

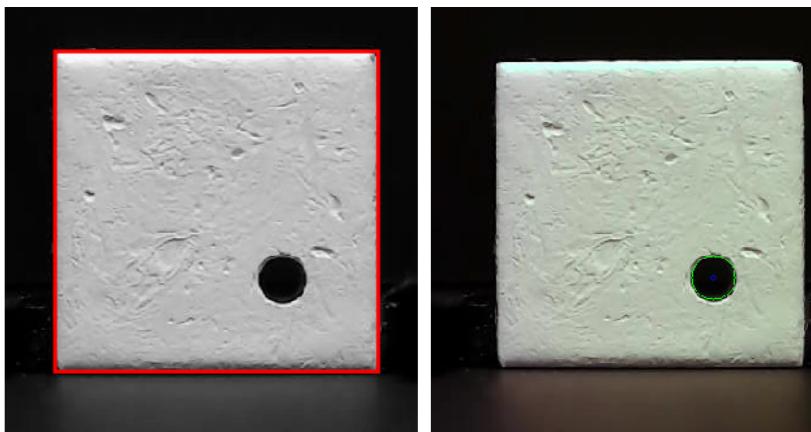
# good / bad cube 選別

2019.10.14 京都大学 谷 真央

# 前回のまとめ

- super-FGD に使う1立方センチメートルのシンチレータキューブの品質チェック
- キューブを撮影し、サイズや穴の位置、半径を得る。
- 前回は撮影台を作り、良いキューブ10個の撮影・解析を行った。

→その後、KEK 小川さん、松原さんからgood / bad キューブそれぞれたくさん(100~200)頂いた。



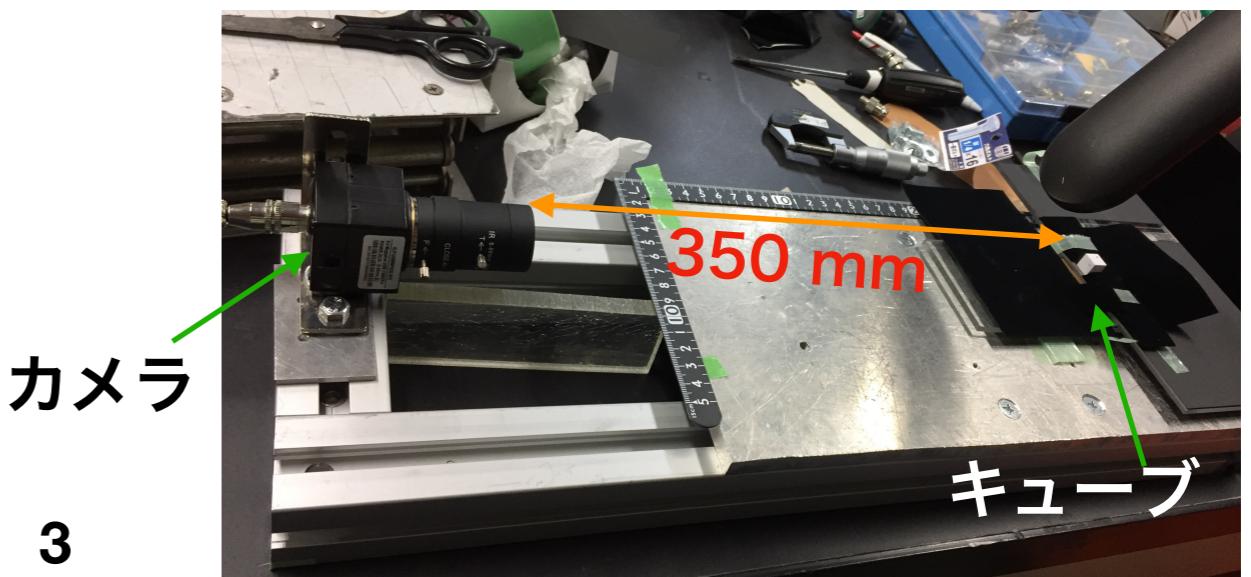
bad cubes

good cubes

# good / bad cube の撮影

- 各 70 個の good / bad cube の撮影
  - ロシアの方法によってすでに判定されているものをサンプルとして用いる
  - 350 mm 離れた位置から焦点を穴に合わせて撮影(穴の位置は今回は右上で固定)
- ROOT による解析で good / bad の選別 (今回のデータは単位を全て pixel 数のままで扱う。)

新しい撮影台



# cube quality-check

- ロシアの方法（現行）
- $14 \times 14$  個のキューブを正方形に並べ、金属の棒を通す
- 金属棒が上手く通らないところがあれば、そこに bad-cube があると判断  
(例：キューブが大き過ぎて隣のキューブと干渉する、穴の位置・方向等が微妙にズれている等)
- 欠点：時間がかかる、定量的な bad-cube の判断が難しい、個人差が出る

