



<http://www.takagi.inf.uec.ac.jp/mmip/>

マルチメディア処理

Multimedia Information Processing

第4回 : 画像・動画のデジタル表現



高木一幸



講義概要

- ・マルチメディアデータ（文書、音声、画像、映像など）の表現方法と処理技術について、基礎的な内容を紹介・解説する。
- ・授業時間中および宿題として、計算機を使った演習を行うことにより実際のデータの扱い方を学び、理解を深める。

第1回：授業の概要説明、
人間の感覚とマルチメディア処理 (4/11)

高木

第2回：文字・テキストの表現と処理 (4/18)

第3回：音声のデジタル表現と処理 (4/25)

第4回：画像・映像のデジタル表現 (5/2)

第5回：マルチメディアデータの符号化とファイル形式 (5/9)

第6回：2次元図形の表現と描画 (5/16)

第7回：画像処理(1)画素ごとの濃淡変換 (5/23)

第8回：画像処理(2)空間フィルタリング (5/30)

廣田

第9回：カメラと写真撮影 (6/13)

第10回：3次元コンピュータグラフィックス (6/20)

(1)形状表現と透視投影

第11回：3次元コンピュータグラフィックス (6/27)

(2)照明効果とシェーディング

第12回：アニメーションと映像制作 (7/4)

第13回：シミュレーションと可視化 (7/11)

第14回：グラフィックパイプラインとシェーダ (7/18)

第15回：マルチメディアの応用例 (7/25)



画像・動画のデジタル表現



1. 画像

1. 標本化、量子化、解像度

2. 画像の種類

1. ウェブで使用するビットマップ形式：GIF, JPG, PNG

2. ウェブで使用するベクター形式：SVG, PDF

3. ウェブで使わない画像形式：AI, PSD, TIFF, BMP, EPS, PICT

2. 色

3. 音楽、音声

4. 動画

5. コンテナフォーマット



画像の種類

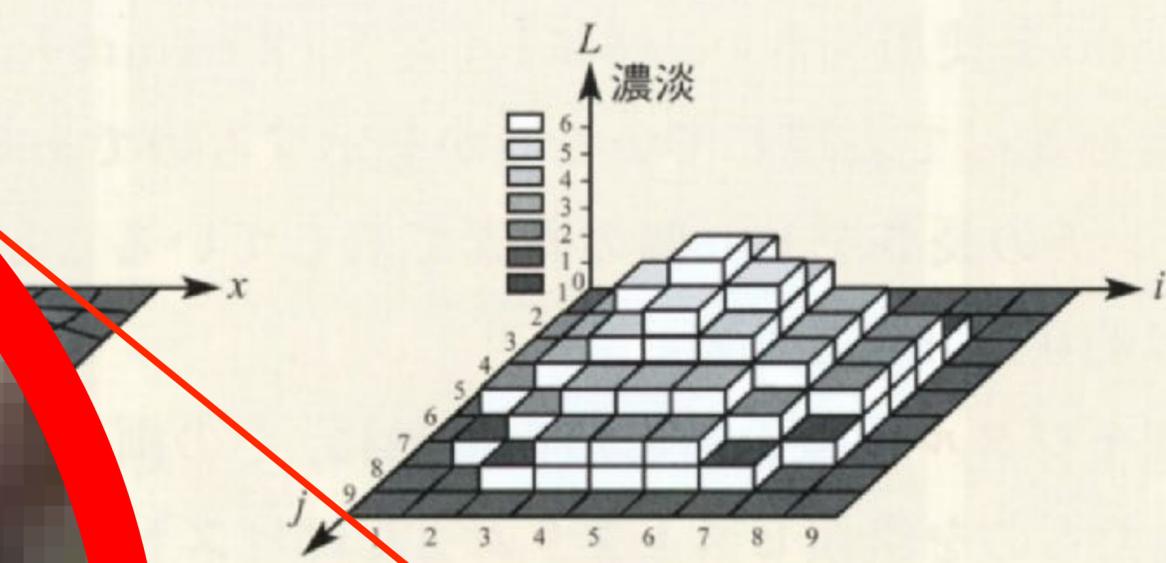
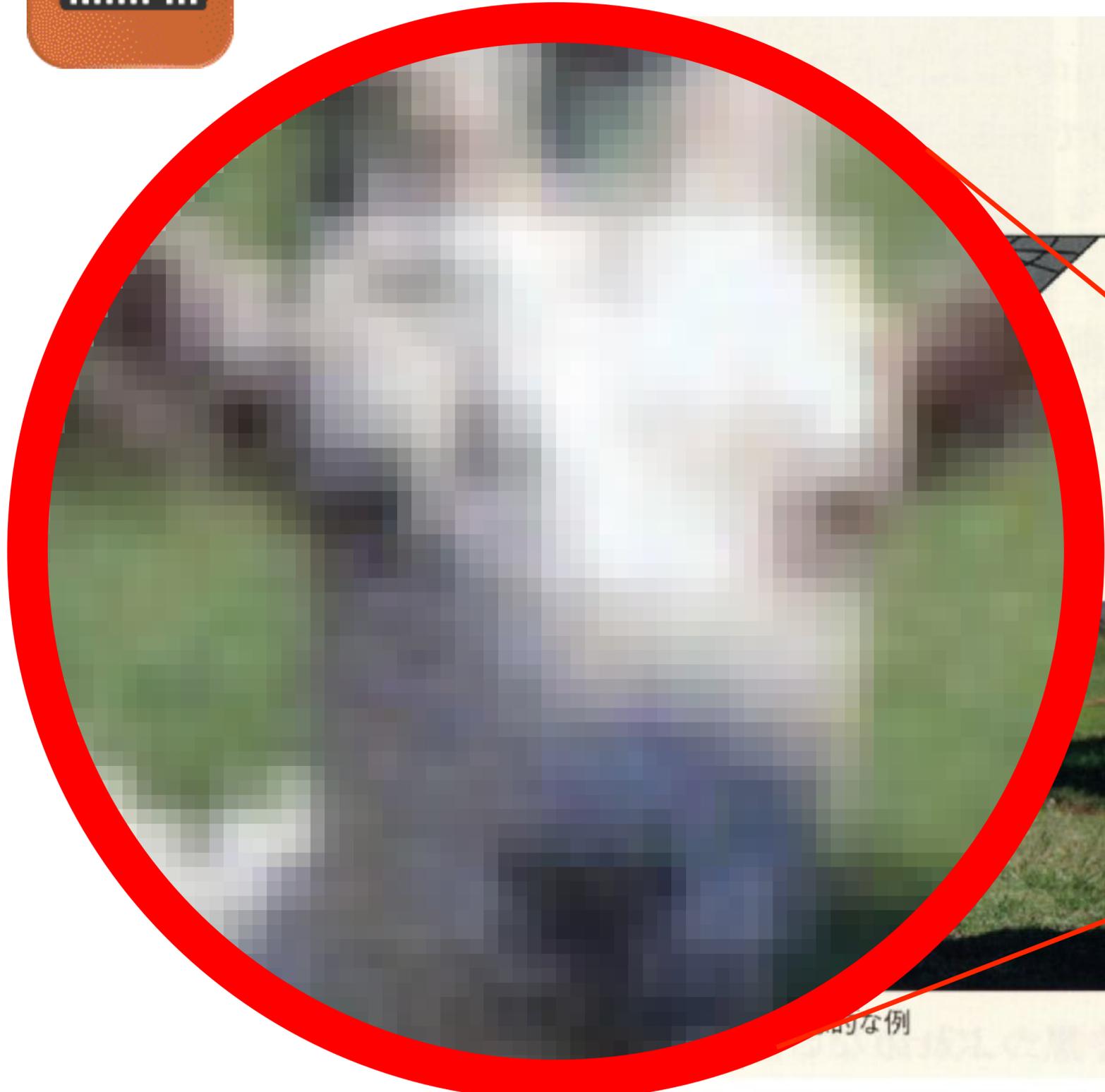
ビットマップ形式

画像を方眼紙のように細かく区切り、色の点（ピクセル）の集まりで表現する形式。ピクセルの密度の細かさを解像度といい、単位はppiやdpi。ウェブ画像では一般的に72ppi (dpi)。拡大すると（点が大きくなり）画像が粗くなる。これを「解像度に依存している」という。

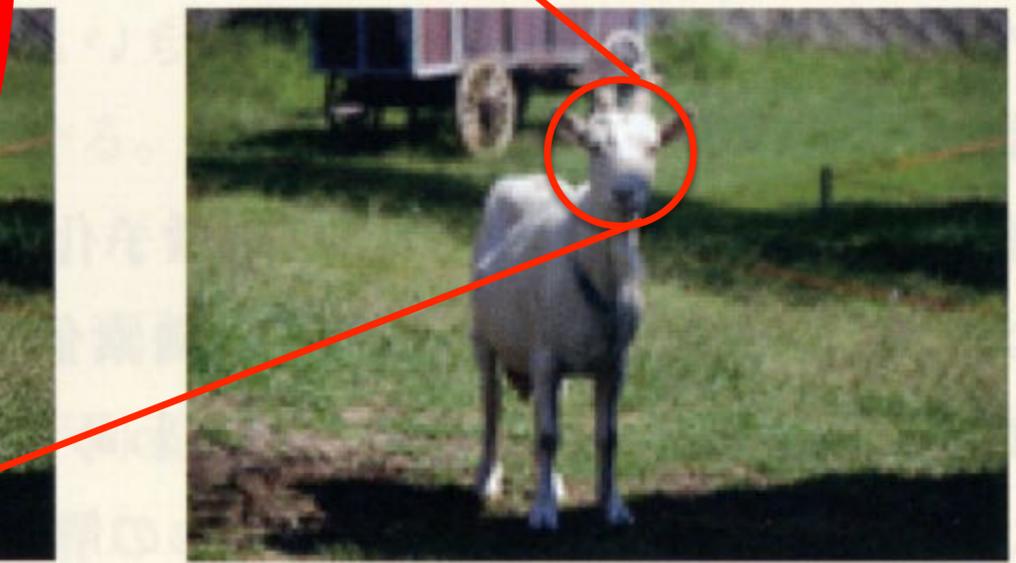
ベクター形式

数式で描くグラフのように拡大しても画像が粗くならない保存形式。ベクトル画像、ベクターイメージ、ベクタグラフィックス、ベクタデータなどともいわれる。点と点を線で結ぶことによって画像を表現するため、綺麗な表現が描ける。Illustratorなどのドローツールで描くことができる。

