

1. フィルムを使って撮影するアナログカメラに対し、フィルムの代わりにデジタルイメージセンサを使うようになったデジタルカメラは、その本質的な撮影方法は変わっていない。

Q1-1. アナログカメラとデジタルカメラの撮影原理をそれぞれ比較し「本質的な撮影方法は変わっていない」ことを説明せよ。

被写体は周囲の光を受け 360 度に光を反射している。(図1)そして記録媒体はすべての角度から記録している。(図2)ピンホールカメラにおいて記録媒体が 1 ピクセルごとにたくさん配置され画像を生成する。図1の様に記録

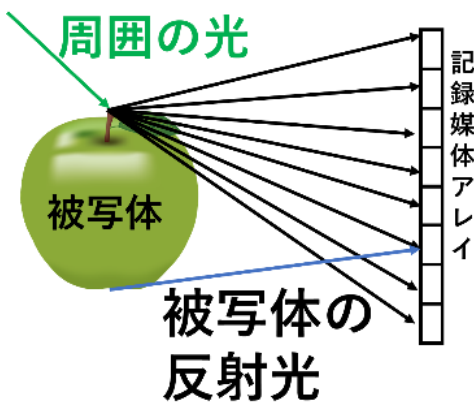


図 1 光の経路

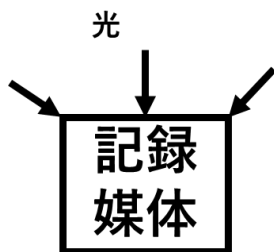


図 2 記録する光

媒体をたくさん並べていくと反射した光がすべての記録媒体に光が入ってしまう。また記録媒体1つではいろんな角度の光を記録していて画像に変換できない。そのため被写体の光を被写体と記録媒体の間に目隠し板を入れて小さい穴(開口)に通して 1 ピクセルごとの記録媒体に少料の光を記録している。

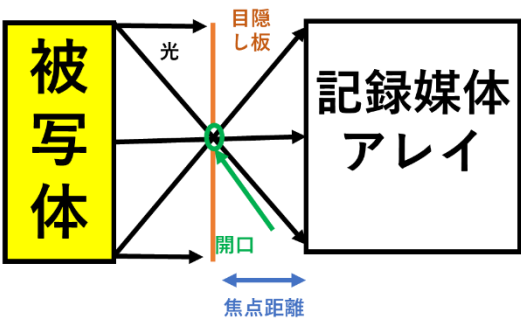


図 3 ピンホールカメラの原理

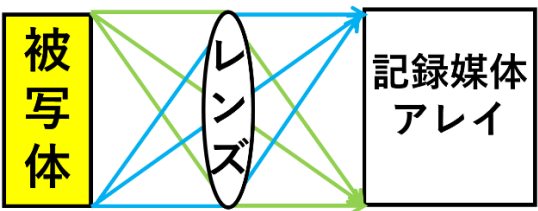


図 4 レンズカメラの原理

また目隠し板によって記録媒体に入る光を減らせる。開口は小さいほど像がボケない。しかし光量が多いためノイズが少なくなる。(図3)そして焦点距離を延ばすと暗く記録媒体に光を写す。そのためレンズを被写体と記録媒体の間に入れてレンズを通して光が 1 つの記録媒体に集約して入るようにしている。記録媒体においてアナログカメラではフィルムが使われている。そしてデジタルカメラではイメージセンサが使われている。(表 1)そのためデジタルカメラとアナログカメラの違いは記録媒体にある。また撮影の原理は根本的には変わらない。

表 1 記録媒体の比較

	アナログカメラ	デジタルカメラ
記録媒体	フィルム	イメージセンサ

Q1-2. コンピュータショナルフォトグラフィは、アナログカメラでは実現が不可能な、デジタルカメラで撮影されたデジタルデータだからこそ実現できる様々な新しい技術が開発されている。コンピュータショナルフォトグラフィの技術の例を挙げてその仕組みを詳述し、なぜアナ