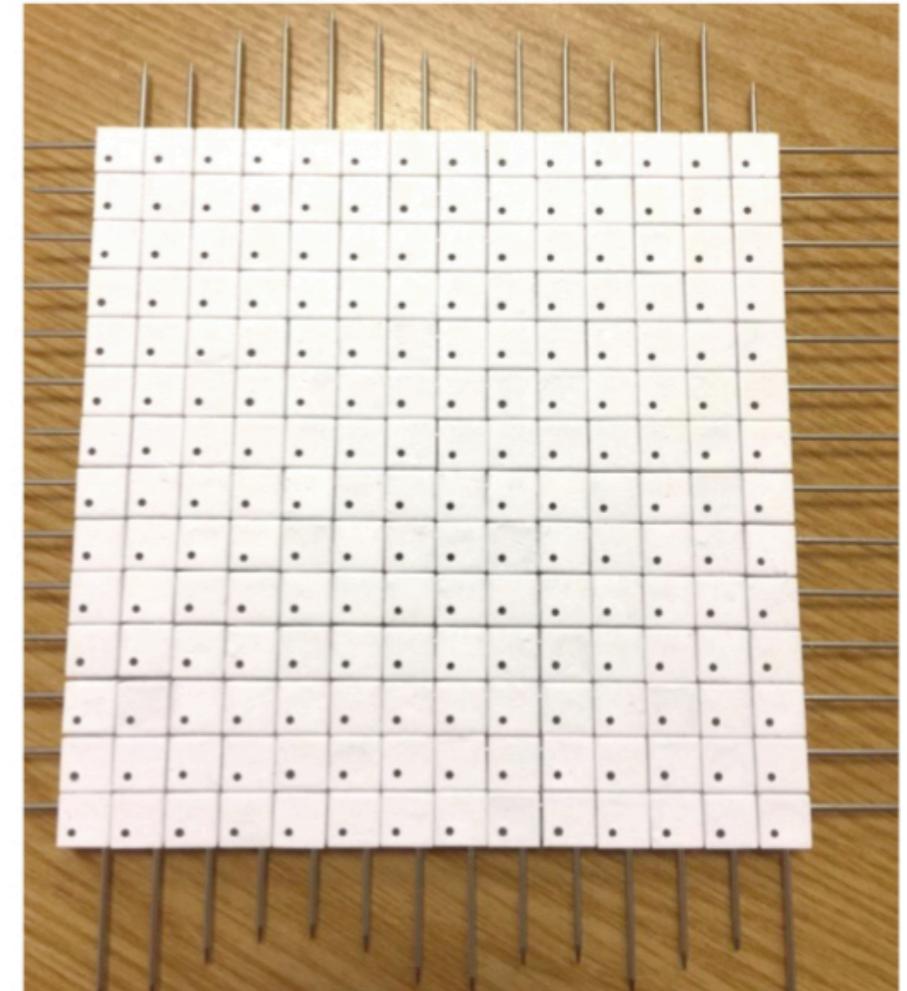
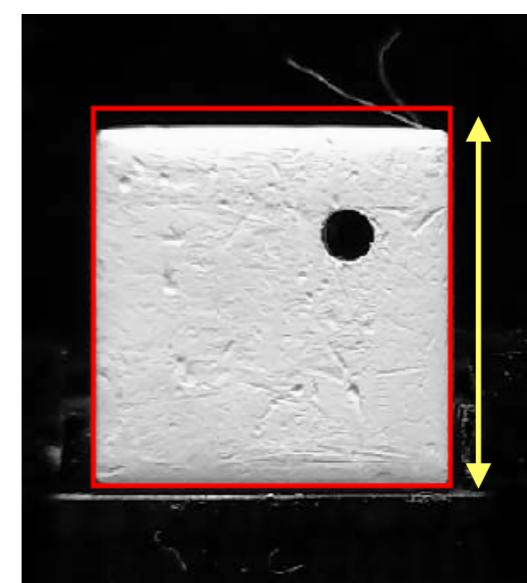
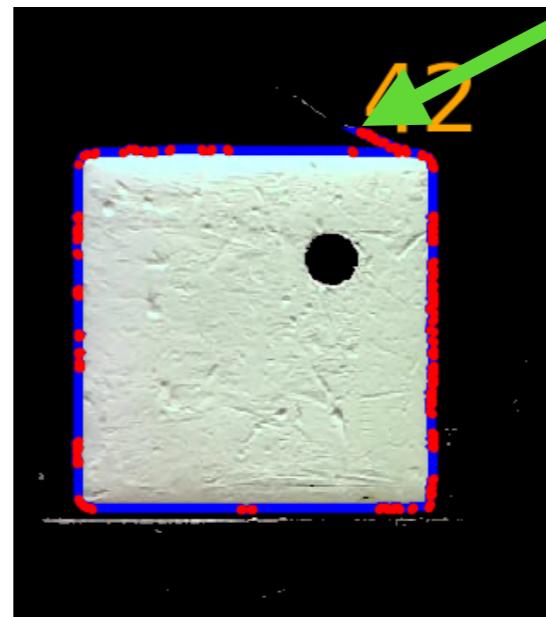
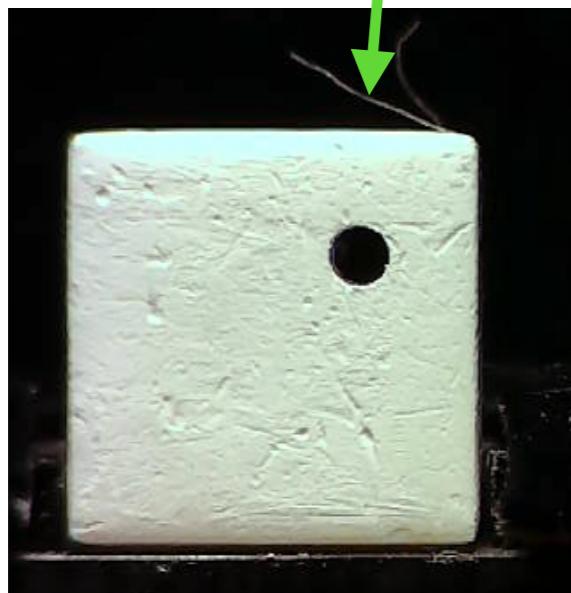


Figure 1: 196-cubes array for quality check.

この面のチェックの後、キューブを90度回転させて第3の穴についてもチェック

# 注意

- 外側にゴミがついているキューブについては、今までの方法ではうまく輪郭を検出できない→コードの改善が必要。



この点をキューブの最も外側  
の点だと認識してしまう

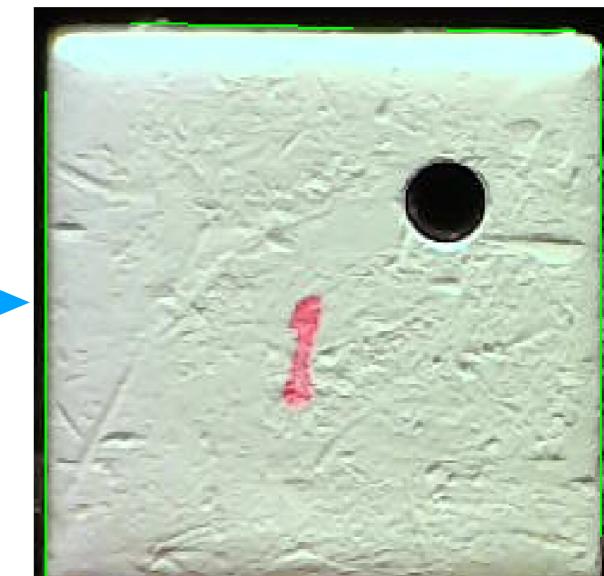
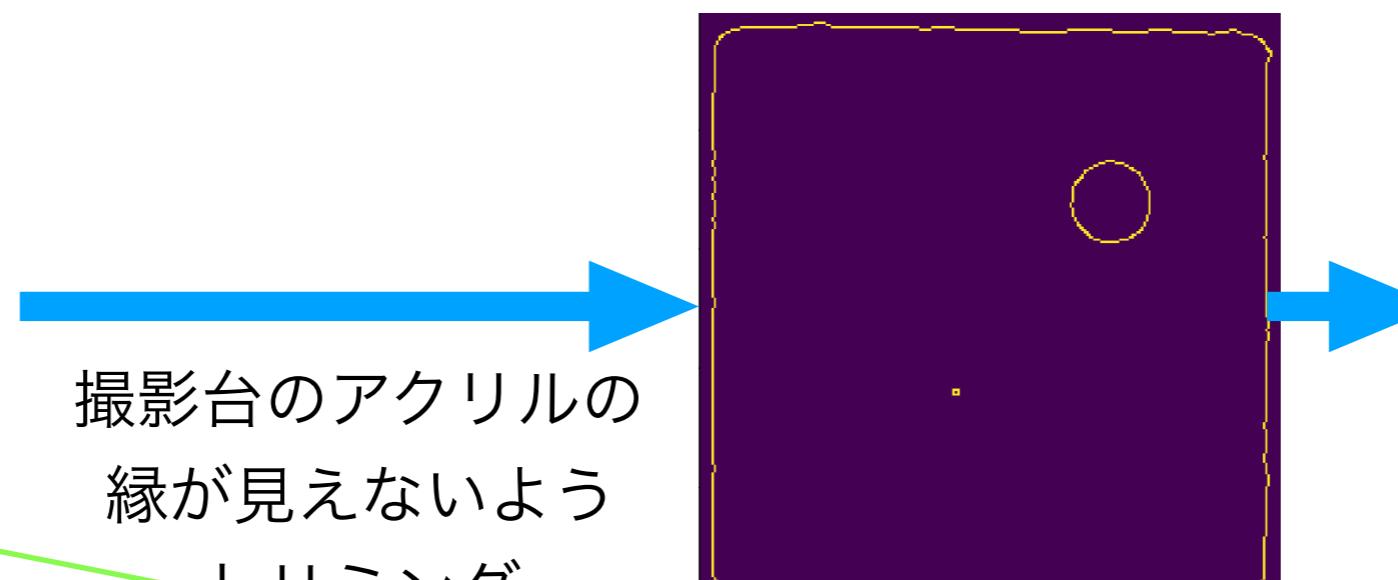
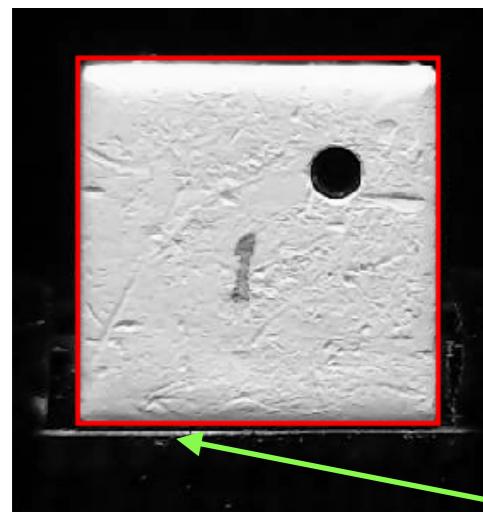
overestimate  
してしまう