アナログ画像とデジタル画像



[b] デジタル画像の濃淡分布 $L(i,j)$
9×9画素に標本化し0~6の7レベルに量子化した場合



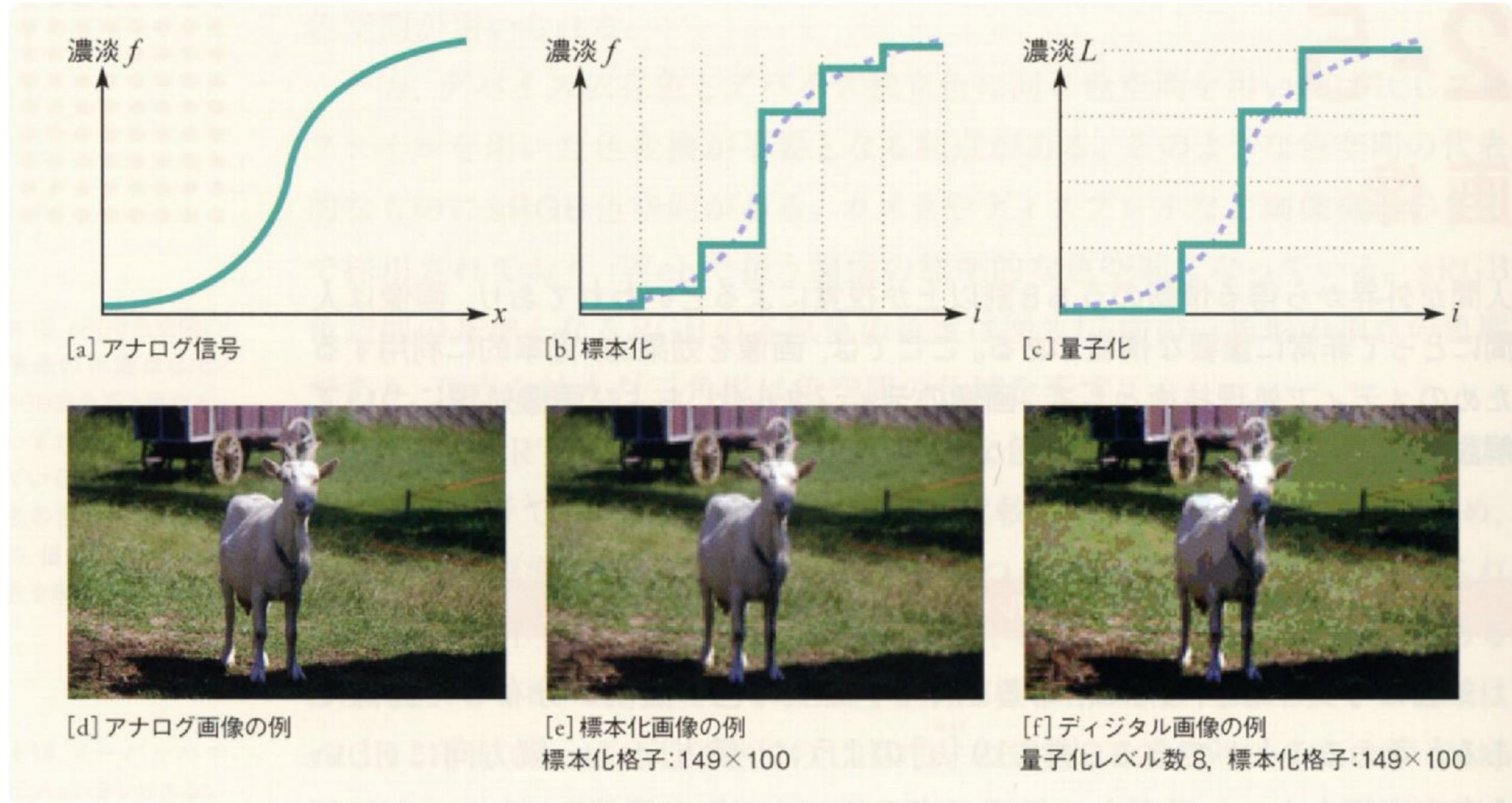
[d] デジタル画像の概念的な例

©実践マルチメディア[改訂新版]画像情報教育振興協会

■図2.19——アナログ画像とデジタル画像

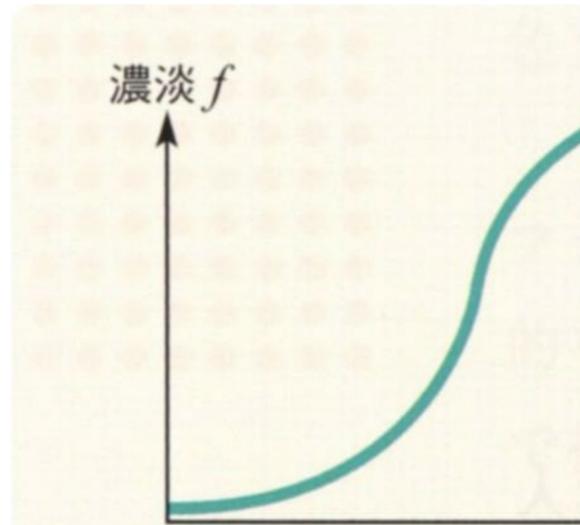


画像における標本化と量子化

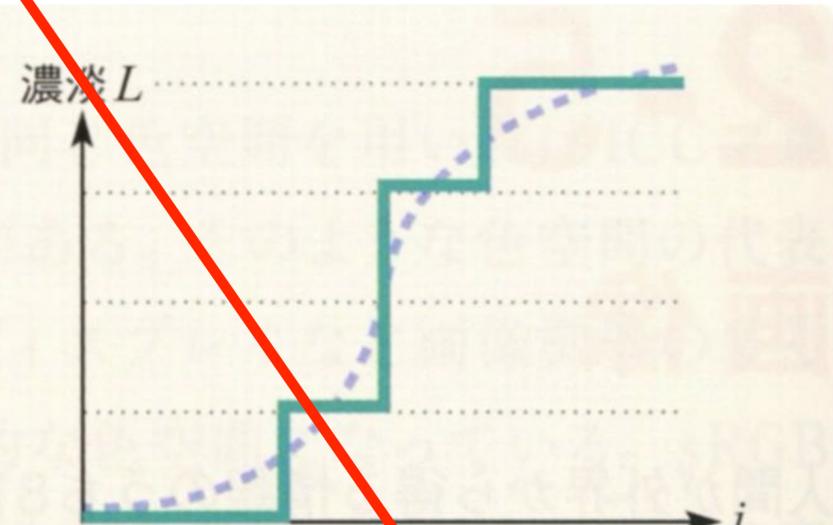


■ 図 2.20 — 画像における標本化と量子化

画像における標本化と量子化



[a] アナログ信号



[c] 量子化



[f] デジタル画像の例
量子化レベル数 8, 標本化格子: 149×100

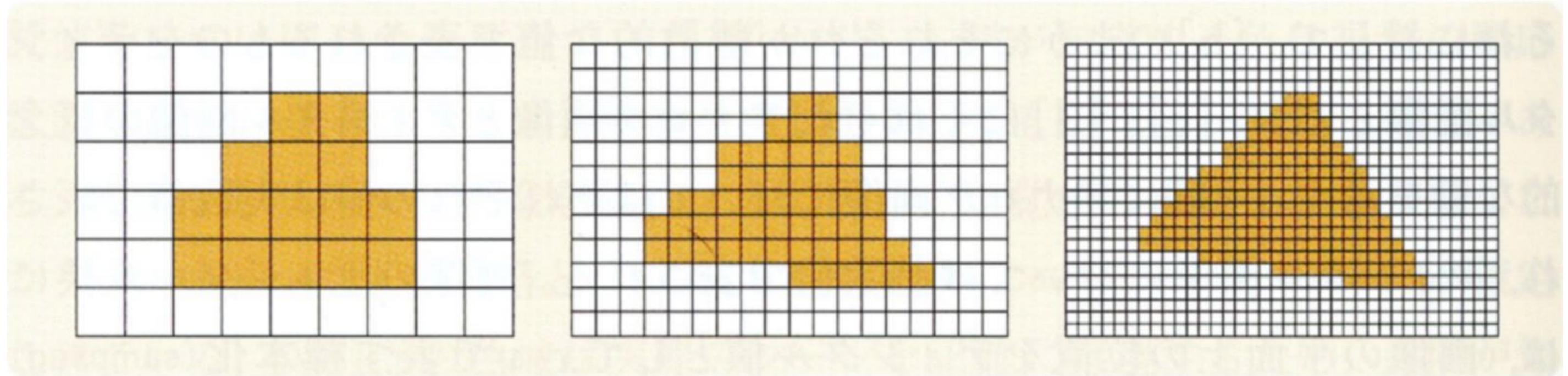
量子化レベル数が小さいと
濃淡・色調の
不連続が目立つ



解像度 (resolution)

標本化において画像をどこまで細かく分割して表現するかの尺度。

画像の解像度は、それを構成する「画素 (ピクセル: pixel)」の数または大きさを表す。



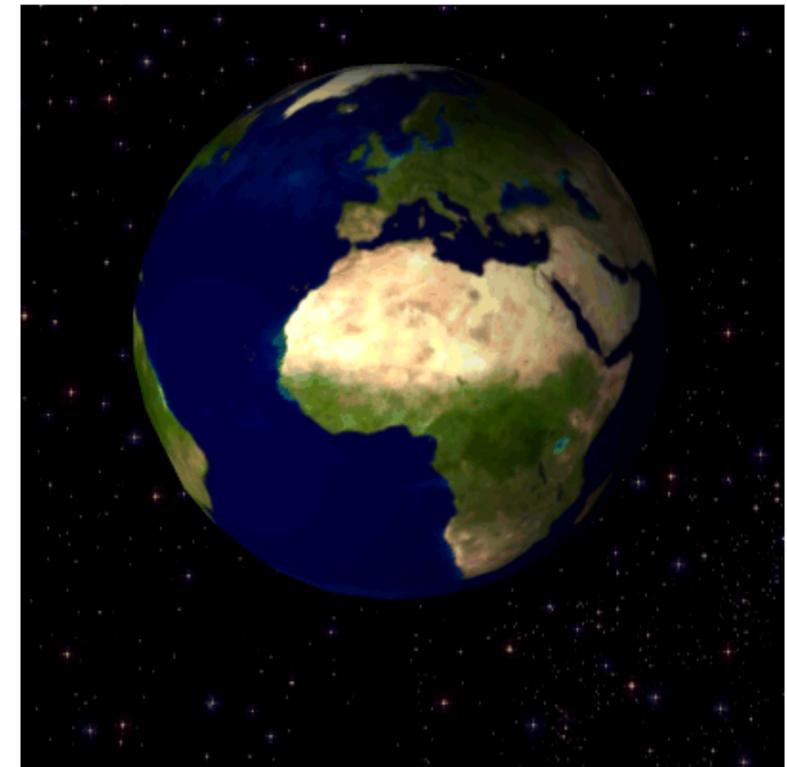
■ 図 2.21 —— 三角形を異なる解像度で表示した例

- ・ 画素 (ピクセル: pixel) ← picture cell
- ・ dpi (dots per inch)



画像 ウェブで使用するビットマップ形式：GIF

- ・ Graphics Interchange Format。
- ・ 8bit / pixel 。1つの画像ファイルの中で色数を256色まで使用可。
- ・ インデックスカラーという方法で色指定。
- ・ 減色することでファイルサイズをカットできる
- ・ 1色のみ透過として扱うことも可能。
- ・ 可逆圧縮。
- ・ 色数が限られているのでイラストやアイコンなどに向いている。
- ・ 拡張機能：透過GIF、アニメーションGIFなど





画像 ウェブで使用するビットマップ形式 : JPEG

- Joint **P**hotographic **E**xports **G**roup
- 画像の圧縮方式のひとつでファイル形式ではない。
- 24bit / pixel (GIFの3倍) 。 1,677,216色 (2の24乗) 。
- 写真やグラデーションなどに向く
- 非可逆圧縮で、8×8ピクセルのブロック単位で扱われる。
- 圧縮率を高くするとデータサイズは小さくなるがブロックノイズなどが発生し画質が低下
- 拡張機能 : プログレッシブJPEGなど





画像 ウェブで使用するビットマップ形式 : PNG

- **P**ortable **N**etwork **G**raphics
- W3Cが開発 (標準化) した画像形式。
- GIFとJPGの性質をあわせもつ。
 - GIFと同じく可逆圧縮。JPGと同じくフルカラー表現が可能。
- 透過はアルファチャンネルという形で情報をもつことができる。
- ファイルサイズはJPGよりも大。
- 拡張機能 : 透過PNG、インタレースPNG、アニメーションMNG





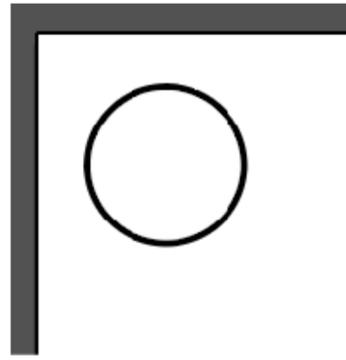
画像 ウェブで使用するベクター形式: SVG

- Scalable Vector Graphics
- XMLによるベクター画像を描画するためのマークアップ言語。
- 2001年にW3C勧告。 W3C: World Wide Web Consortium
- テキスト形式のファイルで画像を表現す
- アニメーション機能有。
- Flashのような簡単なインタラクティブコンテンツが作成できる。





画像 ウェブで使用するベクター形式 : SVG



```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 22.1.0, SVG Export Plug-In . SVG Version: 6.00 Build 0) -->
<svg version="1.1" id="レイヤー_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px"
  y="0px" viewBox="0 0 595.3 841.9" style="enable-background:new 0 0 595.3 841.9;" xml:space="preserve">
<style type="text/css">
  .st0{fill:#FFFFFF;stroke:#000000;stroke-miterlimit:10;}
</style>
<circle class="st0" cx="23.3" cy="23.8" r="14.2"/>
</svg>
```



画像 ウェブで使用するベクター形式: PDF

- **P**ortable **D**ocument **F**ormat
- 2008年にISOによって標準化。
- Adobe社の電子文書フォーマット。
- フォントや文字の大きさ、字飾り、埋め込まれた画像、それらのレイアウトなどの情報を保存できる。

