

# デジタルメディア処理2

担当: 井尻 敬

井尻敬 – takashiijiri.com

**2017 - 現在 : 芝浦工大 准教授**

2017 - 現在 : 慶應義塾大学 SFC 客員研究員

2016 - 現在 : 国立循環器病研究センター 客員研究員 (申請中)

2015 - 現在 : 理化学研究所 客員研究員

**2015 - 2017 : 立命館大学 講師**

2013 - 2016 : 北海道大学 客員准教授

**2009 - 2015 : 理化学研究所 研究員**

**2004 - 2009 : 東京大学 修士/博士**

**2000 - 2009 : 東京工業大学 学士**

井尻敬 – takashiijiri.com

専門 : Computer Graphics / 画像処理 / ユーザインタフェース



モデリング

アニメーション

テクスチャ

領域分割&応用

その他

## ○ 講義の概要:

画像処理は、産業・自然科学・エンタテインメントなど、多種多様な分野の発展に関わる非常に重要な技術です。本講義では、画像処理の基本となる、フィルタ処理・幾何変換・領域分割に関するトピックを紹介します。それぞれの技術に関して、コーディング可能な深さで理解できるよう、ソースコードを交えながら詳細な技術解説を行ないます。また、講義の後半では、Pythonを用いたプログラミング演習を行ないます。

## ○ 達成目標:

1. デジタル画像 – デジタル画像の基礎を理解し、その概要を説明できる
2. フィルタ処理 – 線形/非線形/周波数フィルタなど、フィルタ処理を理解し、その内容を説明できる。
3. 幾何変換 – 画像の幾何学変換を理解し、その内容を説明できる。 B-5
4. プログラミング演習 – Pythonを用いて、簡単なフィルタ処理をプログラムとして記述できる。

## ○ 成績評価:

中間テスト (40%) と 期末テスト (30%) と 演習課題に基づき評価します。出欠も記録します。

## ○ 受講上の注意:

講義中・講義後の質問は歓迎します。(すぐに答えられない場合は私の宿題になります。)

講義中の飲食は周りの迷惑にならない程度に。

私語は周囲の迷惑にならないようオンラインで行なってください。

### ○講義資料：

講義において用いた資料・ソースコードは可能な限りWeb上に公開します。以下のURLを参考にしてください。

[takashijiri.com/classes](http://takashijiri.com/classes)

### ○質問など：

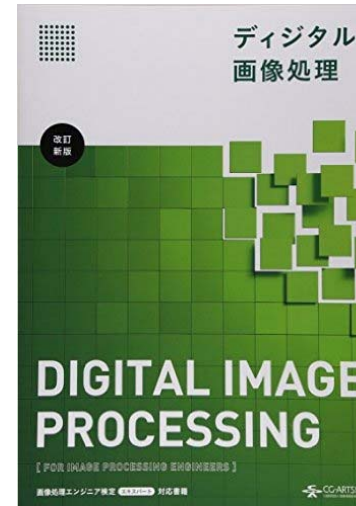
講義に関する質問があれば、講義後またはメールにてご連絡ください。

オフィスアワーは金曜日3限。

takashi.ijiri80 AtMark gmail.ac.jp

### ○シラバス：

皆さんの理解度に合わせて進度を調整します。あらかじめご了承ください。シラバスの内容は、講義が快調に進んだことを想定して作成しています。最後の2回は画像圧縮の話をやめてプログラミング演習にしてもよいけどどうでしょうか？



## 教科書

- CG-Arts協会（画像情報教育進行委員会）
- デジタル画像処理[改訂新版] 大型本
- 日本語で読める画像処理の教科書です
- 画像や例が多く入門者には最適だと思います
- 網羅性が高い反面、説明不足の部分も少しあり  
→ 講義中に丁寧に解説します