今までの方法：

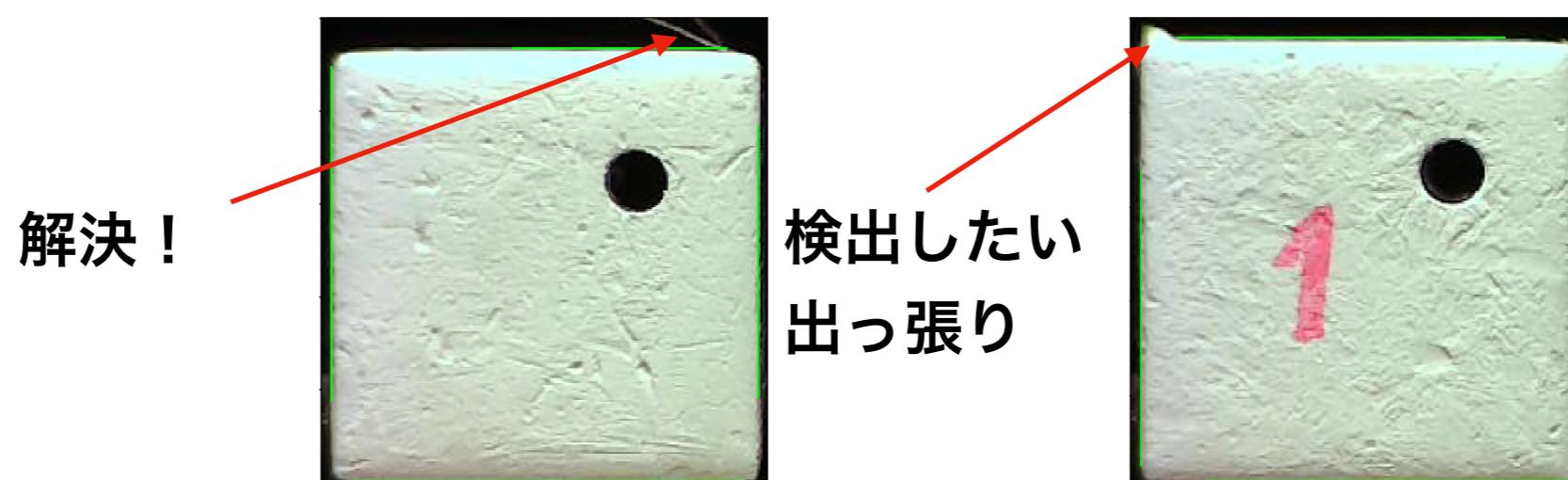
輪郭を検出、キューブの各辺のもっとも出っ  
張ったところを通るような長方形を出力