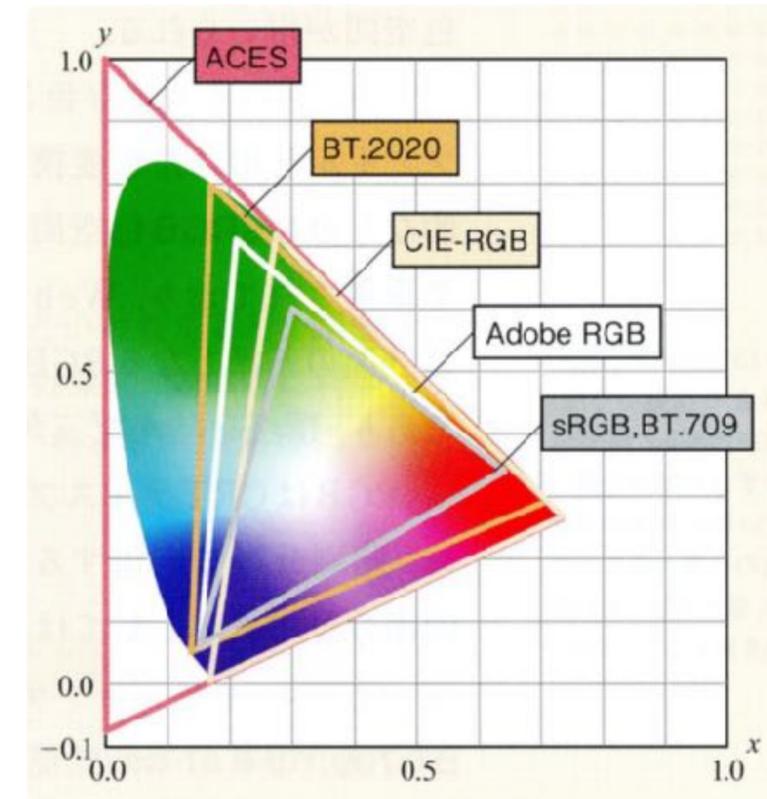


画像・動画のデジタル表現



1. 色

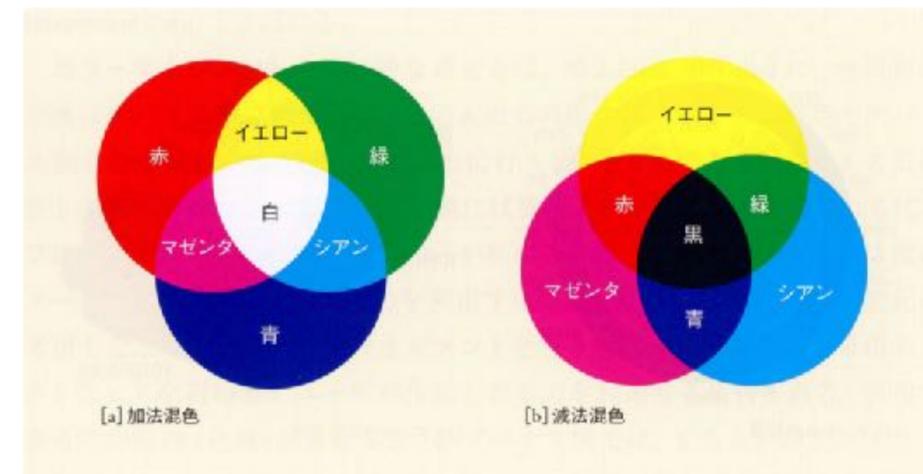
1. RGB
2. CMYK
3. HSB
4. その他のカラーコード
5. カラーマネージメントシステム



■図2.17—xy色度図

2. 画像

3. 動画



■図2.15—三原色による混色



色 RGB

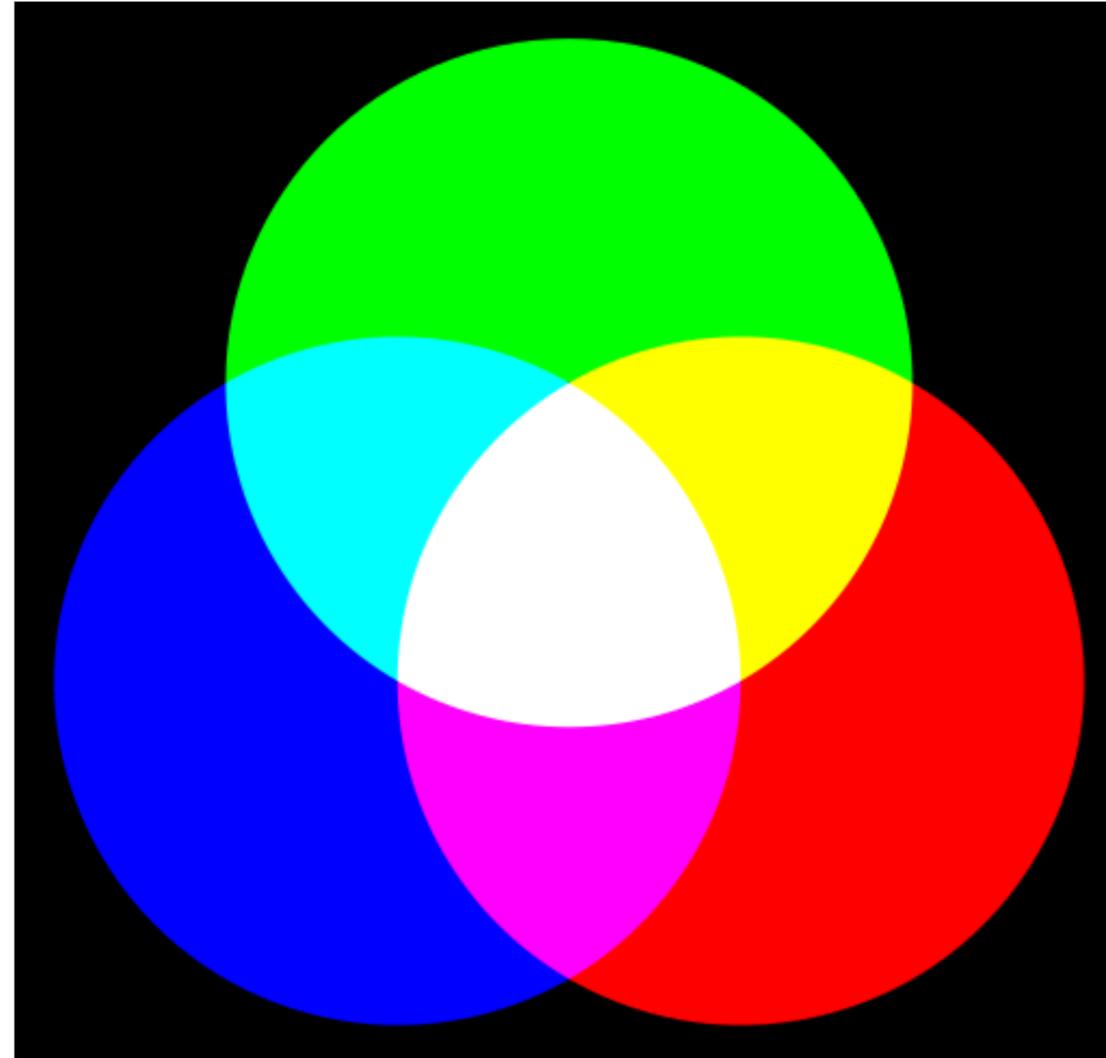
RGB値は16進数で表示され、HTMLでは、#rrggbbとして記述。
00で最小、ffで最大を表します。16,777,216通り。

16進数(24bit)	10進数	色
#000000	0×0×0	黒 (全く光がない)
#ff0000	255×0×0	赤
#00ff00	0×255×0	緑
#0000ff	0×0×255	青
#ffffff	255×255×255	白 (全ての光が重なる)

16進数以外に、以下のような色指定方法があります。

- 10進数で指定する ... rgb (0, 0, 255)
- %で指定する ... rgb (0%, 0%, 100%)

色 RGB

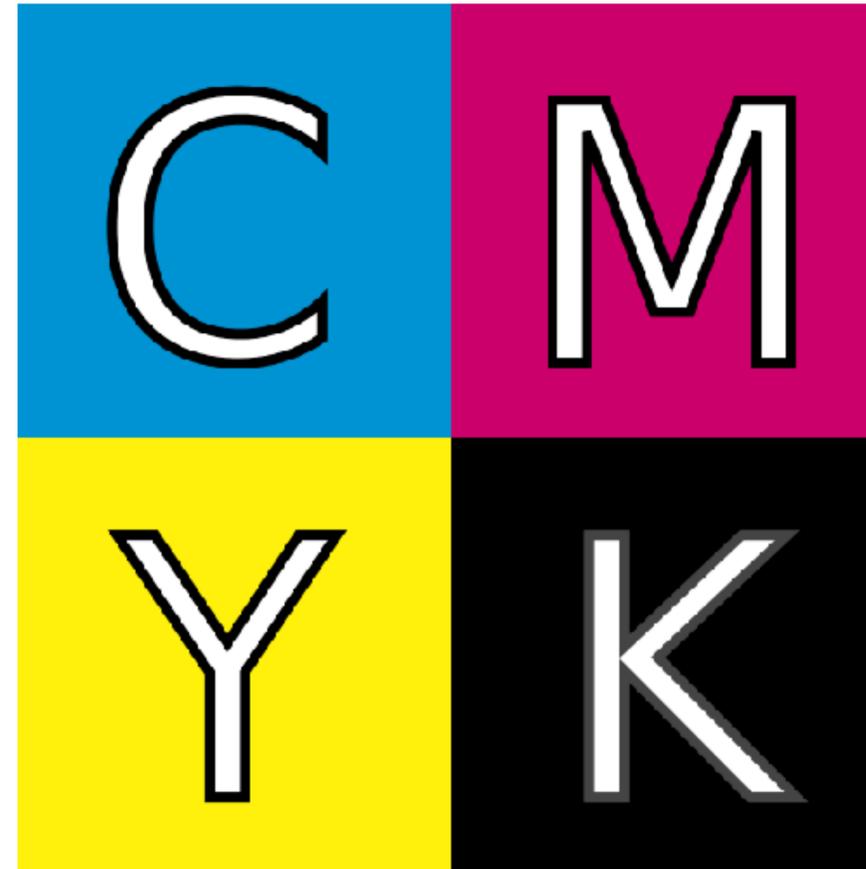


- ・色が重なるごとに白へと近づく。
- ・**加法混色**。ディスプレイやデジタルカメラで使用される。



色 CMYK

- ・シアン、マゼンダ、イエロー、ブラックによる色表現。
- ・これらの色が重なるごとに黒へと近づく。
- ・**減法混色**。印刷媒体で使用される。

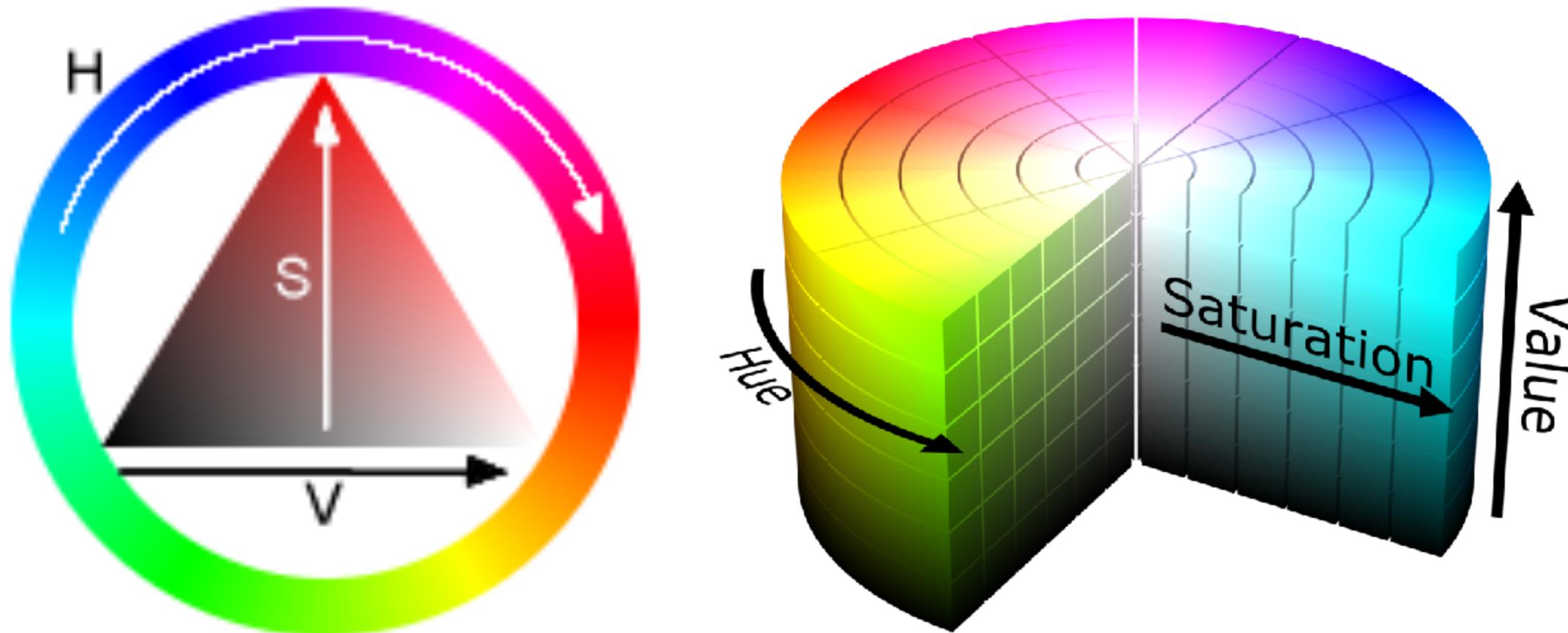


Key Plate



色 HSB

- ・色相 (Hue)、彩度 (Saturation)、明度 (Brightness)
- ・RGBより人間の色覚に合っていると見える。
- ・HSV、マンセル表色系ともいう

