

SuperFGD scintillator cube
光の当たり方、
カメラ毎の見え方の違いによる
穴位置測定値の差の評価

2020.4.14 京都大学 谷
collaborate with 小川さん (KEK)

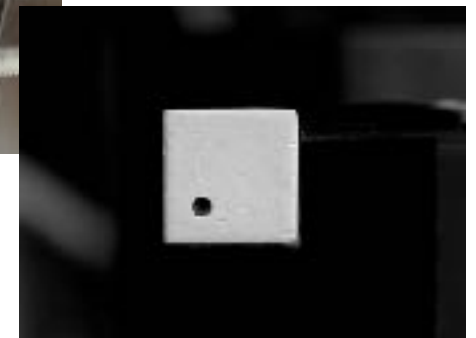
現行の撮影システム

キューブ台座（アクリル、スズノ技研）
黒い布を貼りキューブの周囲をマスキング

リアコンバータ

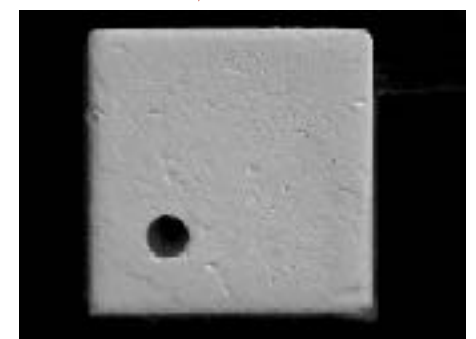


X2 TV EXTENDER JAPAN



使用前

約2倍の分解能



使用後

1 pixel ~ 15 μm

- 3方向からの三面同時撮影
- 焦点距離 5 cmでの撮影
- 各カメラはリング状のLED、拡大レンズ (右図)を装備
- 手でキューブを回転し、
残りの三面の撮影

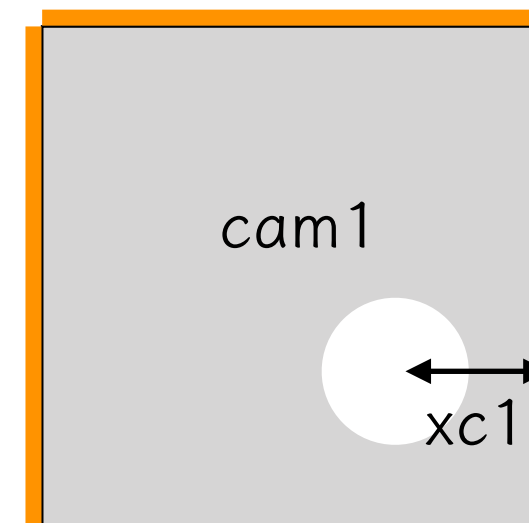
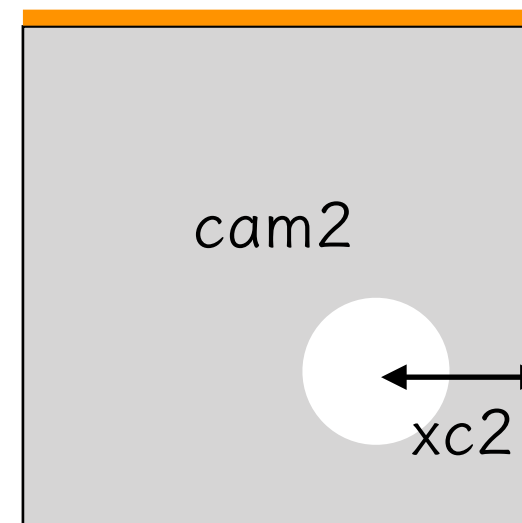
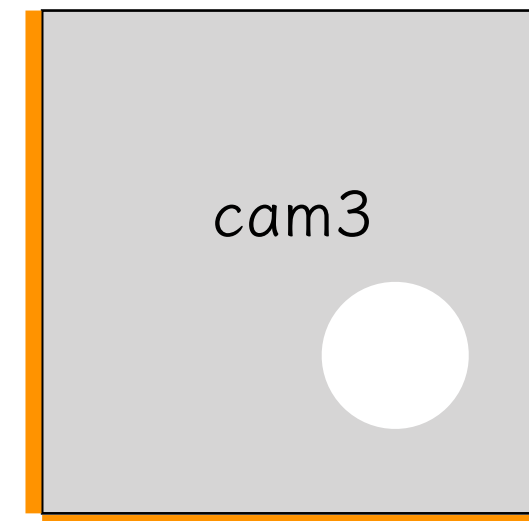
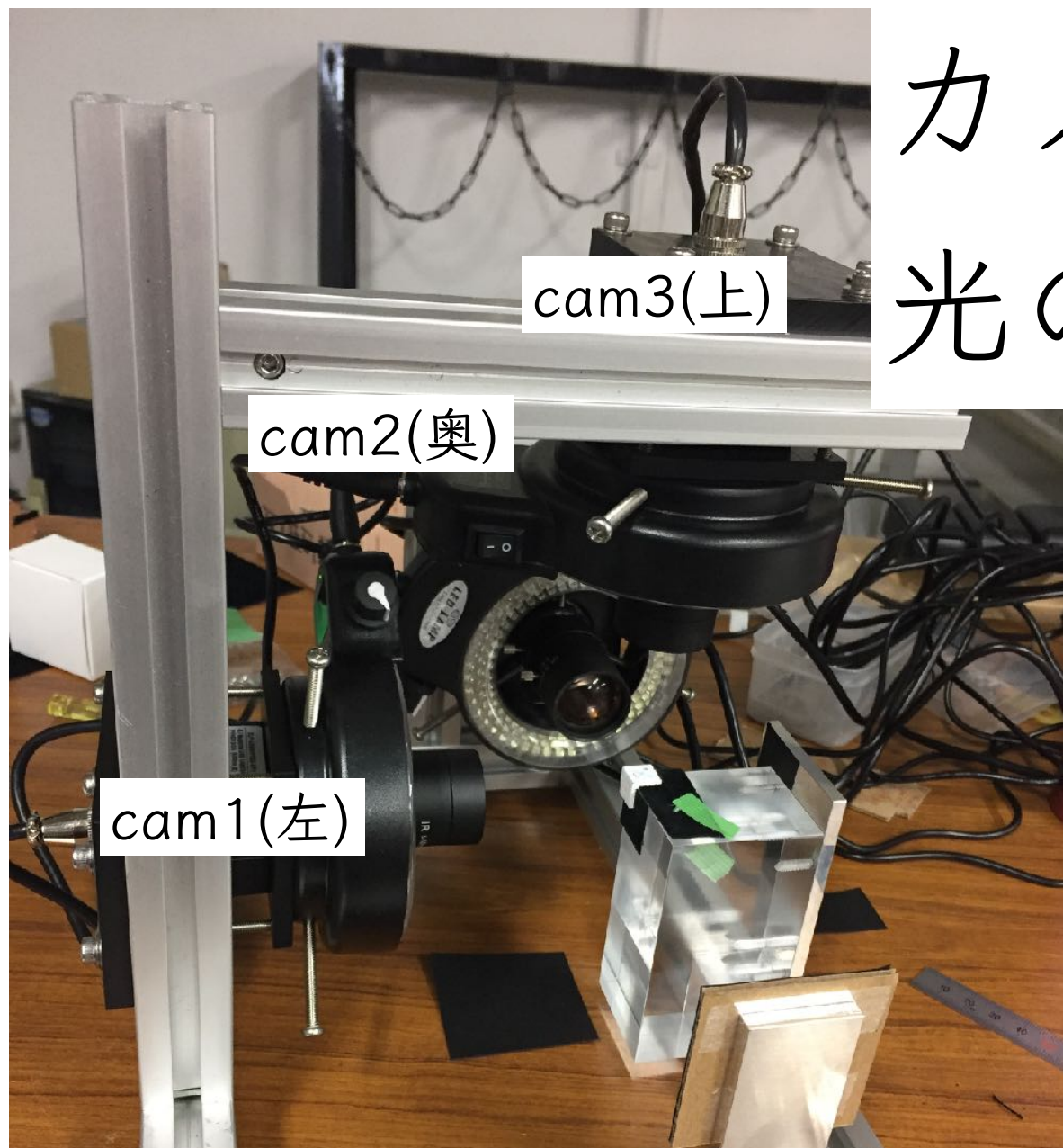
3方向にカメラ＋
リングライト設置

骨組み：太さ25mmの
アルミフレーム

カメラを水平・垂直に固定
するため、専用のカメラジグを
製作（スズノ技研）



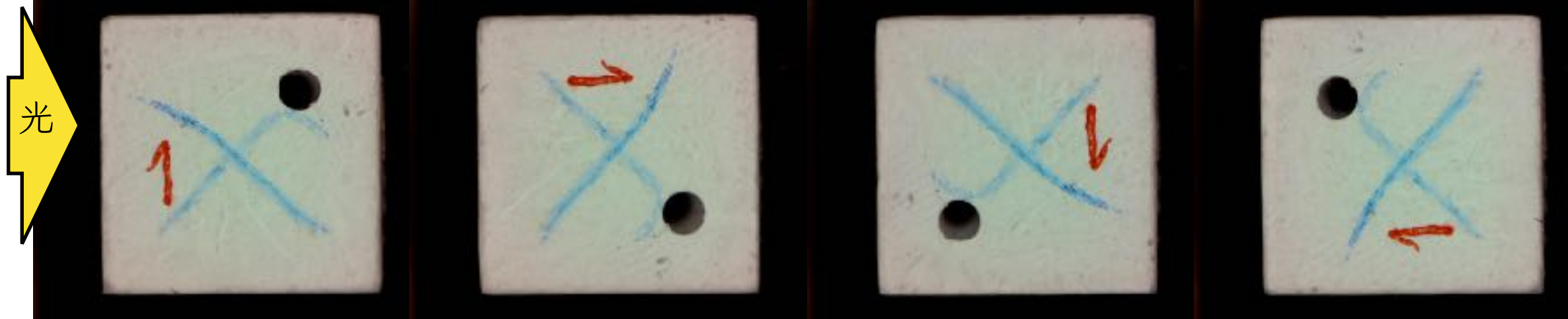
カメラ毎の 光の当たり方



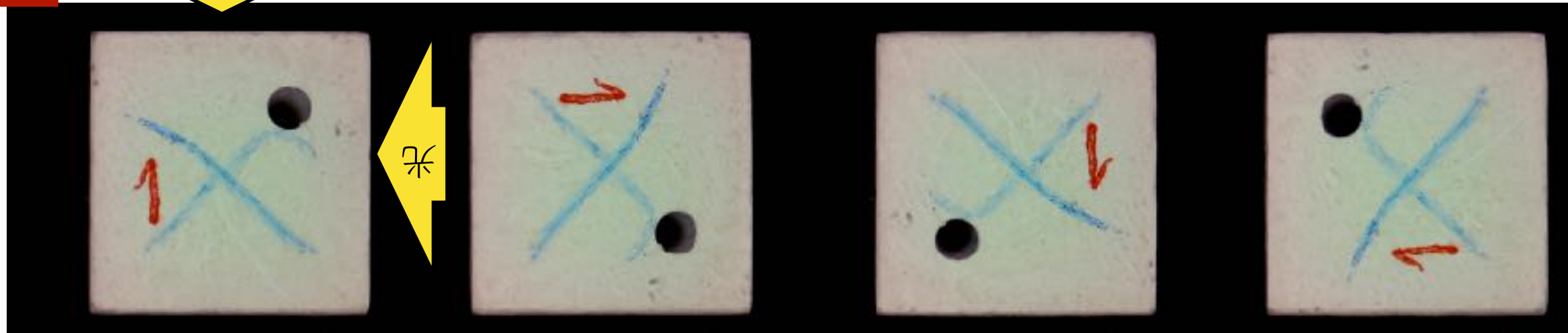
- オレンジの部分に、別方向からの光が当たる。
- 穴の辺からの位置が撮影状況によって変わってしまう
(本来は $xc1=xc2$ のはずなのに、 $xc2$ のほうが大きく見えてしまう)
- 同一の面をそれぞれのカメラで撮影、同一の辺長・穴位置について光の有無によって有意な違いがあるか確認。

