．
    メモリには，image[0][0], image[0][1], .. という順に格納される．

画像の配列での扱い(2)



- 一次元配列を使うことも多い.
  - 右上の例では, 画素数が20個なので, 単に
    unsigned char image[20];
    のようにメモリ確保する.
  - 画素値にアクセスするときは,
    image[y * 5 + x] = 20;
    のように画像の幅を y に掛けてアクセスする.

画像の配列での扱い(3)



- カラー画像の場合
  - unsigned char image[4][5][3];
    image[y][x][c] = 20;
    のような3次元配列による方法(c=0, 1, 2)
  - unsigned char image[60];
    image[(y * 5 + x) * 3 + c] = 20;
    のようなアクセス方法もある.
  - #define でマクロ定義をしておくと良い.