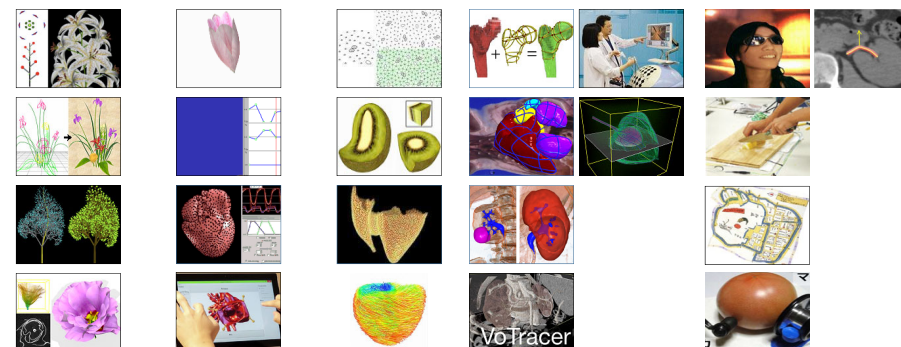


デジタルメディア処理

担当: 井尻 敬

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・フーリエ変換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、講義中に紹介した画像処理技術に関して、Pythonを用いたプログラミング演習を行ないより深い理解を目指します。

○ 達成目標:

1. デジタル画像の基礎 – デジタル画像の取得方法・データ形式に関する基礎的な用語を正しく利用できる
2. フィルタ処理 – 線形/非線形/周波数フィルタといったフィルタ処理の計算法と効果を説明できる。
3. 幾何変換 – 剛体変換やアファイン変換といった画像の幾何学変換を計算でき、その効果を説明できる。
4. 画像圧縮 – 画像圧縮 – 画像圧縮の基本的な仕組みを理解し、その内容を説明できる。
5. プログラミング演習 – フィルタ処理プログラムをPythonを用いて作成できる。

○ 成績評価:

期末テスト (45%)、演習課題 (45%)、ミニッツペーパー (10%) に基づき評価します。
期末テスト (25%)、演習課題 (50%)、小テスト (25%) に基づき評価します。

○ 講義資料:

講義資料・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。
takashiijiri.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.com

○ 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、飲み物とあめガム程度まで許可します。
講義資料閲覧やLSM利用のためのラップトップ・スマホの利用を許可します。
講義中の私語は周囲の迷惑になるため控えてください。必要場合はオンラインで行なってください。
(本学講義において私語で困ったことはないのですが、一応)

○講義資料：

講義で利用する資料・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。

URL：takashijiri.com/classes

○実施方法：

本年度の講義は、反転授業形式（事前動画 + 当日解説）にて実施します。

+ 大学のストーリーミングサービス上に動画をアップロードしscombよりURLを通知します。

+ 好きなタイミングで動画を視聴し、関連する小テストに解答してください

+ 小テストの締め切りは、実際の講義直前までとします。

※小テストは何回でも解答可

※第1回目のみ、小テスト締め切りは講義日までとします

+ 講義日当日は、zoomにて小テストの解説をし、質問があればそれに回答します

+ 質問がある場合は、メール、scomb、当日zoomで直接、の方法で連絡ください。

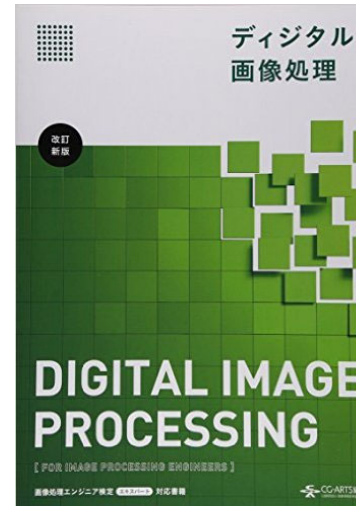
takashi.ijiri80 AtMark gmail.ac.jp

教科書（参考資料）

- 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、説明不足かなと思う部分も多少あります
- 3年前期コンピュータビジョン（デジタルメディア処理2）もこの教科書（参考資料）を利用します