## デジタルメディア処理1、2017（後期）

09/26 イントロダクション1：デジタル画像とは、量子化と標本化、Dynamic Range

10/03 イントロダクション2：デジタルカメラ、視覚特性、色空間

10/10 フィルタ処理1：トーンカーブ、線形フィルタ

10/17 フィルタ処理2：非線形フィルタ、ハーフトーニング

10/24 フィルタ処理3：離散フーリエ変換と周波数フィルタリング

### 11/07 前半のまとめと中間試験

11/14 画像処理演習：python入門（演習室）

11/21 画像処理演習：フィルタ処理（演習室）

11/28 画像処理演習：フィルタ処理（演習室）

12/05 画像処理演習：フィルタ処理（演習室）

12/12 画像の幾何変換1：アファイン変換

12/19 画像の幾何変換2：画像の補間

01/16 画像復元：ConvolutionとDe-convolution（変更する可能性有り）

### 01/23 後半のまとめと期末試験

## ある手法を『理解する』とは？

- 教科書をおぼえた：×
- 人にその手法を説明できる：△
- 例を挙げて人に説明できる：○
- プログラムとして記述できる：◎

→ コードを書こう！

※井尻の偏見に基づきます。異論は認めます。

# Pythonプログラミング演習

以下の日程にてプログラミング演習を行ないます

- 11/14 画像処理演習：python入門 (演習室)
- 11/21 画像処理演習：フィルタ処理 (演習室)
- 11/28 画像処理演習：フィルタ処理 (演習室)
- 12/05 画像処理演習：フィルタ処理 (演習室)

場所：PC教室9, 10

言語：Python

ライブラリ：OpenCV

内容：フィルタ, フーリエ変換, ハーフトニング (予定)

# Contents :デジタル画像とは？

- ラスタ画像とベクター画像
- 量子化と標本化
- 階調数
- HDRI合成 (おまけ)
- 自己紹介 (おまけ)

## Vector Graphics と Raster Graphics



### Vector Graphics

画像を数式(スプライン等)で表現  
計算機で描いたイラスト

例 wmf/ai/cdr/cgm/dfx等



### Raster Graphics

画像をグリッド状の画素で表現  
写真/CT/MRI等の観察画像

例 jpg/png/bmp/gif/tif/等

両者を含む：pdf/DjVu/eps/pict/ps/swf/xaml等

## Vector Graphics と Raster Graphics



### Vector Graphics

制御点のみを保持するため  
データが小さい

拡大しても輪郭がスムーズ

計算機で描いたイラストに向く



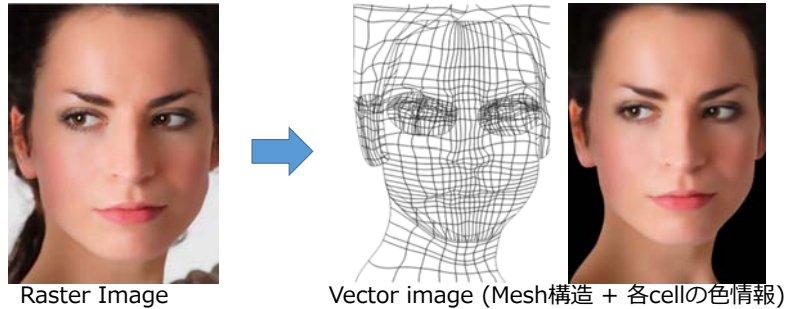
### Raster Graphics

画素情報を保持するため  
データが大きい

拡大したらギザギザ

風景など自然の画像に向く

## Raster → Vector 変換 (Vectorization)



### Gradient mesh ( Adobe Photoshop )

課題 : Raster image (写真等) を Vector image に変換したい

方法 : 画像の特徴線に沿うメッシュを構築し頂点に色情報を保持

各パッチ (四角形) で頂点の色を滑らかに混ぜる

画像の出典 [Jian Sun et. al. Image vectorization using optimized gradient meshes, TOG 2007]

## 標本化と量子化

デジタル画像とは『離散値 (画素) が格子状に並んだデータ』

アナログ情報からデジタル画像を取得するとき

**標本化:** 空間の離散化

**量子化:** 値の離散化

の必要がある

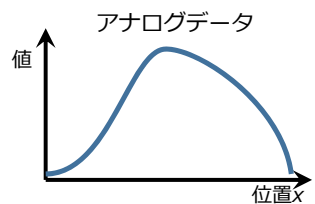


拡大すると画素(pixel)が見える

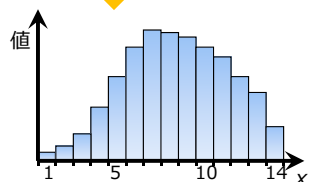


画像の出典 [©2017 Takashi Ijiri, エルサレムで撮影した猫]

