

# 任意の方向からの物体認識用訓練データ生成手法

— 動画とCGを活用した2つのアプローチ —

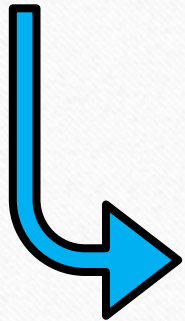
静岡理工科大学情報学部

尾関 政朋, 桑田 彩華, 工藤 司

## 研究の背景

物体認識用の訓練データを生成するには**手間**がかかる

- 2つの手法を比較評価



- 回転台を用いた画像生成手法
- CGによる画像生成手法

## 研究の背景

- ・ホームセンターでレジのアルバイトをしている中で

「レンガの判別が難しい」

という問題を感じていた

その要因として

**レンガの種類が多さ**が挙げられる

・何十種類もあるレンガを全て把握しておくのは難しい



分からないレンガが来たときには.....

**売り場までレンガを確認しに行く**



売り場

20~30m

レジ

・時間がかかるため、お待たせしてしまう



クレームに繋がることも



レンガ全てにJANコード記載のシールを張れば良いのではないか？

- 1つ1つシールを張るのは負担が大きい

- レンガ同士の擦れや雨などによって剥がれたり溶けたりしてしまう



そのような問題を解決するために...

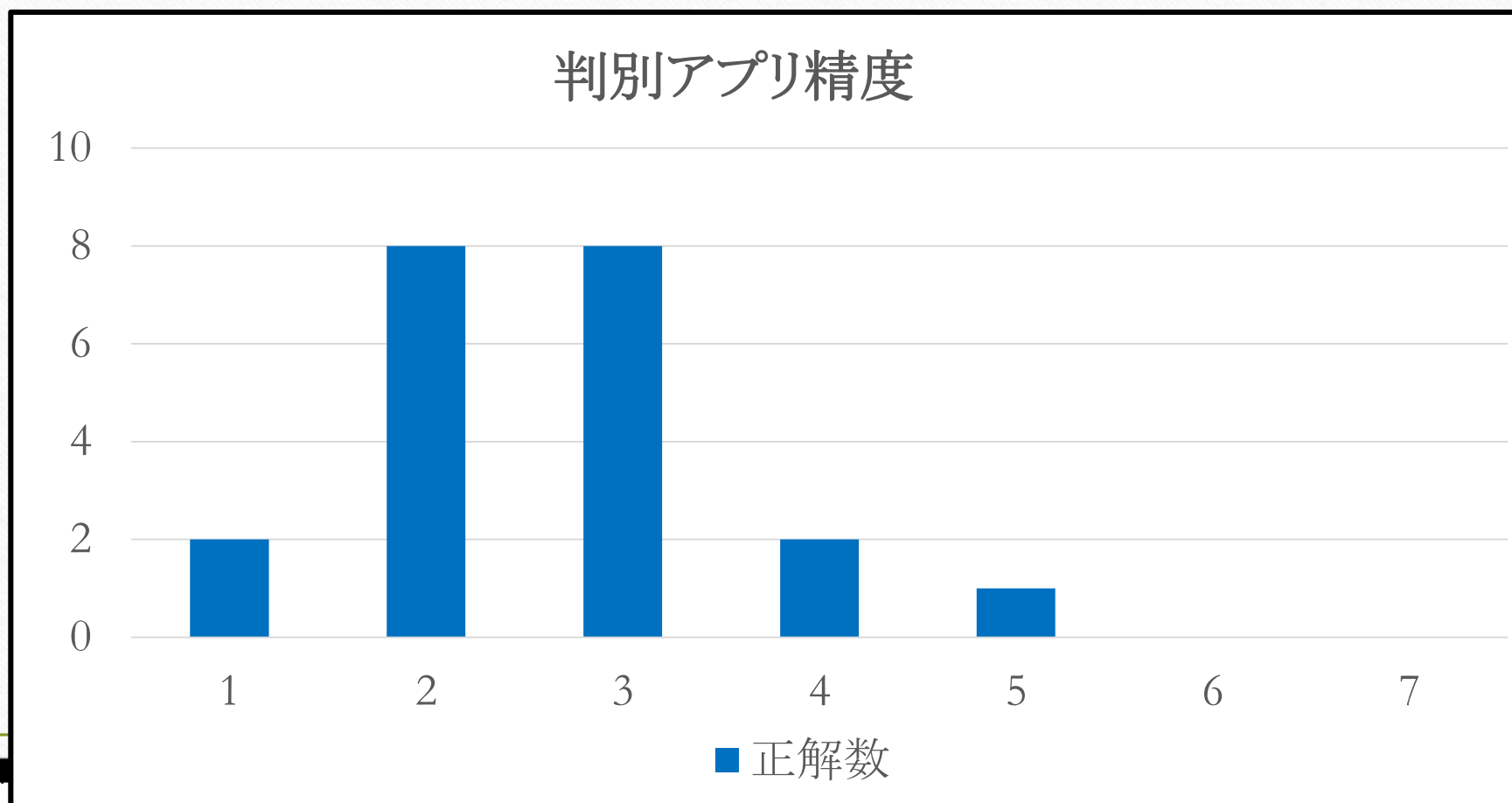


## レンガを識別するアプリが存在

- レンガを撮影すると自動的に判別をする

# しかし、判別の精度が不安定

- ・7つのレンガをそれぞれ10回テストした



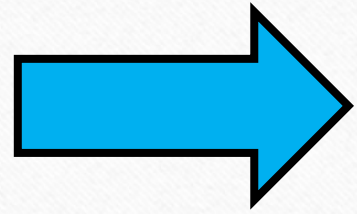


加えて、**天候の変化によって精度が変わる**

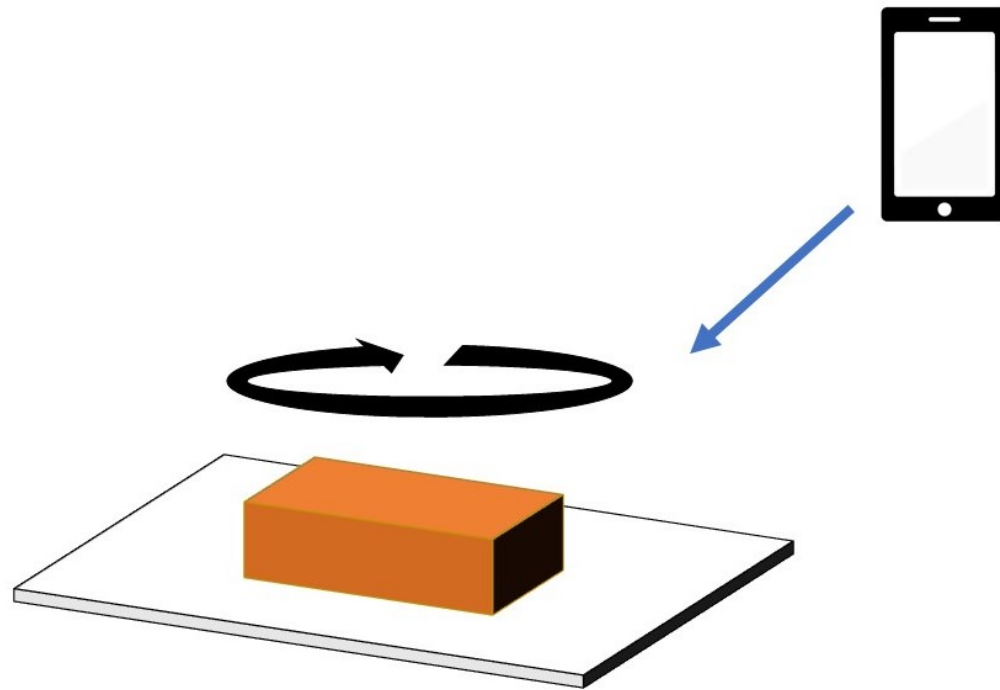
- ・同じレンガでも状態によって精度がバラバラ



精度を上げる方法はないだろうか？

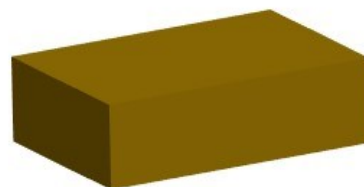


**深層学習**を用いた判別システム



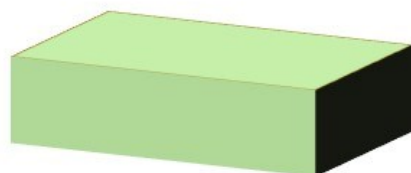
# 目標とするシステム

- ・ どの方向からでも判別できる



- ・ 天候ごとの訓練データの整備

晴れバージョン



雨バージョン



# 対象にしたレンガ



しかし、深層学習には**大量の画像データ**が必要

- 1つのレンガ50枚を手作業で撮影：4分46秒

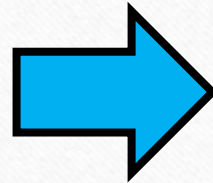
➡ 非効率的



さらに、**多方向から**撮影した画像が必要

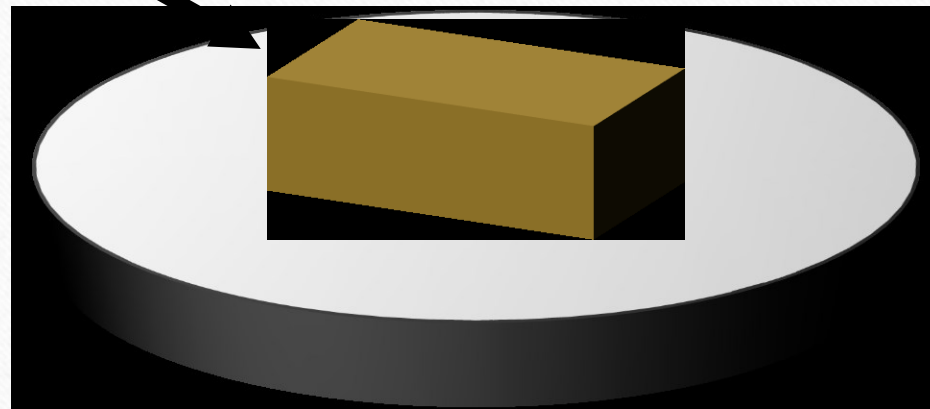
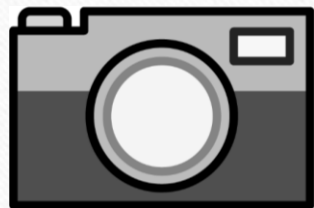