注) 講義では私の用意した資料を中心に利用し、この教科書は講義中にはほとんど利用しません。講義内容やその周辺を学修するための資料として利用してください



スケジュール

| | |
|-------|---|
| 09/29 | 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 |
| 10/06 | イントロダクション2: デジタルカメラ、人間の視覚、表色系 |
| 10/13 | フィルタ処理1: トーンカーブ、線形フィルタ |
| 10/20 | フィルタ処理2: 非線形フィルタ、ハーフトーニング |
| 10/27 | 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 |
| 11/10 | 画像処理演習1: (前半の締め切り11/30 23:59, 後半の締め切り12/14 23:59) |
| 11/17 | 画像処理演習2: |
| 11/24 | 画像処理演習3: |
| 12/01 | 画像処理演習4: |
| 12/08 | 画像処理演習5: |
| 12/15 | 画像の幾何変換: アフィン変換と画像補間 |
| 12/22 | ConvolutionとDe-convolution (進度に合わせて変更する可能性有り) |
| 01/12 | 画像圧縮 (進度に合わせて変更する可能性有り) |
| 01/19 | 後半のまとめと期末試験 |

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

11/10 画像処理演習1:

11/17 画像処理演習2:

11/24 画像処理演習3:

12/01 画像処理演習4:

12/08 画像処理演習5:

方法 : zoom利用

言語等: python + OpenCV

内容 : pythonの初歩的な内容、フィルタ、フーリエ変換、ハーフトーニング
一回目に全ての課題を出題する。
毎回教員TAにて質問を受け付ける。

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。
※困ったことに、実装できて動いているけど原理の詳細はよくわからない。。。という事も実は結構あります

イントロダクション1

到達目標

- デジタル画像のデータ形式に関する基礎的な用語を正しく利用できる
- ラスタ画像・ベクタ画像・量子化・標本化・階調数・エイリアシング・ダイナミックレンジ・HDRI

Contents

- 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

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

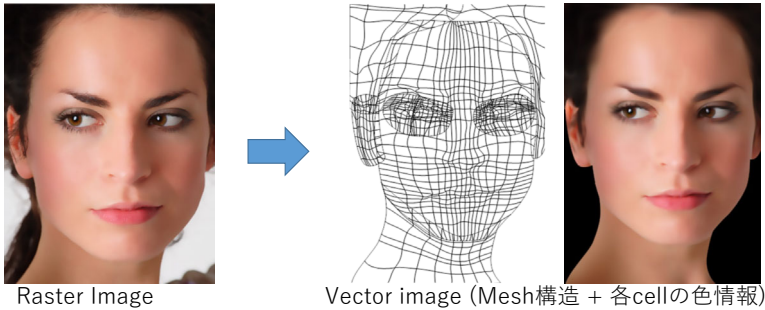
制御点のみを保持するため
データが小さい
拡大しても輪郭がスムーズ
計算機で描いたイラストに向く



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
風景など観測画像に向く

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 (ラスタ画像) 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

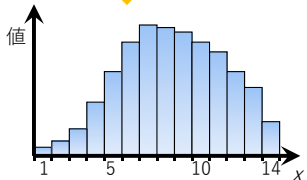
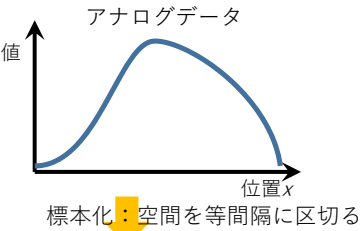
量子化: 値の離散化

の必要がある



拡大すると画素 (pixel) が見える

標本化と量子化



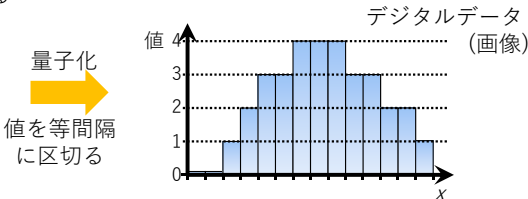
標本点の間隔 : 標本化間隔
画像の場合 , 標本点が画素に対応

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



各画素がとる値の数 : 階調数 ・ 量子化レベル

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8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 $\{0, 1, 2, 3, 4, 5, 6, 7\}$ 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.