ログカメラでは実現できず、デジタルカメラでは実現できるのかを述べよ。  
 コンピュータショナルフォトグラフィの技術に関してリフォーカシングがある。

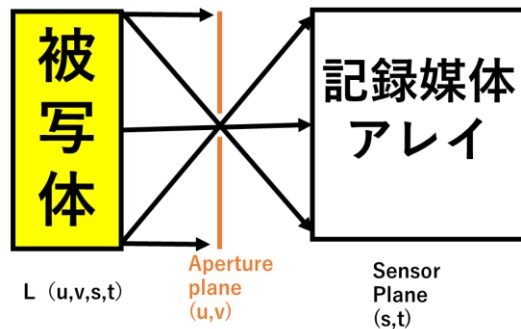


図5 ライトフィールドの概要

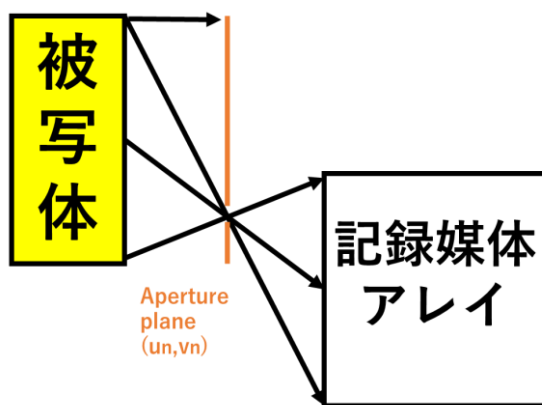


図6 4次元のライトフィールド

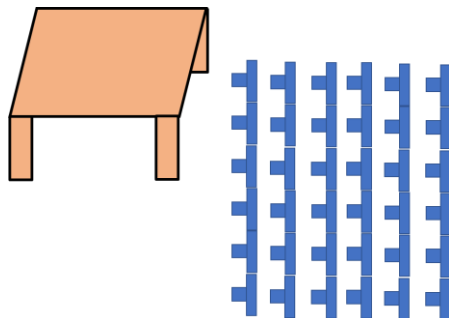
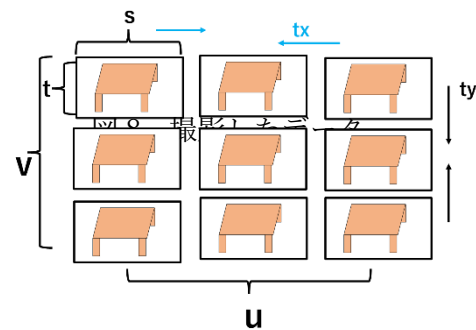


図7 4次元撮影

ライトフィールドは  $L(u,v,s,t)$  の4次元画像である。(図5)ピンホールカメラで  $u,v$  は Aperture plane の穴における位置の座標である。(図8)また  $s, t$  はカメラが撮影したデータの縦、横の長さである。そしてフィルムカメラではライトフィールドの2次元スライスを記録のみしかできない。または Aperture plane でいろんな角度の光を集めると4次元のライトフィールドが作成できる。(図6)4次元のラ



イトフィールドを撮影方法は図7の様にカメラを並べて撮影する。そして撮影した画像は図8の様に真ん中の写真に合わさるように位置を  $ty, tx$  だけずらして輝度値の平均値を合成結果の輝度値とする。この  $tx$  と  $ty$  の長さを調節することでリフォーカシングできる。そしてこのリフォーカシングはアナログフィルムカメラだと輝度値の平均値に調整できない。

2. 連続して短い露光時間で撮影した画像列を加算して1枚の画像にすることで、演算処理によって擬似的に長い露光時間で撮影した画像を生成することが出来る。例えば、ビデオカメラで撮影された動画像を、連続して短い露光時間で撮影した画像列と見なし、これらの画像列を加算(正確には画像枚数で割り算、つまり平均画像を求めることで白飛びを防ぐ)することで、動画像から1枚の長時間露光画像が生成される。

Q2-1. 横断歩道を撮影した動画像[1]から生成した長時間露光画像を[2]に示す. この画像の特徴を挙げ, なぜこのような画像が生成されるのかその原理を説明せよ.