同一キューブの同一面の見え方の違い(回転補正前)

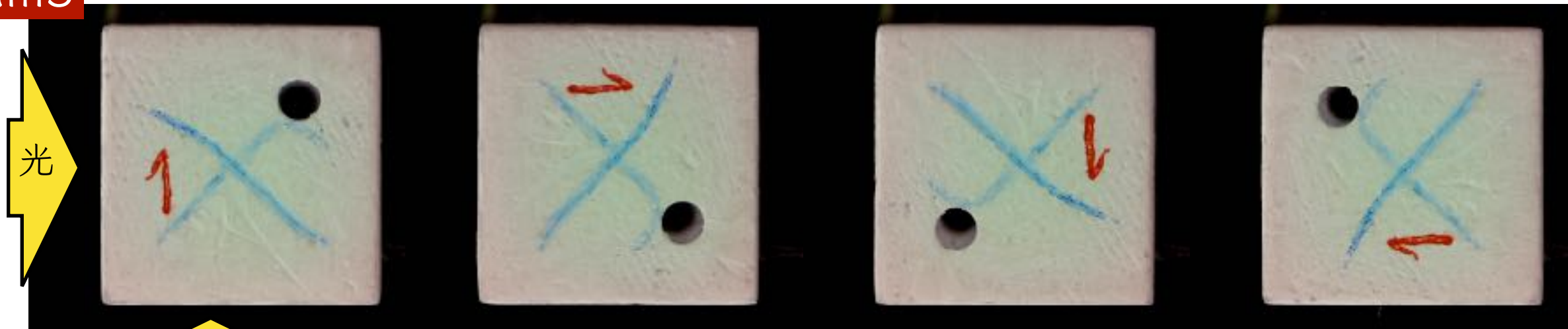
cam1



cam2

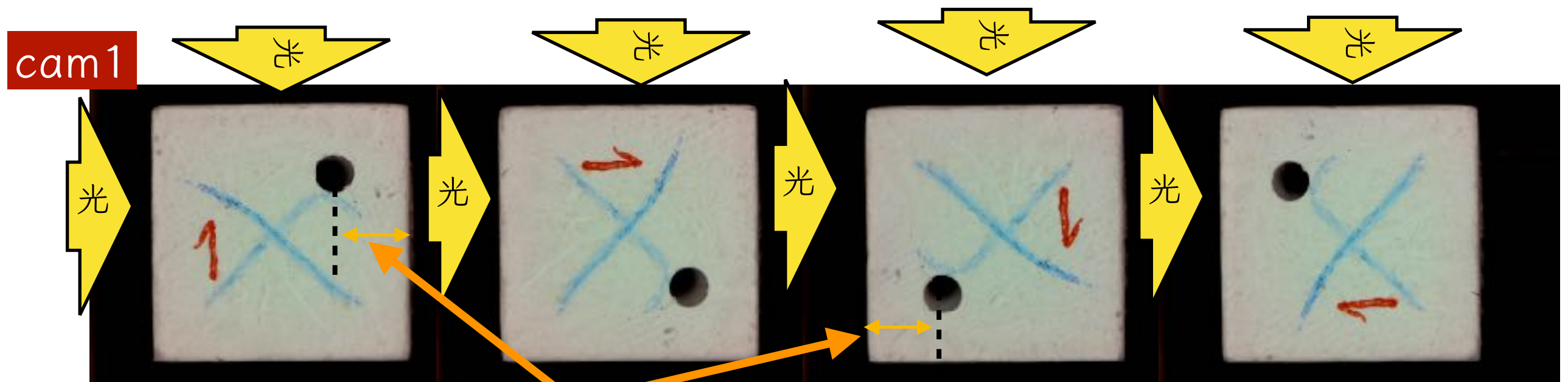


cam3



データ組について

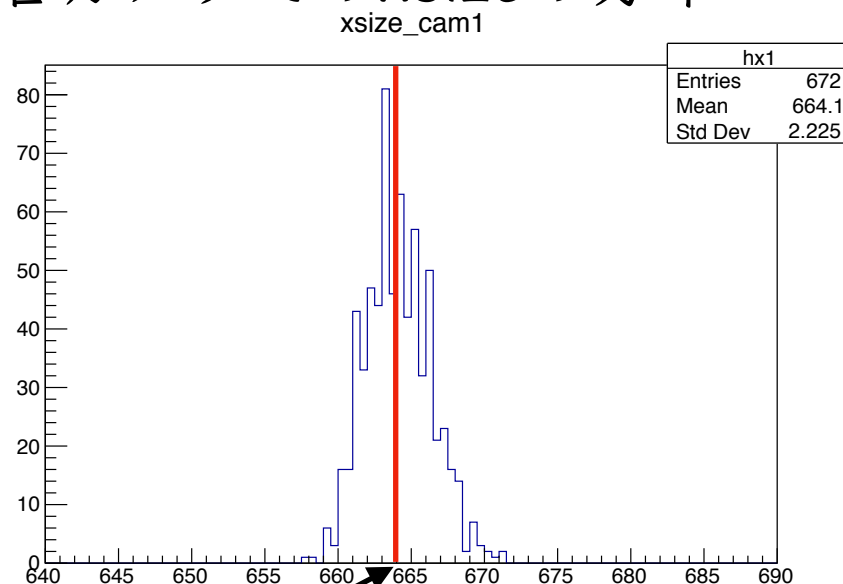
- 同一キューブにおける同じ箇所を、側面に光が当たる場合に測定した値と、側面に光の当たらない場合に測定した値の相関を調べる。
- 今回は、各カメラ毎の、同じ箇所の測定値をひとつのデータとした（カメラ1とカメラ2の組、などは無く、カメラ1での回転したものどうしの組など）。
- 1面につき12枚の写真、1キューブで $6 \times 12 = 72$ 枚、以前頂いた good 28キューブ撮影 $72 \times 28 = 2016$ 枚 のサンプルを使用した。



例：この2つの長さを一組のデータにする（y長さでも同様）

- 実際には、穴の位置の違いを見る以前に、**キューブの大きさ**がどのカメラで見ても一定であることが保証される必要がある。
- xsize, ysizeともに12枚の写真で一定であるかどうか確認する。
- カメラごとの違いを除くために、各カメラでのxsize,ysize の平均で割った値を用いる。