上から見た場合



横から見た場合

# 回転台を使用して多方向から撮影した レンガの動画を収集



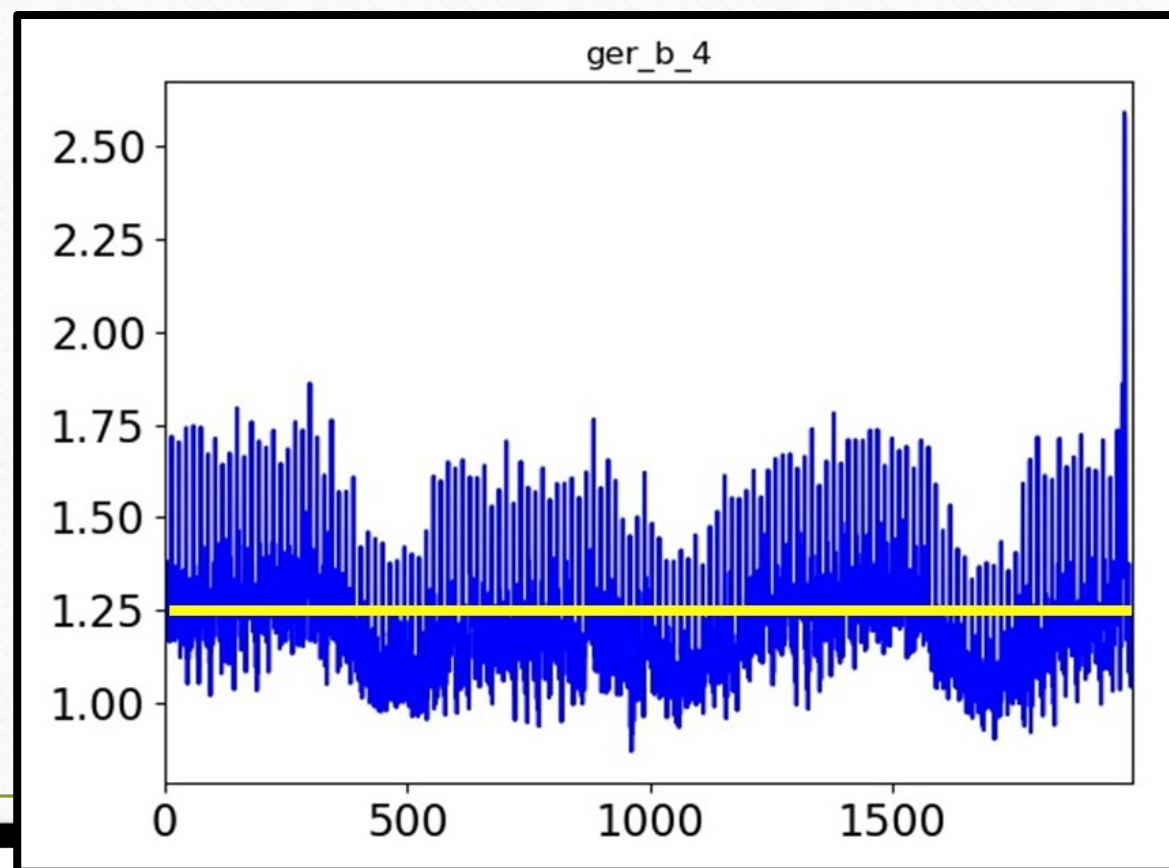
# 撮影の様子





# 撮影した動画から画像を抽出

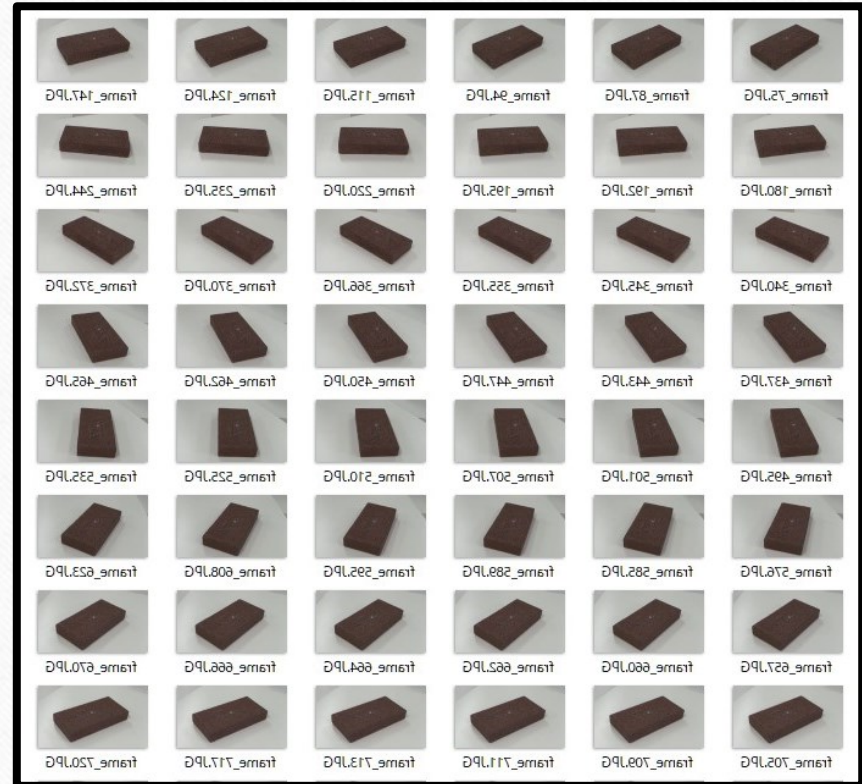
- ・指定した閾値以下の区間ごとの最小フレームを抽出



# 多方向から撮影した**大量の画像**を **自動的に生成**することが出来た



# 左右反転した画像で水増し



乾いた状態の晴れバージョン:10,064枚



濡れた状態の雨バージョン:10,080枚



約60分で訓練に必要な画像を生成できた

## 生成した画像を使用してモデルの訓練

75%を訓練データ

(その内の90%:訓練データ、10%:検証データ)

25%をテストデータ

※晴れ・雨バージョン共に同じ条件

## 訓練後のモデルで判別テスト

- 検証データとテストデータによる判別精度



訓練データと同一環境の画像

- 別環境の画像による判別精度



訓練データとは別環境の画像

# 環境を変えるとレンガの見え方がかなり違ってくる



訓練画像と同じ環境



訓練画像とは別環境

## 検証データを使用した精度

- ・晴れバージョン・雨バージョンどちらも  
**100%の精度**で判別がされた

モデルの評価：ベストモデル  
loss= 1.9614707009494015e-08  
accuracy= 1.0

実画像晴れバージョン

モデルの評価：ベストモデル  
loss= 0.0  
accuracy= 1.0

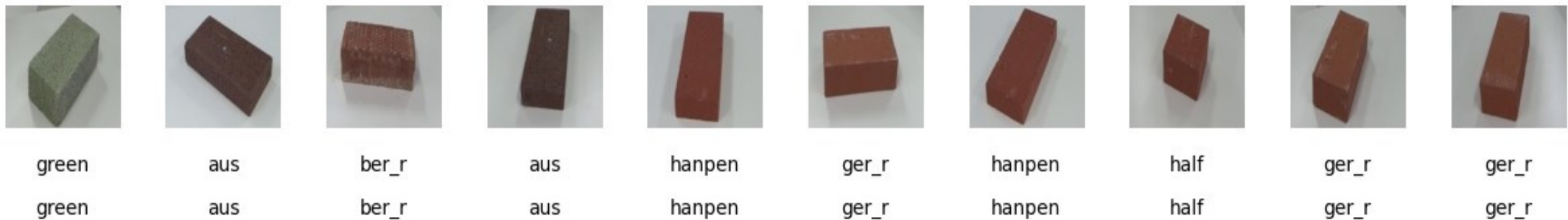
実画像(雨バージョン)