## 標本化と量子化



↓ 標本化



標本点の間隔: 標本化間隔  
画像の場合, 標本点が画素に対応

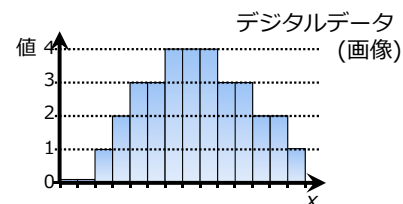
**標本化 (sampling):** 空間の離散化

等間隔の標本点を画素と呼ぶ

**量子化 (quantization):** 値の離散化

画素が保持する値の数を階調数と呼ぶ

↓ 量子化



各画素がとる値の数: 量子化レベル

## 標本化に伴うエイリアシング

### 標本化定理

周波数  $f_{max}$  に帯域制限されたアナログ信号は,  
 $2f_{max}$ 以上の周期で標本化すれば再構成可能

### エイリアシング

標本化周期が  $2f_{max}$ 以下のとき, 元信号には  
含まれない偽信号(alias)が現れる



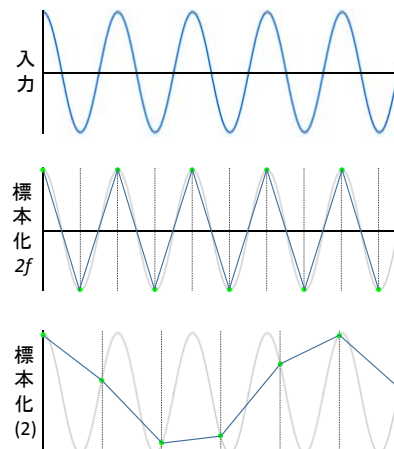
画像の出典 [Photo by Maksim / CC BY-SA 3.0]

## 標本化に伴うエイリアシング

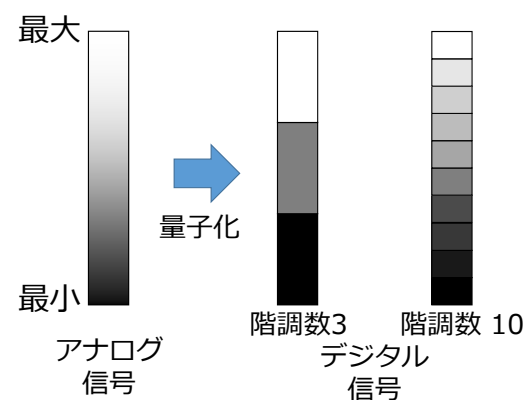
元信号が含む最大周波数が  $f_{max}$   
→ 周波数  $2f_{max}$  で標本化すれば元信号を復元可