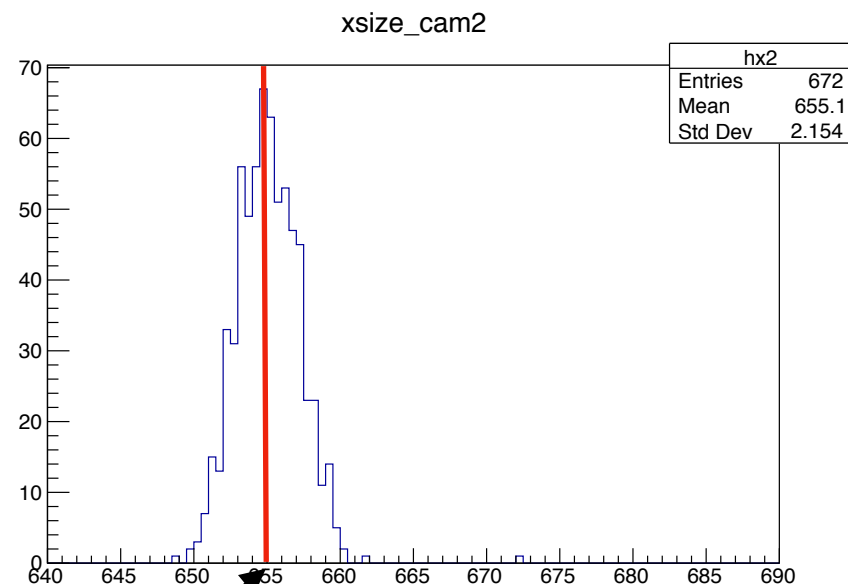
各カメラでのxsizeの分布



xsize(pixel); cam1

cam1の平均xsize: 664.1pix

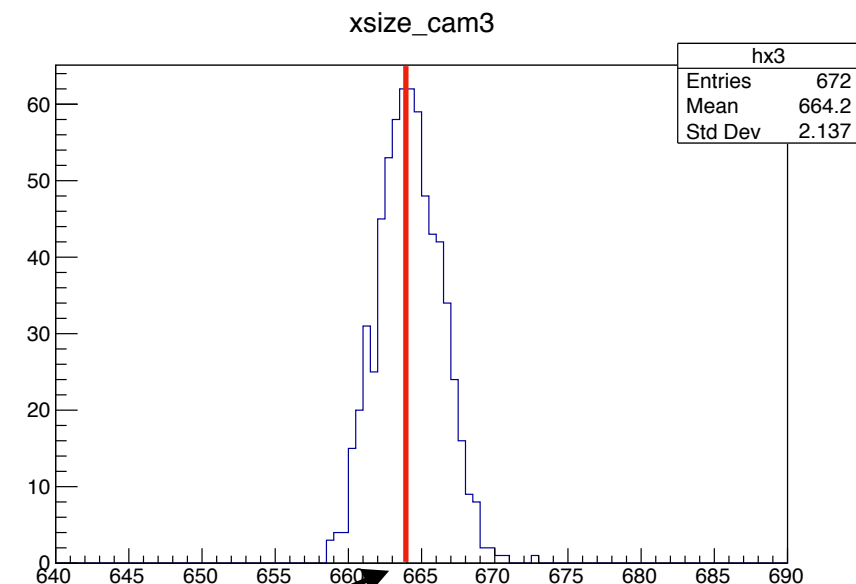
約15.4 μ m/pix



xsize(pixel); cam2

cam2の平均xsize: 655.1pix

約15.6 μ m/pix



xsize(pixel); cam3

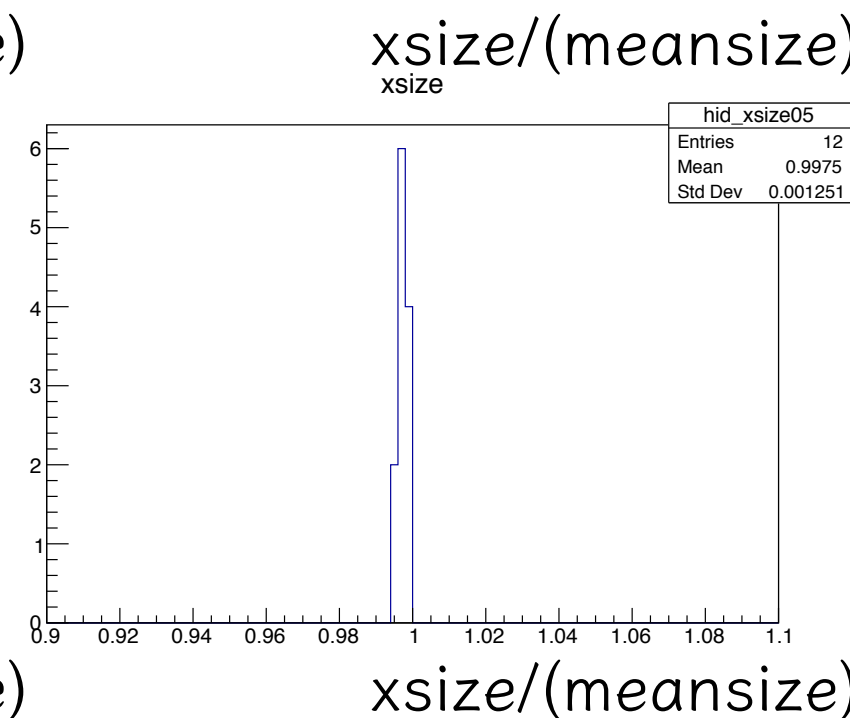
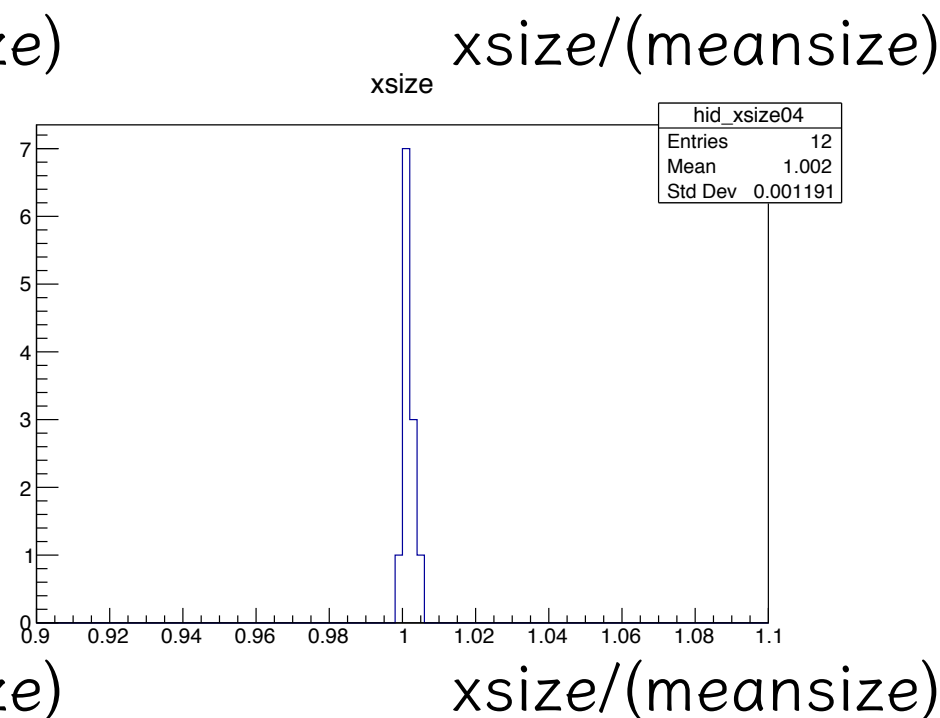
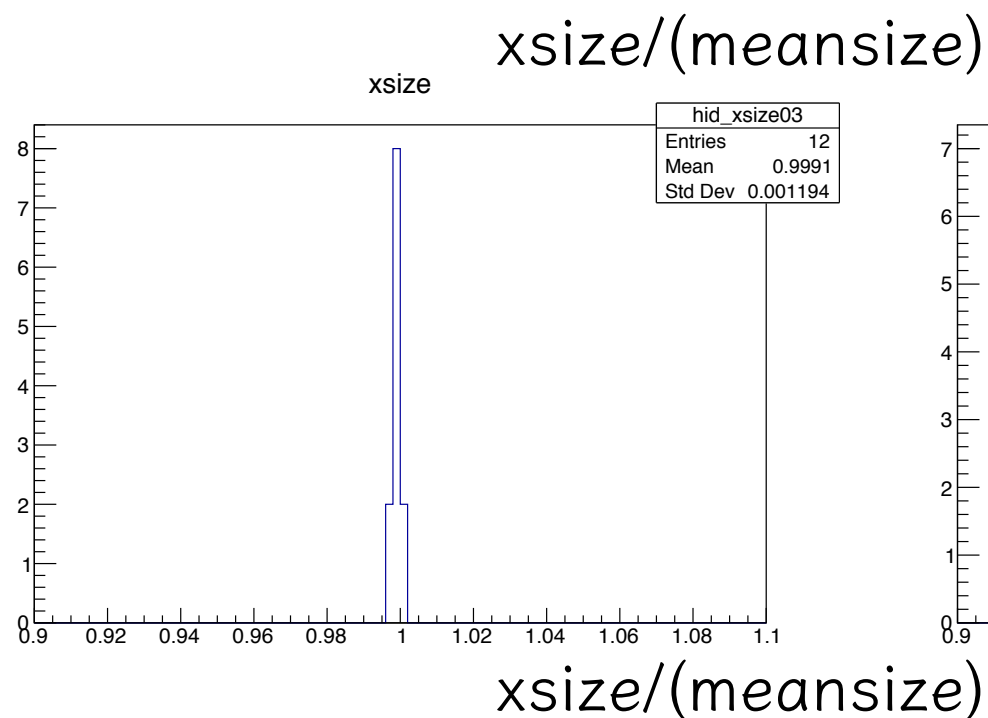
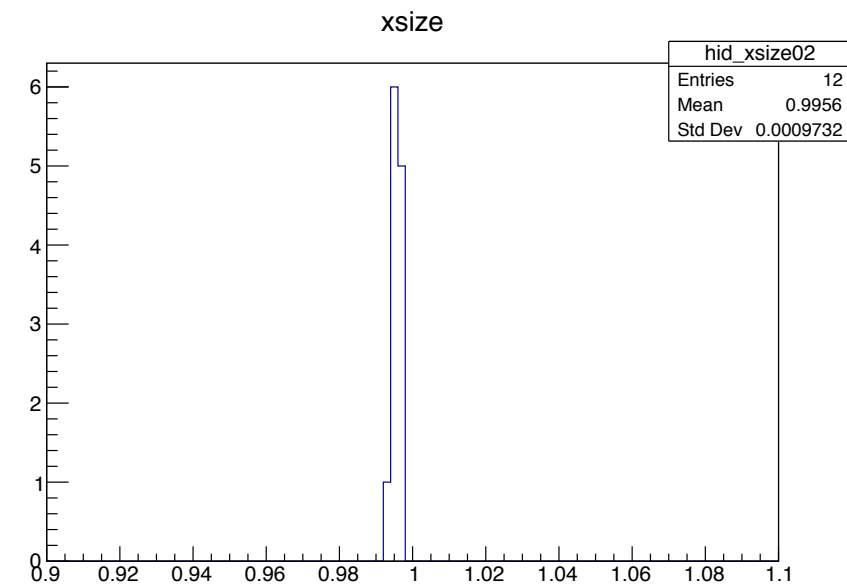
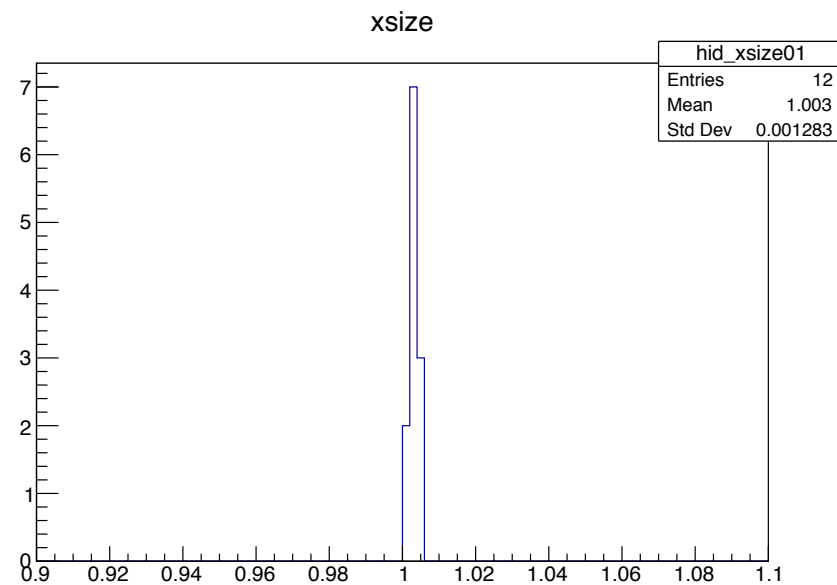
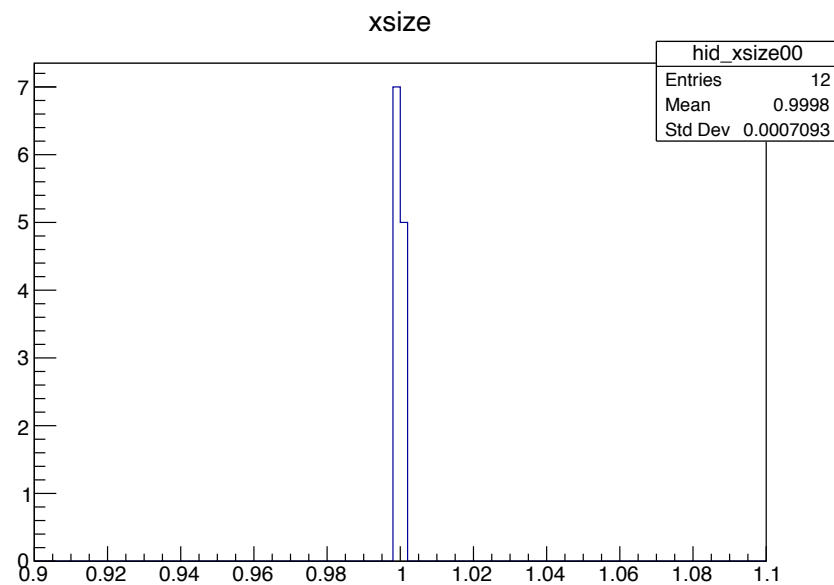
cam3の平均xsize: 664.2pix

約15.4 μ m/pix

以降は各カメラでのサイズの平均値で割った値を横軸に用いる。

- xsize, ysizeともに12枚の写真で一定であるかどうか確認する。
- カメラごとの違いを除くために、各カメラでのxsize,ysize の平均で割った値を用いる。単位は1=1.02 cm(=キューブサイズの平均)

良い例：キューブ1の6つの面のxsize の、（カメラ毎の）平均に対する割合



- xsize, ysizeともに12枚の写真で一定であるかどうか確認する。
- カメラごとの違いを除くために、各カメラでのxsize,ysizeの平均で割った値を用いる。