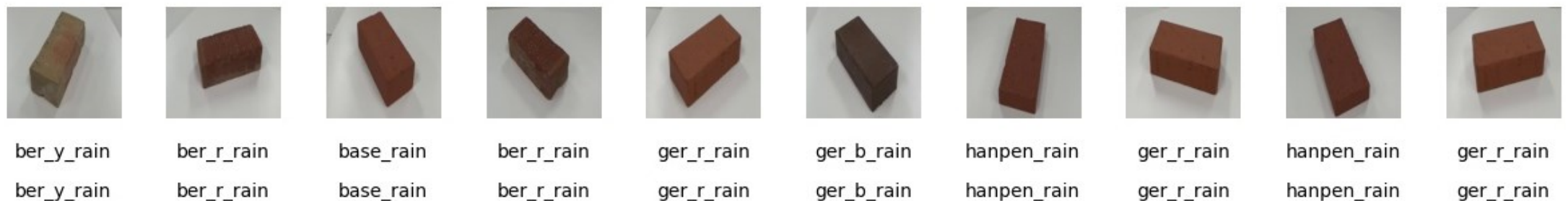
# テストデータを使用した精度

- ・晴れ,雨どちらも**100%の精度**で判別がされた

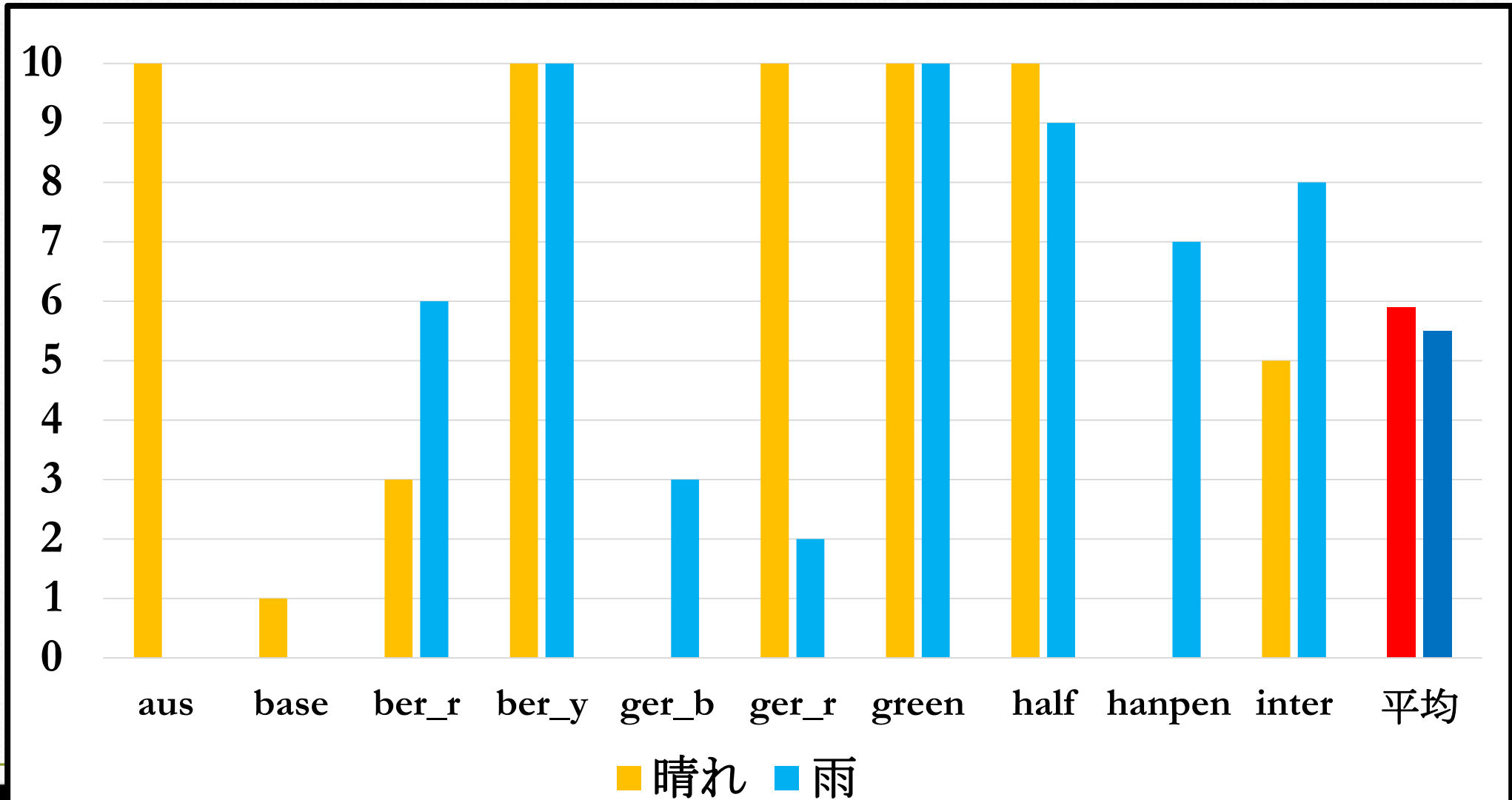
## 晴れver



## 雨ver



# 別環境の画像を使用した判別精度



## 判別結果まとめ

- 同じ環境であれば100%の精度で判別できる
- 一方で環境が変わってしまうと、  
精度は半分くらいまで劣化してしまう

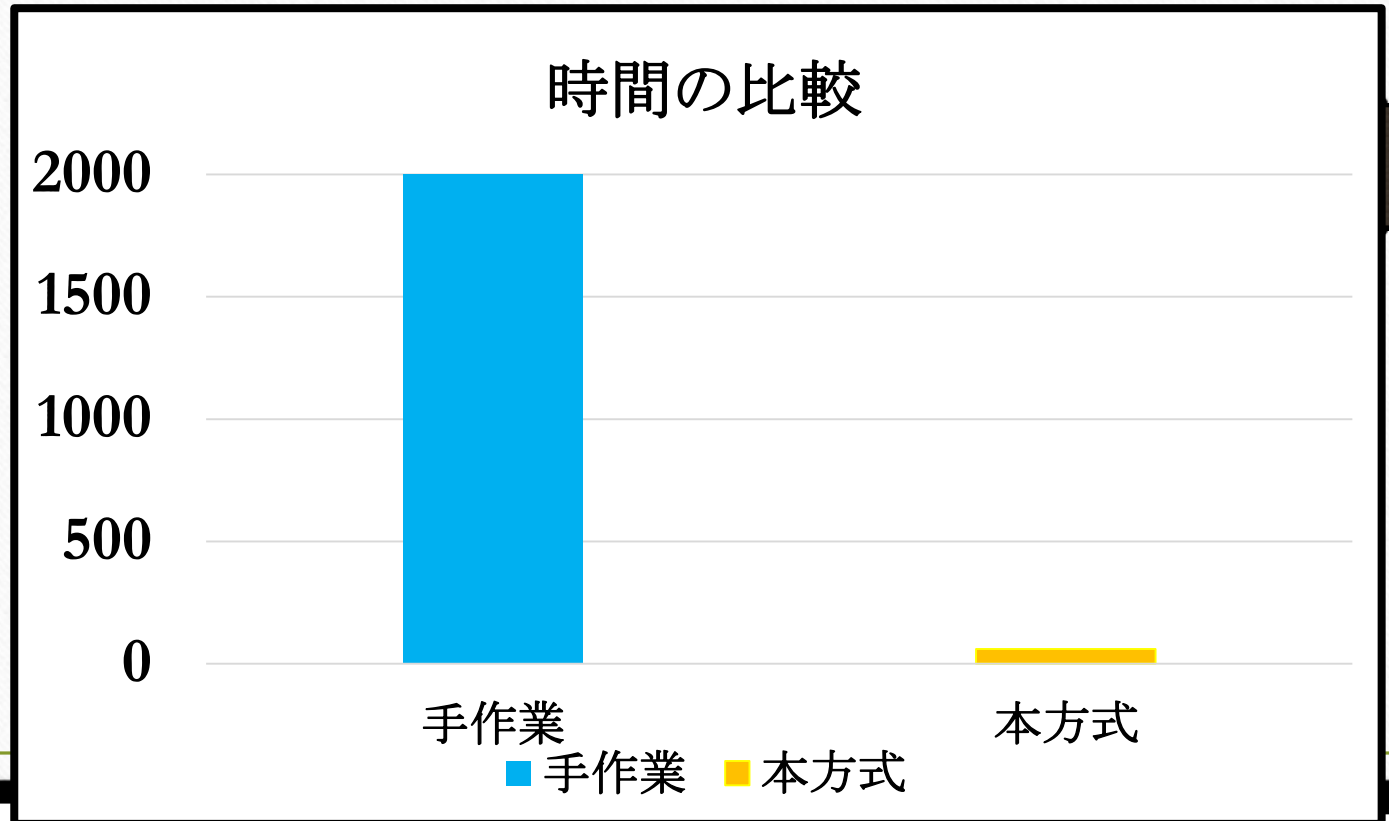
# 考察とまとめ

- ・回転台を用いた本方式は、手作業で撮影する方法の33倍効率的に画像を生成することができた

・それぞれ2万枚を撮影したのにかかる時間を計算

・手作業2000分(約33時間)

・本方式60分

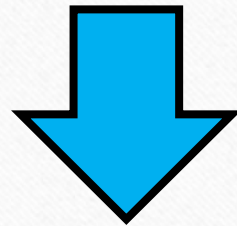


検証データを使用した判別テストでは100%の精度  
で判別がされた



- 回転台を用いて生成した画像を使ったからといって、  
精度が悪くなったということはない
- 本方式は大量の画像を効率的に生成できるだけでなく、  
有用性のある画像も生成できると考える