# サイズ検出の改善

- 直線を検出するメソッドを用いる。



- 前ページの問題は解決したが、今度は出っ張りが検出で  
きない。

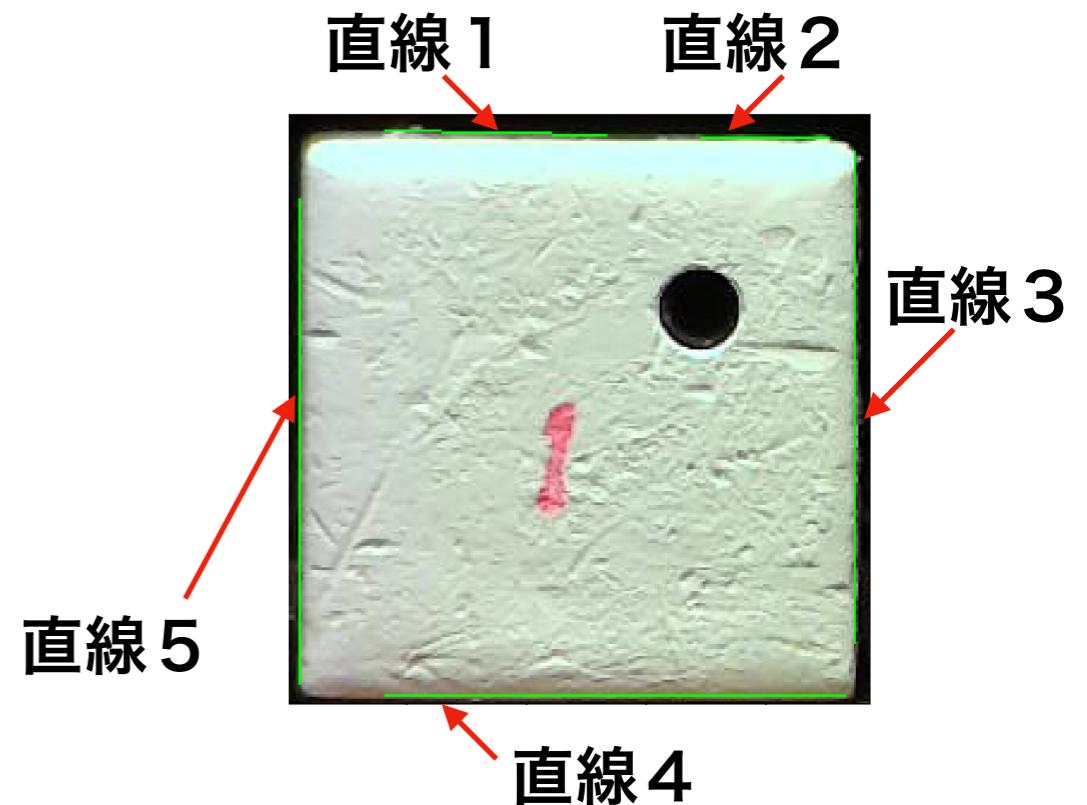


# good / bad 選別の方針

- good の分布と見比べて、badから
  - ① 検出サイズが分布から外れているものをはじく。
  - ② 穴が傾いているものをはじく。
  - ③ ①ではじけなかったもののうち、穴の位置がずれているもののをはじく。
  - ④ 穴の大きさが分布から外れているものをはじく。
- good / bad キューブを排除する割合を調べる

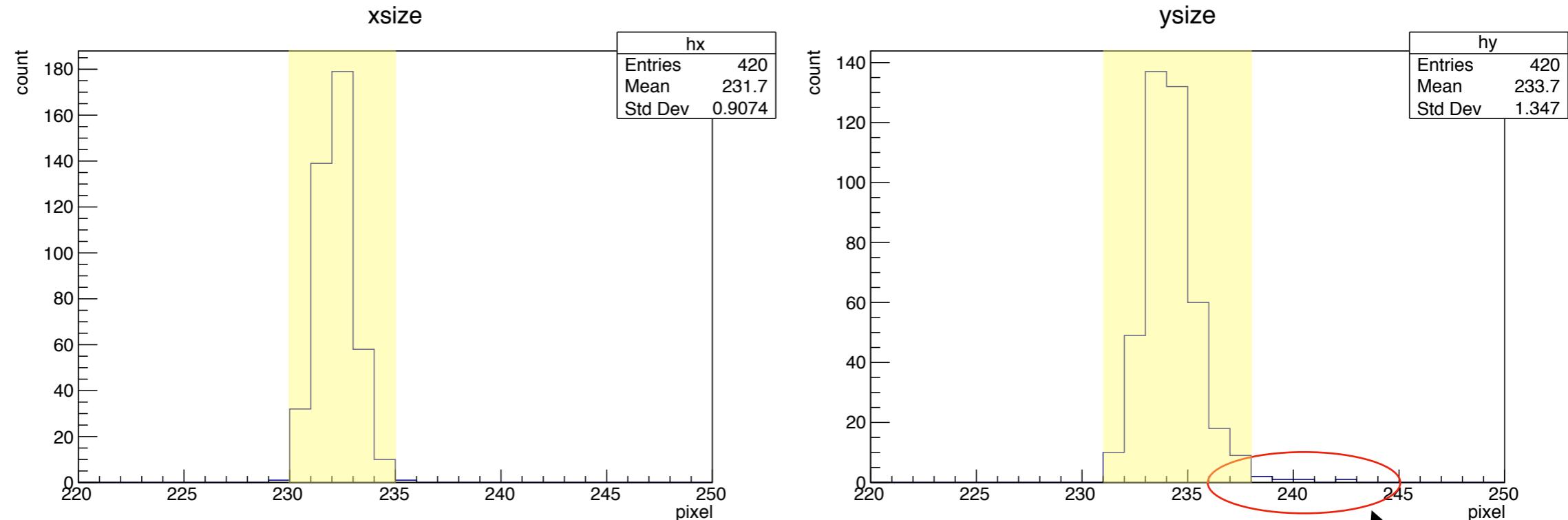
# 検出サイズの計算

- 各検出直線の始点( $x_1, y_1$ )・終点( $x_2, y_2$ )を得る
- 各直線上の中心の点  $(\bar{x}_i, \bar{y}_i)$  を計算
- $xsize, ysize$  をそれぞれ  
 $\bar{x}_{max} - \bar{x}_{min}, \bar{y}_{max} - \bar{y}_{min}$  で計算