直線検出と、それをもとにした傾き補正の際にバグがあり、分布から外れてしまうものがまれにあった。

悪い例：キューブ3の6つの面のxsizeの、（カメラ毎の）平均に対する割合

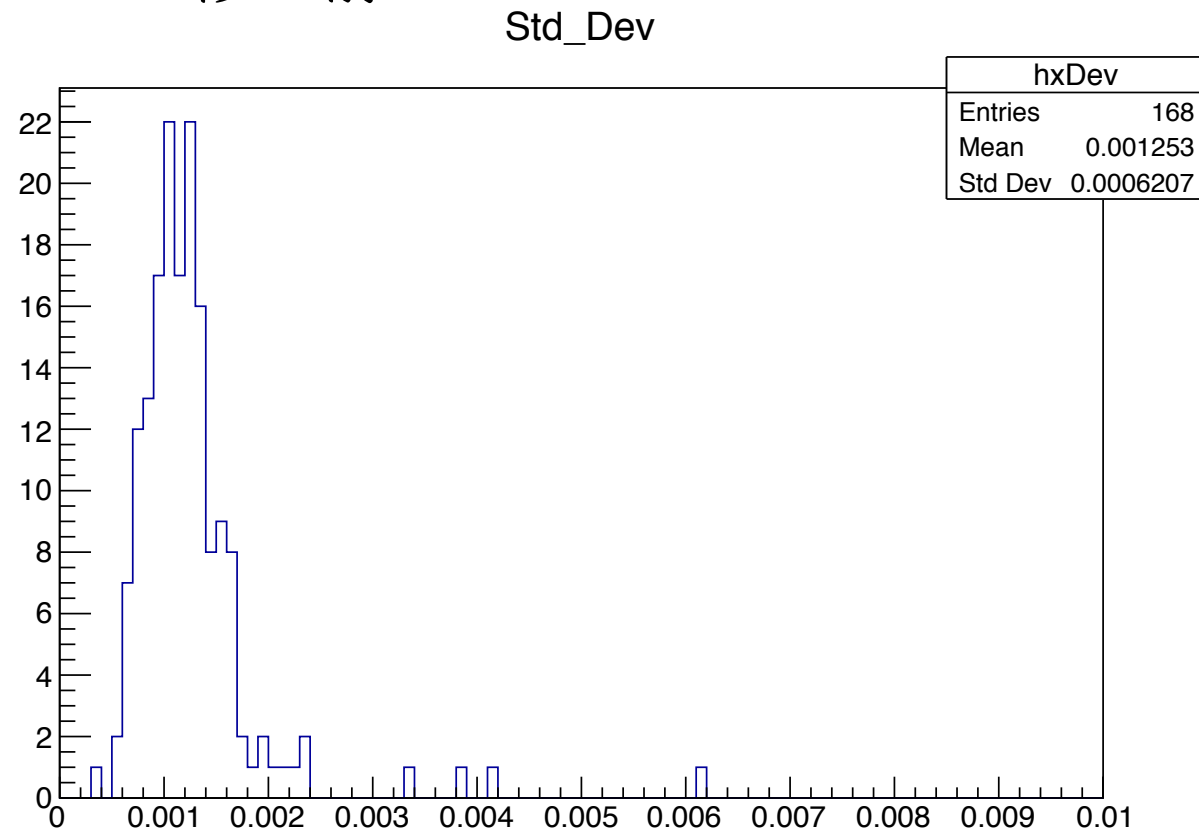


12枚の写真全てで同じように見えている場合は、Std Devは~0.002程度だが、正しく取れていない場合はこの値は大きくなる。

キューブの見え方の“広がり”についての分布

- 前ページの分布のStd Dev の分布（キューブ28個×6面）
- 解析コードを修正することにより改善 (ysize についても同様)。

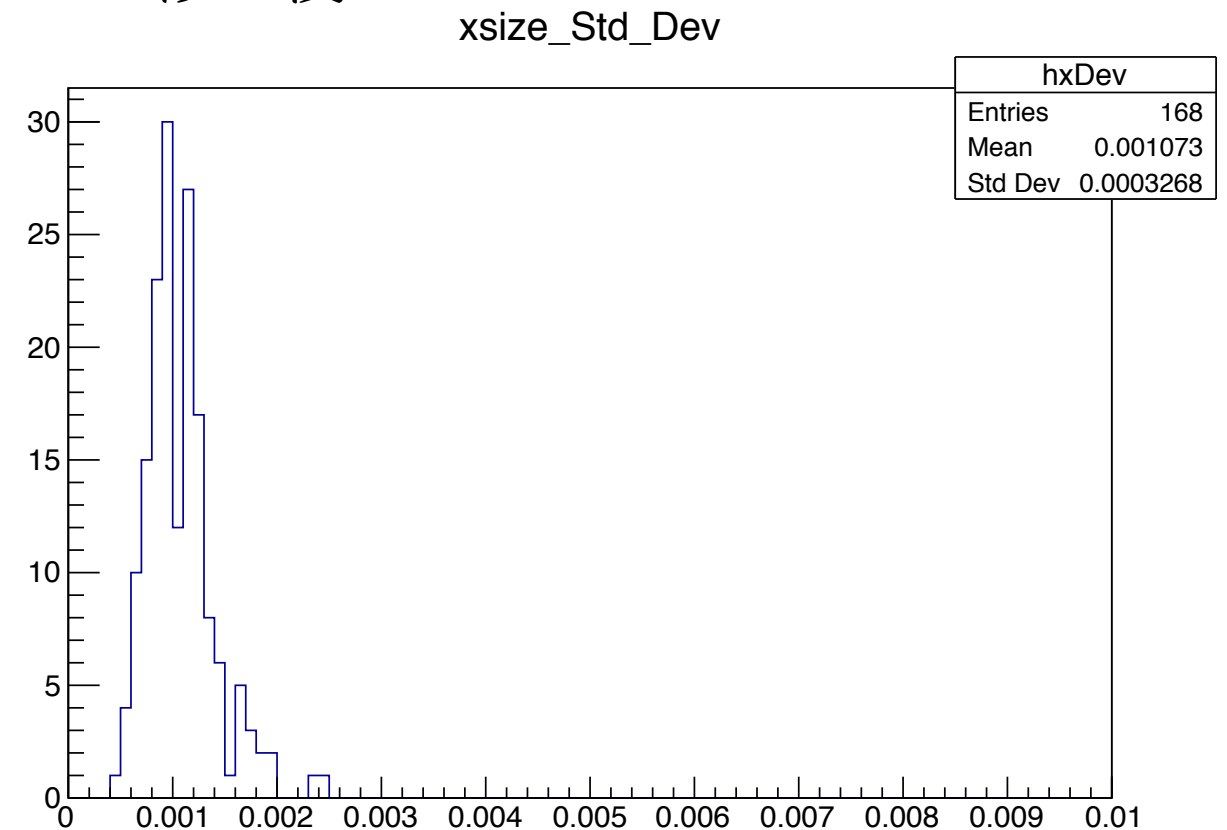
修正前



修正前

- 4イベント（全体の2%）、正しく値を取れていない。
- 平均の広がりは12 μm 程度

修正後

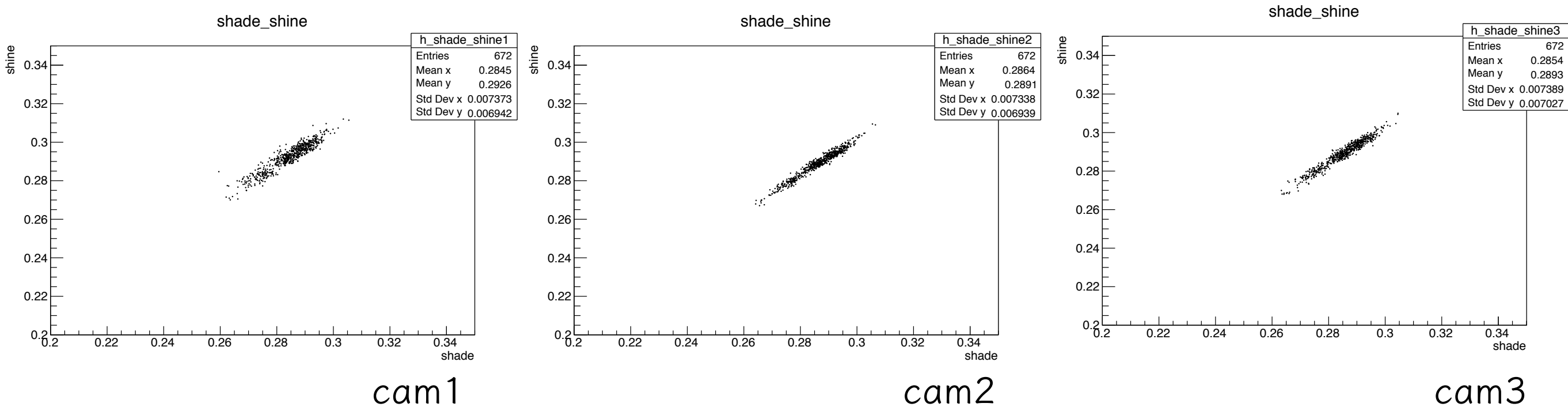


修正後

- サイズの誤認は解消された。
- 平均の広がりは11 μm 程度と小さくなった。

穴の位置の見え方の補正

- はじめの仮定では、光が当たっているか当たっていないか、さえ補正すれば、全て等しく扱えると考えていた。
- しかしデータを確認していくと、穴についてもカメラ毎の見え方に無視できない程度の違いがあることがわかった。



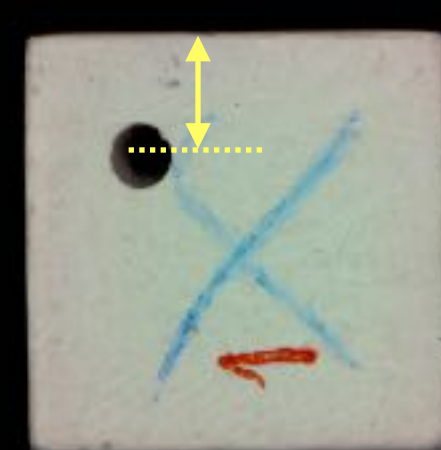
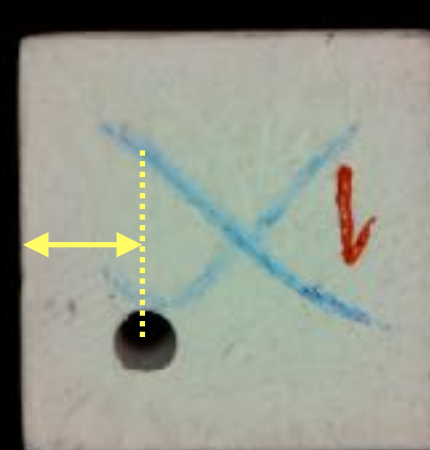
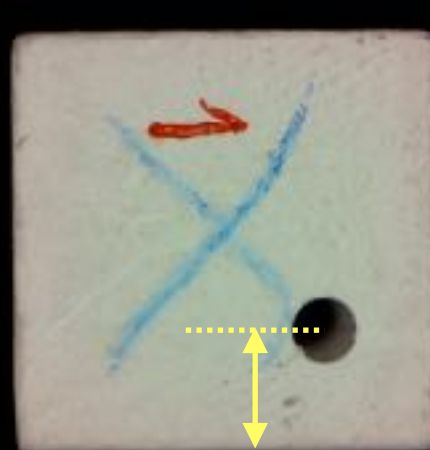
各カメラでの光の当たる/当たらないでの穴の位置の見え方
(各カメラでのサイズの平均値で割った値)

