

情報基礎 画像

1

イメージ形式



ビットマップ(bitmap)
or
ラスタ(raster)形式



ベクタ(vector)形式

2

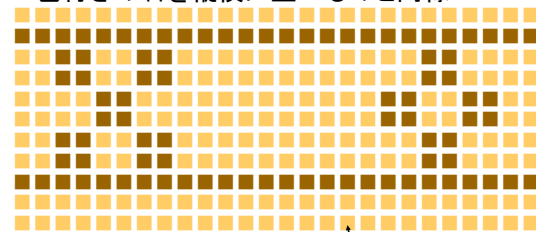
ペイント(ビットマップ/ラスタ形式)と ドロー(ベクタ形式)

- ペイント(Paint)型ツール
 - 絵筆で描いていくようなツール
 - 縦横の画素数を決めたキャンバスに描く
 - 画素のひとつひとつに対する色情報
 - ビットマップ(bitmap)方式、ラスタ(raster)方式ともいう
- ドロー(draw)型ツール
 - さまざまな図形(オブジェクト)を配置して描く
 - 図形を配置する座標情報で描く
 - ベクタ(vector)方式ともいう

3

ビットマップ/ラスタ形式

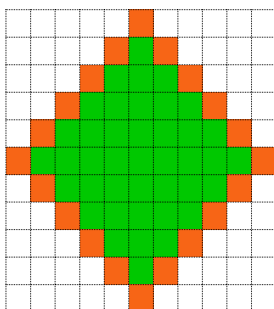
- 色付きの石を縦横に並べるのと同様



ピクセル(pixel)、画素

4

ペイント(ビットマップ/ラスタ形式)



- 斜めの線は階段状
- 縮小とは画素を間引くこと
- 拡大とは1画素の色情報を4画素に割り当てること(200%拡大の場合)
- 拡大・縮小で画質を維持することはできない
- 目的に合ったキャンバスの大きさ(縦横の画素数)を決めることが重要
- 保存するのはすべての画素の色情報

5

代表的な ビットマップ/ラスタ画像形式

形式	拡張子	圧縮	色数	その他の特徴
TIFF	.tiff .tif	可逆	フルカラー	CMYK対応
BMP	.bmp	しない	フルカラー	
PNG	.png	可逆	フルカラー	RGBA対応
JPEG	.jpeg .jpg	非可逆	フルカラー	CMYK対応
GIF	.gif	可逆	256	動画形式あり

6

ソフトウェアに依存する ビットマップ/ラスタ画像

ソフトウェア例	拡張子
GIMP	.xcf
PhotoShop	.psd

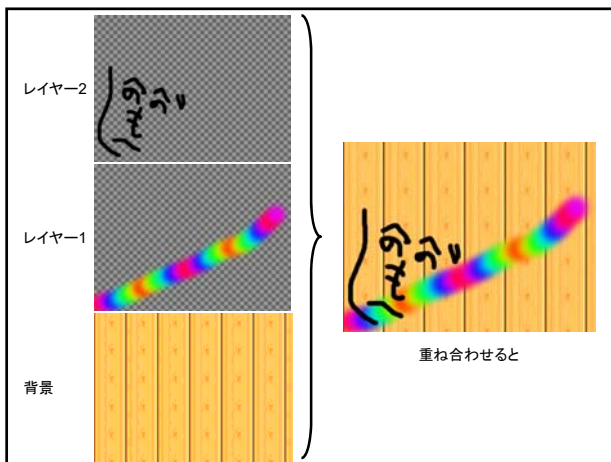
- レイヤー構造をもつ
- レイヤーを利用した画像は、レイヤー情報を保持したまま保存すること
 - レイヤーあり ⇒ レイヤーなし、の変換は簡単

7

レイヤー

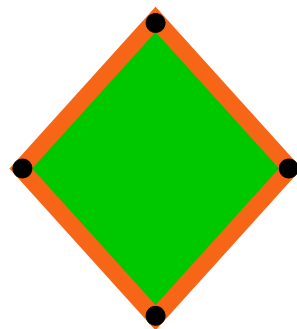
- 特にペイント型ツールでの描画に柔軟性を持たせる工夫のひとつ
- 基本的な使い方
 - 背景レイヤー(画用紙)に背景を描く
 - 上に透明のレイヤーを重ね、見せたり除いたりしたい絵を描く

8



9

ドロー(ベクタ形式)



- 拡大・縮小すると座標値を計算しなおして再描画
- 画像全体の拡大・縮小も、個々の図形(オブジェクト)の拡大・縮小もできる
- 保存するのは、図形の頂点座標、辺の色、塗りつぶしの色、描画する順番など