元信号が含む最小周期が  $T = 1/f_{max}$   
→ 間隔  $T/2$  で標本化すれば元信号を復元可能

より詳しくは  
『金谷健一:これなら分かる応用数学教室』  
を参照



## 量子化レベル（階調数・画素深度・色深度）



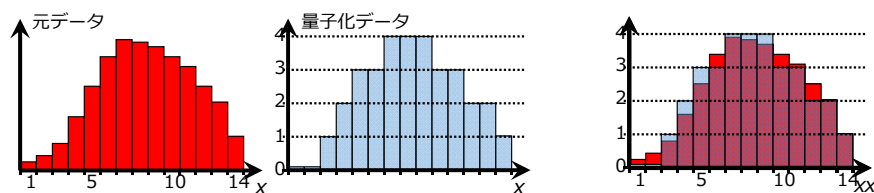
量子化レベルとは  
各画素の色数のこと  
最小値と最大値の分割数

量子化レベルが大きいと…

- 微妙な色の変化を表現可能
- データが大きくなる

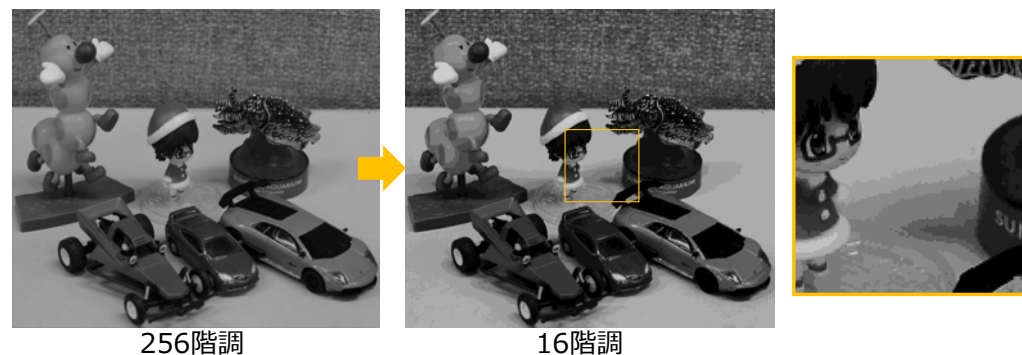
## 量子化誤差

量子化では、連続値が離散値に置き換わるので、誤差が生じる  
これを**量子化誤差**と呼ぶ



## 量子化による擬似輪郭

階調数が極端に小さい場合、疑似的な輪郭が生まれることがある  
写真ならまあ良いけど、医用画像などでは深刻な場合もありうる





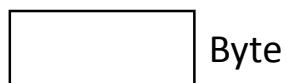
## 画像のデータサイズ（未圧縮なら）

例1) グレースケール画像

量子化レベル 8bit (1Byte) [0,255]

画像幅 W pixel

画像高さ H pixel

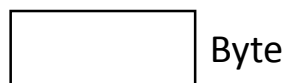


例1) カラー画像

量子化レベル RGB各色 8bit [0,255]

画像幅 W pixel

画像高さ H pixel



※これは未圧縮bmpの場合、圧縮画像の場合はもっともっとデータサイズは小さくなる。

※画像データはヘッダ情報も含むので上の値よりは少し大きくなる

## 画像フォーマットの階調数

### ビットマップ(.bmp)

1bit bitmap : モノクロ画像

4/8bit bitmap : 16/256色のカラーパレット(インデックスカラー)

16/24bit bitmap : RGB毎に 5/8-bit 階調

### Portable Network Graphics (.png)

グレースケール : 1, 2, 4, 8, 16-bit階調

カラー : 24bit (RGB毎に8bitの階調数), 48bit

インデックスカラー : 1, 2, 4, 8個のカラーパレット

### Nikon D7000(rawデータ)

14bit

### 某社 X線マイクロCTの生データ (rawデータ)

12bit階調 (階調数 = 濃度分解能)

## 練習：関数の標本化・量子化

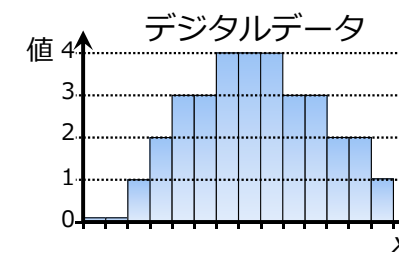
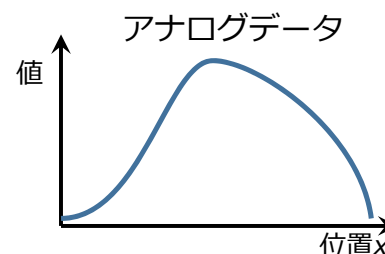
関数  $f(x) = -x^2 + 5x$  を標本化間隔1, 量子化レベル6で標本化・量子化せよ。