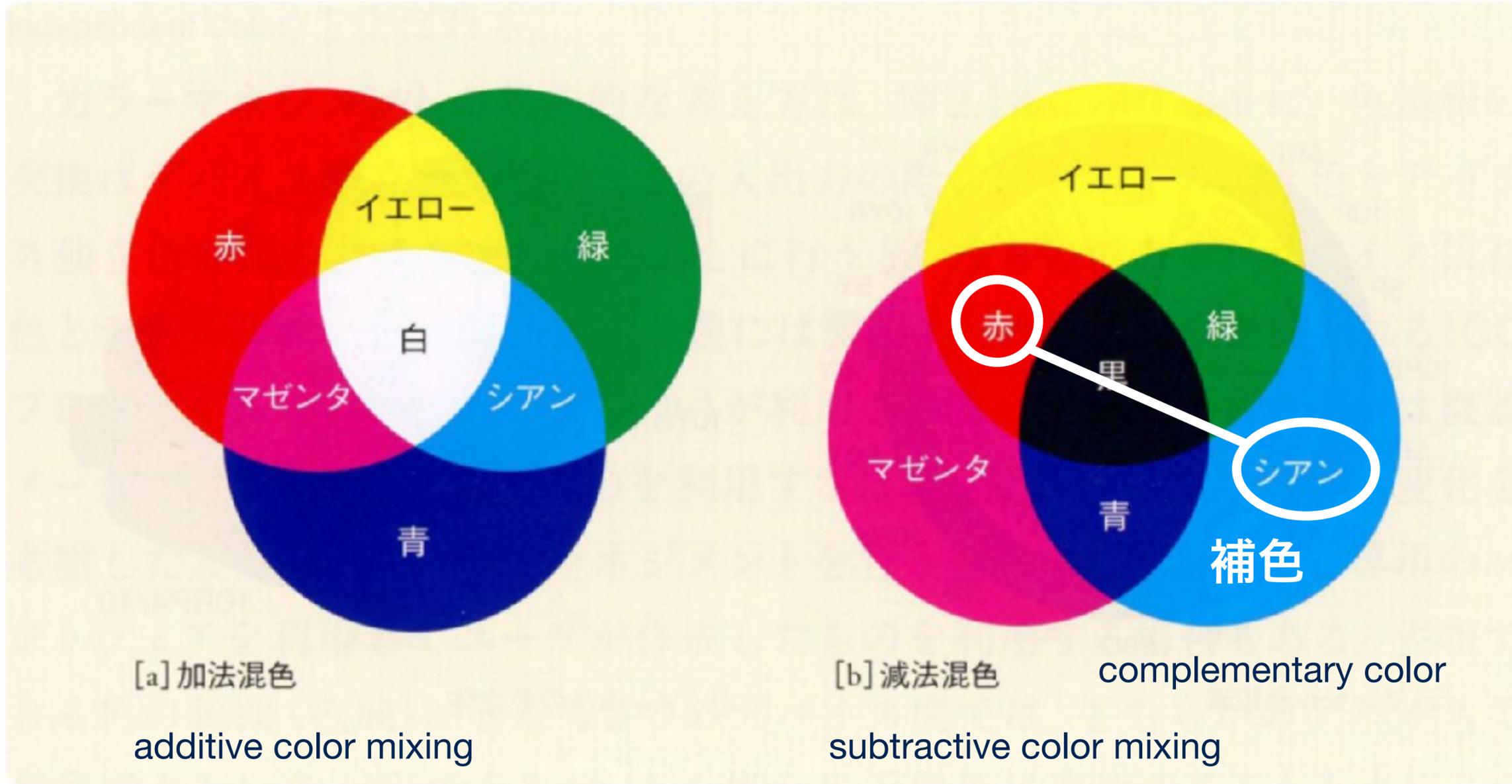




色 ... 加法混色と減法混色

color mixing

three primaries



■ 図 2.15 — 三原色による混色





色 その他のカラーコード : ウェブセーフカラー

- ・RGBのうち、MacとWindowsのOSが用意したカラーの重複分「216色 (=6×6×6)」を抜き出したもの。
- ・ブラウザやOSの違いに左右されない。
- ・各種ソフトのカラーパレット等に使われる。





色 その他のカラーコード：カラーネーム

- ・「white」「pink」などのカラーネームによる色指定。
- ・140種類。
- ・一般的には使用しない。

black #000000	aliceblue #f0f8ff	darkcyan #008b8b	lightyellow #ffffe0	coral #ff7f50
dimgray #696969	lavender #e6e6fa	teal #008080	lightgoldenrodyellow #fafad2	tomato #ff6347
gray #808080	lightsteelblue #b0c4de	darkslategray #2f4f4f	lemonchiffon #ffffac	orangered #ffa500
darkgray #a9a9a9	lightslategray #778899	darkgreen #006400	wheat #f5deb3	red #ff0000
silver #c0c0c0	slategray #708090	green #008000	burlywood #deb887	crimson #dc143c
lightgray #d3d3d3	steelblue #4682b4	forestgreen #228b22	tan #d2b48c	mediumvioletred #c71585
gainsboro #dcdcdc	royalblue #4169e1	seagreen #2e8b57	khaki #f0e68c	deeppink #ff1493
whiteSmoke #f5f5f5	midnightblue #191970	mediumseagreen #3cb371	yellow #ffff00	hotpink #ff69b4
white #ffffff	navy #000080	mediumaquamarine #66cdaa	gold #ffc700	palevioletred #db7093
snow #ffaafa	darkblue #00008b	darkseagreen #8fbc8f	orange #ffa500	pink #ffc0cb
ghostwhite #f8f8ff	mediumblue #0000cd	aquamarine #7fffd4	sandybrown #f4a460	lightpink #ffb6c1
floralwhite #ffaaf0	blue #0000ff	palegreen #98fb98	darkorange #ff8c00	thistle #d8bfd8

色 その他のカラーコード : カラーネーム

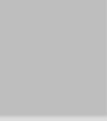
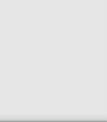


lightyellow #ffffe0	coral #ff7f50
lightgoldenrodyellow #fafad2	tomato #ff6347
lemonchiffon #ffffac	orangered #ff4500
wheat #f5deb3	red #ff0000
burlywood #deb887	crimson #dc143c
tan #d2b48c	mediumvioletred #c71585
khaki	deeppink

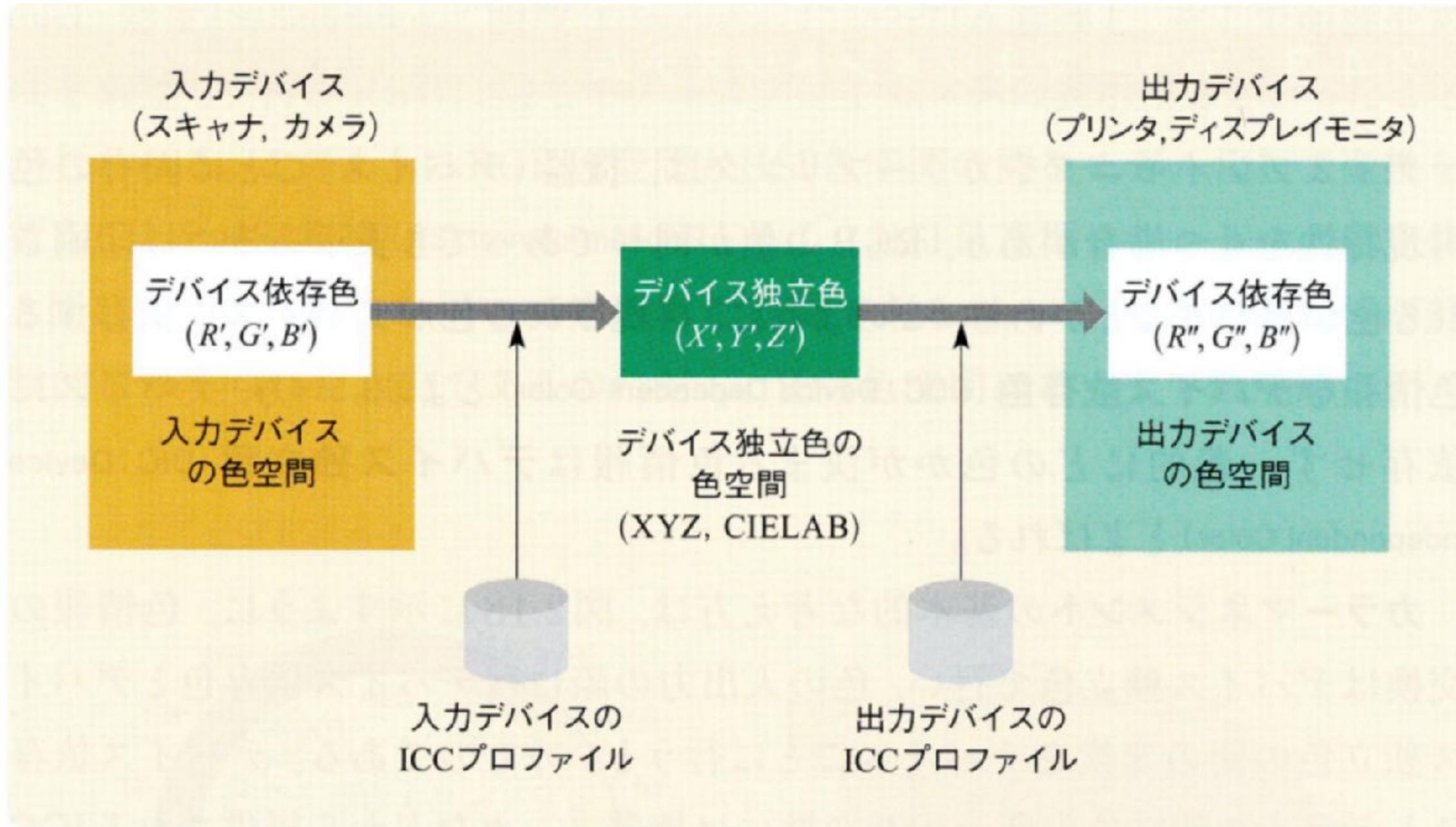


色 その他のカラーコード: システムカラー

- ・ユーザーのGUI環境で使用される28種。
Windows、Mac、Linux等のOSが保持しているシステム情報
- ・CSSにキーワードとして指定することが認められている。