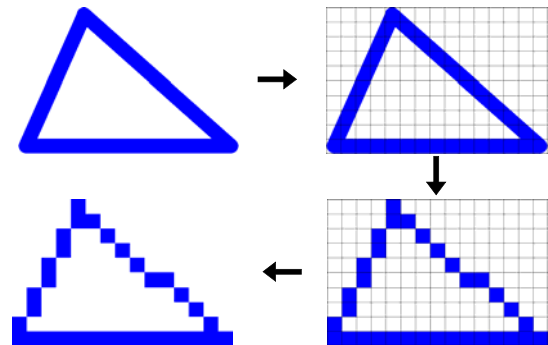
10

ベクタ画像形式

名称またはソフトウェアの例	拡張子	特徴
StarSuite図形描画	.sxd	このソフトウェアの独自形式
Windows Metafile (WMF)	.wmf	Microsoft Wordのクリップアートなど
Scalable Vector Graphics (SVG)	.svg	Webブラウザで表示可能
Encapsulated PostScript (EPS)	.eps	PostScriptで記述された画像データ

11

ディスプレイ表示 ベクタもラスタ化



12

ディスプレイの規格と解像度

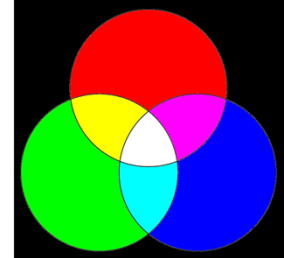
- ディスプレイが描画する縦横の画素数
- ディスプレイの大きさは直接関係ない

規格	解像度
VGA	640×480
SVGA	800×600
XGA	1024×768
SXGA	1280×1024
UXGA	1600×1200

13

加色三原色

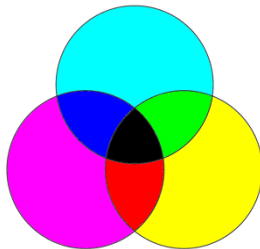
- 光の三原色ともいう
- 自ら光を発する器具
- ディスプレイ
- Red 赤
- Green 緑
- Blue 青
- RGB



14

減色三原色

- 色の三原色ともいう
- 光を吸収して色を作る
- 吸収されずに残った色を見ている
- カラープリンタ、絵の具
- 減算混色
- Cyan 水色
- Magenta 紫
- Yellow 黄
- CMY



15

CMY と CMYK

- CMYの減算混色だけでグレーを再現するのは
 - 現実には容易ではない
 - CMYのインクやトナーを浪費する
- CMYに黒(K)を加えたCMYK

16

2進数

- 0と1の2つで数を表わす
- $2^1=2$ [0, 1]
- $2^2=4$ [00, 01, 10, 11]
- $2^3=8$ [000, 001, 010, 011, 100, 101, 110, 111]
- ...
- $2^8=256$ [00000000, 00000001, ...]

17

表現できる色数

- コンピュータグラフィックスでは通常RGB方式
- RGBそれぞれの on off だけなら $2^3 = 8$ 色
- RGBそれぞれ256 ($= 2^8$)段階、 $256^3 \doteq 1678$ 万色
 - 人間の目で識別可能な限度を越えるため、フルカラーともいう
 - 各色 $256 = 2^8$ 、すなわち8bit用いて表現するため、24bitカラーともいう
 - 0 (発光しない) から 255 (最強の発光) で表現

18

RGB方式による表現例

	R		G		B		結果
247	■	101	■	22	■	⇒	■
187	■	28	■	222	■	⇒	■
29	■	93	■	158	■	⇒	■
190	■	236	■	22	■	⇒	■
239	■	141	■	137	■	⇒	■

19

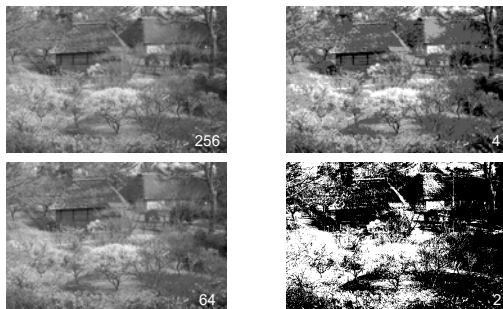
グレースケール(モノクロ)



- 黒から白までの濃淡だけで表現する
- 墨絵
- カラー表現同様、用いるbit数で階調数がきまる
- 8bitグレースケールでは
 - 256 ($=2^8$) 階調
 - 0 (真っ黒) から 255 (真っ白) で表現

20

階調数による違い



21

RGB方式でのグレースケール

- RGBとも同じ数値で表現する

	R		G		B		結果
255	■	255	■	255	■	⇒	□
200	■	200	■	200	■	⇒	■
100	■	100	■	100	■	⇒	■
0	■	0	■	0	■	⇒	■

22

Webの場合(RGB)

- #rrggbb
- 16進数
 - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f

#000000	black	#ffffff	white
#ff0000	red	#00ff00	lime
#0000ff	blue		

23