ただし、定義域は  $x \in [0, 5]$ , 量子化後の値は  $\{0, 1, 2, 3, 4, 5, 6\}$  とせよ。

また、関数値は画素の中心で評価し、量子化においては整数値となるよう小数点を切り捨てよ。

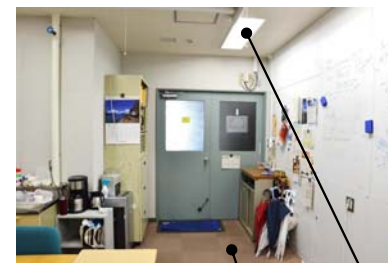
## まとめ: デジタル画像とは

『Vector graphics』『Raster Graphics』『標本化』『量子化』『量子化レベル』『量子化誤差』『擬似輪郭』について解説した。



## HDRI合成 (おまけ)

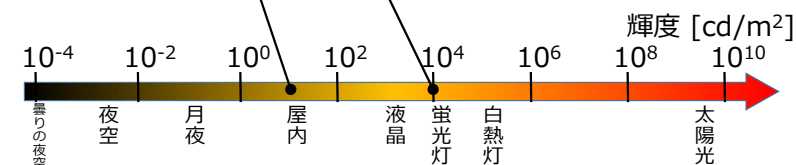
## 輝度(Luminance) - とは



人の感じる物体の明るさ

ある光源に対して

- 単位方位角あたり
  - 見かけの単位面積あたり
- 『人の感じる』明るさ



## ダイナミックレンジ - とは

信号をセンサーで計測するとき

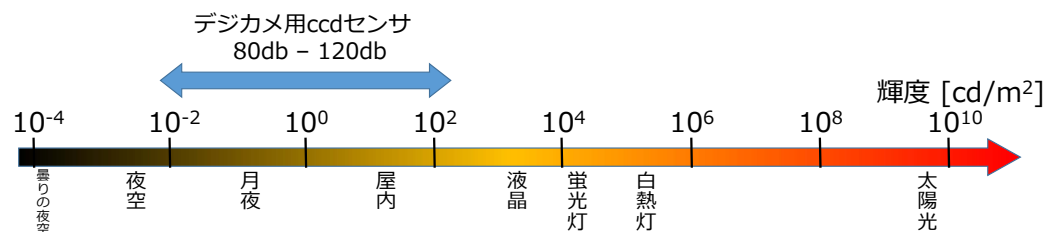
計測可能な最小輝度値 $I_{min}$ と最大輝度値 $I_{max}$ の幅のこと

$$D = 20 \log_{10} \frac{I_{max}}{I_{min}} \text{ (db)}$$

人の視覚のダイナミックレンジは

- ある視野内で100db程度
- 順応を考慮すると200db以上

[奥田: 高ダイナミックレンジ画像. 2010]



## HDRI : Motivation



露光時間 1/500 秒  
黒つぶれ (アンダー)



露光時間 1/4 秒  
白飛び (オーバー)

デジタルカメラのセンサは  
『凄く明るいところ』と『凄く暗いところ』  
を同時に撮影できない

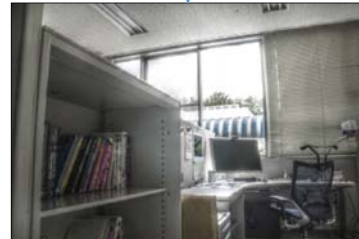
## HDRI : Motivation

『凄く明るいところ』と『凄く暗いところ』の情報を持つ画像  
を取得して…

暗いところがよく見える画像を合成したい



白飛び・黒つぶれの無い  
画像を合成したい



## HDRI : HDRIとトーンマッピング

**HDRI**



黒つぶれ・白飛びがなく  
大きな/小さな値をもつ画像

階調数(RGB毎) : 16bit  
輝度値の幅 :  $[10^{-1}, 10^4]$



表示のため階調数を落とす処理が必要  
『トーンマッピング』

**表示  
デバイス**



液晶モニタ・プロジェクタ等  
階調数(RGB毎): 8bit  $[0, 255]$

## HDRIの取得

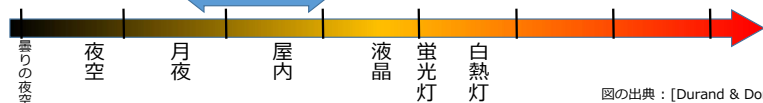
最小輝度値と最大輝度値の幅の広い画像  
露光を変えた写真の撮影から合成できる



HDRI(実数値)



階調数の大きい  
ファイルで保存する  
.hdr 32bit階調  
.exr 48bit階調



図の出典 : [Durand & Dorsey SIGGRAPH 2002]

## HDRIを自作してみる

Luminance HDR

露光時間が可変のカメラ - NIKON D7000

HDRI合成ソフト - Luminance HDR (ver 2.3.1)