- 本方式により得られた画像で訓練を行ったところ、同じ環境の画像であれば、100%の精度で判別ができることが分かった



- 撮影環境を一定にすることが出来れば、現状の問題点を解決する1つの方法として提案できると考えている

## 研究背景・課題

- 近年、深層学習を利用した画像認識の研究が行われている  
しかし、学習させるためのデータを集めることが大変である



データをCGで作ることはできないか

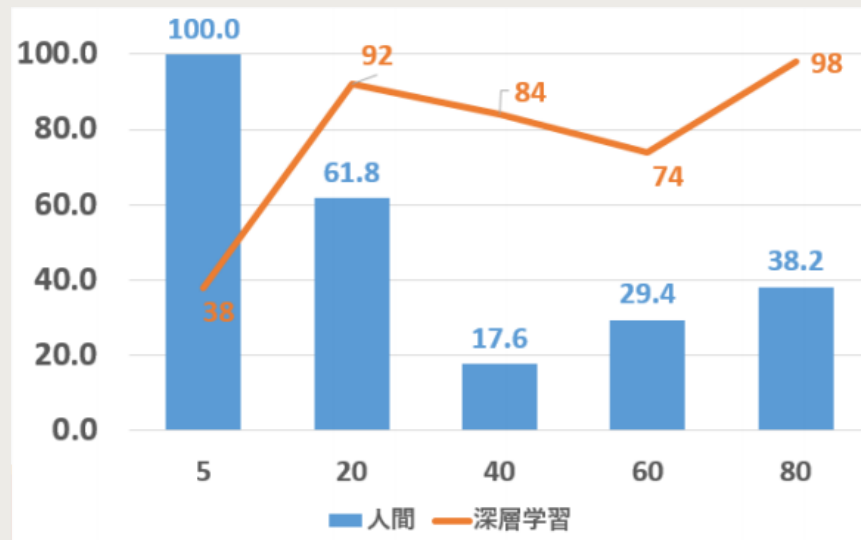
## 先行研究

### 深層学習を用いた在庫数量判定

使用対象：ビー玉



例：ビー玉60個



精度評価

数量が増えるに従い人間に比べて高い判定精度になる



## 先行研究

### 訓練データの作成時間

訓練データの作成枚数 = 1,600枚

作成方法 ⇒ 手作業

作成時間 ⇒ 25時間



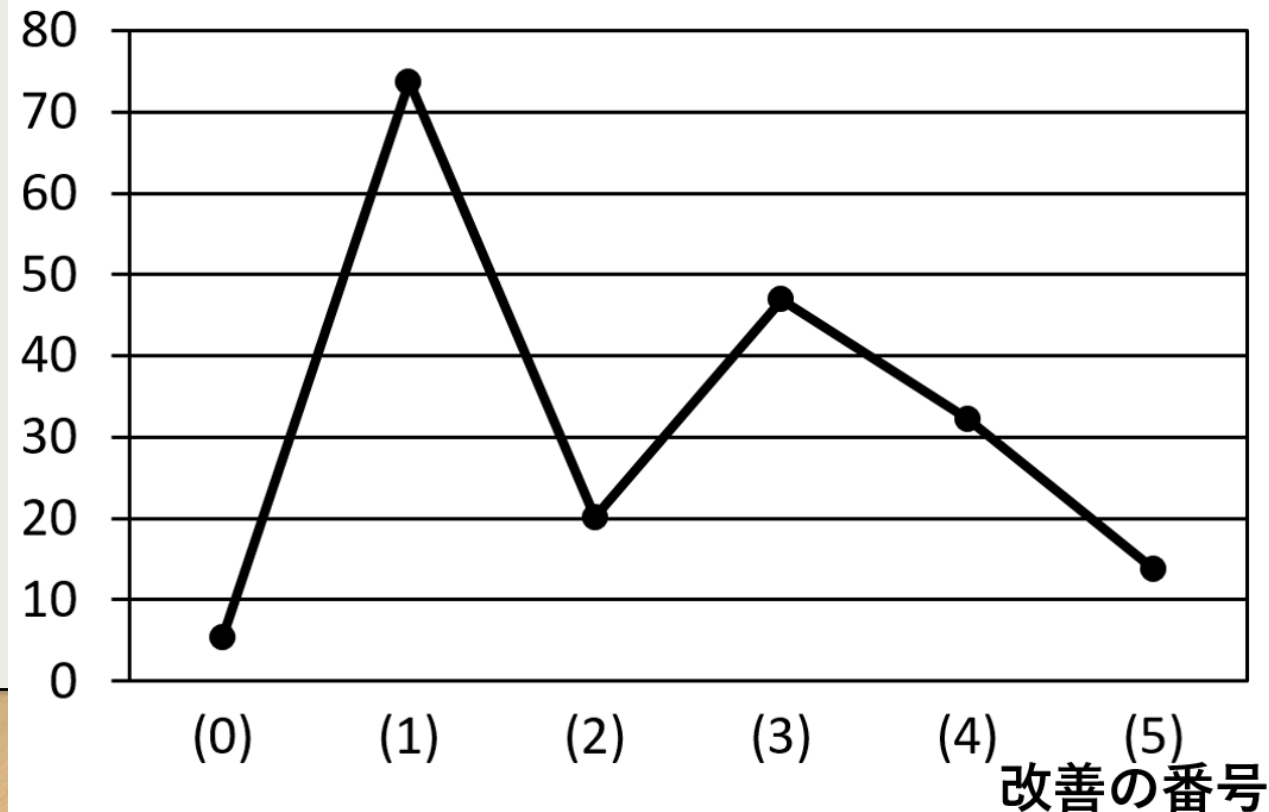
学習データを3DCGにすることでデータ作成時の負担を軽減できないか



## 先行研究

### CG画像で訓練した場合の数量判定精度評価

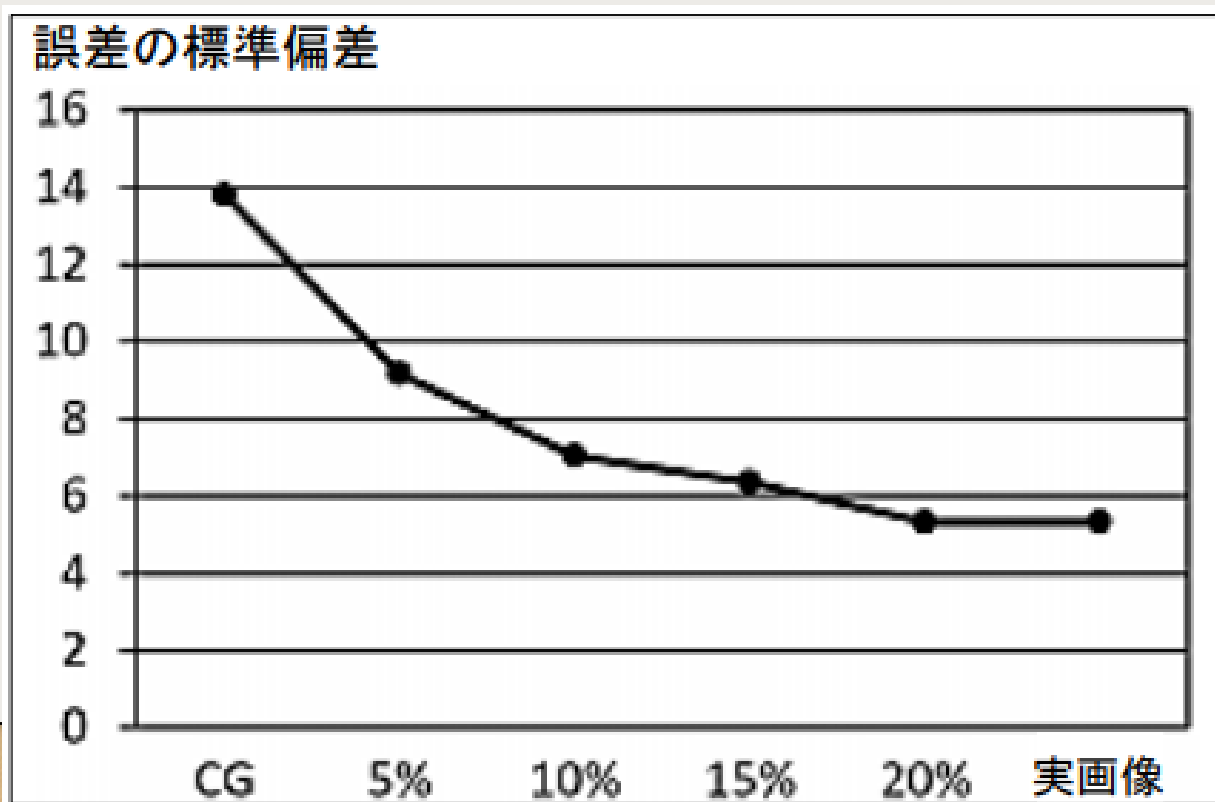
#### 誤差の標準偏差



- (0) 実画像
- (1) ラフな画像
- (2) 位置の改善
- (3) 照明の改善
- (4) 材質の改善
- (5) 全改善反映

## 先行研究

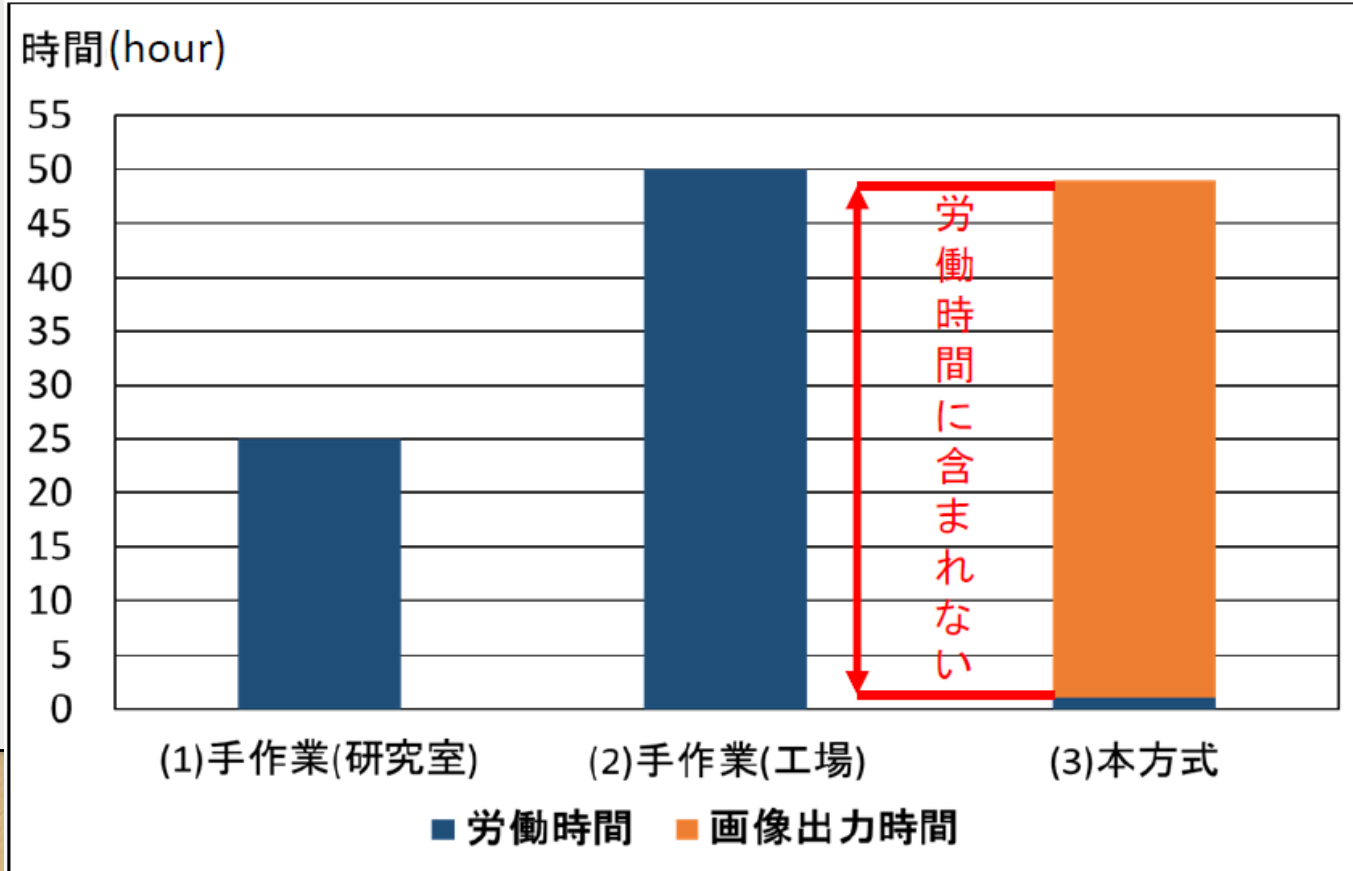
### CG画像の一部を実画像に置き換えた場合の精度



20%混ぜたとき実画像と同等の精度

# 先行研究

## 時間効率の評価



1/50の作業時間の短縮ができた

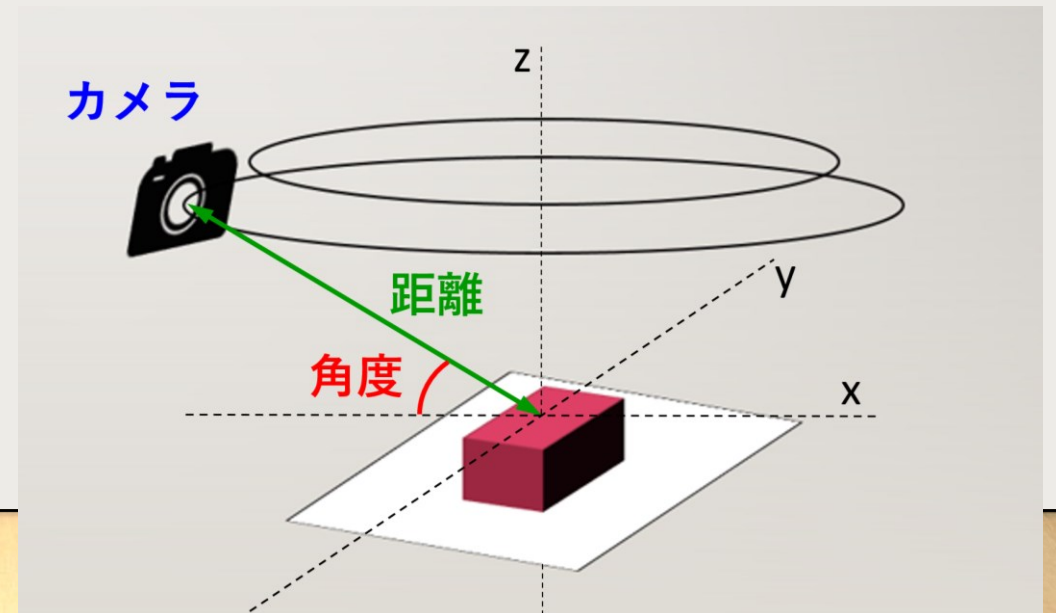
## 研究の目的

- 一方向からでは認識できないものがある→立体のもの
- 立体のものを、任意の方向から認識できるようにするために、多方向からの訓練データをCGで作成する



## 研究の提案

- カメラが物体のほうを向きながら、円上を回ることによって画像を撮影する
- さらに、円を少しずつ動かす  
→ 様々な角度のデータを得ることができる



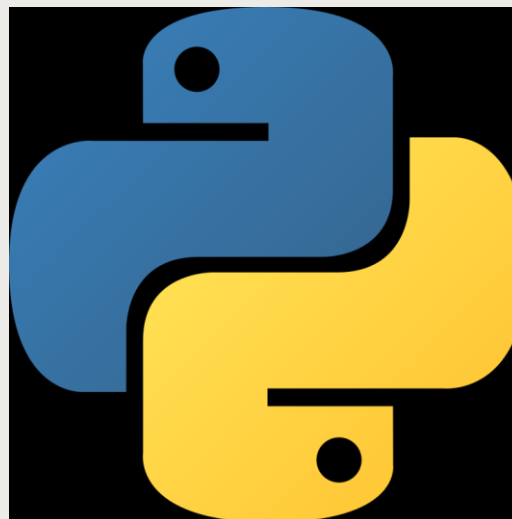
## 本研究の概要

- 「回転台を用いた画像生成手法」  
に基づいて実験を行う(晴の場合だけを対象とする)
- レンガ10種類を斜めから撮影する
- 環境は均一であり対象物を変えて撮影する