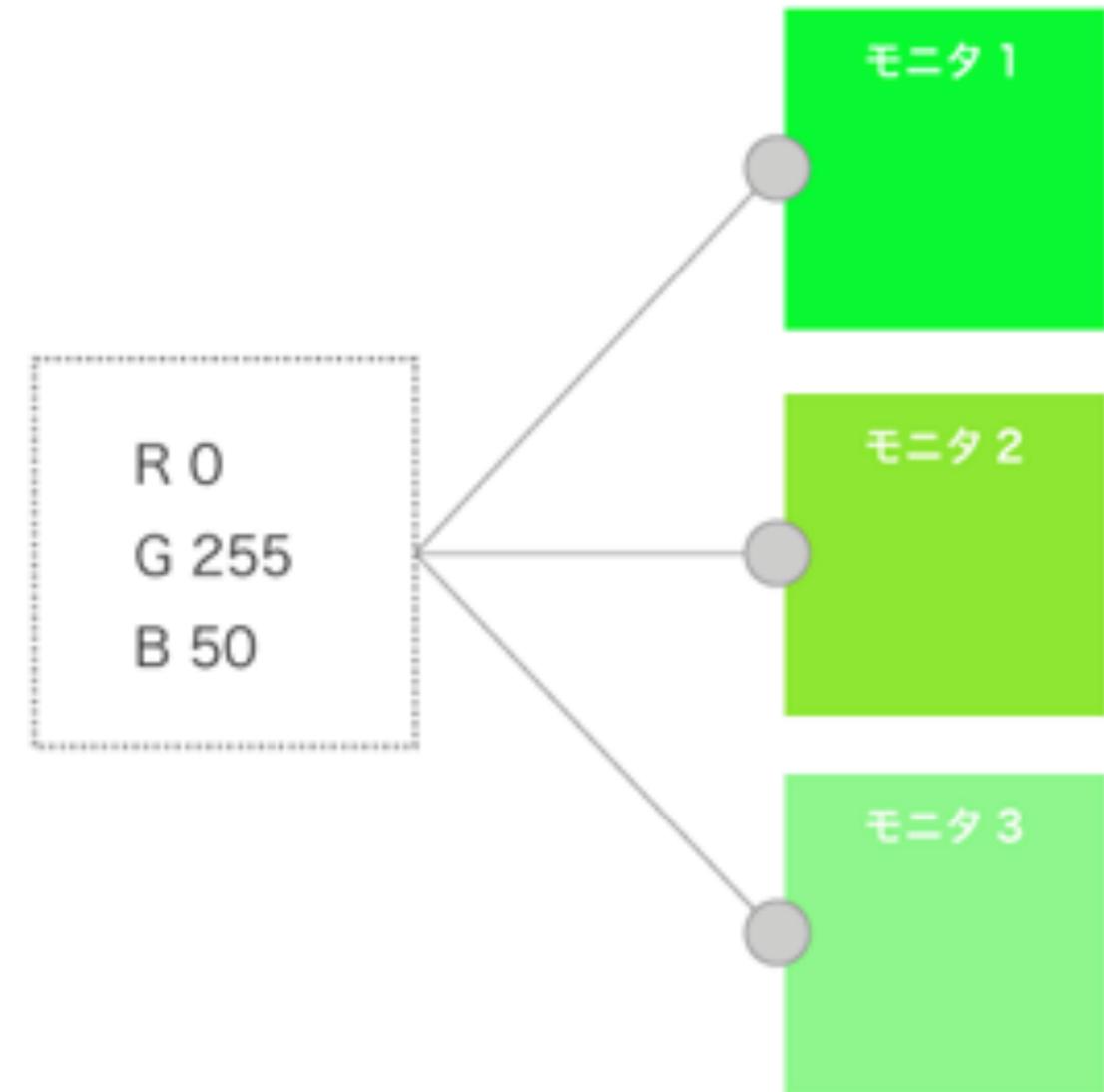
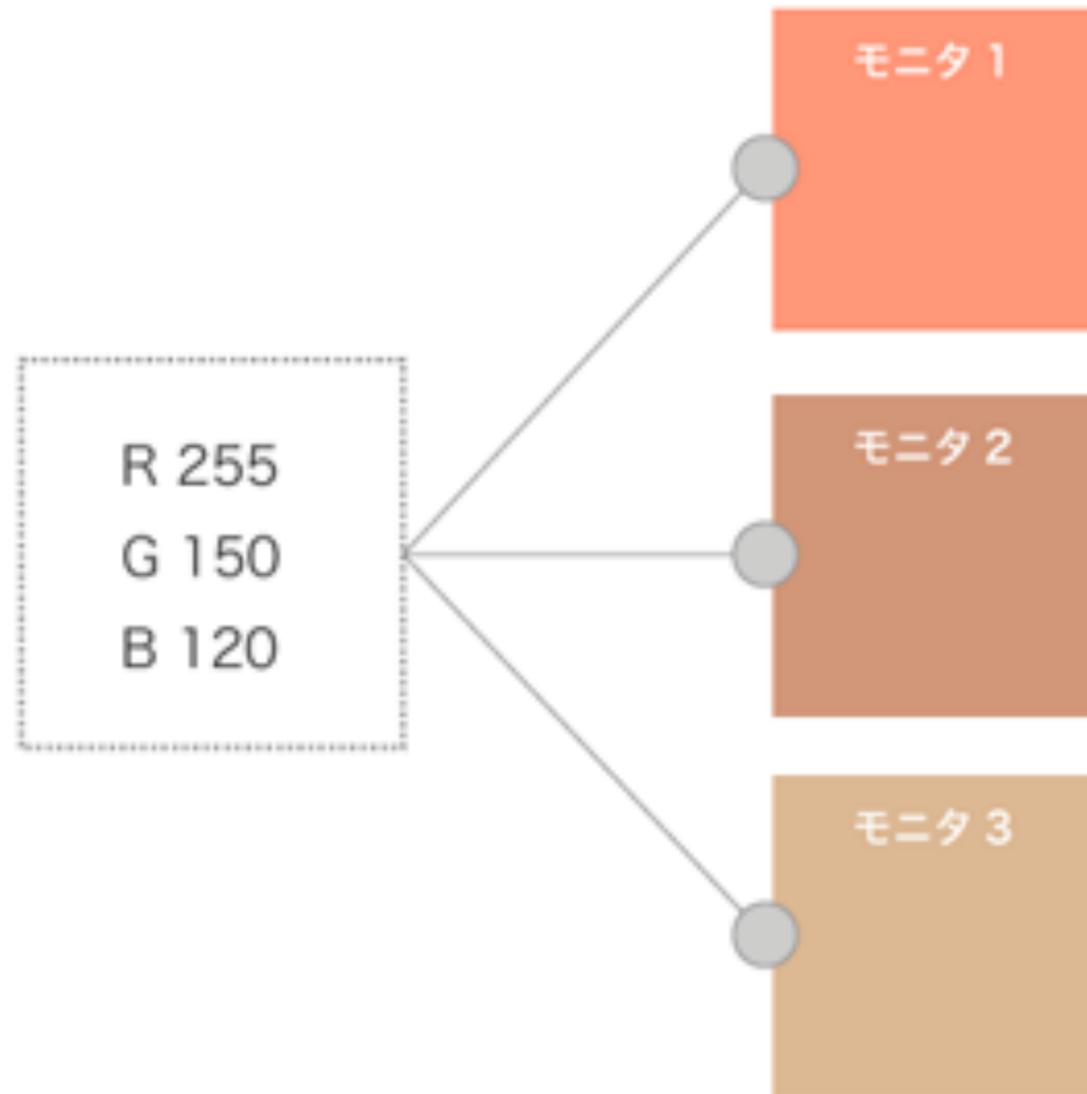
	ActiveBorder	アクティブなウインドウ枠の色
	ActiveCaption	アクティブなウインドウのタイトルバーの色
	AppWorkspace	アプリケーションウインドウの背景色
	Background	デスクトップの背景色
	ButtonFace	ボタンの色
	ButtonHighlight	選択されたときのボタンの色
	ButtonShadow	ボタンの影の色
	ButtonText	ボタンのテキストの色
	CaptionText	タイトルバーのテキストの色

カラーマネージメントシステム



■ 図2.18 — ICCプロファイルを用いたカラーマネージメントのしくみ International Color Consortium

カラーマネージメントシステム



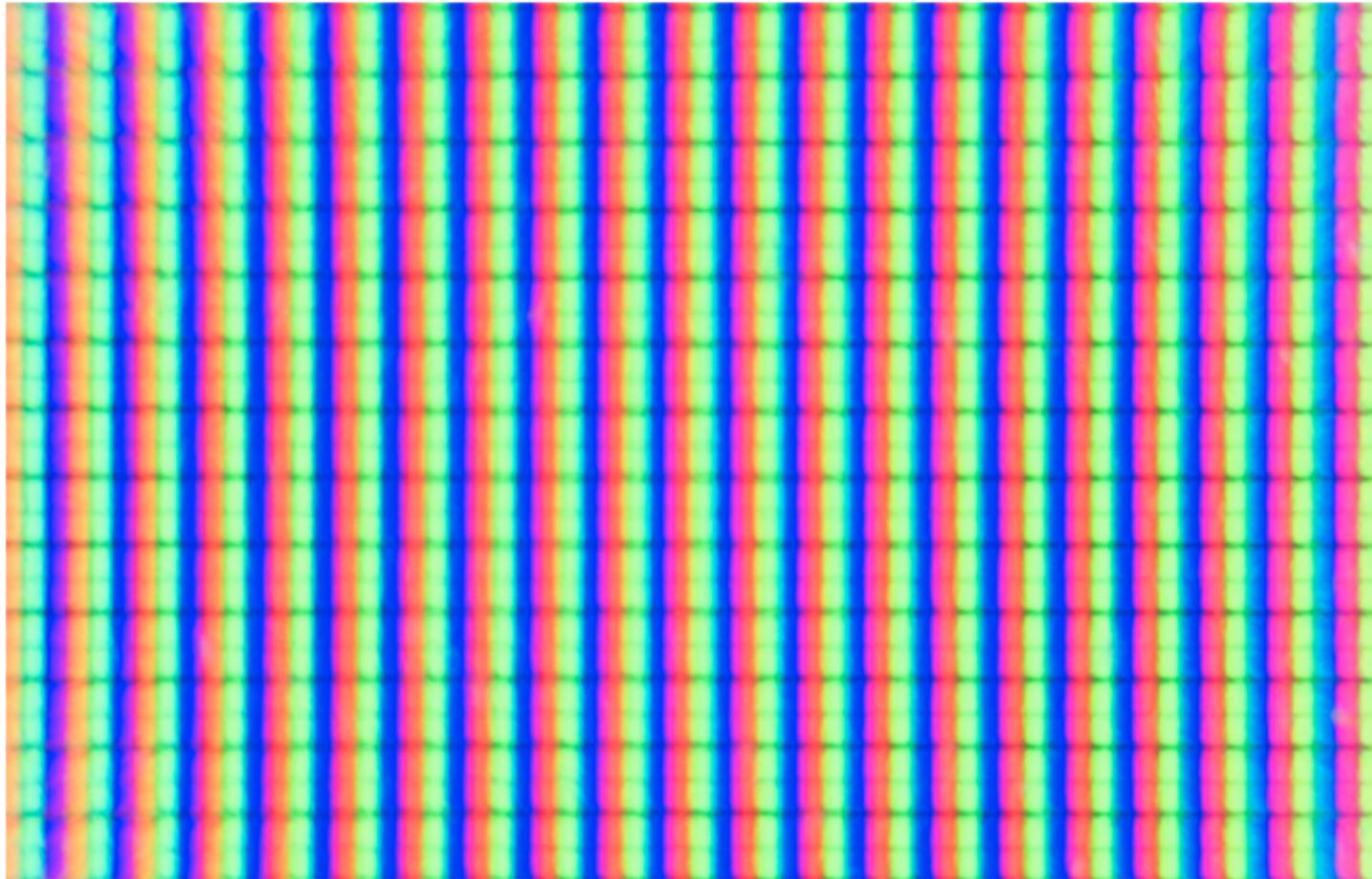
トレンド&テクノロジー / PERCH長尾の知っ得！デザインビズ必読ポイント！

第28回：カラーマネージメントの基礎知識 RGBの値が同じでも、色は変わる

https://area.autodesk.jp/column/trend_tech/designviz_point/28_rgb_color-management/



カラーマネージメントシステム



モニタによって Red, Green, Blue の各色の発色特性は異なります。より鮮やかな赤だったり、暗い赤だったり、朱色（少し黄色がかった赤）だったり、機種ごとにバラバラです。これはモニタに利用されている液晶の特性と、その発色を管理しているモニタ内部の組み込みコンピューターによって発色特性が異なるからです。