明らかに分布の形が異なる。

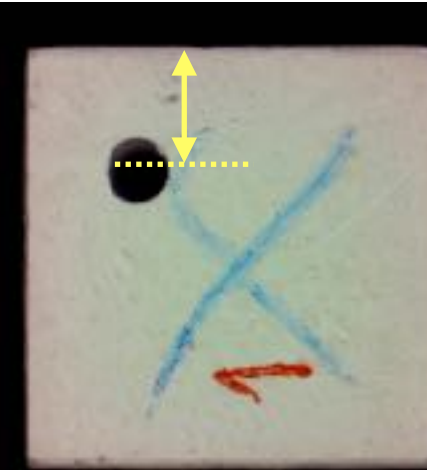
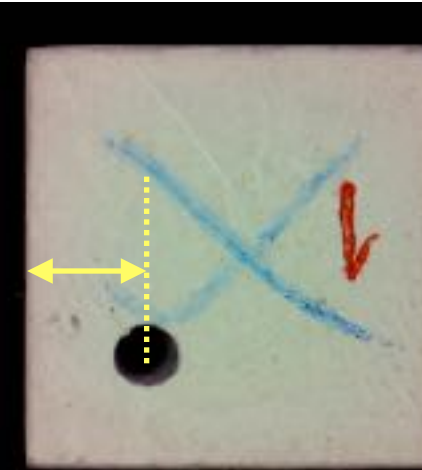
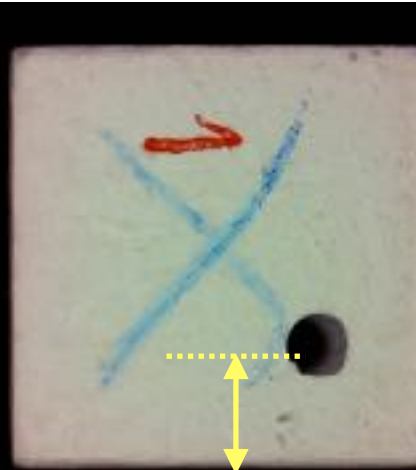
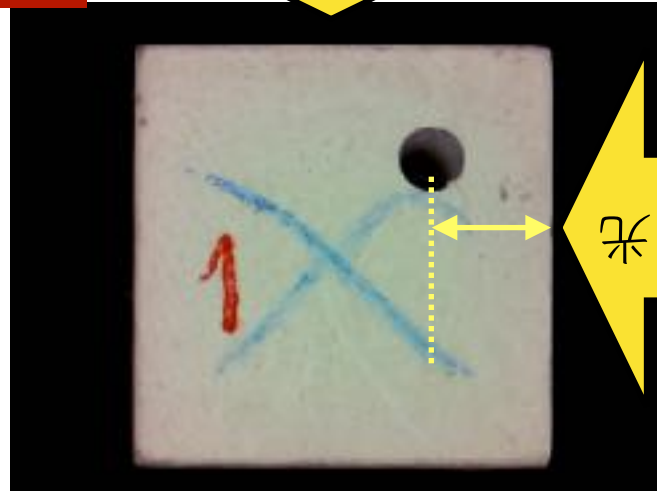
cam1 に関しては、更にその中でも分布に広がりがあるように見える。

原因の異なる見え方の違いを一度に補正：12枚の写真に対し、
全て左上の画像と見え方が一致するように補正を行う。

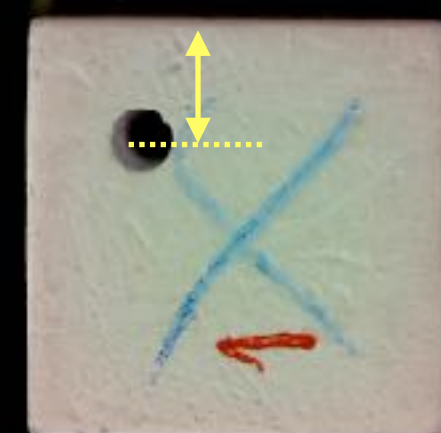
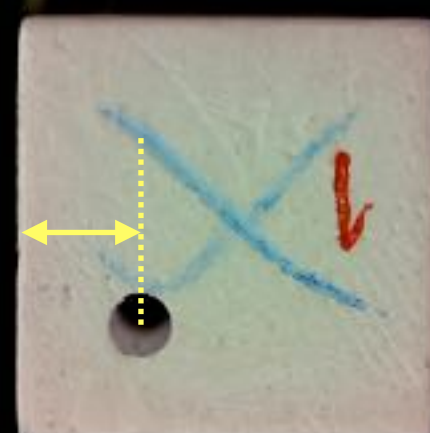
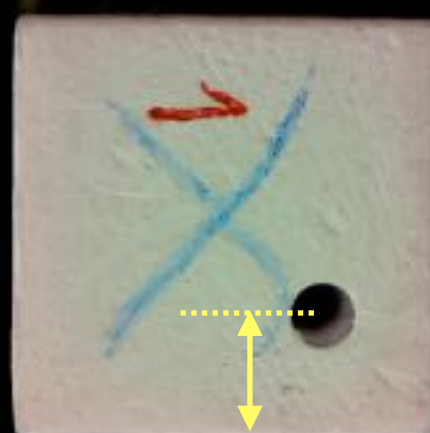
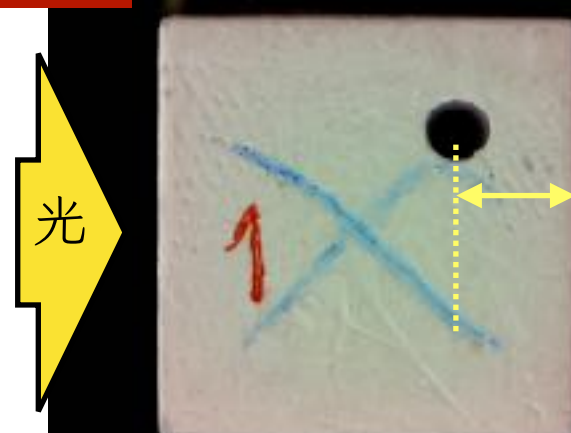
cam1



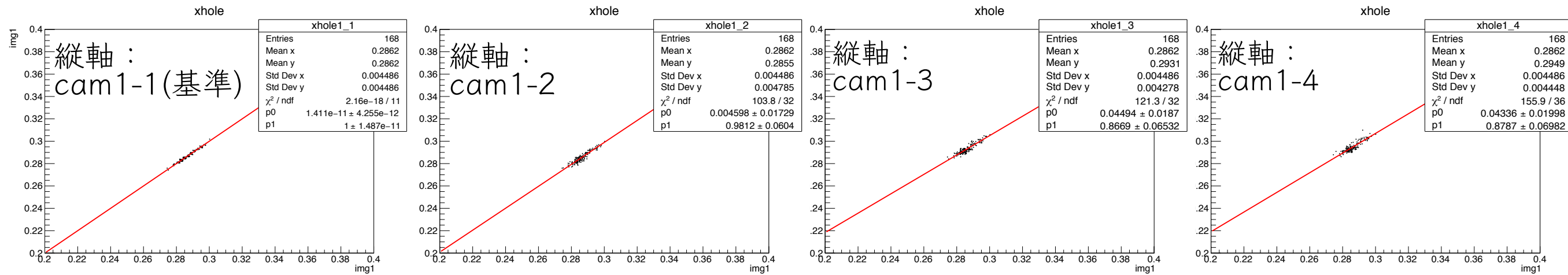
cam2



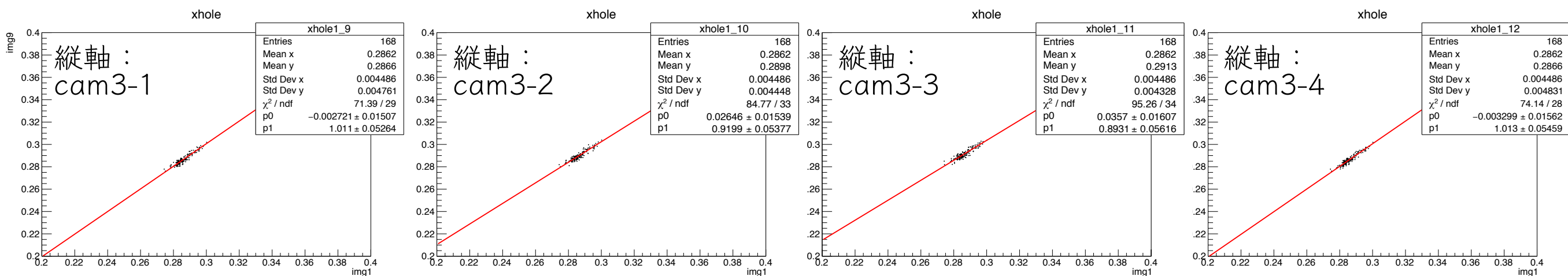
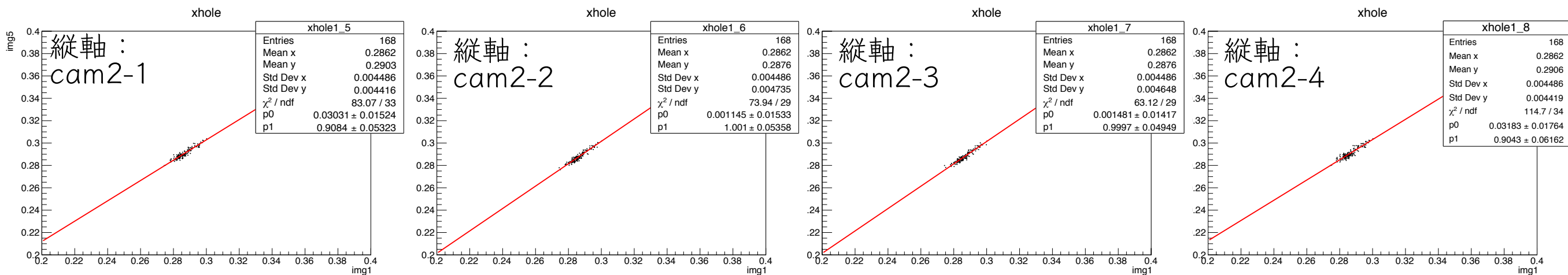
cam3



- 穴の位置 xhole について、基準となるcam1での1つ目の写真とその他の写真で比較、1次関数でフィッティング

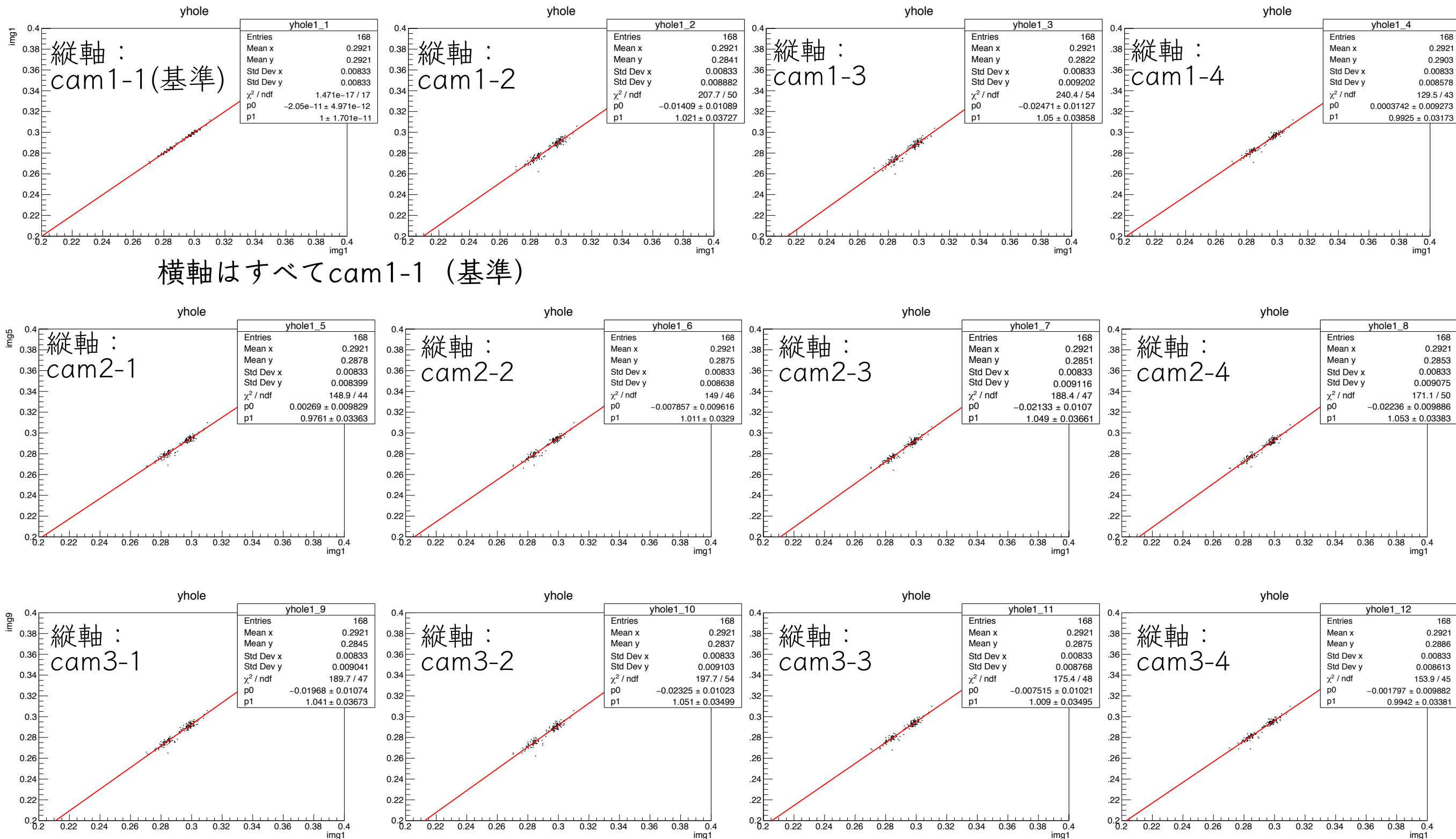


横軸はすべてcam1-1 (基準)