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、 $2f_{max}$ 以上の周波数で標本化すれば再構成可能

エイリアシング

周波数 f_{max} に帯域制限されたアナログ信号を、 $2f_{max}$ より小さい周波数で標本化すると、元信号には含まれない偽信号(alias)が現れることがある



by Maksim / CC BY-SA 3.0

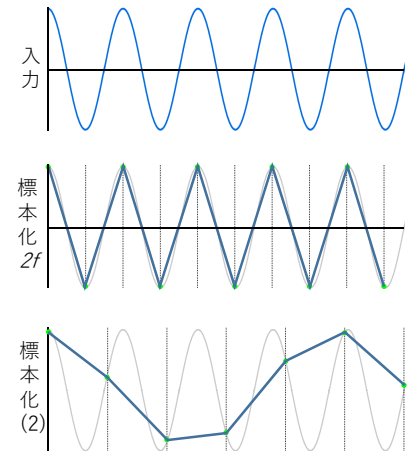
標本化に伴うエイリアシング

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

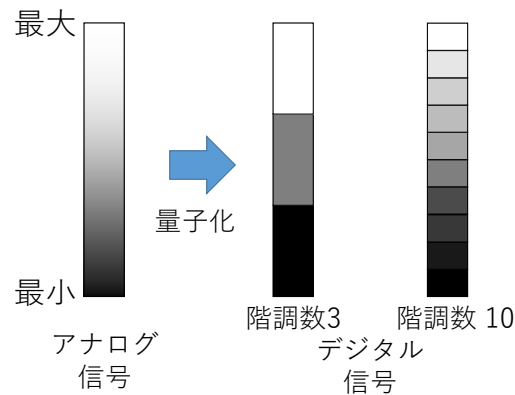
元信号が含む最小周期が $T = 1/f_{max}$

→ 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）



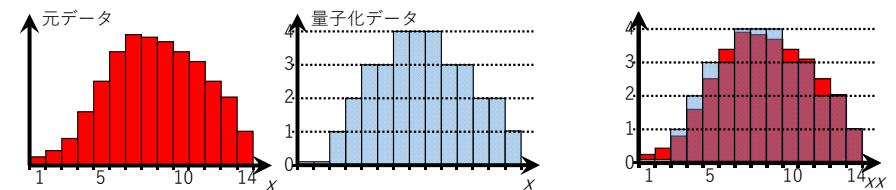
量子化レベルとは
各画素の色数のこと
最小値と最大値の分割数

量子化レベルが大きいと…

- 微かな色の変化を表現可能
- データが大きくなる

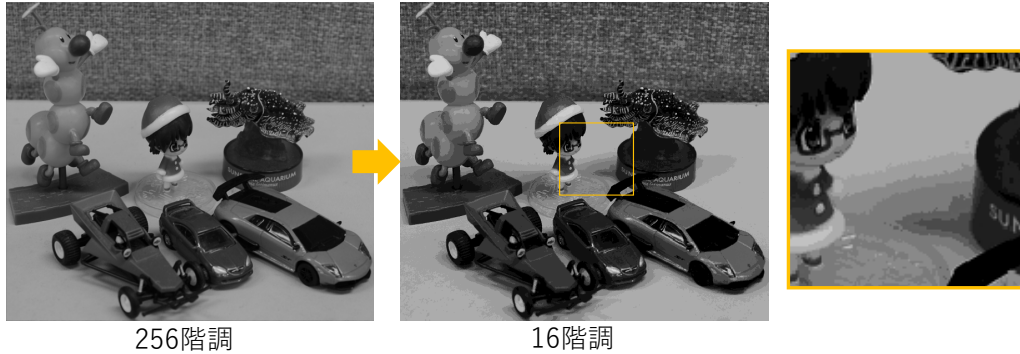
量子化誤差

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
※自然科学・医用画像などでは深刻な影響が出る場合もありうる



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2) 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)

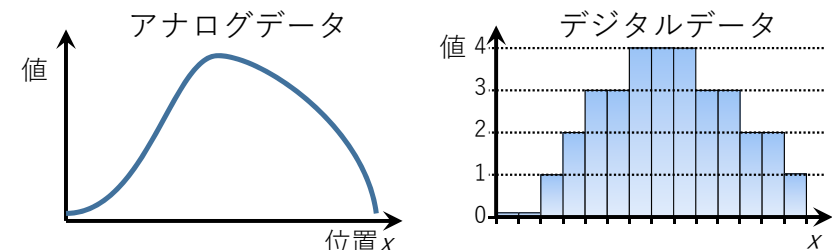
14bit

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)