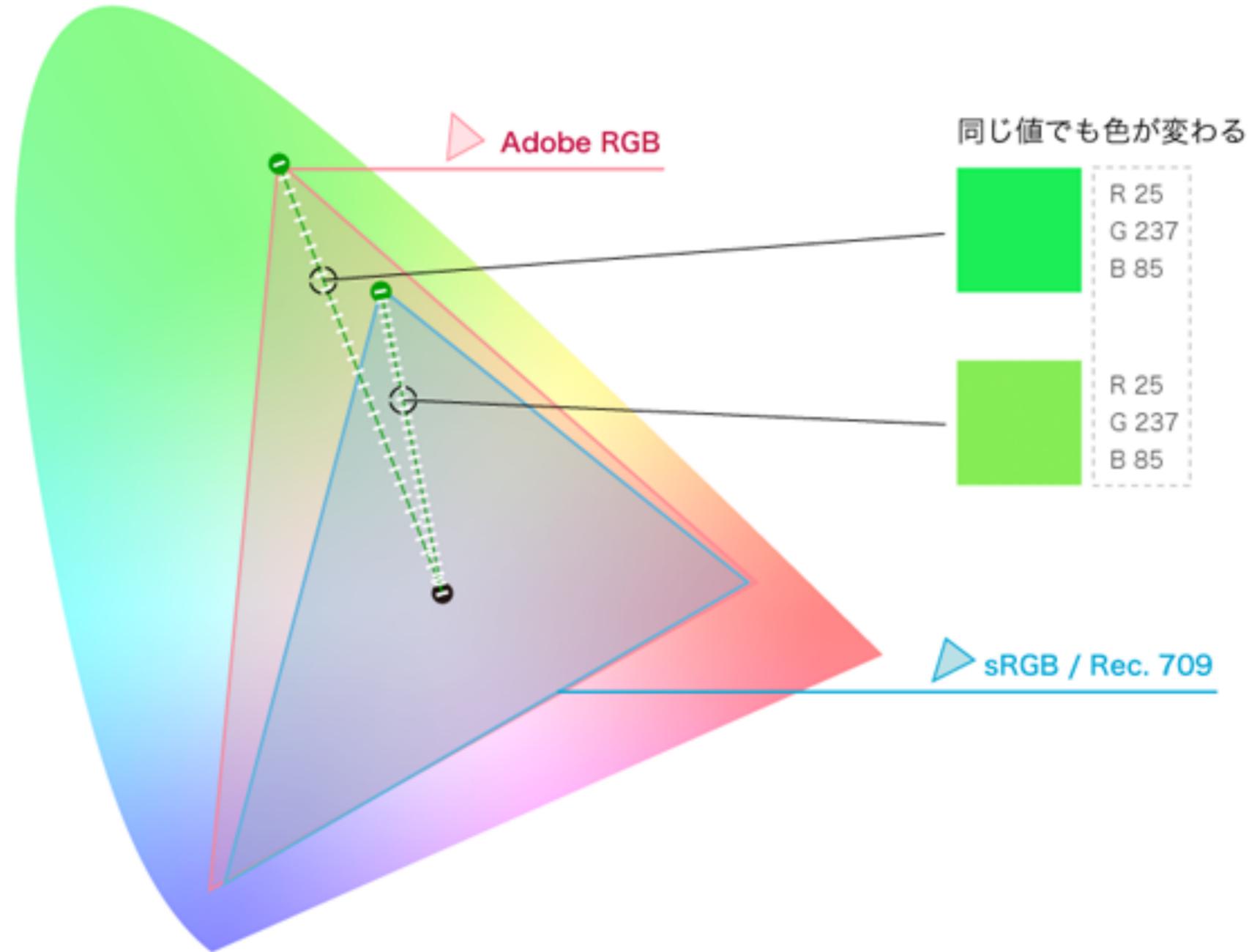
Red, Green, Blue の3色の発色特性が異なるわけですから、色も変わってきます。

トレンド&テクノロジー / PERCH長尾の知っ得! デザインビズ必読ポイント!

第28回: カラーマネージメントの基礎知識 RGBの値が同じでも、色は変わる

https://area.autodesk.jp/column/trend_tech/designviz_point/28_rgb_color-management/

カラーマネージメントシステム



RGB は8ビットで利用することが多いと思います(より階調がなめらかな16ビットも最近は多く利用されるようになりました)。8ビットは256パターンを管理できるので、Red, Green, Blueの発色特性(最大出力)から黒(出力ゼロ)までを256段階で再現しています。そのため256の最大出力の色が異なるだけでなく、中間の値であっても色が異なります。

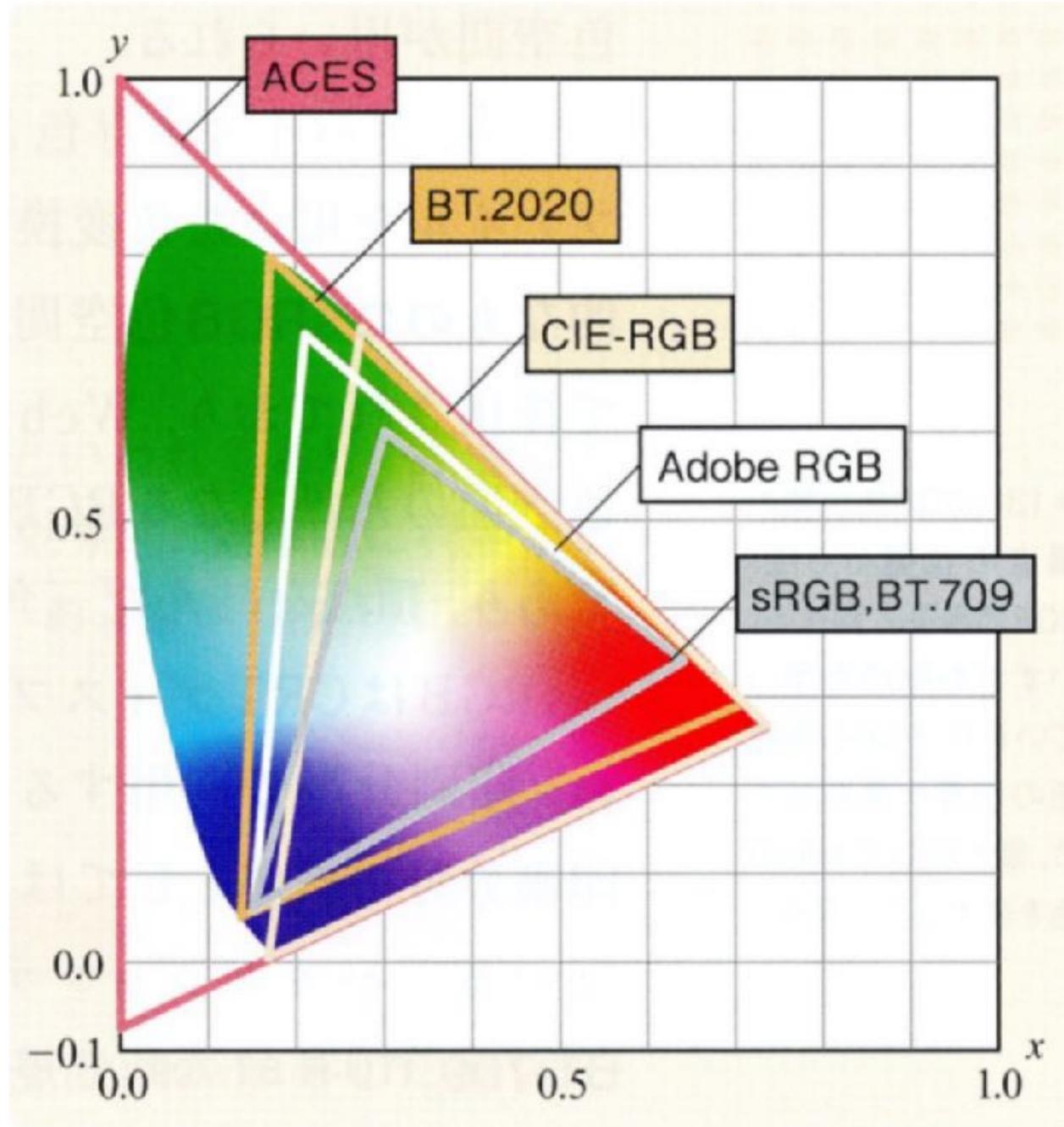
トレンド&テクノロジー / PERCH長尾の知っ得! デザインビズ必読ポイント!

第28回: カラーマネージメントの基礎知識 RGBの値が同じでも、色は変わる

https://area.autodesk.jp/column/trend_tech/designviz_point/28_rgb_color-management/



RGB表色系



■ 図 2.17 — xy 色度図

ACES (Academy Color Encoding System)

映画芸術科学アカデミーの支援の下で開発された無料でオープンな、デバイスに依存しないカラーマネジメントおよび画像交換システム。

BT.2020 UHDTV放送方式の映像信号を規定する勧告ITU-R BT.2020。4K, 8Kテレビの色域。

CIE-RGB CIE (国際照明委員会) が定めた実在する光の三原色の混合比による加法混色によるカラーシステム。

Adobe RGB Adobe Systems社が発表した色空間

sRGB Hewlett-PackardとMicrosoftが提唱。Windows98以降で採用。

BT.709 ITU-R (国際電気通信連合無線通信部門 ITU Radiocommunication Sector) によって高精細度テレビジョン放送 (2Kテレビ) のエンコードと信号特性について制定された標準規格。