## 実装

- 使用3DCGソフトウェア 「Blender」
- 使用する言語 Pythonスクリプト





41

## 実装手順

環境作成



カメラの設定



オブジェクト  
の作成



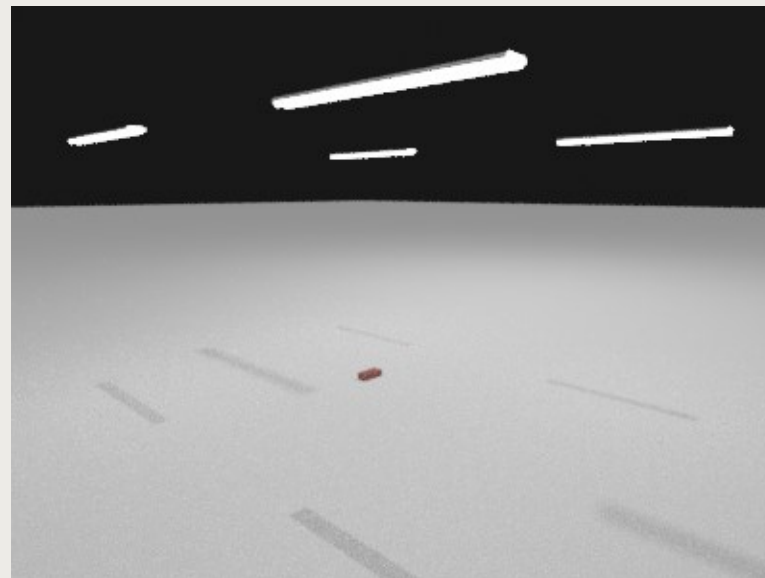
画像データの  
作成

## CG上の環境

- 環境は均一であり対象物を変えて撮影する→環境は先に作っておく



実際の環境



CG上の環境

43

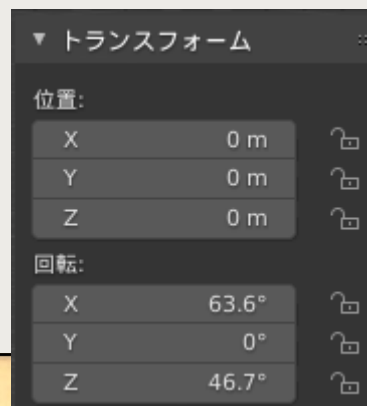
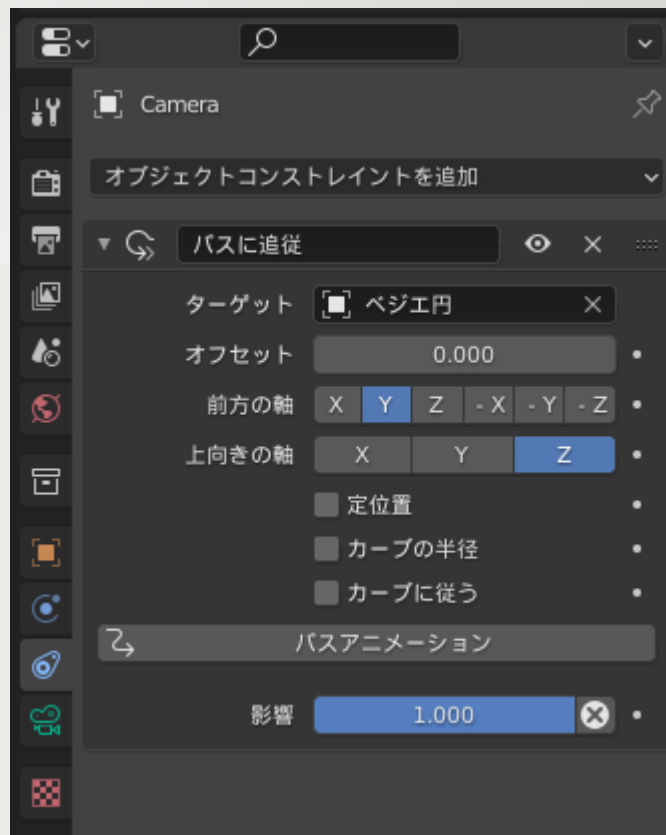
## カメラの追従

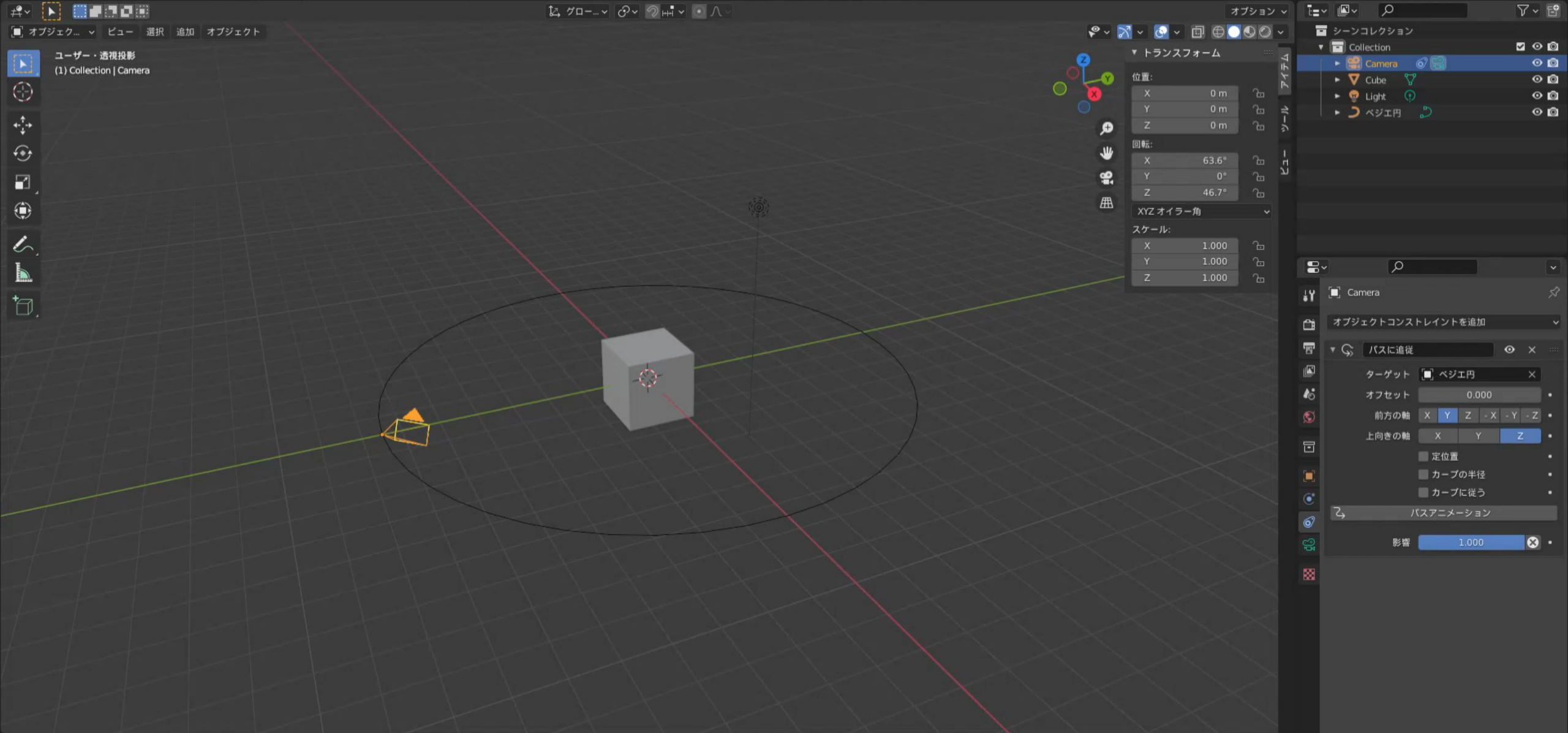
円上をカメラが動くようにする

「パスアニメーション」を使用

カーブの「円」を追加→

カメラの「パスに追従」を選択する





オプション

▼ トランスフォーム

位置:

X	0 m
Y	0 m
Z	0 m

回転:

X	63.6°
Y	0°
Z	46.7°

XYZ オイラー角

スケール:

X	1.000
Y	1.000
Z	1.000

シーンコレクション

- Collection
  - Camera
  - Cube
  - Light
  - ベジエ円

Camera

オブジェクトコンストレイントを追加

パスに追従

ターゲット: ベジエ円

オフセット: 0.000

前方の軸: X Y Z -X -Y -Z

上向きの軸: X Y Z

定位置

カーブの半径

カーブに従う

パスアニメーション

影響: 1.000

再生 キーイング ビュー マーカー

1 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250

開始 1 終了 100

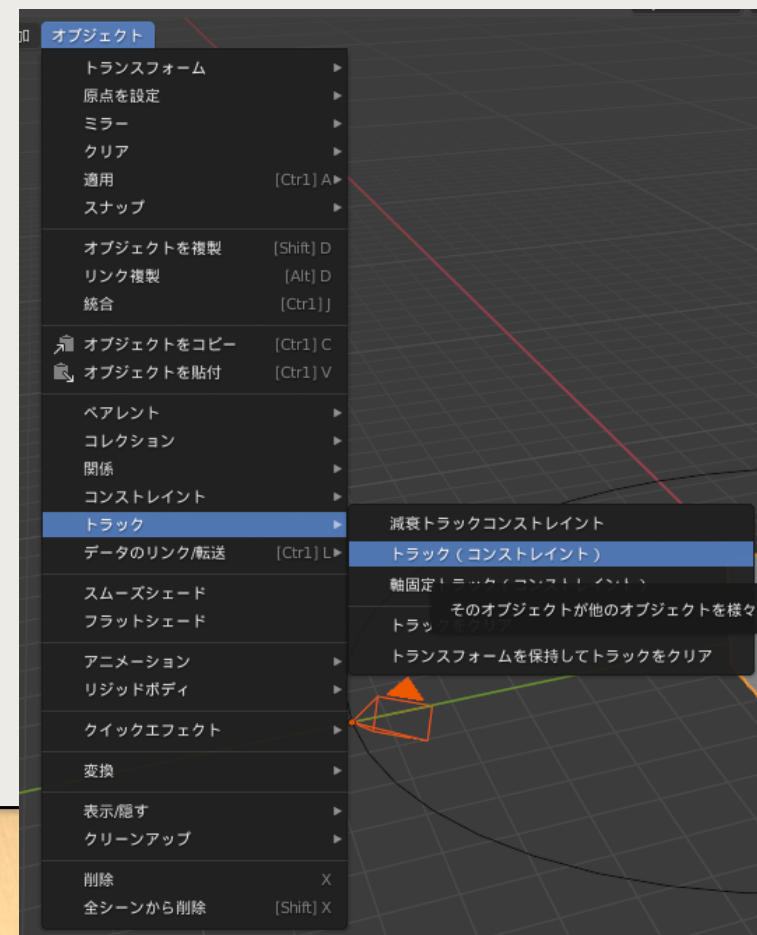
## カメラの追従

カメラを常に対象物に向き続けられるように設定

「トラッキング」を使用

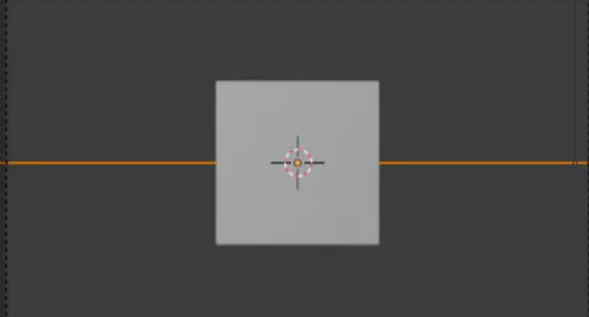
「カメラ」→「被写体」の順で複数選択

→「トラック(コンストレイトン)」を選択



オブジェクト... ビュー 選択 追加 オブジェクト

カメラ・透視投影  
(1) Collection | ベジエ円

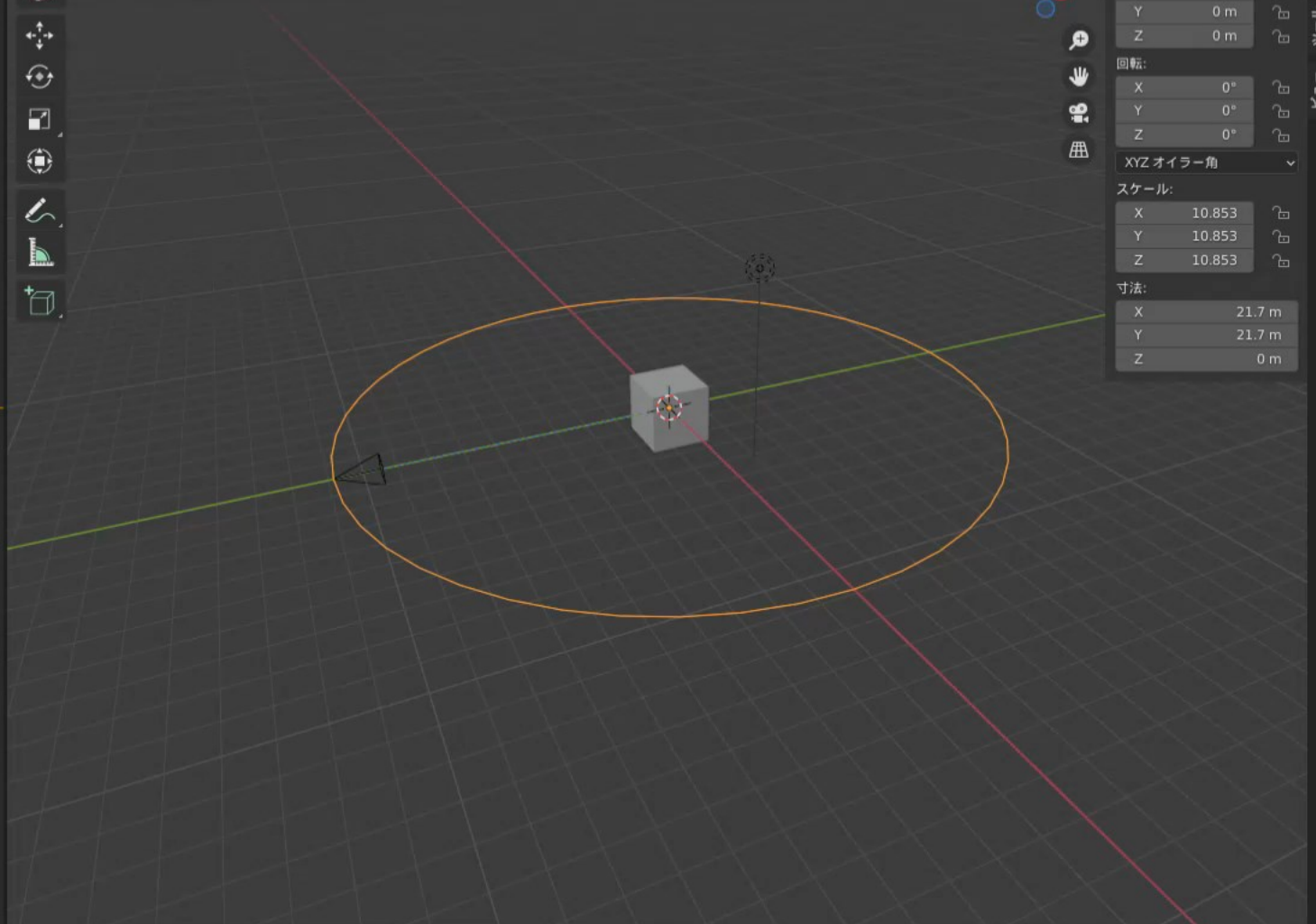


オブジェクト... ビュー 選択 追加 オブジェクト

ユーザー・透視投影  
(1) Collection | ベジエ円

トランスフォーム

位置:	X	0 m
	Y	0 m
	Z	0 m
回転:	X	0°
	Y	0°
	Z	0°
XYZ オイラー角		
スケール:	X	10.853
	Y	10.853
	Z	10.853
寸法:	X	21.7 m
	Y	21.7 m
	Z	0 m



シーンコレクション

- Collection
  - Camera
  - Cube
  - Light
  - ベジエ円

ベジエ円

オブジェクトコンストレイントを追加

再生 キーイング ビュー マーカー

1 開始 1 終了 100

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250

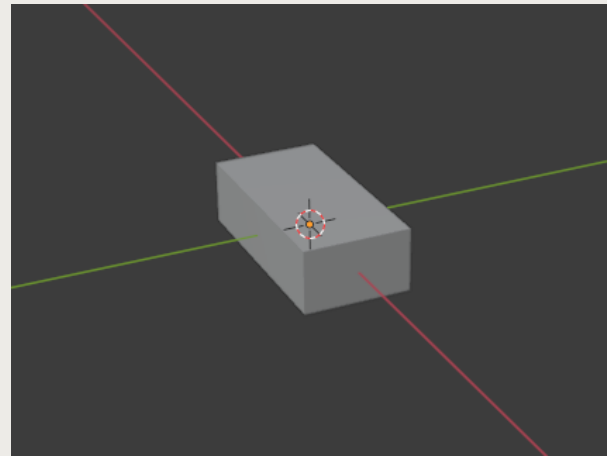
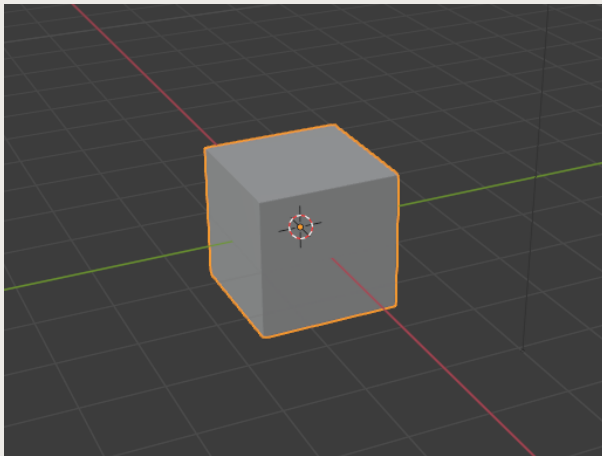
## カメラの軌道

- 距離：20cm～40cm 角度：30° ～60°
- 10回、位置を変えてデータを作成
- 1つの円につき100枚のデータ  
(1つのレンガにつき1,000枚×10種類=10,000枚)
- 円を動かすことで様々な角度のデータを得られる

角度	距離
45	28.3
51	32.0
36	25.0
35	30.8
45	35.4
54	30.8
53	25.0
45	25.5
39	28.4
43	30.4

## オブジェクト作成

- 質感を再現するために、実際の写真を撮り、テクスチャを作成する
- レンガの表面すべての写真を撮り一つの画像にする



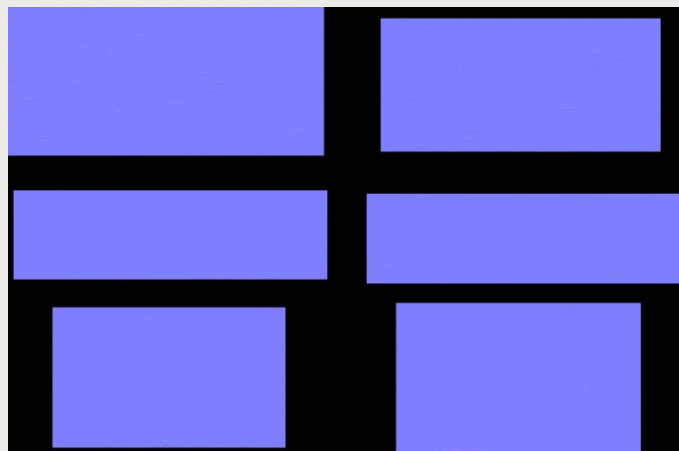


## オブジェクトの作成

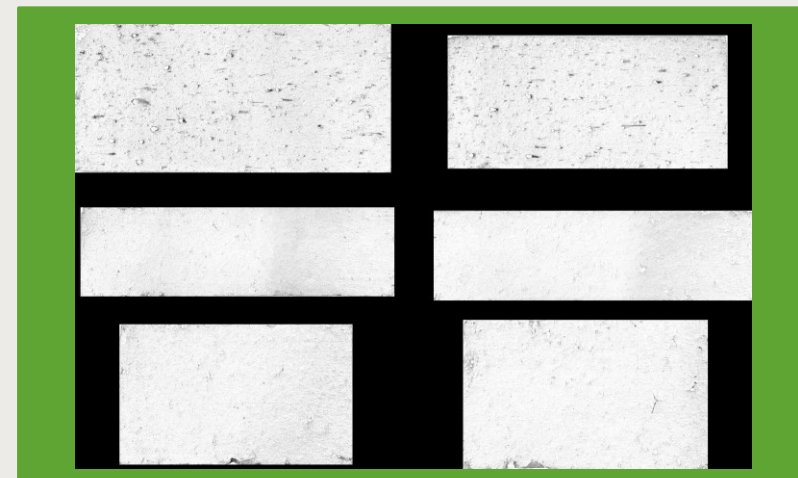
- テクスチャを貼ることで光の反射(凹凸)を表す
- GIMPを使ってRoughness mapとNormal(法線) mapを作成