横軸は xhole を各カメラでのサイズの平均値で割った値。

- 穴の位置 yhole についても同様、1次関数でフィッティング
どの図でも2つ山があるように見える(18ページ参照)

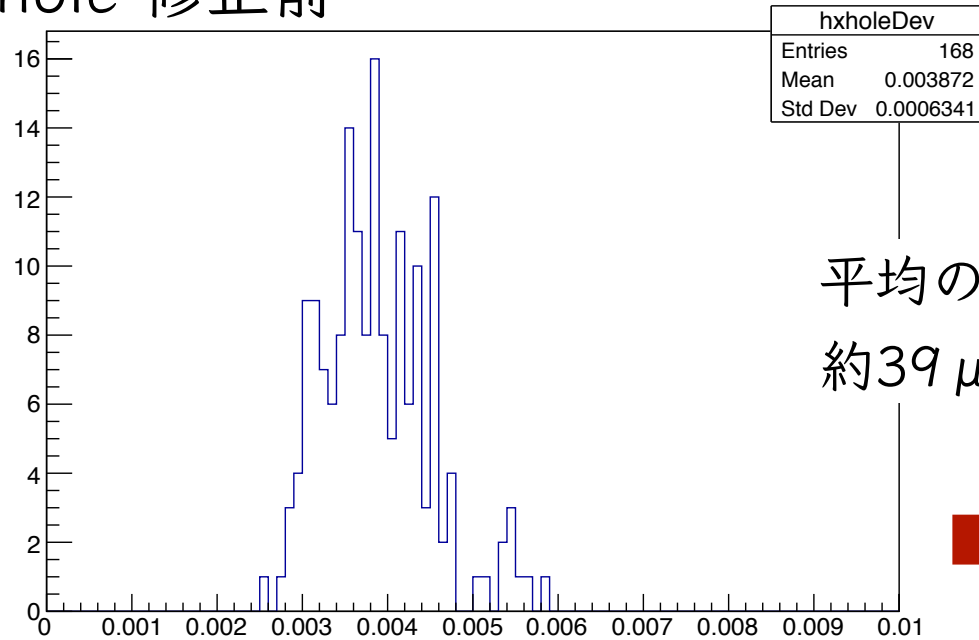


横軸は yhole を各カメラでのサイズの平均値で割った値。

キューブの穴の見え方の“広がり”についての分布

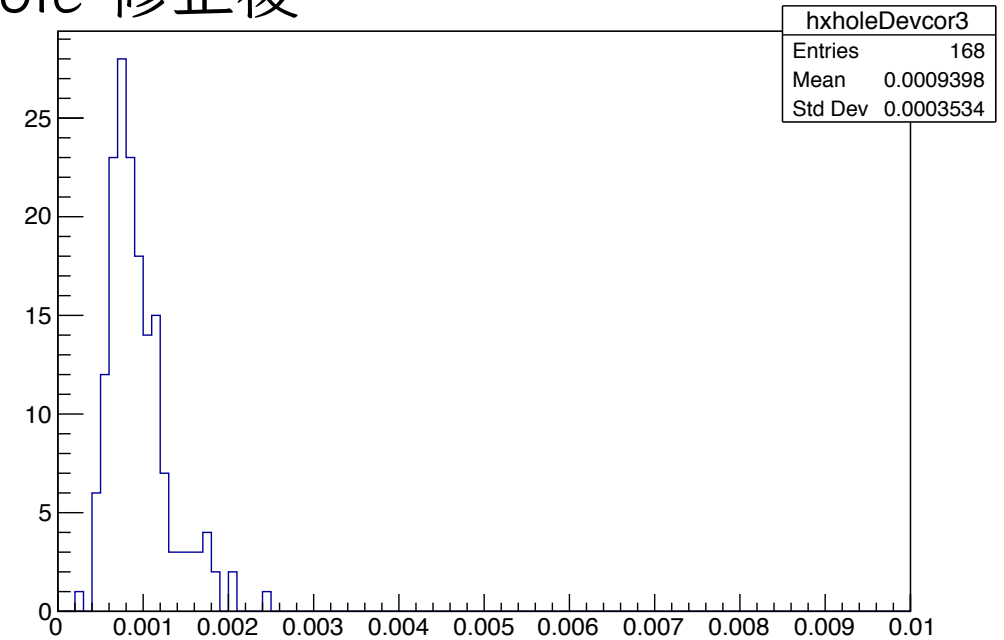
- キューブの穴の見え方についても、Std Dev の分布（キューブ 28個×6面）
- 解析コードを修正することにより改善。