画像・動画のデジタル表現

1. 画像

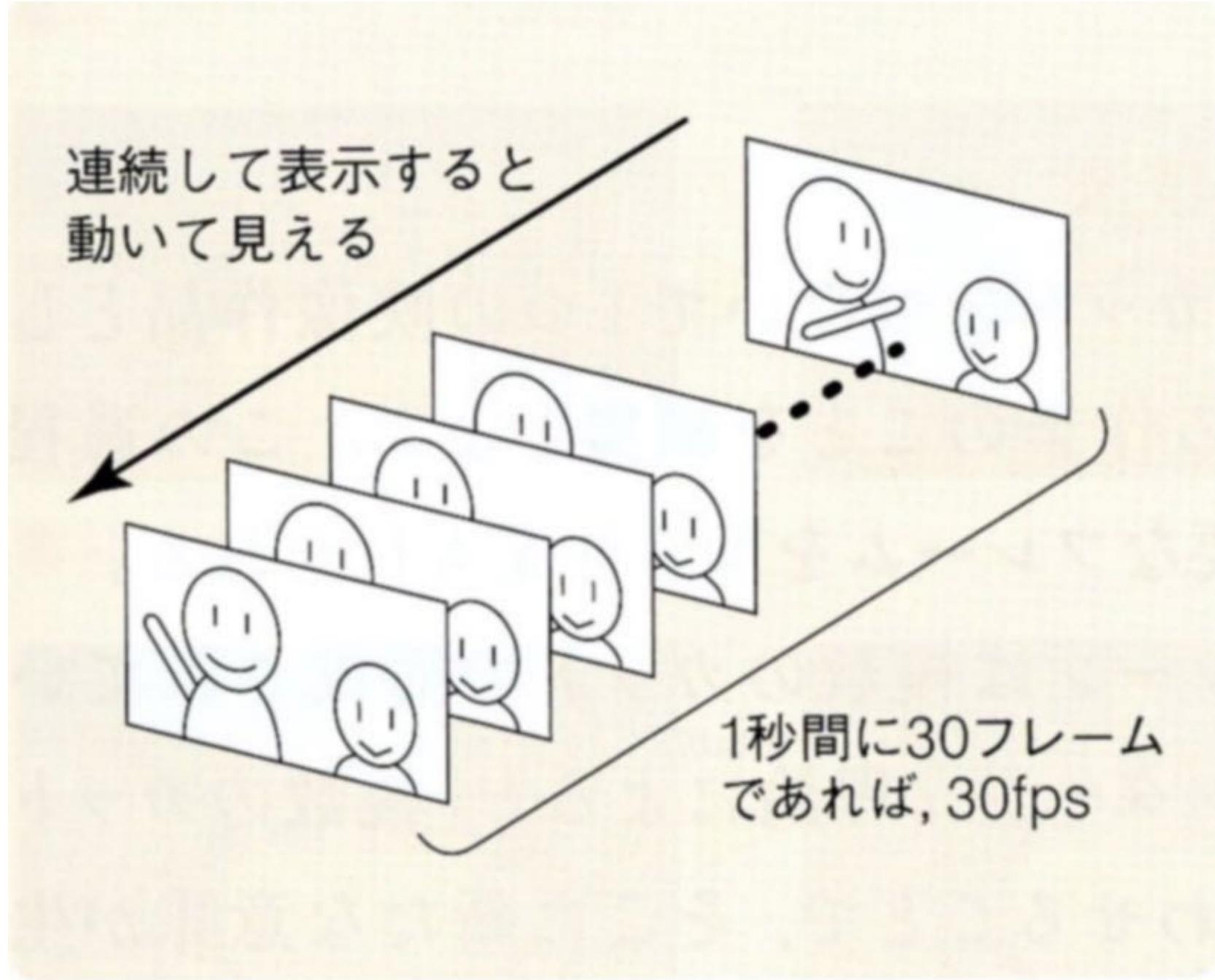
2. 色

3. 動画

1. 動画と音声の値
2. 動画コーデック
3. 音声コーデック
4. コンテナフォーマット
5. ストリーミング再生とは
6. Flash



動画のデジタル表現

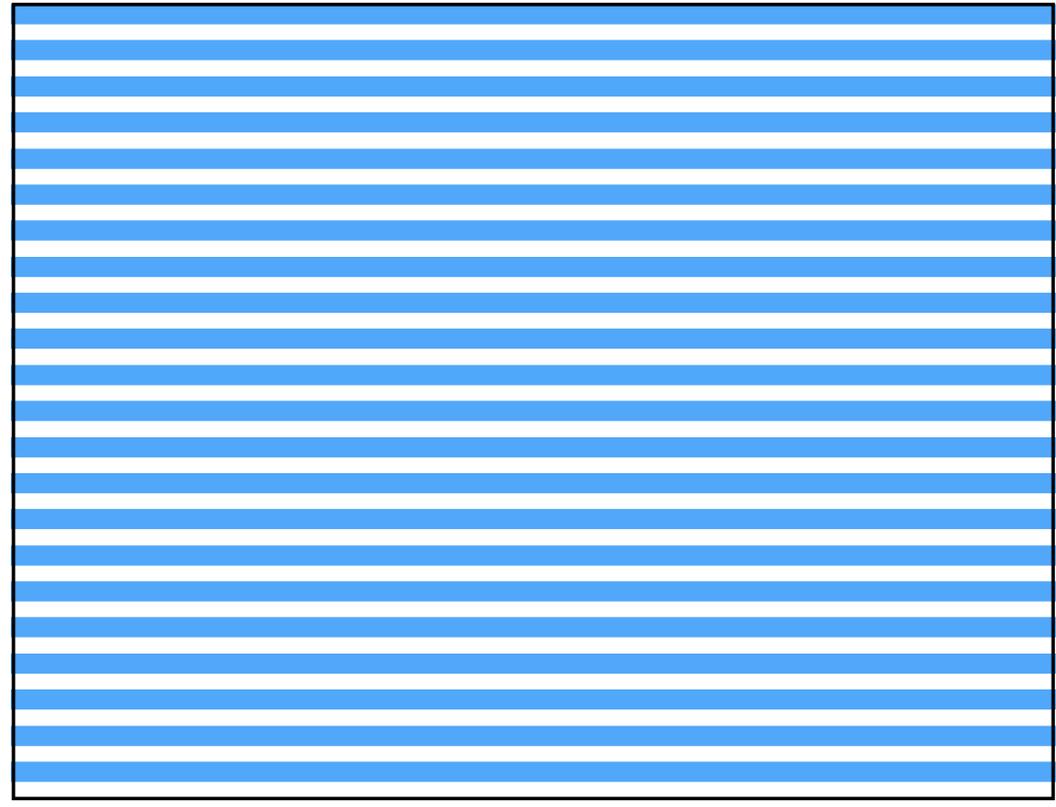


■図 2.55 —— 静止画と動画



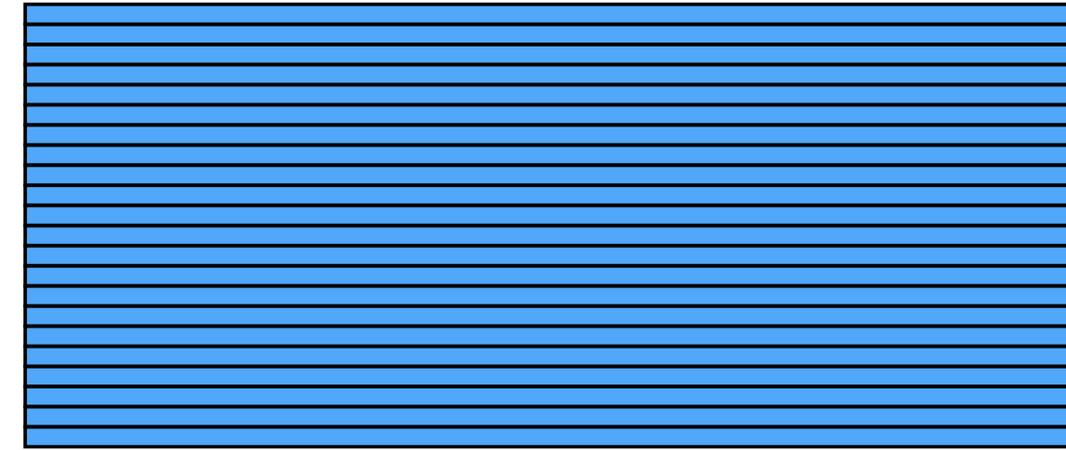


フレームの構造

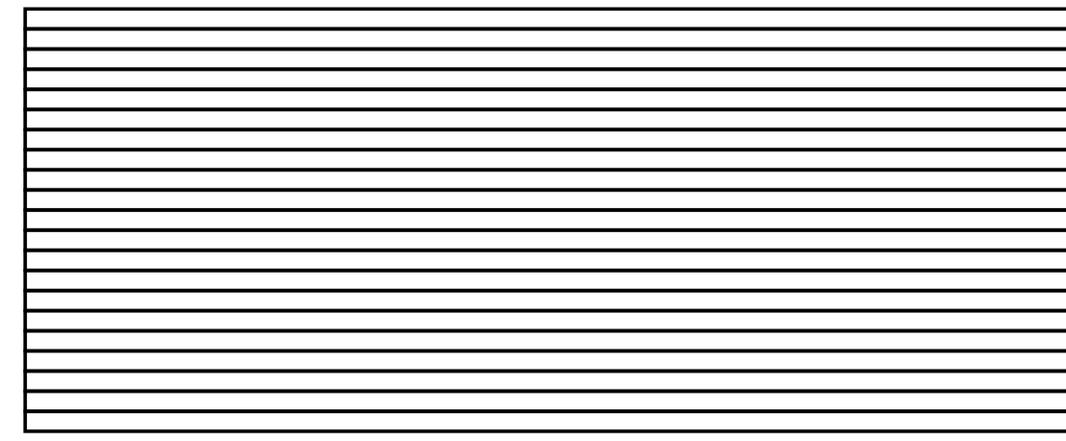


フレーム

- 標準テレビ：525本
- HDTV：1125本
- 4K：2160本
- 8K：4320本



奇フィールド



偶フィールド





4K, 8Kの高画質

HDR: High Dynamic Range



SDRの「黒つぶれ」

室外に露出を合わせた場合



SDRの「白飛び」

室内に露出を合わせた場合



自然でリアルなHDR

室外も室内も良く写っている



画像・動画のデータ量



3080

5472

単純計算

$$5472 \times 3080 \times 24 = 385.8\text{MB}$$

file size= 8.5MB 1/45



720

1280

単純計算

$$1280 \times 720 \times 24 \times 29.97 \times 31 = 19.1\text{GB}$$

file size= 28.2MB 1/694



動画・音声の符号化 動画と音声の値

フレームレート

映像データにおいて、1秒あたりの描画回数をfps (フレーム毎秒 : frame per second) で表した値。フレームレートが30fpsの場合、1秒間に30回描画が行われる。

サンプリングレート

音声データにおいて、1秒あたりのデジタル変換回数をHzで表した値。

ビットレート

映像データや音声データにおいて、1秒あたりのデータ送受信 (転送) 量をbps (ビット毎秒 : bits per second) で表した値。



音声 コーデックの種類：WMA

- **Windows Media Audio**
- 可逆圧縮の音声コーデック。
- Microsoft社が開発したWindowsの標準音声圧縮方式で、Windows Media Playerで再生可能。





音声 コーデックの種類 : MP3

- MPEG-1 Audio Layer 3
- CDと同程度の音質を保ったまま約1/11 (128kbps) に圧縮。
- CDのビットレートは $44.1 \text{ kHz} \times 16 \text{ bit} \times 2 \text{ channel}$
= 1411.2 kbps





音声 コーデックの種類 : AAC

- **A**dvanced **A**udio **C**oding
- ISO/IECのMPEGで策定された音声圧縮方式。
- MPEG-4 に含まれる。MP3より効率的な圧縮が可能。





音声 コーデックの種類 : Vorbis

- ・ MP3などの代替のパブリックドメインのフォーマット
- ・  xiph.org
- ・ 主にWebM、Ogg (オック) コンテナで使用される。
- ・ 動画コーデックにVP9、音声コーデックにVorbis、コンテナにWebMを使用した動画が、Youtubeの標準形式として採用されている。





動画

コーデックの種類

└── 符号化・復号の形式 CODEC

1. H.264 (MPEG-4)

携帯コンテンツ等で使用される通信用の低画質ファイル形式。



2. WMV

MPEG-4を基にMicrosoftが開発。Windows Media Player 標準サポート

3. Sorenson H.263 : Flash Video用。



4. On2 VP6 / FLV

Flash Video、Skype Video などに対応。広範囲の帯域で良い性能。

5. Theora

オープンソースの動画コーデック。FireFoxではプラグインなしに再生が可能。



コンテナフォーマットの種類



- ・さまざまな種類のデータや標準的なデータ圧縮方法を使って圧縮したデータを保持できるファイルフォーマット。
- ・単純なコンテナフォーマットは、異なる種類の音声ファイル形式データを複数含むことができる。
- ・先進的なコンテナフォーマットは、さまざまなストリーミングを再生し直すのに必要な同期情報とともに、音声・動画・副題・章（チャプター）・字幕・メタデータ（タグ）などに対応する。

MOV



AVI



ASF



FLV



3GPP / 2GPP2



コンテナフォーマット : MOV



- Apple社が開発した QuickTime のコンテナフォーマット。
- <拡張子> .qt / .mov / .moov
- <格納できるコーデック> MP3、AAC、WMA、On2VPシリーズ、MPEG など





コンテナフォーマット : AVI

- **A**udio **V**ideo **I**nterleaving
- Microsoft社が開発したWindowsの標準的な動画ファイル形式。
- ウェブサイトで動画を配信するには不適切。Windows上でマルチメディアデータの格納に使われるRIFFというフォーマットを応用している。
- <拡張子> .avi
- <格納できるコーデック> MP3、AAC、WMA、On2VPシリーズなど



コンテナフォーマット : ASF



- **A**dvanced **S**treaming **F**ormat
- Microsoft社が開発したAVIの後継で、ストリーミング配信するためのコンテナフォーマット。
- <拡張子> .asf .wmv .wma
- <格納できるコーデック> MP3、AAC、WMA など



コンテナフォーマット: FLV



- ・ **F**lash **V**ideo
- ・ Adobe Flashが標準で対応しているコンテナフォーマット。
- ・ インターネットでアップロードされる際にこの形式に変換される。
- ・ <拡張子> .flv
- ・ <格納できるコーデック> H.263、AAC など



コンテナフォーマット: 3GPP / 2GPP2

3G携帯電話向けのコンテナフォーマット。

<拡張子> .3gp / .3g2

<格納できるコーデック> H.263、AAC など



3GPP: 3rd Generation Partnership Project

第三世代携帯電話(3G)システム 及び

それに続く第3.9世代移動通信システムに対応するLTEや、

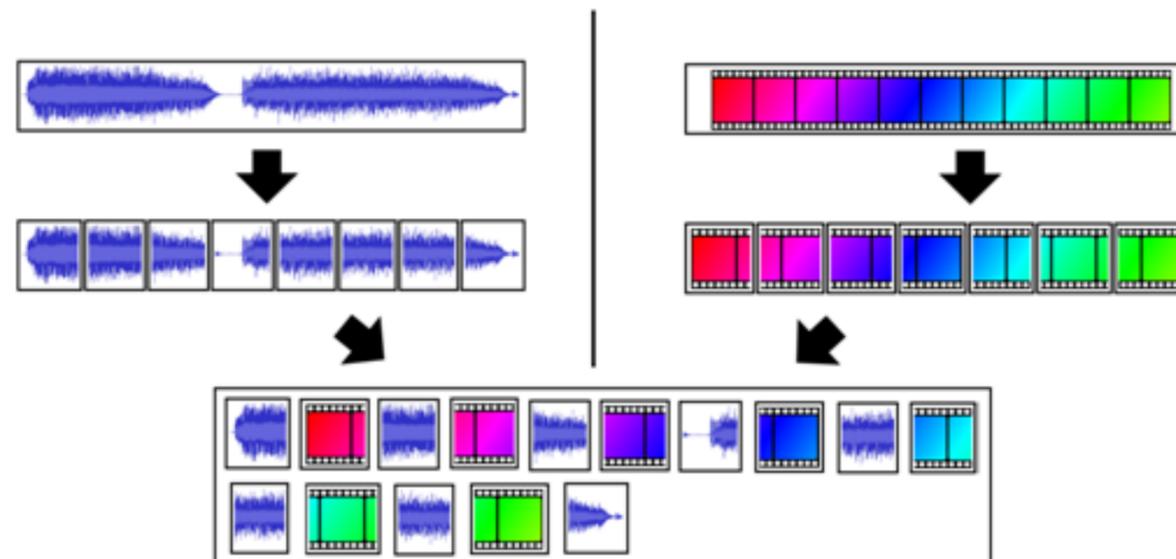
第4世代移動通信システムに対応するLTE-Advanced、さらに

次の世代である第5世代移動通信システム(5G)の仕様の検討・作成を行う標準化プロジェクト。

画像・動画のデジタル表現

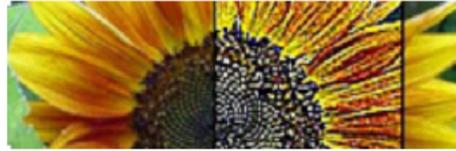
今日の要点

1. 画像のデジタル化 : 2次元標本化、濃淡を量子化
2. マルチメディアはデータ量が膨大…圧縮必須
3. メディアで表示できる色は限られる
4. 色の値が同じでも見え方が異なることがある
5. codecとcontainer