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。

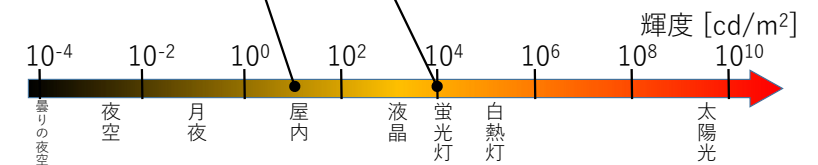


HDRI合成

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



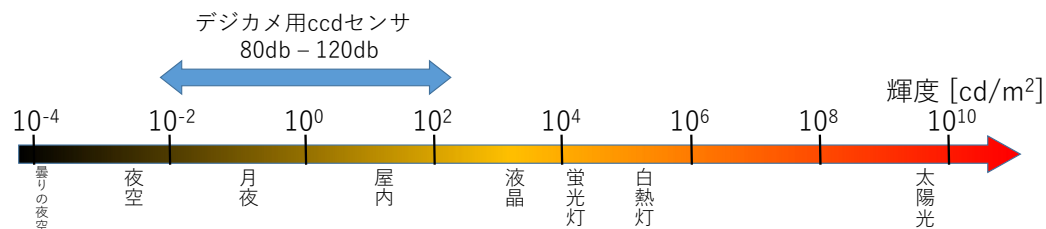
ダイナミックレンジ - とは

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

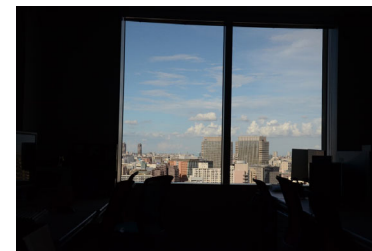
$$D = 20 \log_{10} \frac{I_{max}}{I_{min}} \text{ (db)}$$

- 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

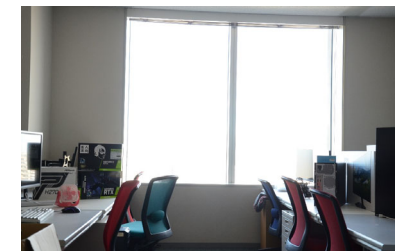
[奥田: 高ダイナミックレンジ画像, 2010]



HDRI : Motivation



露光時間 1/8000 秒
黒つぶれ (アンダー)



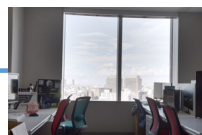
露光時間 1/100 秒
白飛び (オーバー)

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

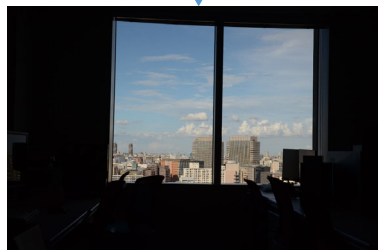
HDRI : Motivation

『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…

明るいところがよく
見える画像を合成したい

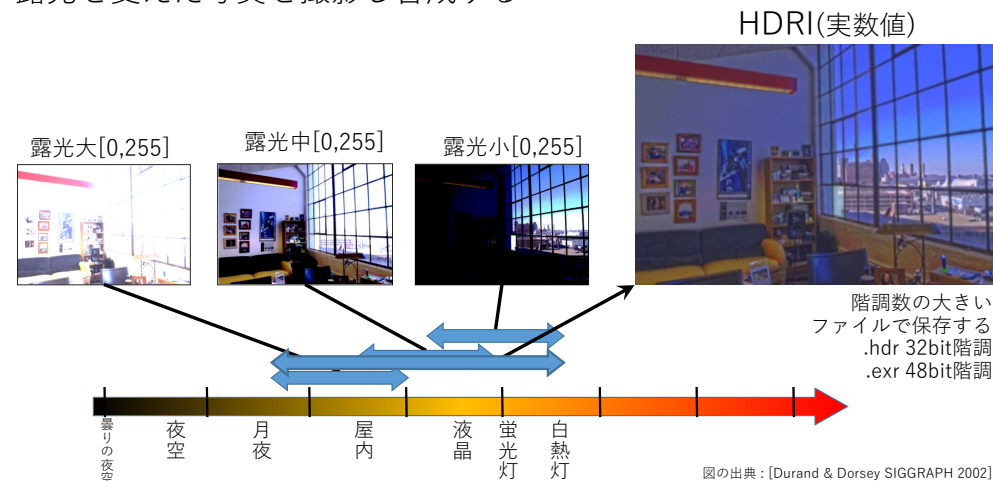


白飛び・黒つぶれの無い
画像を合成したい



HDRIの取得

露光を変えた写真を撮影し合成する



同時に2種類の露光の撮影が可能なカメラもある

→ HDR動画の撮影が可能になる

• <https://www.sony.co.jp/SonyInfo/News/Press/201208/12-107/>

HDRIを自作してみる

Luminance HDR

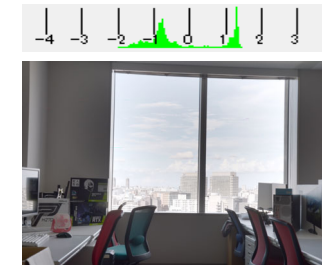
露光時間が可変のカメラ - NIKON D7000

HDRI合成ソフト - Luminance HDR (ver 2.5.1)