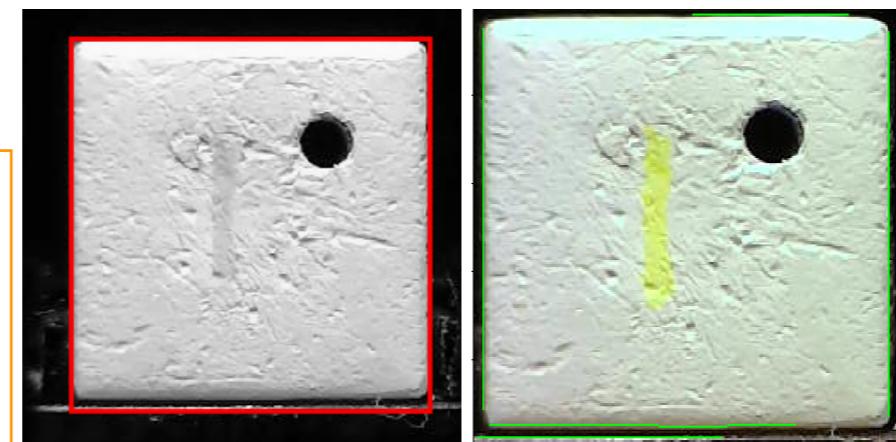


# 検出サイズの分布 (good)

- good キューブのサイズ分布を見る。



ちなみに、good キューブのサイズ~10.15 mm (前回) を用いると、  
約 0.044 mm/pixel

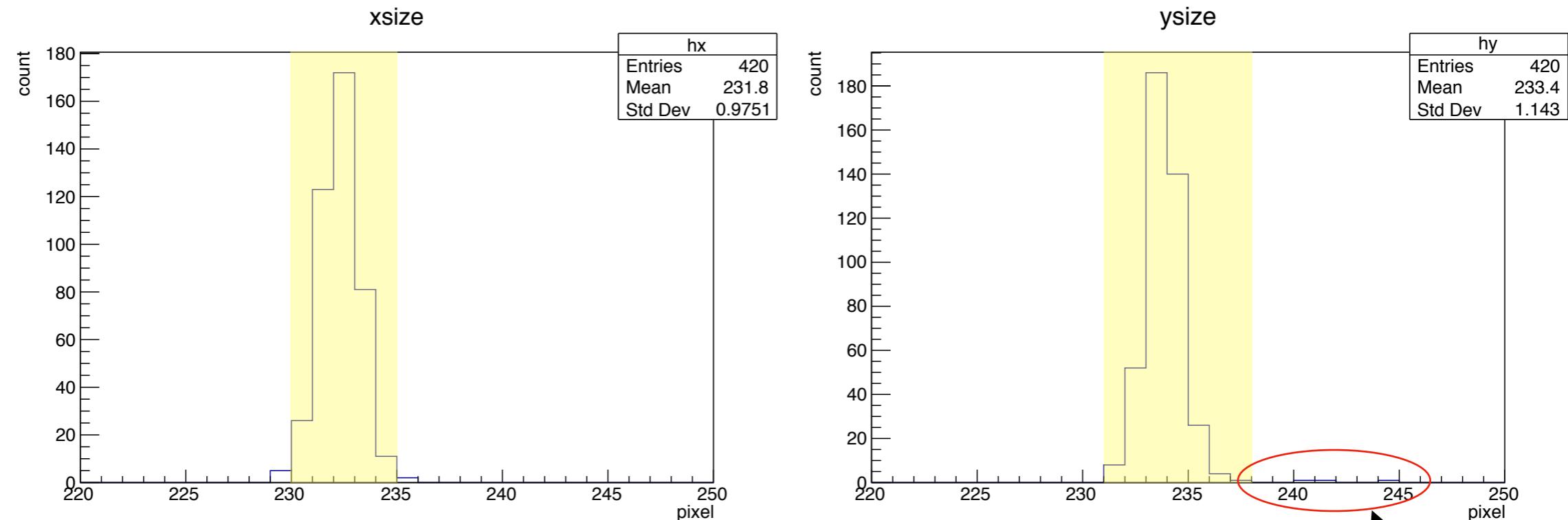


good : 70 cubes → **66 cubes**

撮影台のアクリルが見えないよう、トリミングしたが、そもそも輪郭を誤検出している場合、正しい結果が得られない→改善の余地あり

# 検出サイズの分布 (bad)

- bad キューブをサイズで選別する



good の結果：

$230 \leq xsize \leq 234$

$231 \leq ysize \leq 237$

でカットをかける。

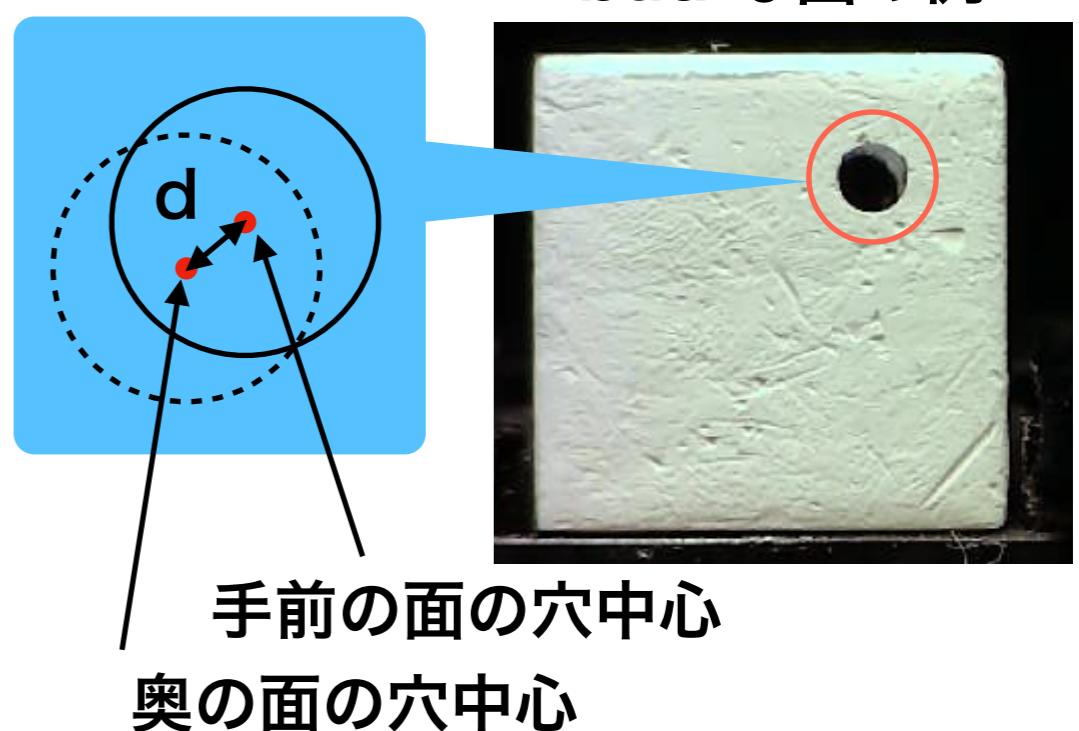
6面の各サイズについて、どれか1つでもカットがかかれればそのキューブをはじく。

bad に関しても誤検出が

**bad : 70 cubes  $\rightarrow$  63 cubes** いくつか見られる

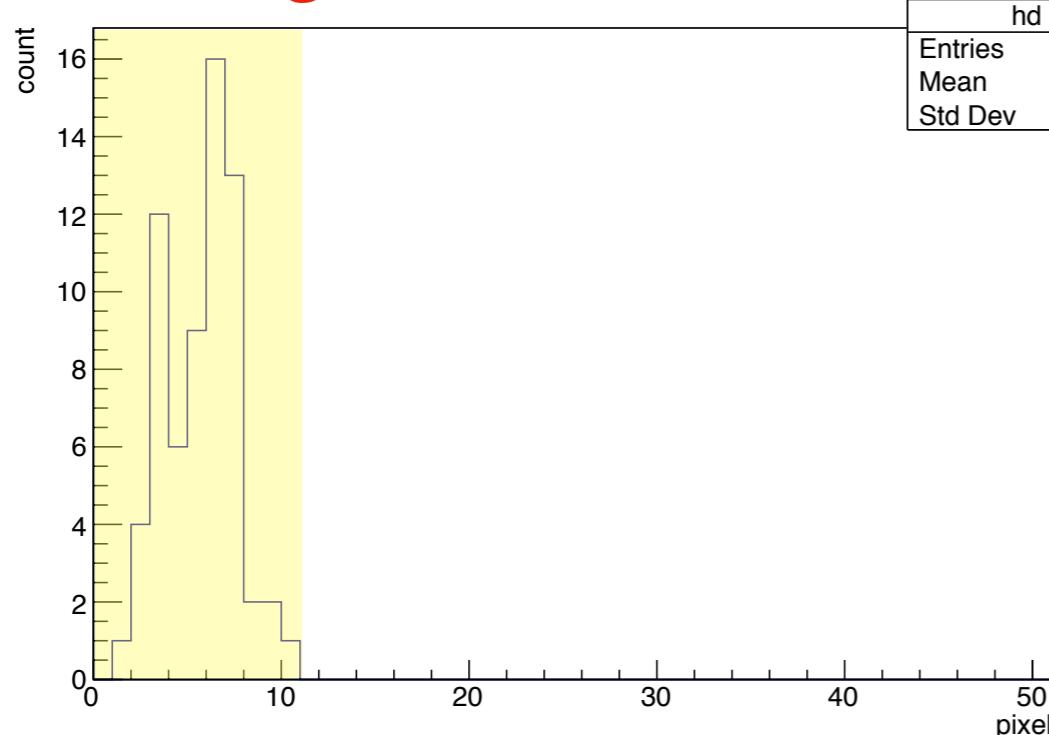
# 穴の傾きについて

- 穴が傾いている場合、ファイバーが通りにくい→ bad と判定したい
- 各穴について、 $d = \sqrt{dx^2 + dy^2}$  を定義  
( $dx$ 、 $dy$  : x、y方向の穴の位置のずれ)
- 3つの穴の  $d$  のうち、最大のものを  $d_{max}$  と定義
- good、bad について  $d_{max}$  の分布を比較する。