BaseColor  
色の表現



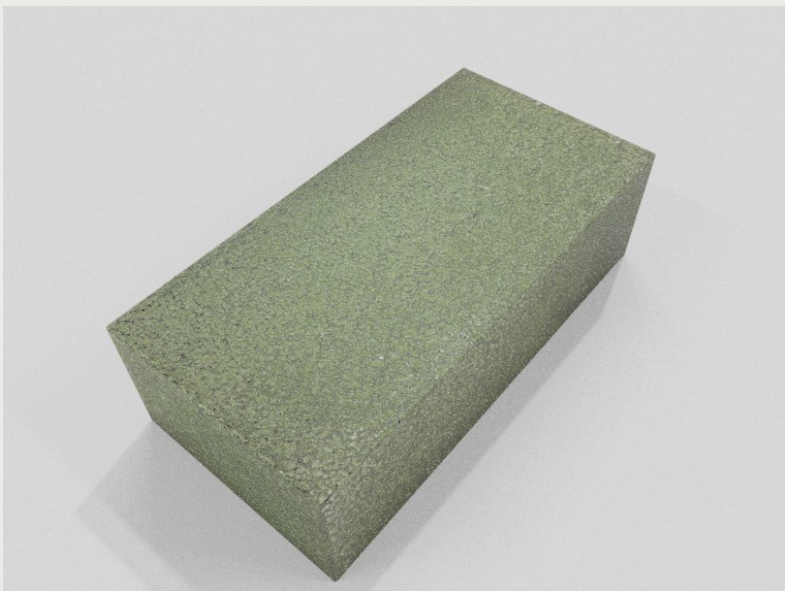
Normal  
RGBで凹凸を表す



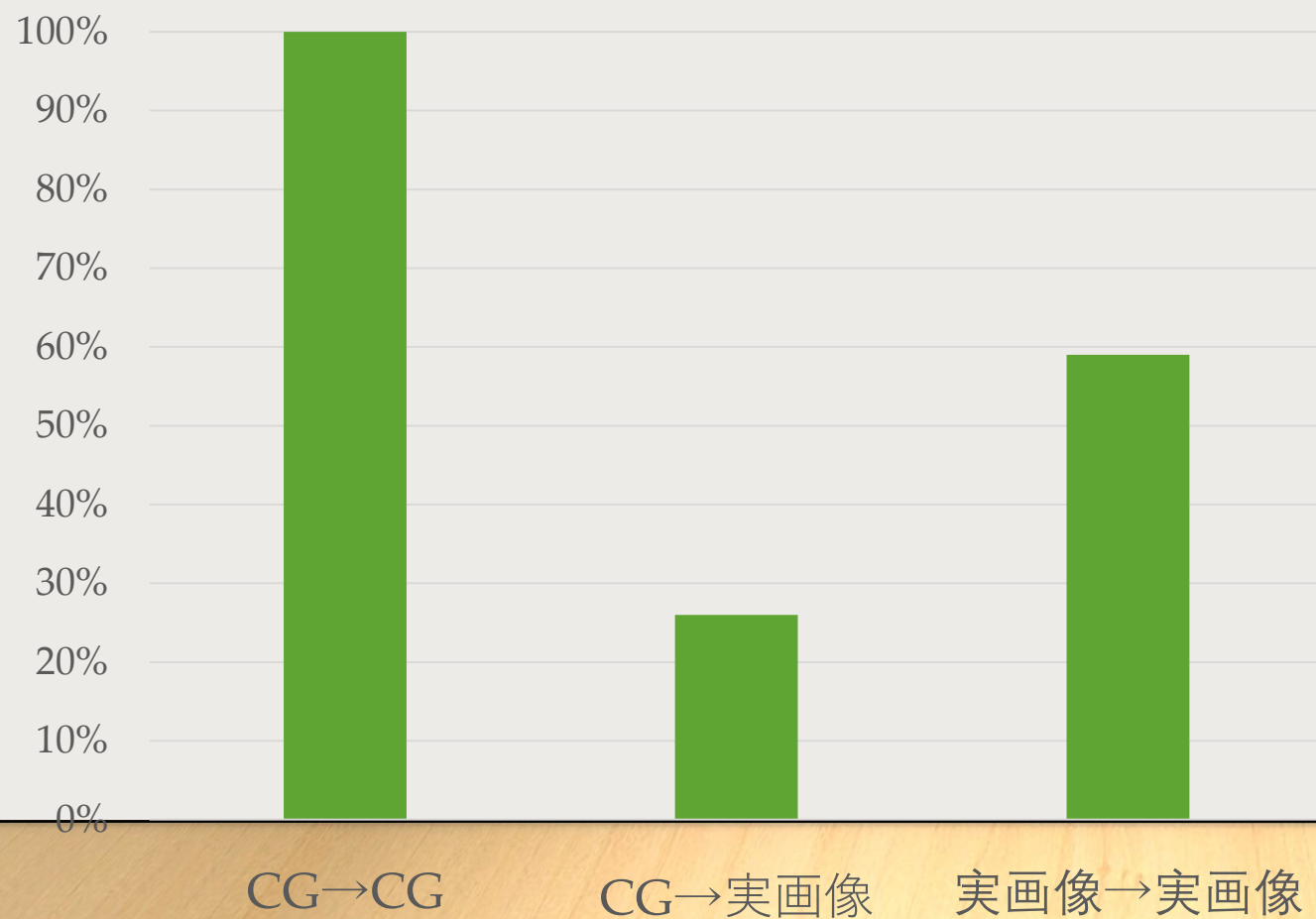
Roughness(粗さ)  
1 ~ 0 で表現

## 画像データの作成

- 作成したCG画像と実画像

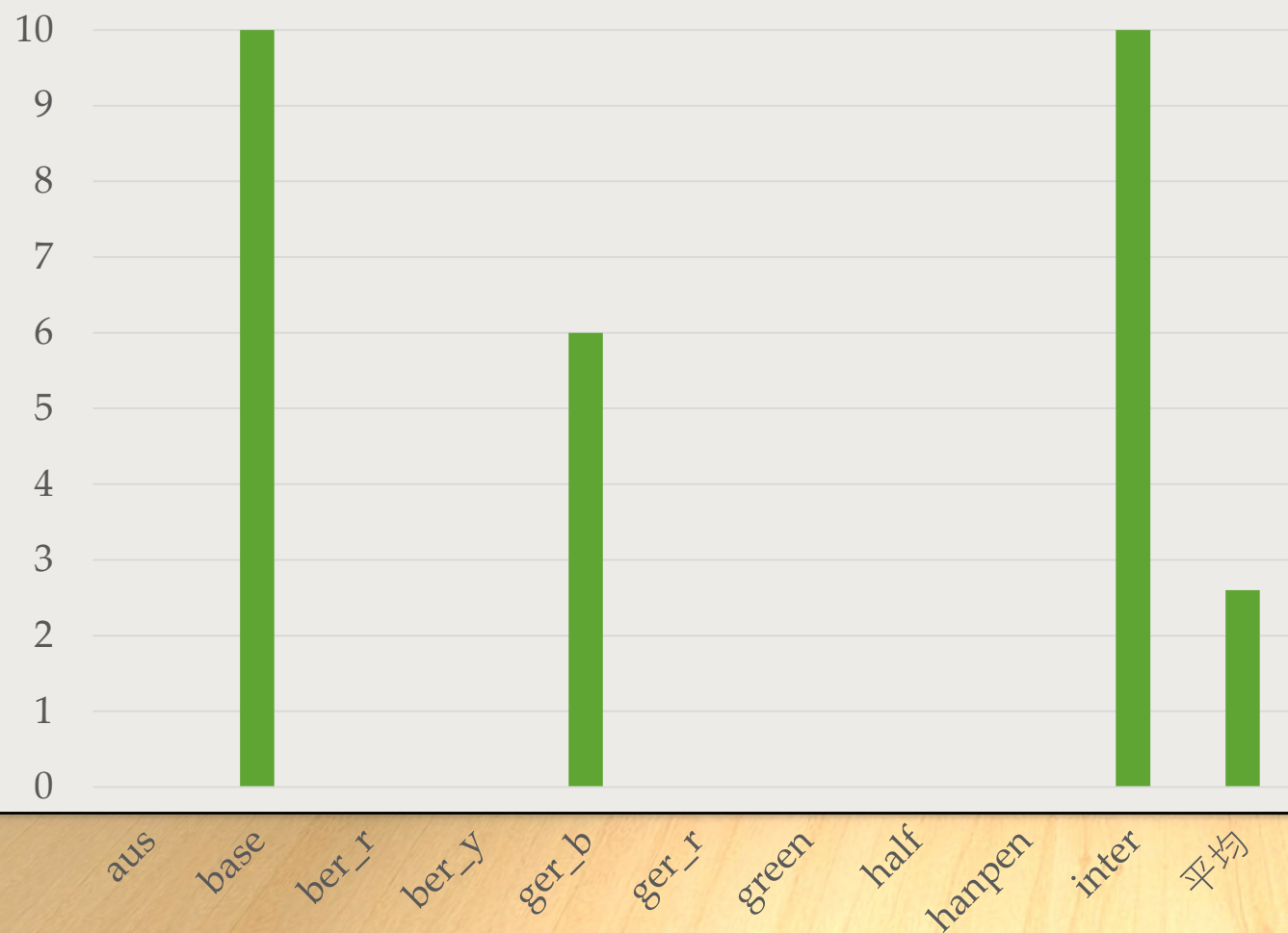


## 評価：精度



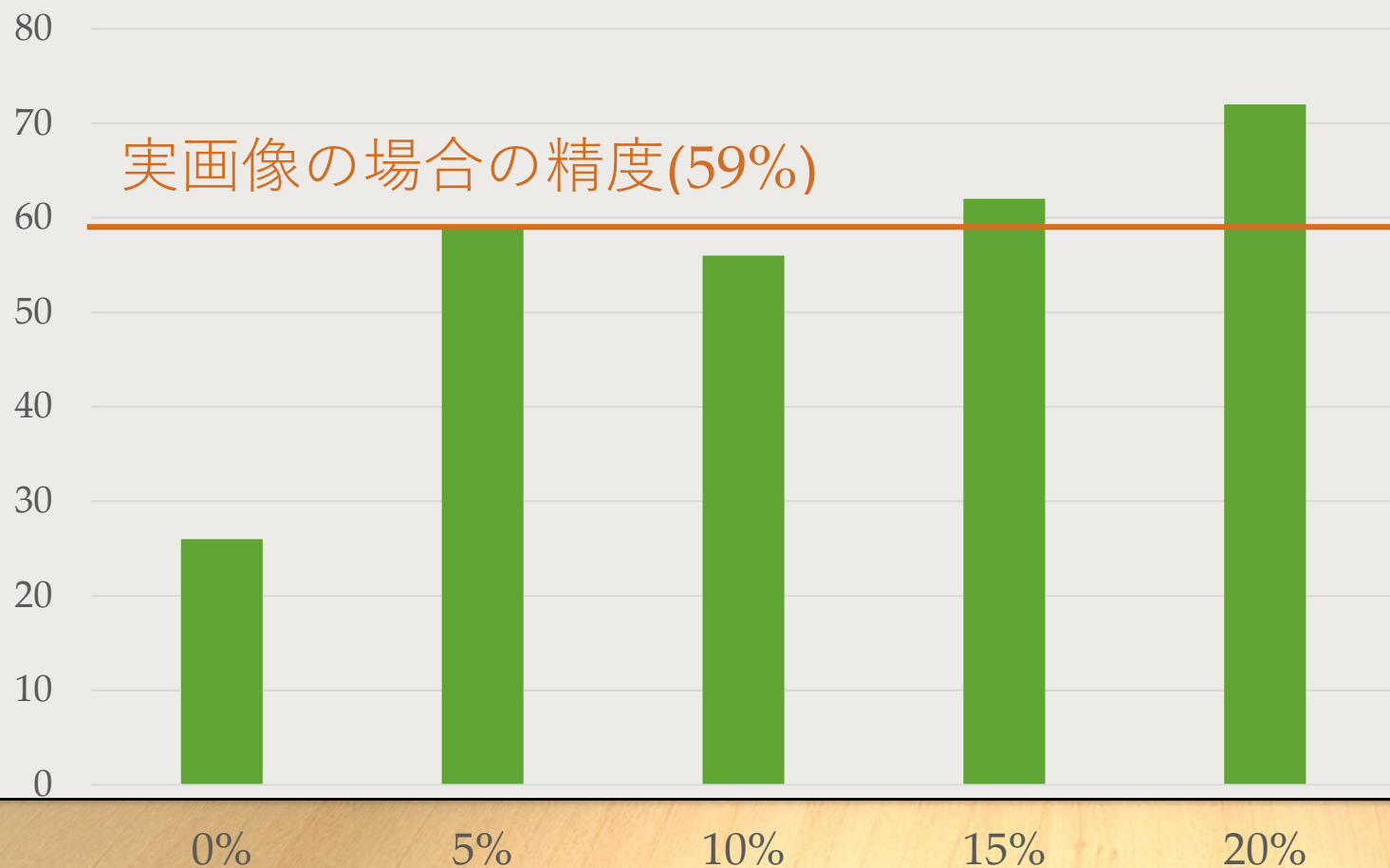
- CG—CG — 100%
- CG—実画像 — 26%
- 実画像—実画像 — 59%

## 評価：精度



- base、interのレンガは全て認識した
- ger\_bは半数は認識した
- それ以外のレンガは全て認識しなかった

## 評価：精度

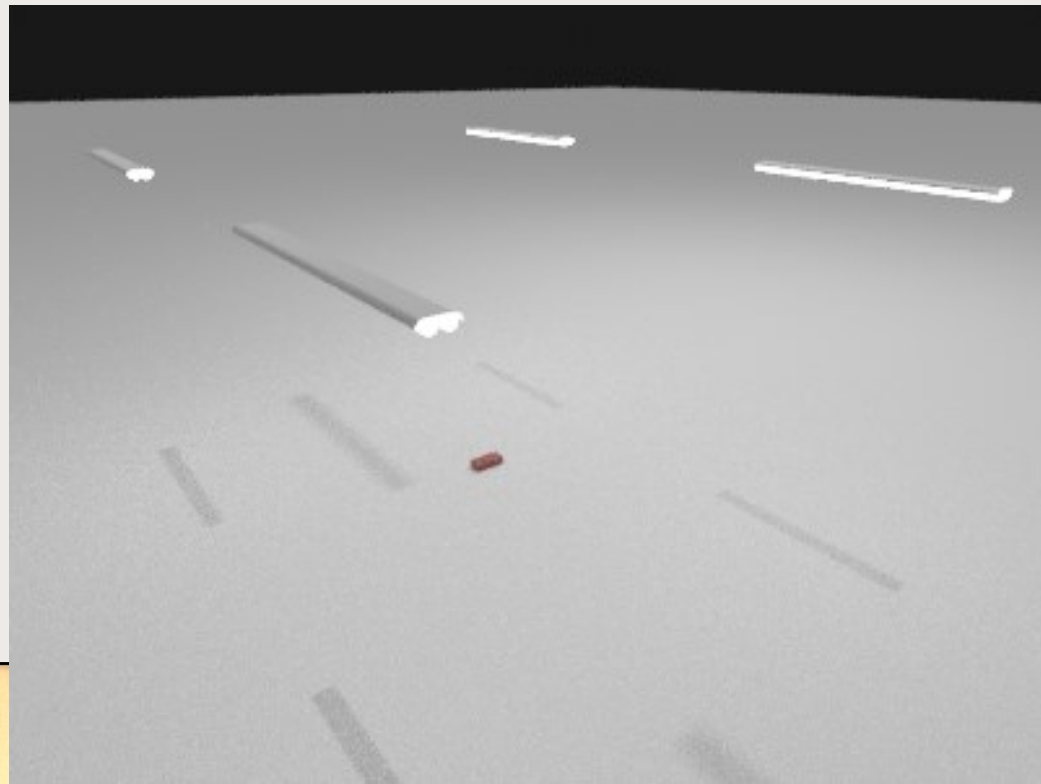


- CGの訓練データセットの中に、実画像を混ぜて訓練した
- 割合が増加するに従って、精度が向上している
- 15%以上で、実画像で訓練したモデルを上回った

54

## 評価：時間

- 均一な環境でのデータ作成のため、作業時間としてレンガの作成時間を評価する



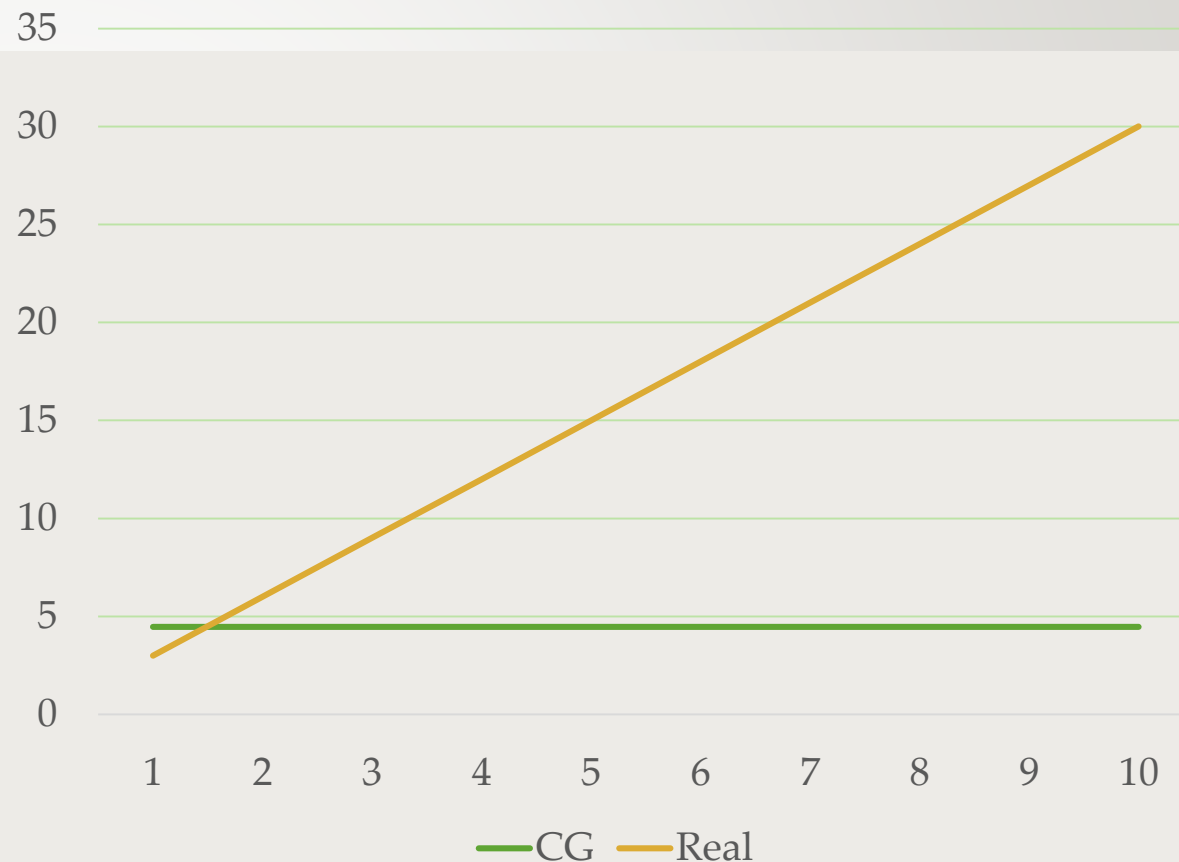
55