1. 露光時間を変え撮影

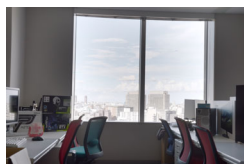


2. 一枚のHDRI画像に合成



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』

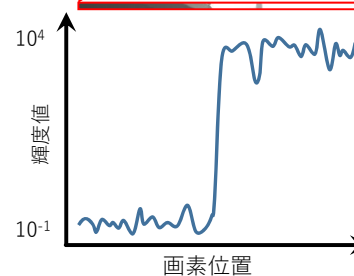
**表示
デバイス**



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

トーンマッピング (線形)

高ダイナミックレンジ画像

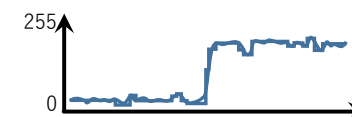


全レンジ
を量子化

256階調画像

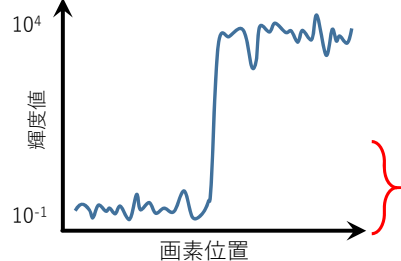


細かな情報が潰れる



トーンマッピング (線形)

高ダイナミックレンジ画像

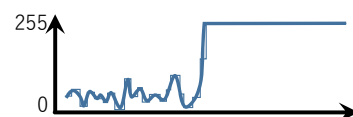


低い部分を
量子化

256階調画像

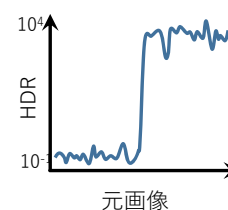
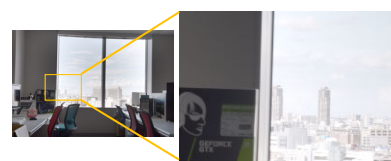


暗い部分が明瞭だが白とび



トーンマッピング (HDR合成)

高ダイナミックレンジ画像



振幅の小さな波

+

振幅の大きな波

=

エッジ保存平滑化



256階調画像



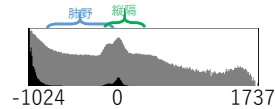
振幅の小さな波を足す

振幅の大きな波を圧縮

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



min:-2000 max:2000

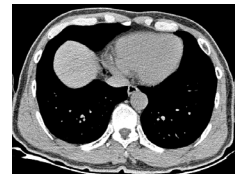
肺野条件

Min : -1200.0 Max: 0.0
Window level : -600 (中心)
Window size : 1200 (幅)



縦隔条件

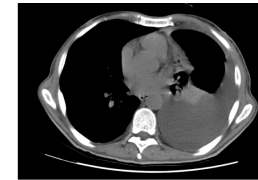
Min: -110.0 Max: 190.0
Window level : 40
Window size : 300



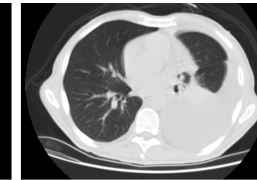
画像は理化学研究所生体力学シミュレーションチームより

CT / MRI 画像のトーンマッピング

縦隔条件



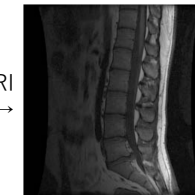
肺野条件



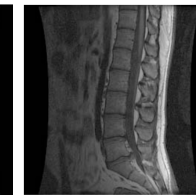
非線形トーンマッピング



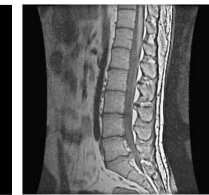
MRI
→



Linear



Bilateral Filtering

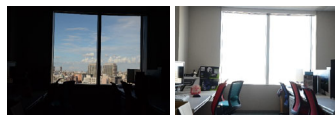


Local Sigmoid

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

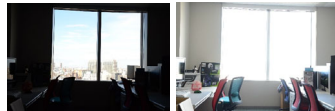
画像の出典 (下) [Park et al. "Evaluating Tone Mapping Algorithms for Rendering Non-Pictorial (Scientific) High-Dynamic-Range Images", JVCIR 2007.]