Processingで画像を扱う



1. 画像をファイルから読み込んで表示

`PImage`, `loadImage()`, `image()`

`PImage img;` 画像データを扱うデータタイプの変数 `img` を宣言。

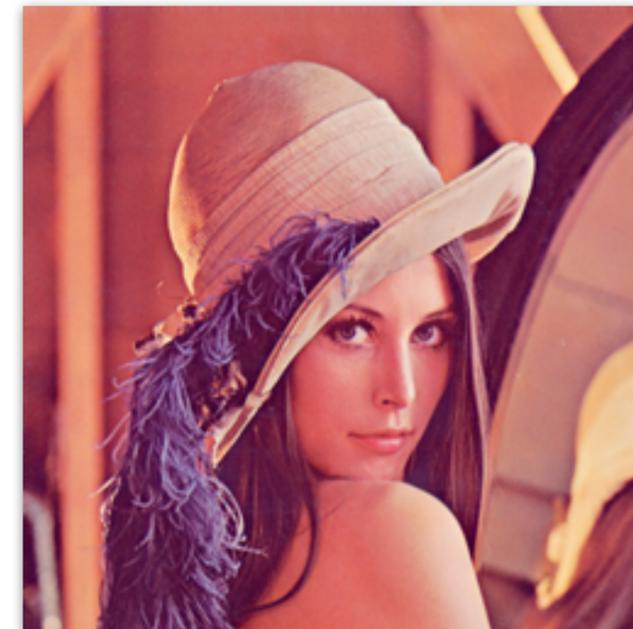
```
setup() {
    size(256,256);
    img = loadImage("Lenna.bmp");
}
```

"Lenna.bmp" というビットマップ画像を `img` に読み込む。

```
void draw() {
    background(0);
    image(img,0,0);
}
```

Processingでは画像処理の関数が提供されている。講義第7回と第8回ではこれらを用いず説明する。第2回のレポート課題ではProcessingの画像処理関数は使用禁止。

表示画面の `(0,0)` に `img` の画像を表示する。





Processingで色を扱う



1. 図形の色を指定する(1)

`background()`, `stroke()`, `fill()`

```
size(200, 200);
```

画面サイズは200x200ピクセル

```
background(255);
```

背景は白

```
stroke(0);
```

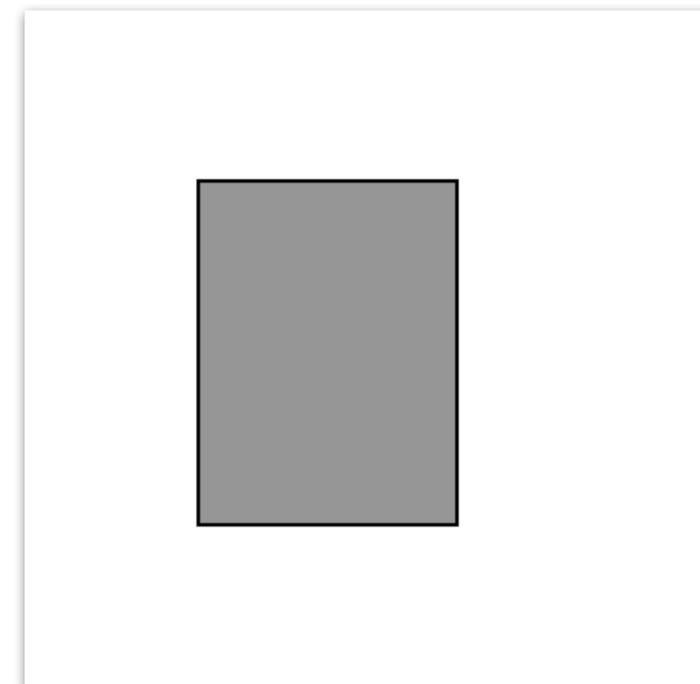
線あるいは図形の縁は黒

```
fill(150);
```

図形塗りつぶし色は150の灰色

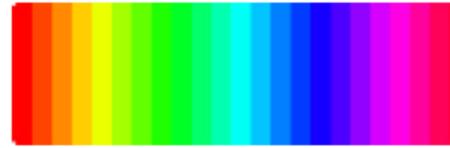
```
rect(50, 50, 75, 100);
```

(50, 50) から幅75ピクセル、高さ100ピクセルの矩形を描く





Processingで色を扱う



2. 図形の色を指定する(2)

`noStroke()`, `noFill()`

`background(150);`

背景は150の灰色

`stroke(0);`

線の色は黒

`line(0,0,100,100);`

左上角から右下角へ直線を引く

`stroke(255);`

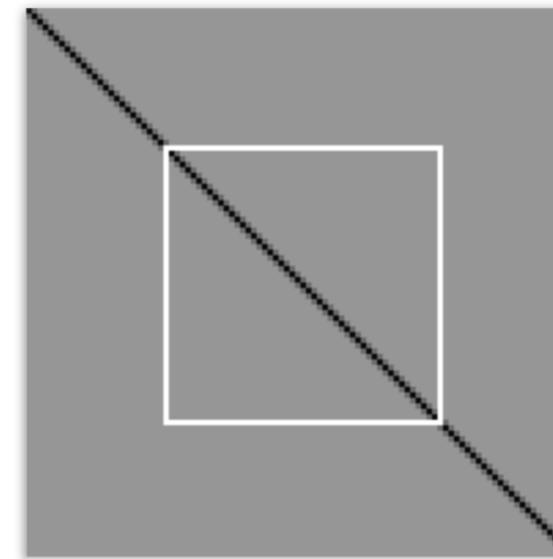
線の色は白

`noFill();`

塗りつぶさない

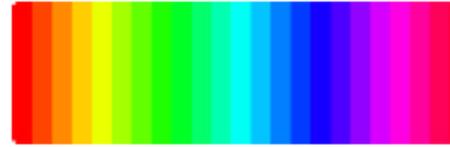
`rect(25,25,50,50);`

(25, 25) から幅50ピクセル、高さ50ピクセルの矩形を描く





Processingで色を扱う



3. 図形の色を指定する(3) RGBカラー

`noStroke()`, `noFill()`

```
background(255);
```

```
noStroke();
```

```
fill(255,0,0);
```

鮮やかな赤

```
ellipse(20,20,16,16);
```

円を描く

```
fill(127,0,0);
```

暗い赤

```
ellipse(40,20,16,16);
```

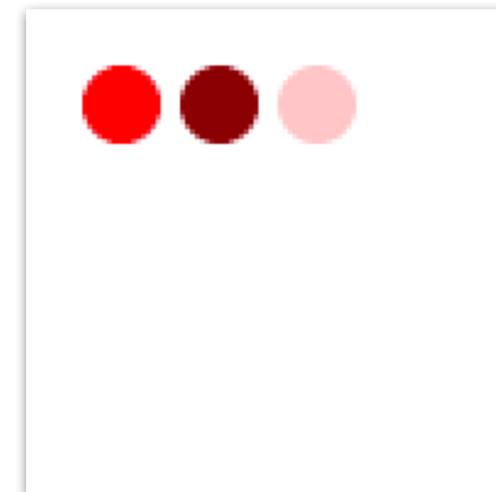
円を描く

```
fill(255,200,200);
```

ピンク

```
ellipse(60,20,16,16);
```

円を描く





Processingで色を扱う



4. 図形の色を指定する(4) 透明度

alpha transparency 0:透明~255:不透明

```
size(200,200);
background(0);
noStroke();
fill(0,0,255);
rect(0,0,100,200);
fill(255,0,0,255);
rect(0,0,200,40);
fill(255,0,0,191);
rect(0,50,200,40);
fill(255,0,0,127);
rect(0,100,200,40);
fill(255,0,0,63);
rect(0,150,200,40);
```

背景は黒

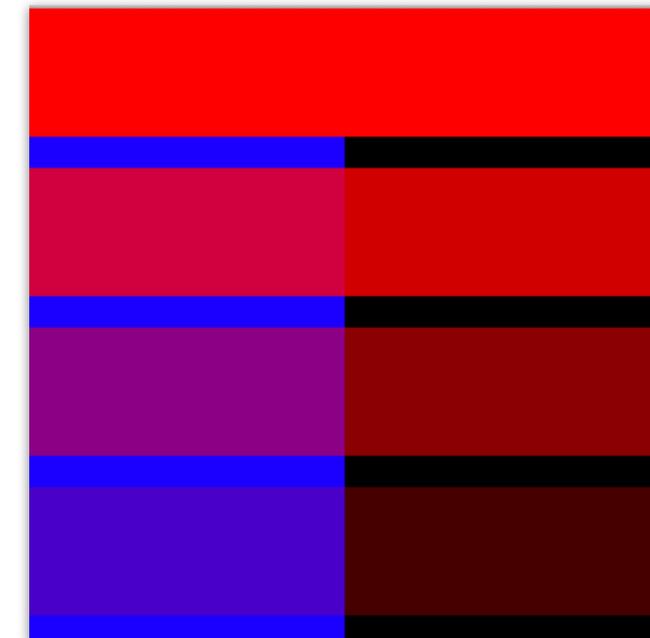
画面の左半分を青100%で塗りつぶす

一番上に200×40ピクセルの不透明な赤100%の矩形を描く

その下に200×40ピクセルの75%不透明な赤100%の矩形を描く

その下に200×40ピクセルの50%不透明な赤100%の矩形を描く

その下に200×40ピクセルの25%不透明な赤100%の矩形を描く





Processingで PDF



1. 直線を描きPDFファイルとして保存 画面非表示

```
import processing.pdf.*;
```

PDF Exportライブラリを呼び込む

Processingから直接PDFファイルを生成するためのライブラリ

```
void setup() {
```

```
  size(400, 400, PDF, "filename.pdf");
```

```
}
```

400× 400ピクセルのPDFファイル "filename.pdf" を定義

```
void draw() {
```

```
  line(0, 0, width/2, height);
```

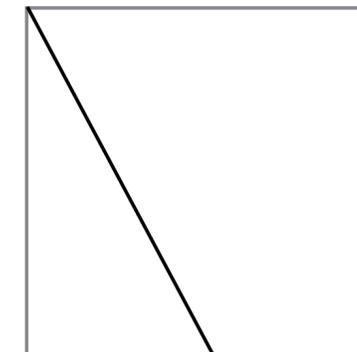
```
  println("Finished.");
```

```
  exit();
```

コンソールに終了メッセージを表示

```
}
```

(0,0)



(width,height)



Processingで PDF



2. 直線を描きPDFファイルとして保存 画面にも表示

`beginRecord()`と`endRecord()`関数を使う。

```
import processing.pdf.*;
```

```
void setup() {  
  size(400, 400);  
  noLoop();
```

```
  beginRecord(PDF, "filename.pdf");
```

```
}
```

新しいPDFファイルを開き、これ以降の画面への描画を全てファイルに記録する。

```
void draw() {  
  line(0, 0, width/2, height);  
  endRecord();
```

ファイル記録を終了する。

```
}
```

<https://processing.org/reference/libraries/pdf/> による



レポート課題 [前半第1回]

課題

1-1 文字・テキストの表現と処理

新聞記事など身近なテキストデータを形態素解析し、出現する形態素ごとにその形態素が出現する頻度を数えよ。2つ以上の分野のテキストデータの結果を比較すること。元のテキストの特徴を表しているか？（分析に用いるテキストデータは多ければ多いほど良い。）

1-2 音声のデジタル表現と処理

スマホやパソコンの音声入力機能を用いて「音声インチキ」と発音してみよ。不明瞭に発音したときに音声認識の結果はどう変わるか。なぜそのような変化が起きるのか説明せよ。同様の例を1つ以上考え、同様の実験と考察をせよ。

ただし、授業スライド（第3日24頁・25頁）に記載の例は除く。

提出

課題を解いてレポート形式にまとめ、PDF形式の文書ファイルで mmip@takagi.inf.uec.ac.jp宛に提出すること。メールの件名は「MMIP Report #1」とすること。提出期限は2025年**5月15日 (木)** 23:59とする。レポート点は提出点（2点：0.5点刻み）と内容点（4点：0.5点刻み）からなる。提出が遅れた場合は遅れの週数に応じて提出点を減点（-0.5/週）する。





レポート課題 [前半第1回] 参考リンク



Web茶まめ (国立国語研究所)

<https://chamame.ninjal.ac.jp/>



形態素解析ウェブアプリUniDic-MeCab

解析器MeCabと辞書UniDicによる形態素解析

<http://www4414uj.sakura.ne.jp/Yasanichi1/unicheck/>



形態素解析ツールMeCab公式サイト

<https://taku910.github.io/mecab/>