1. 露光時間を変え撮影



1/500秒



1/4秒

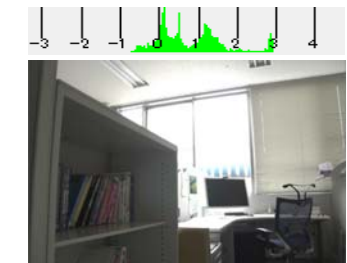


1/60秒



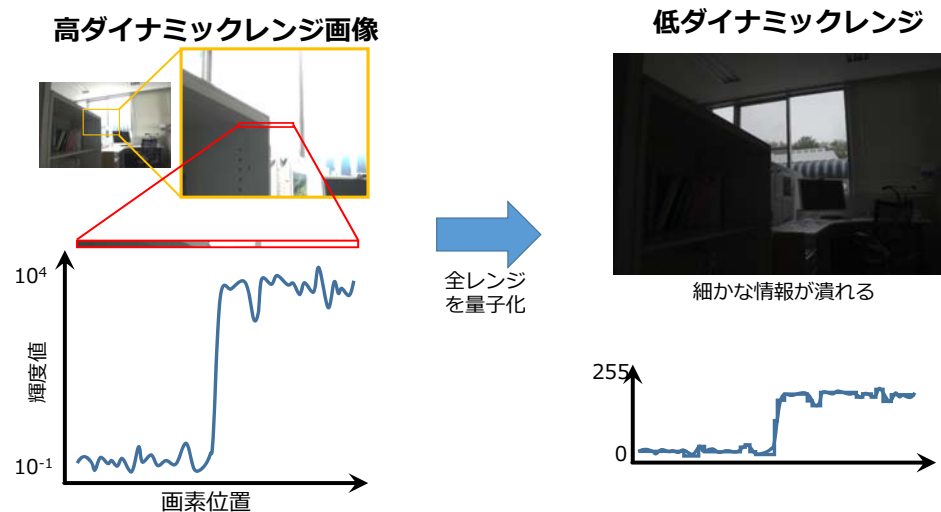
1秒

2. 一枚のHDRI画像に合成

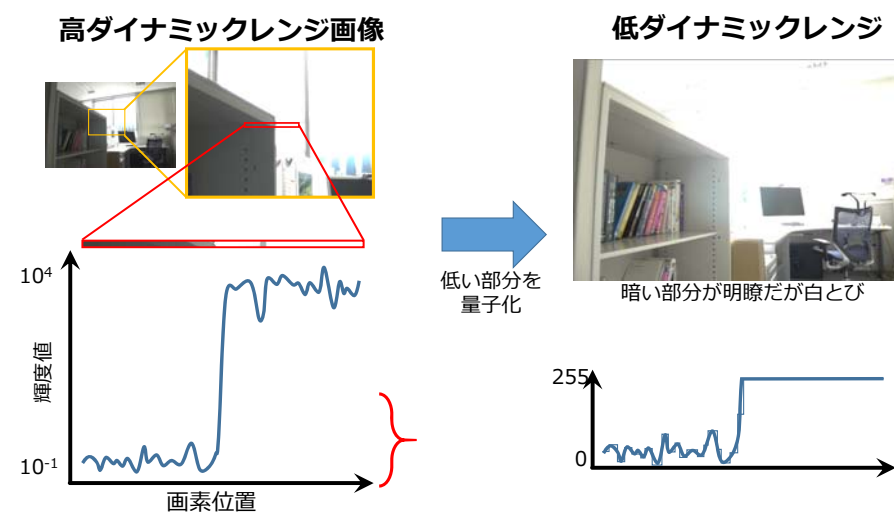




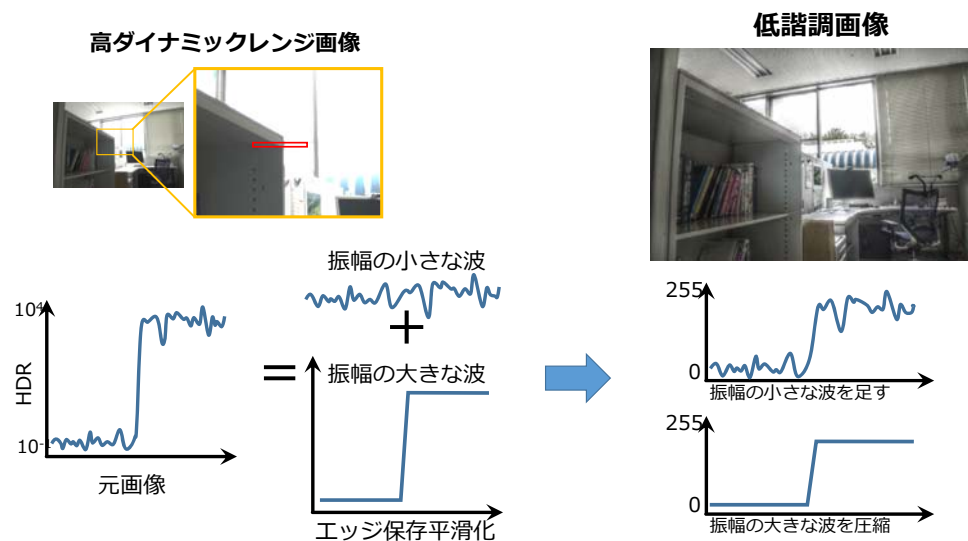
## トーンマッピング（線形）



## トーンマッピング（線形）



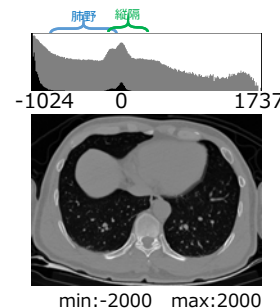
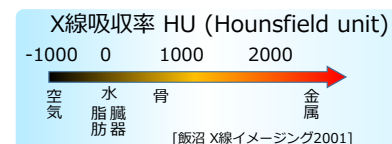
## トーンマッピング（HDR合成）