xhole 修正前 xhole_Std_Dev

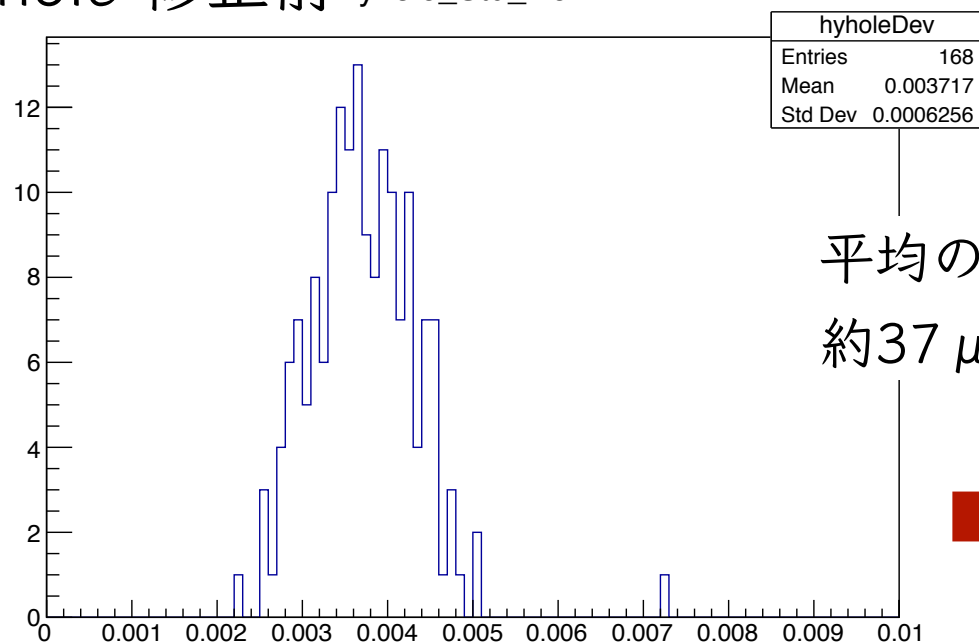


平均の広がり：
約39 μm → 約9 μm

xhole 修正後 xhole_Std_Devcor3

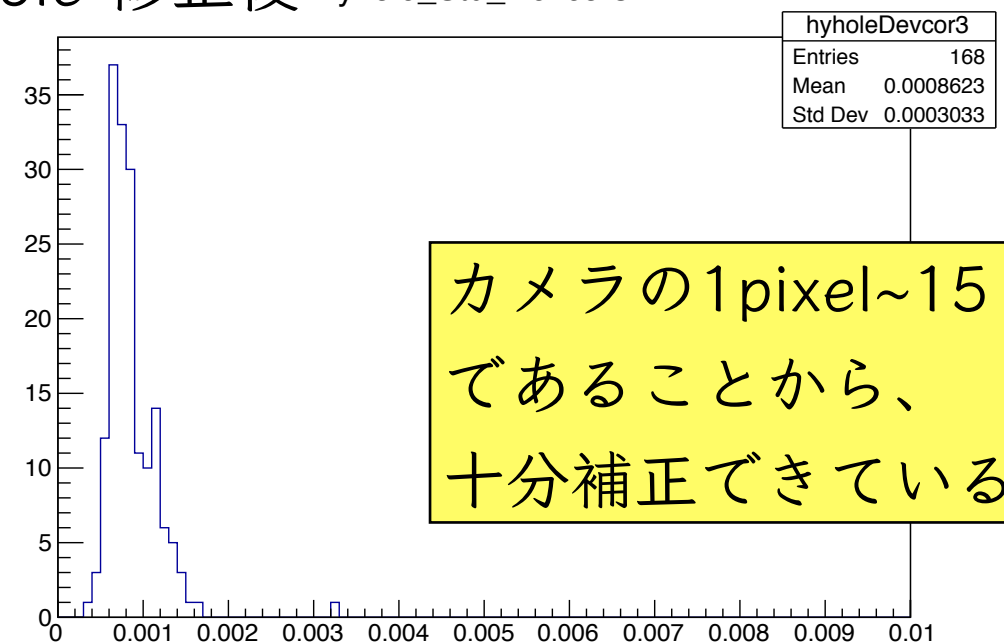


yhole 修正前 yhole_Std_Dev



平均の広がり：
約37 μm → 約9 μm

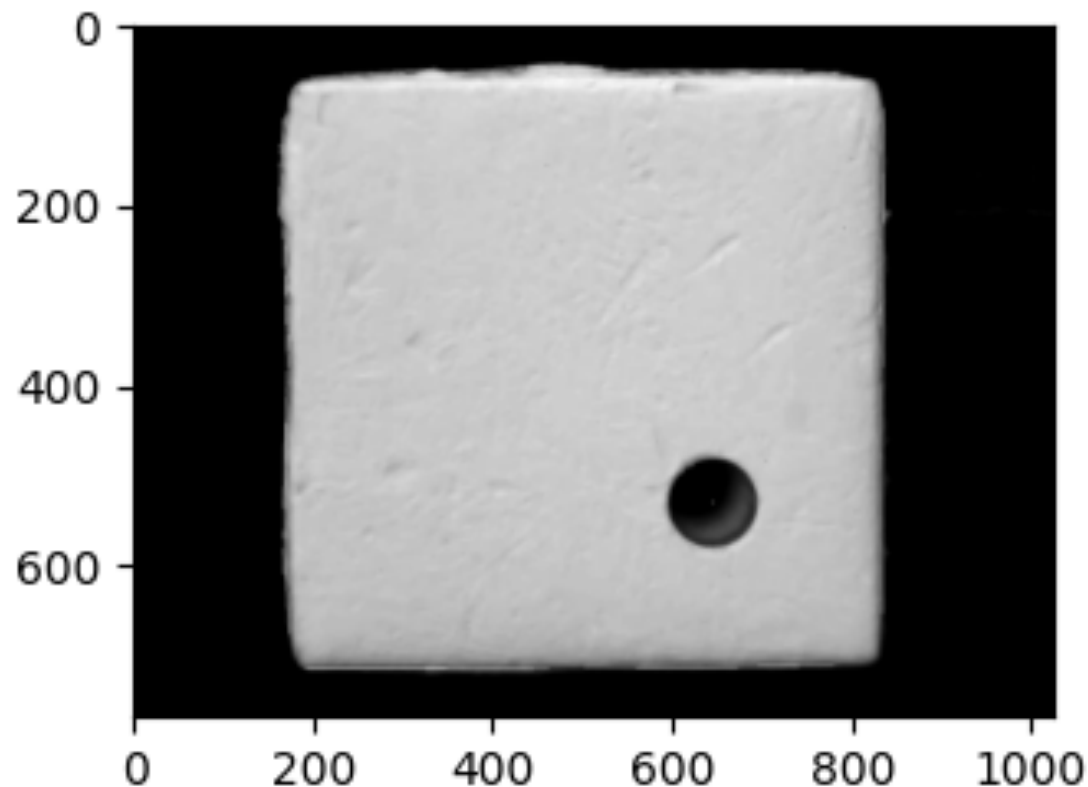
yhole 修正後 yhole_Std_Devcor3



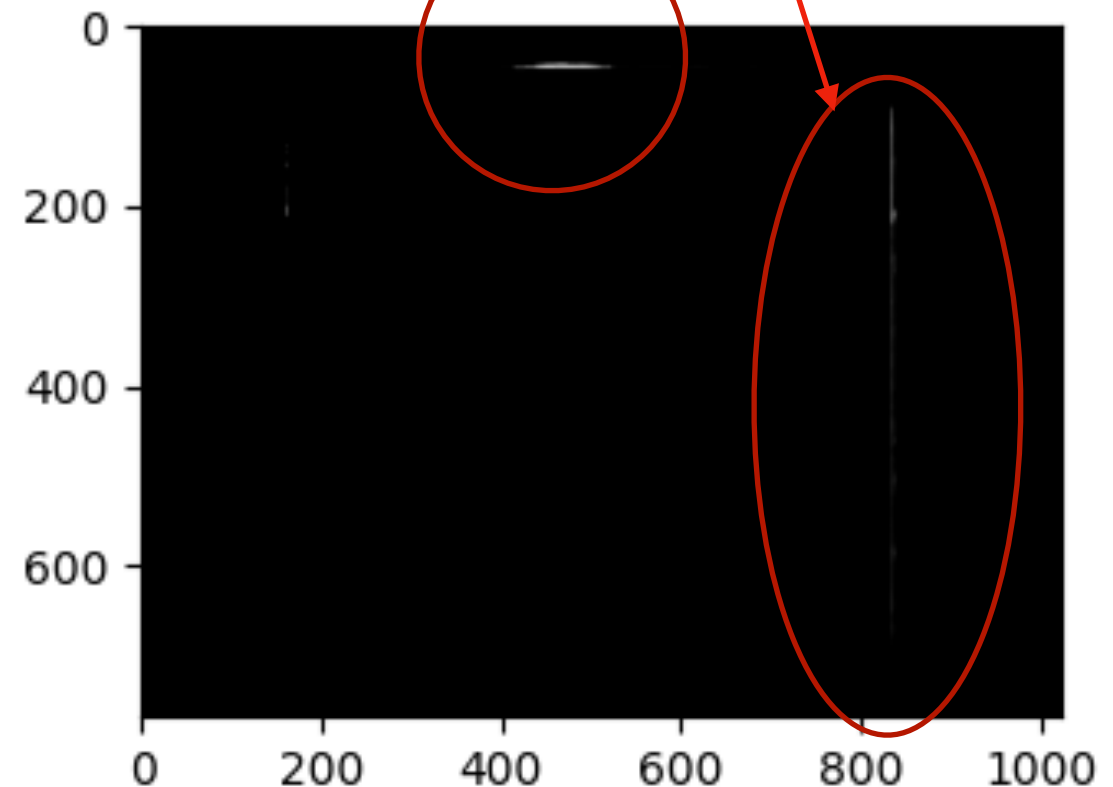
カメラの1pixel~15 μm
であることから、
十分補正できている

残りの課題

- 出っ張り検出法の確立
 - 輪郭検出で見たキューブの輪郭と、直線検出で見たキューブの辺を比較、はみ出た白い部分の面積を求める
- 撮影システムの製作
 - カメラ等の追加購入はした。
 - カメラの角度をつけた固定方法はまだ考えていない。
- 学振



この出っ張りは
正しく取れている



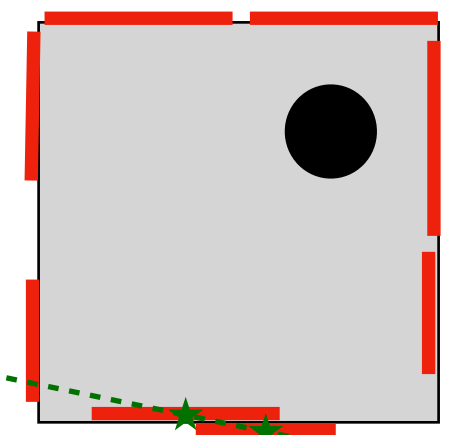
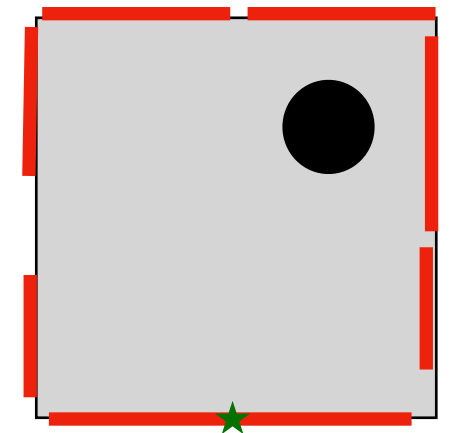
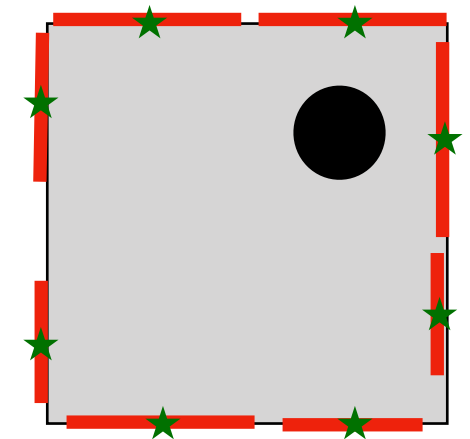
キューブの縁の白い部分を見てしまっている

back up

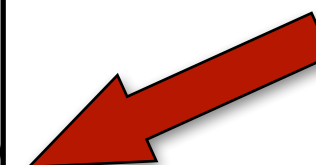
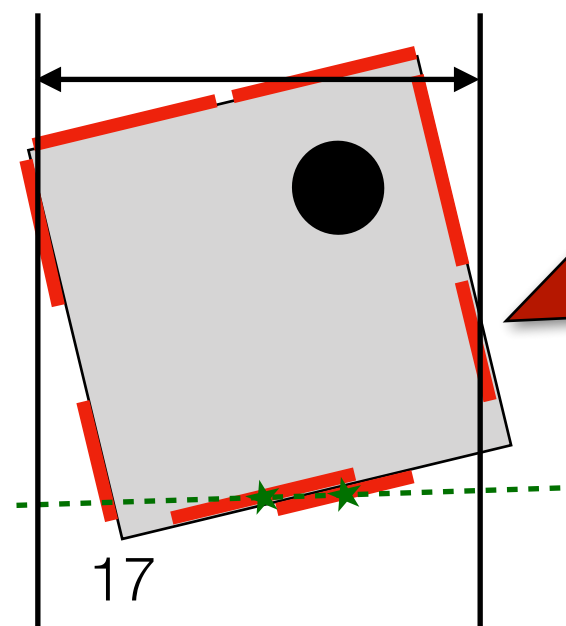
傾き補正の際のキューブの辺検出

★：各直線を中心

- 全ての辺について2本以上直線として検出されれば各辺の傾きが定義できる。
- ある辺について直線が1本のみ検出された場合は、その辺については傾きを計算しない（残りの辺の傾きだけを用いる）。
- 問題：
ある辺について直線が2本以上”重なって”検出された場合、2つの中心の点が近くなり、見かけの傾きがとても大きくなってしまう。



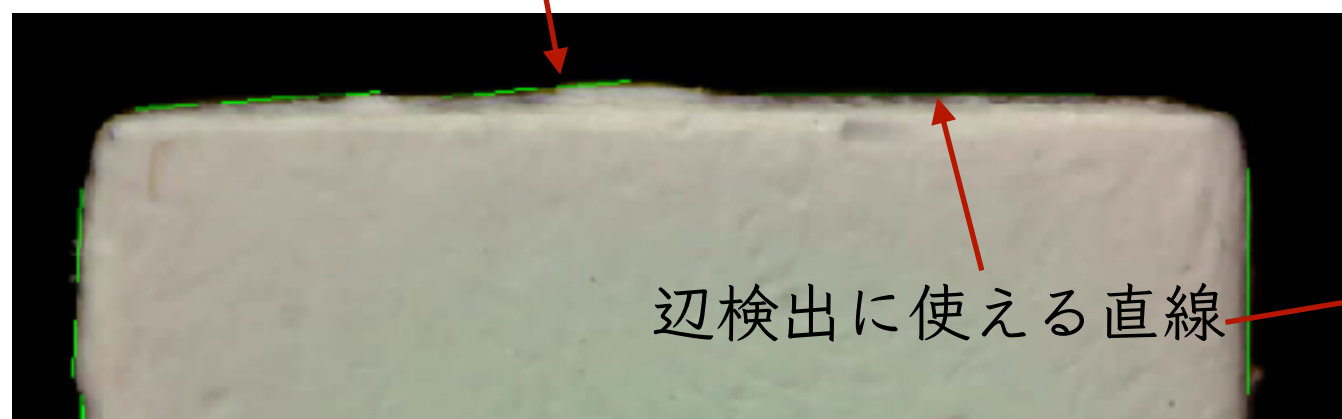
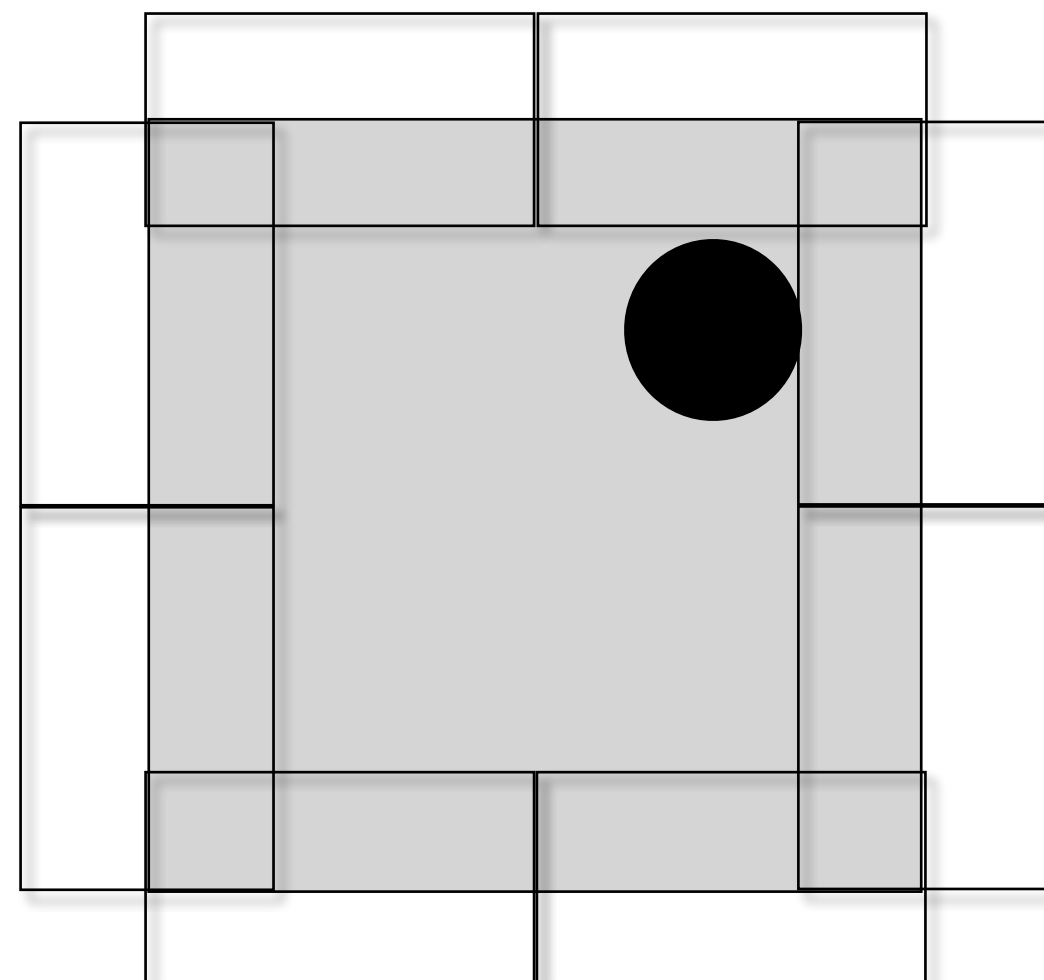
見かけの傾きに対して補正を行うと
失敗してしまう