## 評価：時間

- レンガモデリングを作成するのに、4分46秒を要した
- 実画像では、1つの角度でデータファイルを作るのに3分要した



- 角度と距離が多くなると、時間のCGが効率的である



- CG画像の一部を実画像に置き換えた場合、15%以上で実画像のみで訓練する場合よりも高い精度が得られた
- 1つの視点の場合、回転台を活用した方が短時間でデータは得ることができるが、多様な視点の場合は、CGで行った方が効率的にデータは得ることができる



## 目的

- 任意の方向からの訓練データをCGで作成する

## 実験・評価

- 精度は実画像に比べて劣化する、CG + 実画像は実画像だけの時以上の精度になる
- CG画像でデータを作成する方が、多様な角度のデータを効率よく得ることができる

## 結論

- 多様な視点からの学習データは、CGの活用により効率的に作成できる

## 将来研究

- CG画像を実画像に近づけること  
→マテリアルやモデリングの改善
- 様々な環境にも対応した訓練データを生成する  
→光源を動かすことで環境を変化させる

# 全体のまとめ

- 一定の環境の場合

回転台を利用した実画像の学習データが有効

- 多様な環境の場合

実画像とCG画像を混合した学習データが有効

## 将来研究

- 最も効率的に作る方法

CGにより様々な環境での学習データを自動生成する方法

ご清聴ありがとうございました