# 穴の傾きの分布

good cube dmax

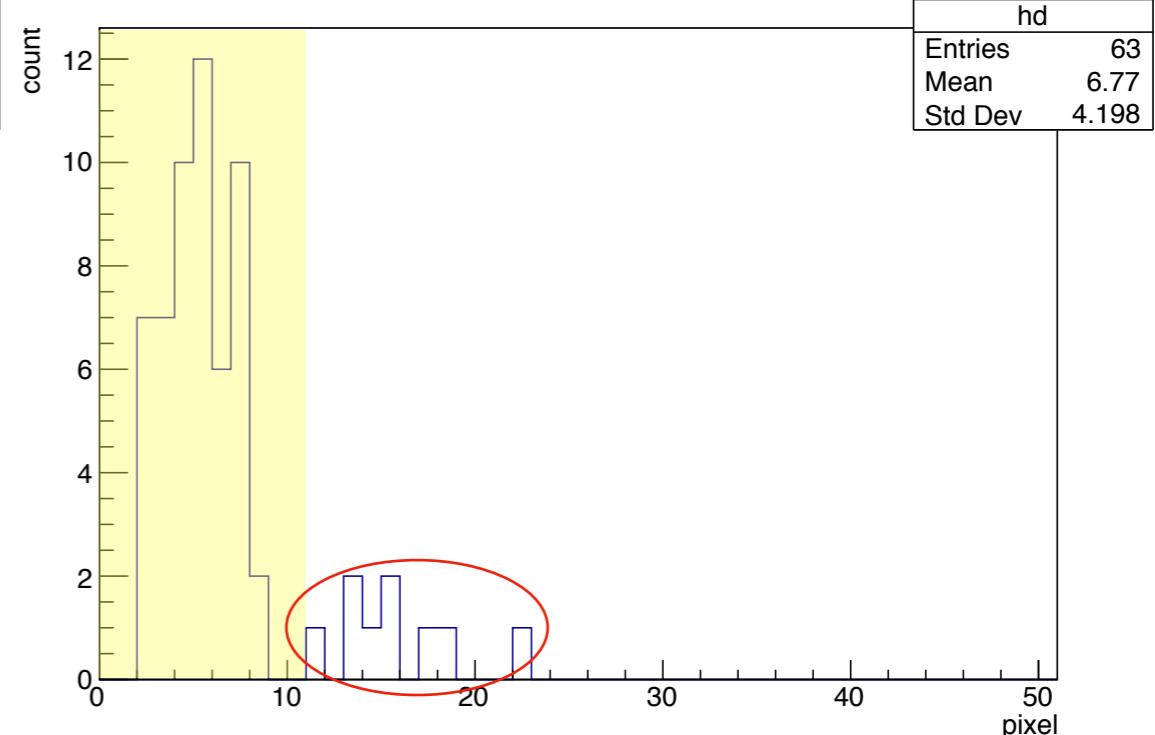


good の分布を見て、  
pixel 数

$dmax \leq 11$

でカットをかける

bad cube dmax



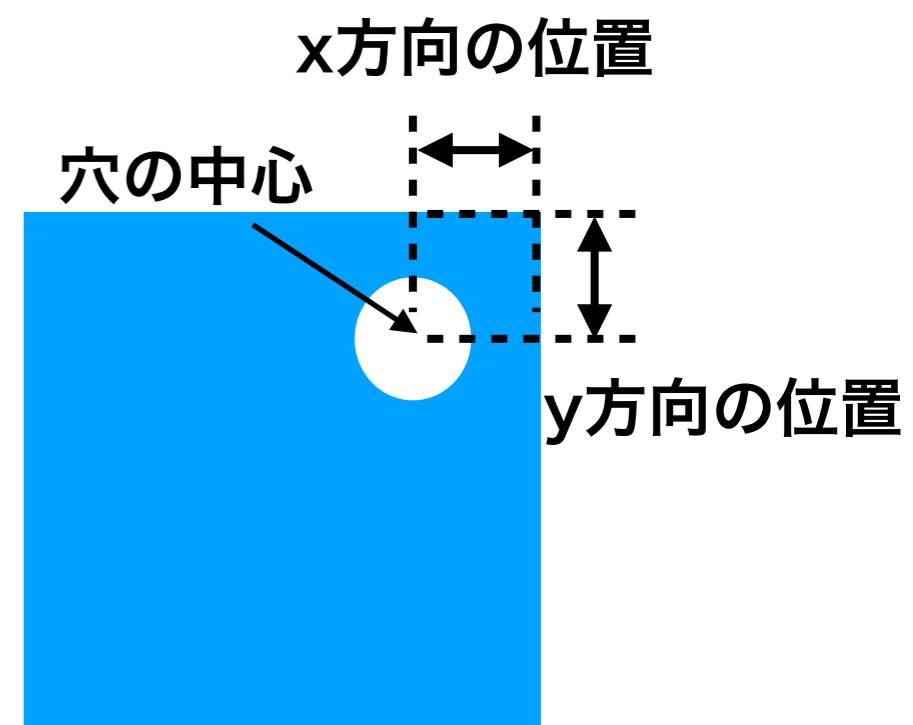
$dmax$  が大きいもののはじく

good : 66 cubes  $\rightarrow$  66 cubes

bad : 63 cubes  $\rightarrow$  54 cubes

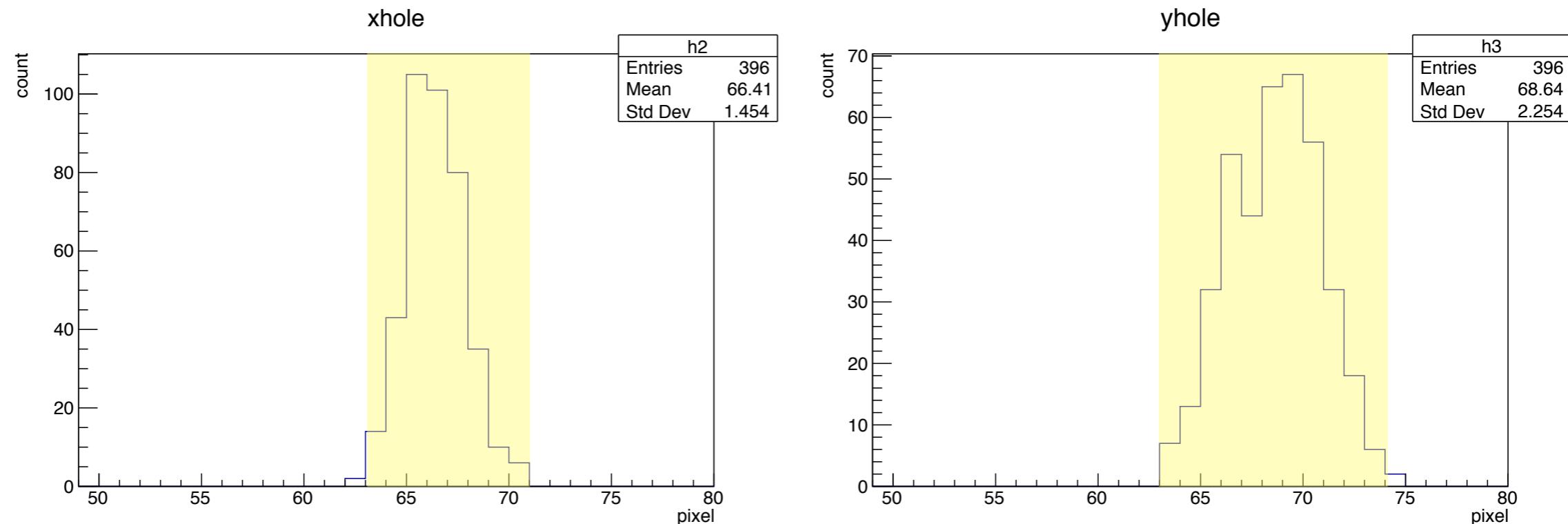
# 穴の位置について

- 穴が傾いていない場合でも、そもそも位置がずれていればファイバーは通りにくい
- good、bad について穴の位置の分布を比較する。



# 穴の位置分布 (good)

- good キューブの穴の位置分布を見る



good の結果：