図8においてカメラの露光時間が長くなる場合にはシャッターが開いている間に被写体で

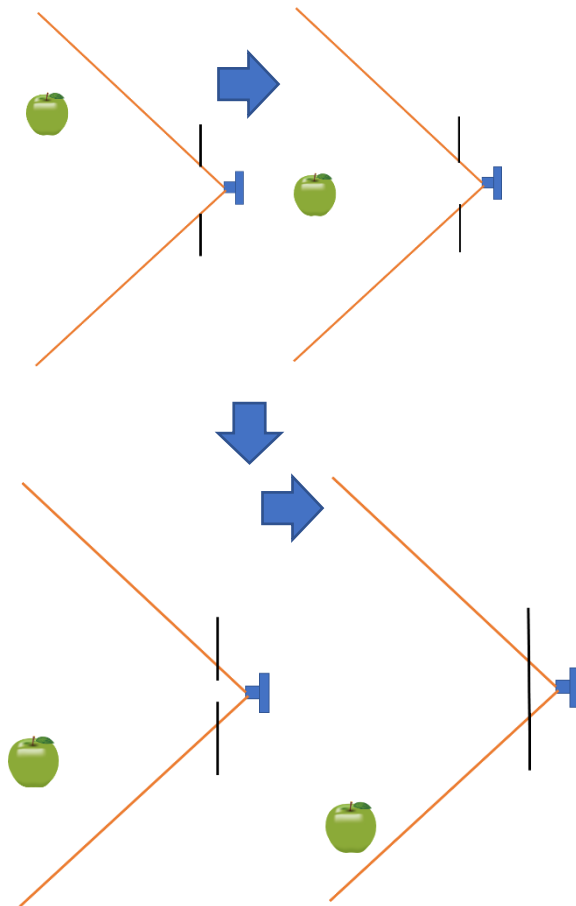


図8 動作ボケの仕組み

あるリンゴが動くと記録媒体に写る被写体はボケが生じる。これはリンゴの発している光が動いているため周りよりずれてかつ周りより少なく記録される。よって図9の様に現象を行うと見える。そのため露光時間が長いときにシャッターが開いている間に被写体が動くとボケが発生する。また被写体の光が動いていない周りのものより少なくなり薄く見える。



図9 動作ボケの撮影結果 (イメージ図)

Q2-2. このような長時間露光画像を生成するプログラムを作成し, ソースコードを示せ.

OpenCV で動画像[1]を読み込むこともできるが, 簡単のために連番のファイル名をつけた静止画も用意した[3]ので, これらを入力として使用しても構わない. なお, プログラムは Python(Google Colaboratory)で作成することを推奨するが, 他の方法を用いても構わない.

PNG ファイルの読み込みをした際 RGB の値が1つずつと4つ目の値があった。この値のすべての時間画像の RGB の平均輝度値を求めて生成した。

```

import numpy as np
import cv2
from matplotlib import pyplot as plt
from tqdm import tqdm

#画像数
img_times=158

#画素数
height=720
width=1280
picture=np.zeros([img_times,height,width,4])

# 画像を読み込
for count in tqdm(range(img_times)):
    if(count<10):
        name_str='00'+str(count)
    elif(count<100):
        name_str='0'+str(count)
    else:
        name_str=str(count)
    filename = 'crossing/img'+name_str+'.png'
    img = cv2.imread(filename, -1)
    picture[count]=img
    picture[count,:,:]=picture[count,:,:].astype(np.float32)/255.0

#出力画像の格納
out_img=np.zeros([height,width,4])

#画像の合成
for height_count in tqdm(range(height)):
    for width_count in range(width):
        out_img[height_count,width_count,0]=np.average(picture[:,height_count,width_count,0])
        out_img[height_count,width_count,1]=np.average(picture[:,height_count,width_count,1])
        out_img[height_count,width_count,2]=np.average(picture[:,height_count,width_count,2])
        out_img[height_count,width_count,3]=np.average(picture[:,height_count,width_count,3])

#0~255 までの表記戻す
img_8bit=256*out_img

#0 未満を 0、255 より大きい数を 255 にする
img_8bit=np.clip(img_8bit,0,255)

#データの型を uint8 にする
img_8bit=img_8bit.astype(np.uint8)

#ファイル書き込み
cv2.imwrite('output.png',img_8bit)

```

図 10 露光時間が長い写真の生成プログラム

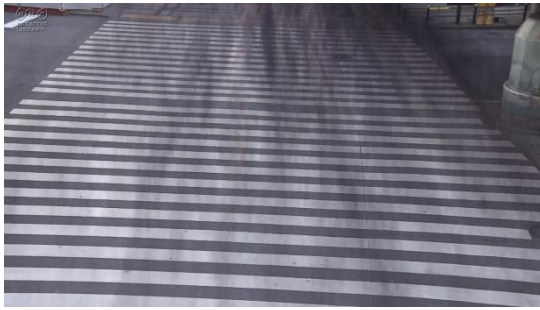


図 11 生成した PNG ファイル



図 14 生成した写真

Q2-3. 自分で動画像を撮影し, 同様の方法で長時間露光画像を生成せよ. OpenCV で動画像を読み込む方法は記事[4]を参照すること.

E T ロボコン 2019 に出た時のロボットのテストの様子を長時間露光画像として撮影した。今回使用した動画は 30fps の動画であった。そして動画から 1 秒間に 3 枚の画像を抽出した。これには 47 秒の動画を使用したため 141 枚の画像の RGB の平均輝度値を取った。(図 14) この動画はロボットがブロックを動かす様子をチェックしているものである。

図 12 のような時と時図 13 の時がありこの時の露光が残っている。



図 12 47 秒目の写真



図 13 19 秒目の写真