まとめ : HDRI合成



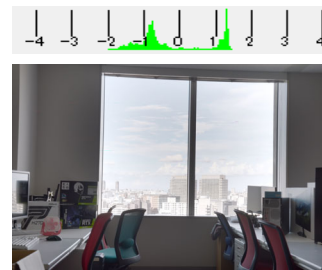
1/8000秒

1/100秒



1/1250秒

1/50秒



- ダイナミックレンジの広い（黒つぶれ白とびのない）写真を取得する方法
 - 露光の異なる写真から合成できる
- トーンマッピング：表示用にHDRIの階調数を落とす処理のこと

豊洲キャンパスから撮影したスカイツリー
HDRI合成結果



ハイダイナミックレンジ画像の構築(1/4)



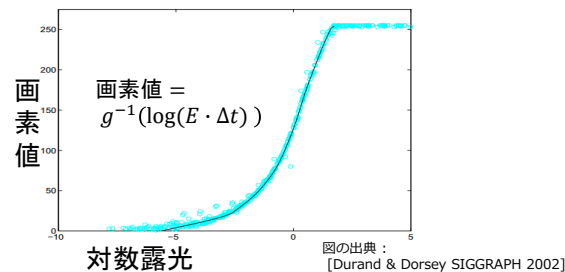
HDRI合成の補足資料
講義内では取り扱わない予定

ハイダイナミックレンジ画像の構築(2/4)

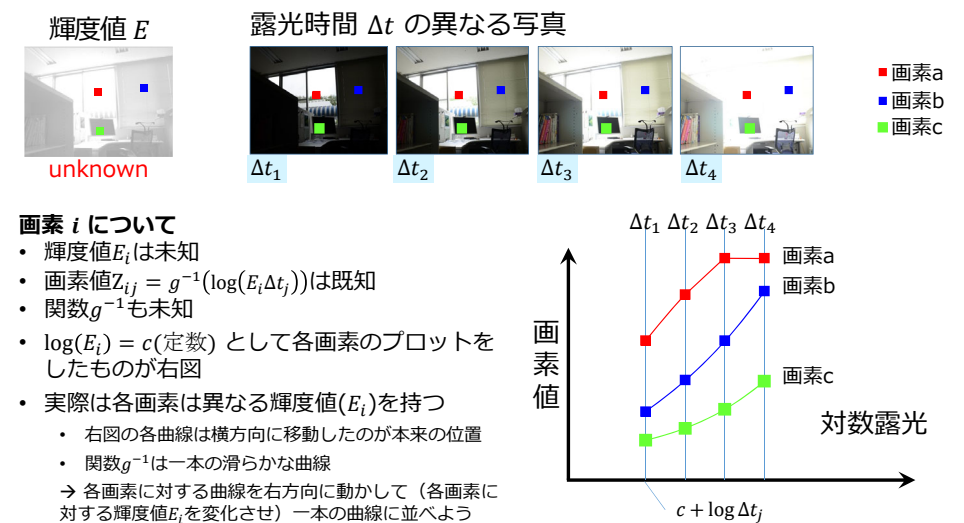


応答関数 g^{-1}

『対数露光 $\log(E \cdot \Delta t)$ 』と
『画素値』には、非線形の
関係がある

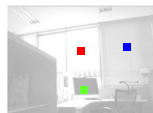


ハイダイナミックレンジ画像の構築(3/4)



ハイダイナミックレンジ画像の構築(4/4)

輝度値 E

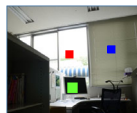


unknown

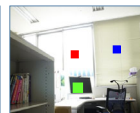
露光時間 Δt の異なる写真



Δt_1



Δt_2



Δt_3



Δt_4

- 画素a
- 画素b
- 画素c

各画素のなすプロットは、
一本の曲線（応答関数）に乗るはず

全画素のなすプロットが一本の曲線に乗るように
横軸方向に平行移動（輝度値 E_i の値を定数ではなく
変数として考える）

→ 各画素の輝度値 E_i が得られる

※カラー画像の場合、R・G・Bチャンネルごとに
輝度 E_i を計算