$63 \leq xhole \leq 70$

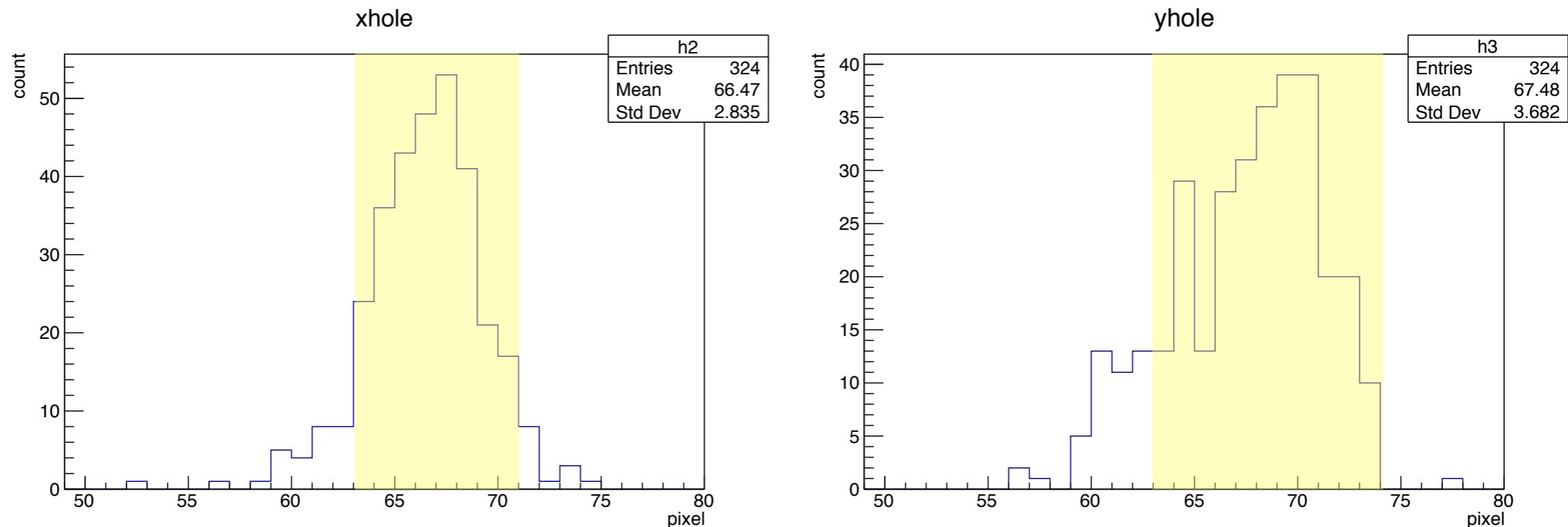
$63 \leq yhole \leq 74$

でカットをかける。

good : 66 cubes  $\rightarrow$  62 cubes

# 穴の位置分布 (bad)

- bad キューブを穴の位置分布から選別する



good の結果：

$63 \leq xhole \leq 70$

$63 \leq yhole \leq 74$

でカットをかける。

6面の xhole, yhole について、どれか1つでもカットがかかればそのキューブをはじく。

bad : 54 cubes  $\rightarrow$  19 cubes

# 穴の大きさについて

- ・ 穴はドリルで開けているので、本来ならばドリル径（あるいはそれより大きめ）の穴が空いているはず
- ・ 検出した穴の半径が小さ過ぎる、大き過ぎる場合、穴の状態が悪く、正しく検出できていないと仮定