## CT画像のトーンマッピング

### CT画像

階調数 : 12 - 16 [bit]  
 レンジ : -1000 - 1500 [HU]  
 → トーンマッピングの必要有



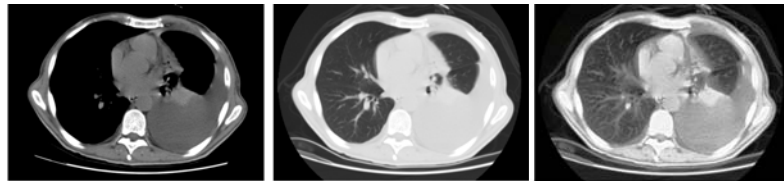
画像は理化学研究所生体力学シミュレーションチームより

## CT / MRI 画像のトーンマッピング

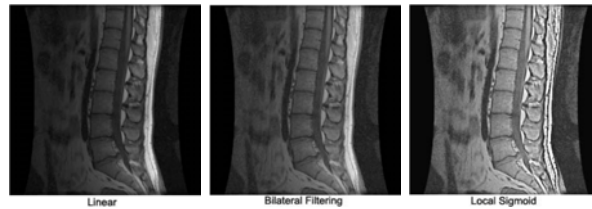
縦隔条件

肺野条件

非線形トーンマッピング

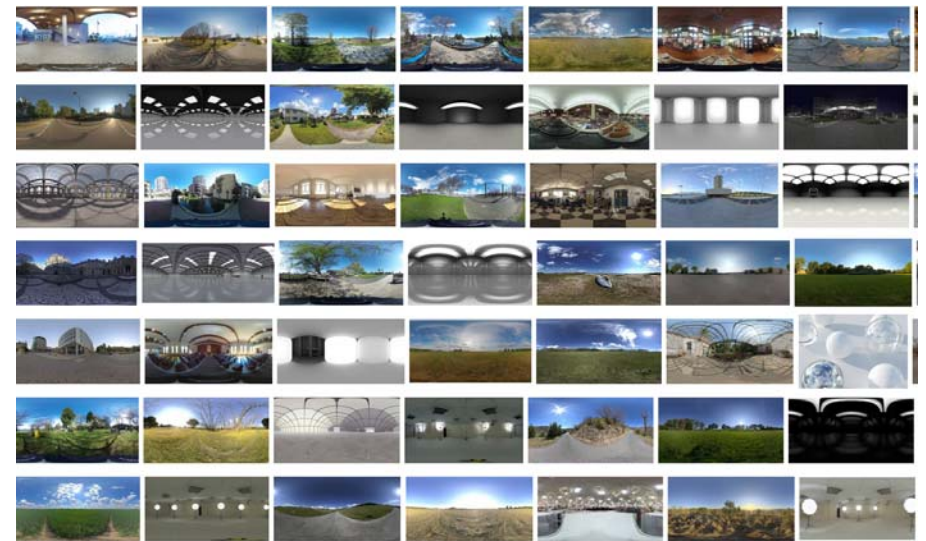


MRI  
→



画像の出典 (上) [F. Edward Boas, "High dynamic range images in radiology 2007" [www.stanford.edu/~boas/science/pub\\_list.html](http://www.stanford.edu/~boas/science/pub_list.html)]

画像の出典 (下) [Park et al. "Evaluating Tone Mapping Algorithms for Rendering Non-Pictorial (Scientific) High-Dynamic-Range Images", JVCIR 2007. ]



『HDRI』でGoogle画像検索した結果

## ハイダイナミックレンジ画像の構築(1/4)

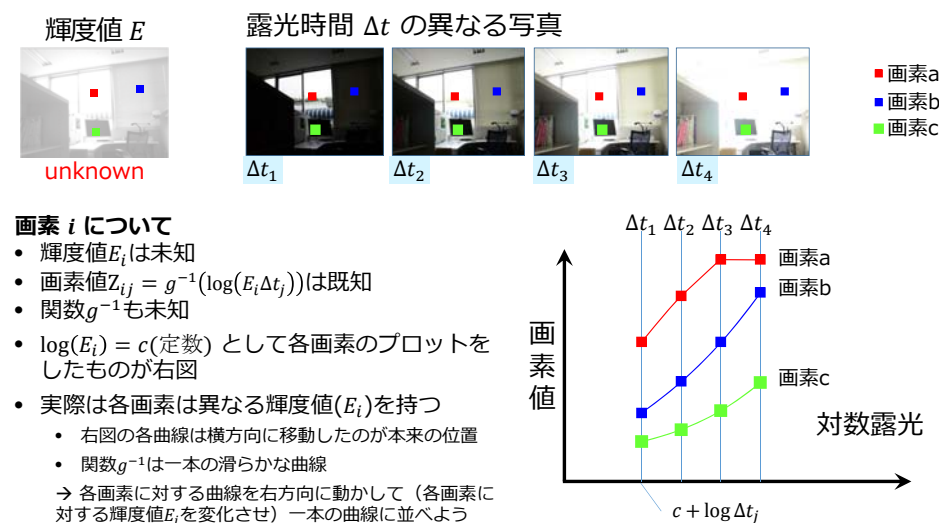


HDRI合成の補足資料  
講義内では取り扱わない予定

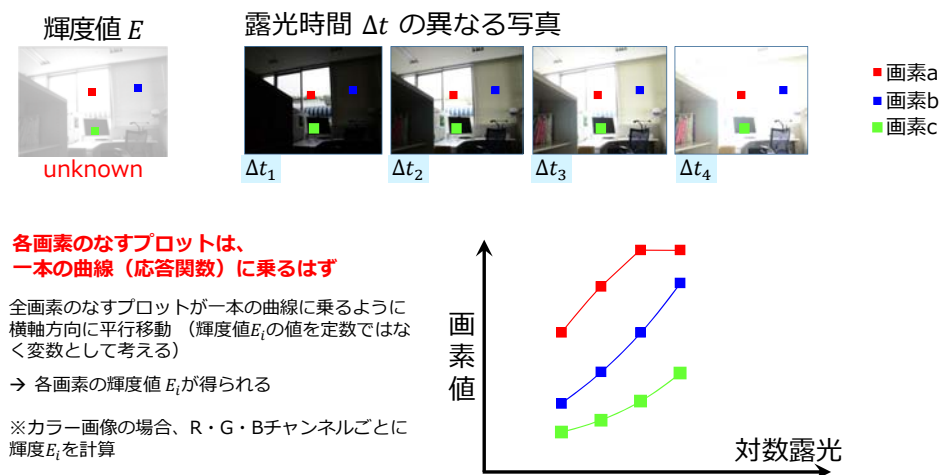
## ハイダイナミックレンジ画像の構築(2/4)



## ハイダイナミックレンジ画像の構築(3/4)



## ハイダイナミックレンジ画像の構築(4/4)



## 実装したい人のために...

### 入力(既知)

$\Delta t_j$  : 露光時間

$Z_{ij}$  : 露光時間  $\Delta t_j$  で撮影された画像の画素  $i$  の値. 8bit階調  $[0, 255]$

### 出力(未知)

$E_i$  : 画素  $i$  の輝度値. 実数値.

1) 応答関数  $g$  は対数露光と画素値の関係を表す.  $N$ :画素数,  $P$ :画像枚数

$$g(Z_{ij}) = \log E_i + \log \Delta t_j, \quad i = 1, \dots, N, \quad j = 1, \dots, P$$

2)  $g(\cdot)$  の引数  $Z_{ij}$  は  $[0, 255]$  の整数なので、 $g(\cdot)$  を以下の通り数列  $g_k$  に離散化

$$g(k) = g_k, \quad g_k \in R, \quad k = 1, \dots, 255$$

3) 以下のエネルギー関数を最小自乗法で最適化

$$\min_{E_i, g_k} E(E_i, g_k) = \sum_{i=1}^N \sum_{j=1}^P w(Z_{ij}) (g_{Z_{ij}} - \log E_i - \log \Delta t_j)^2 + \lambda \sum_{k=1}^{255} w(k) (g_{k-1} - 2g_k + g_{k+1})^2$$

$$w(x) = \begin{cases} 255 - x & x \geq 128 \\ x - 0 & x < 128 \end{cases}$$

3') この問題は、 $\frac{\partial E}{\partial E_i} = 0, \frac{\partial E}{\partial g_k} = 0$  より一次方程式 ( $Ax=b$ ) になり解ける.

平行移動の曖昧性削除のため条件  $g_{128} = 0$  が必要