よい穴の例：若干小さく検出している  
ように見えるが、許容範囲とみなす

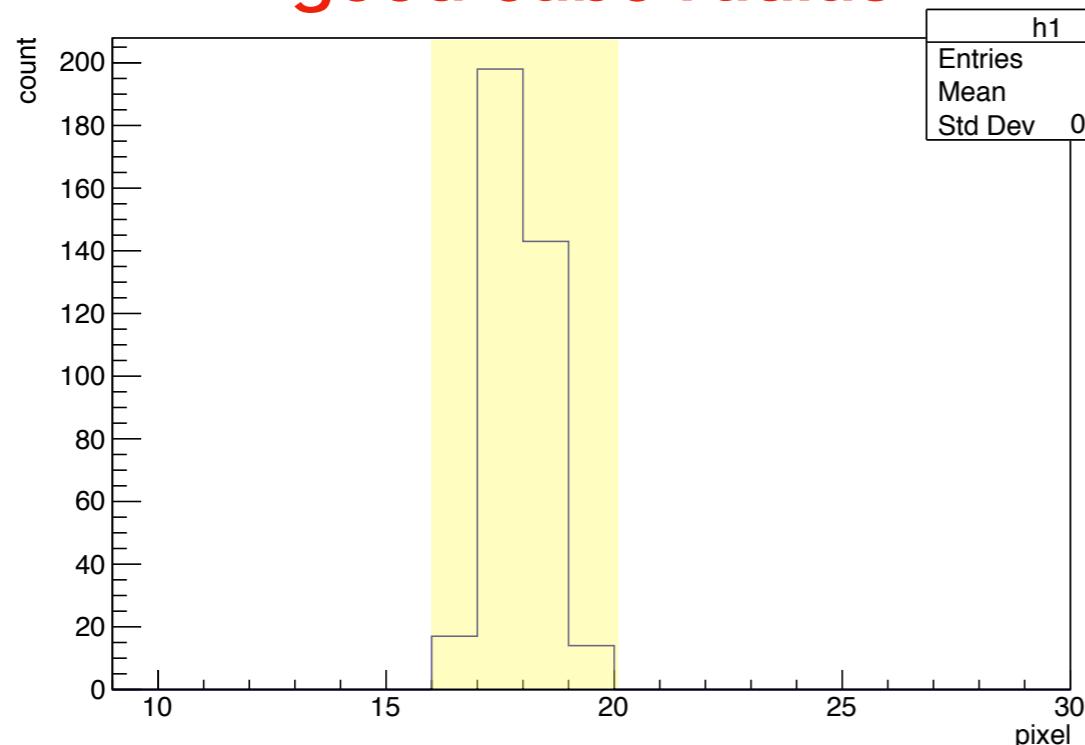


検出半径の大き過ぎる例

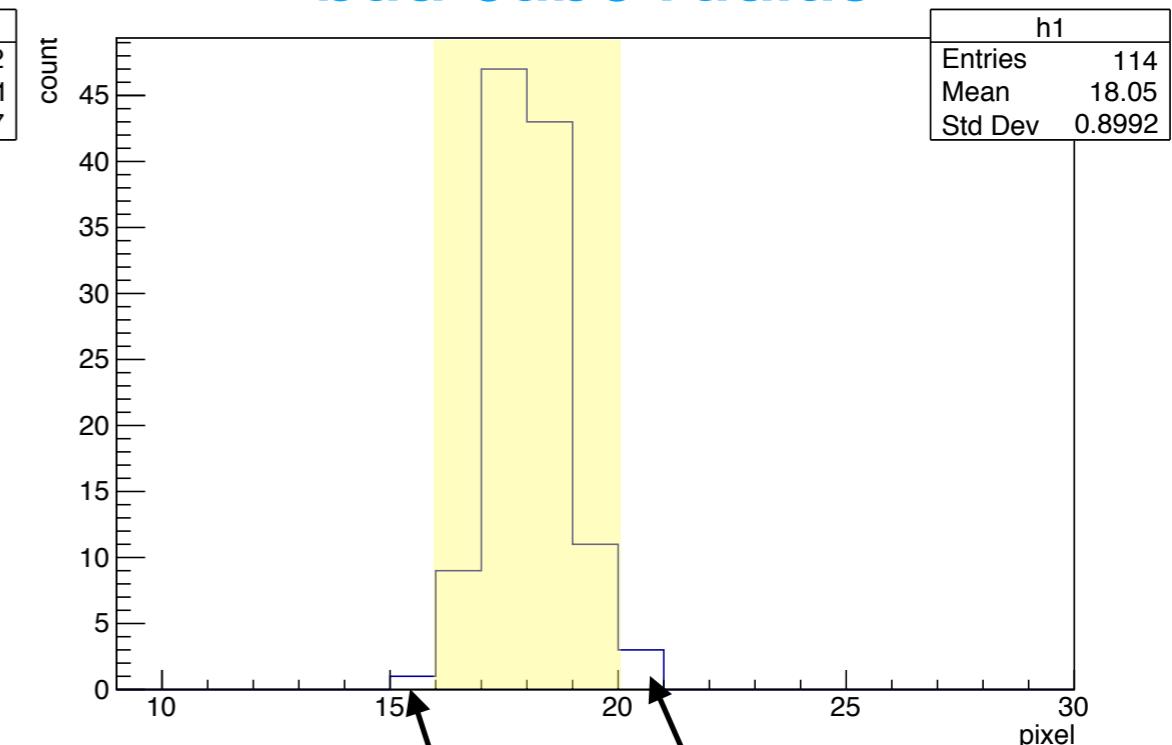


# 穴の大きさの分布

good cube radius



bad cube radius



good の分布を見て、

pixel 数

$16 \leq \text{radius} \leq 21$

でカットをかける

小さすぎる 大きすぎる

good : 62 cubes  $\rightarrow$  62 cubes

bad : 19 cubes  $\rightarrow$  15 cubes

# ここまでの中のまとめ

- 以上の選別で、

good : 70 cubes → 62 cubes

bad : 70 cubes → 15 cubes

となった。

- bad を排除できた割合は  $\frac{70 - 15}{70} = 78.6\%$

- good を誤って排除してしまった割合は  $\frac{70 - 62}{70} = 11.4\%$

- もう少し条件を追加して、精度を高めたい。

# 今後の方針

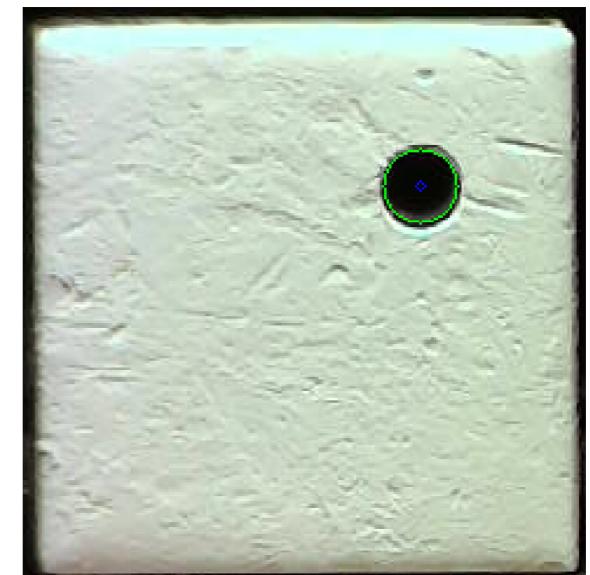
- 穴の内部のピクセルの明るさのデータを用いたキューブの選別ができるかもしれない。

(図参照)

- 今回は焦点距離 350 mm 、レンズ軸を穴に合わせた撮影

→今後レンズ軸をキューブ中心に合わせた撮影・解析を試す。

良い例：理想的な穴では、内部には黒の背景のみが映る



悪い例：穴の状態や傾きによって、穴の内部に白い部分が見られる。