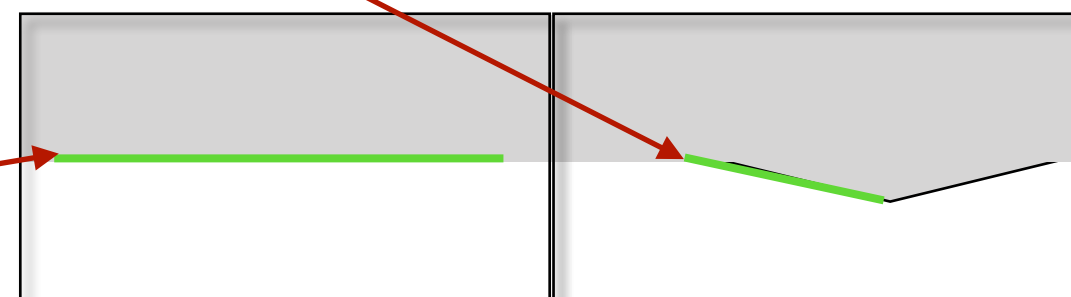
見かけの傾き

解決策：直線検出領域を分割、窓ごとに検出

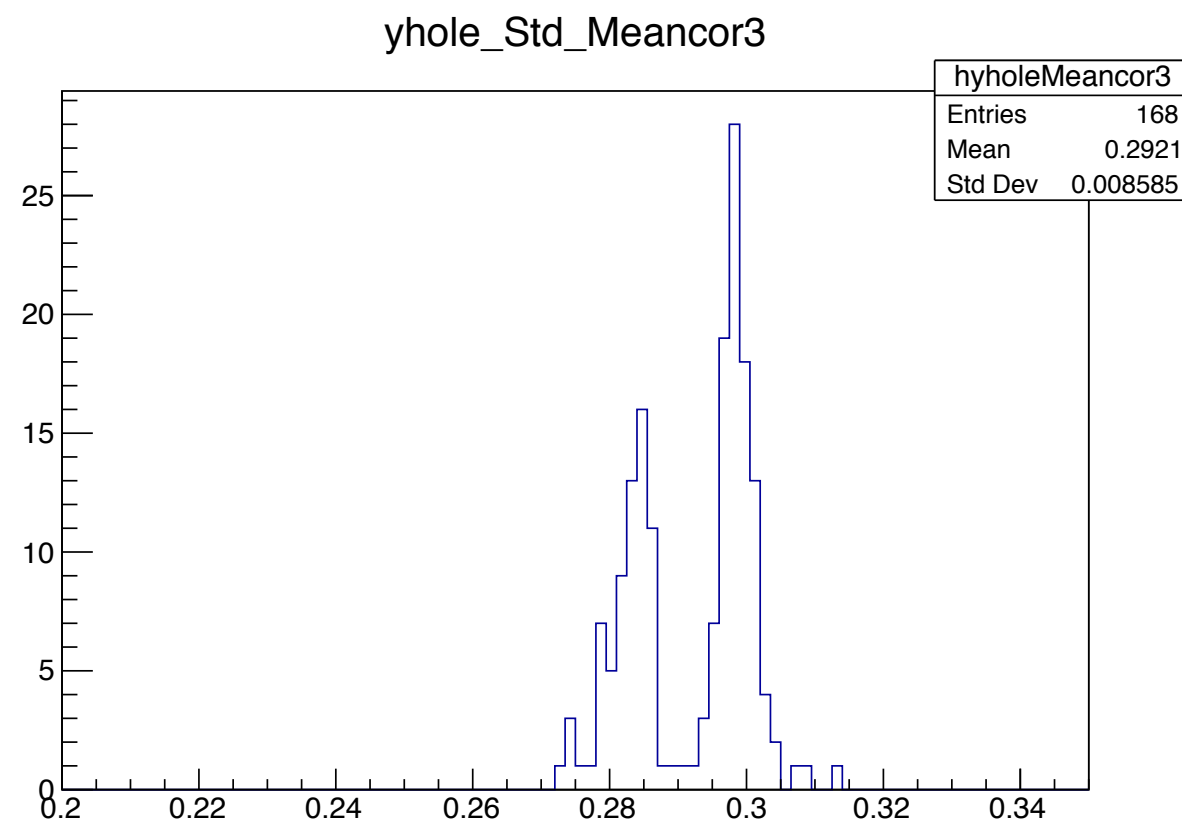
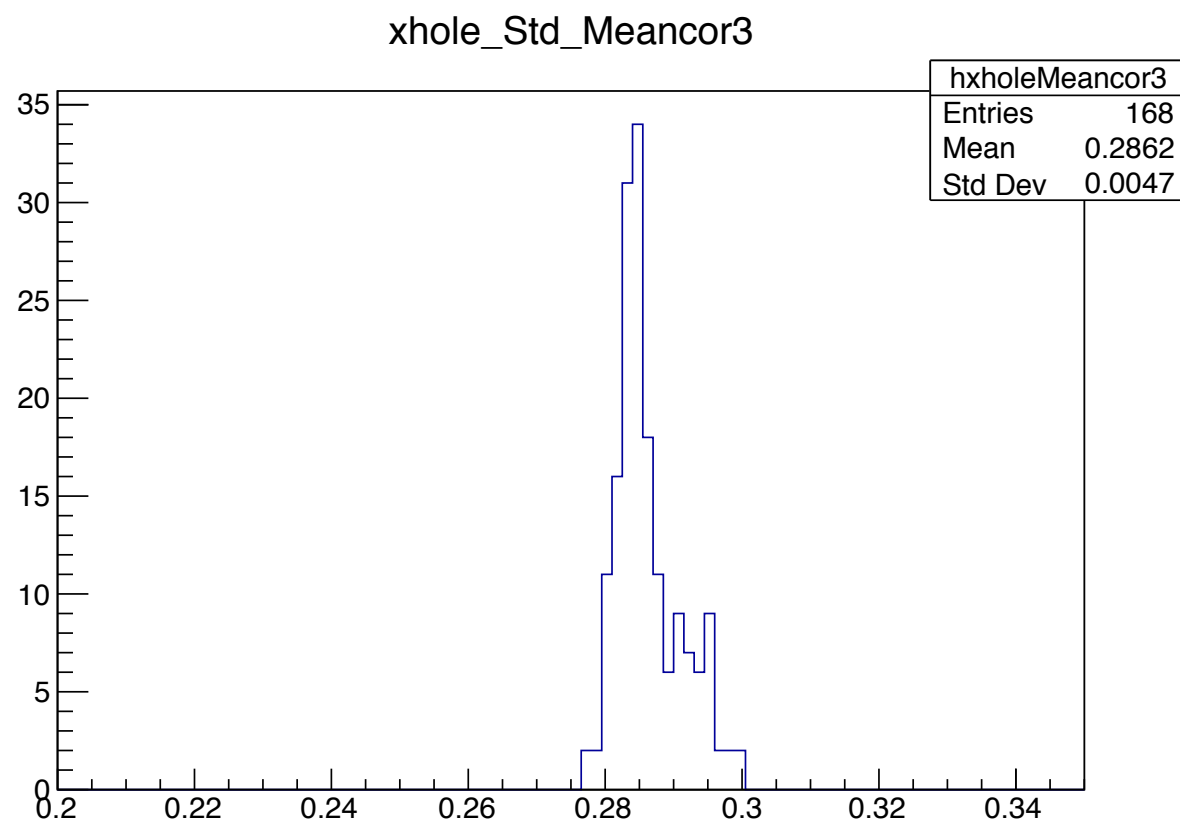
- 今までは1つの辺につき1つ以上直線検出すればよかったので、長さに関するthreshold等、粗く設定していた。
- 今回は”1つの窓につき”1つ以上の直線を検出。より慎重にthresholdを設定した。
- 新しい問題：
出っ張りの側面を直線で検出してしまう。→短い直線に対しては、斜め度合いの条件を厳しくすることで対処：
 $|x_1 - x_2| < 150\text{pixel}$ のときは $|y_1 - y_2| < 5\text{pixel}$ 、など



辺検出に使える直線

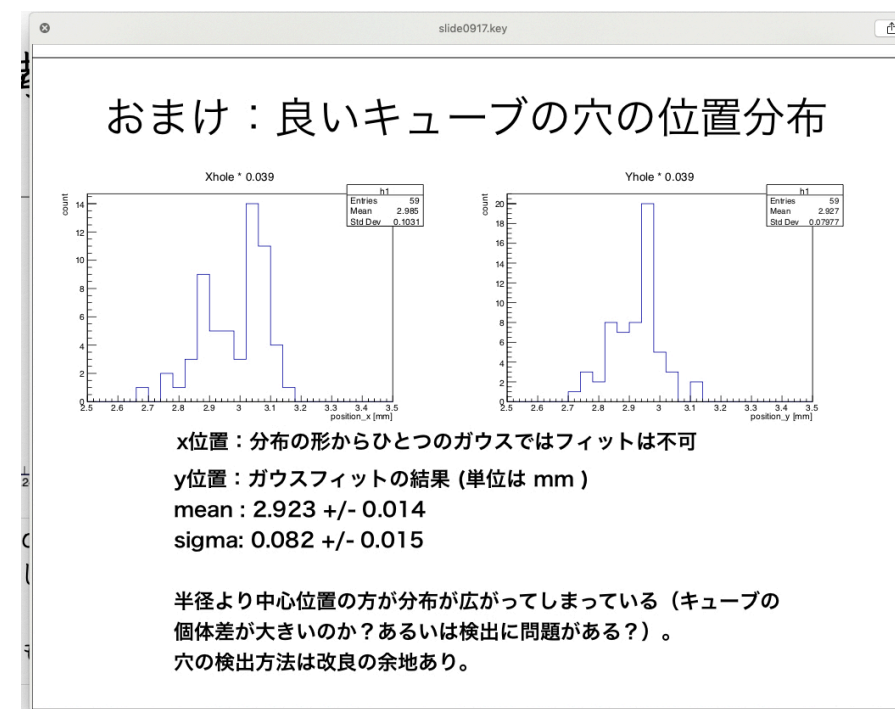


参考：補正後の xhole, yhole の分布

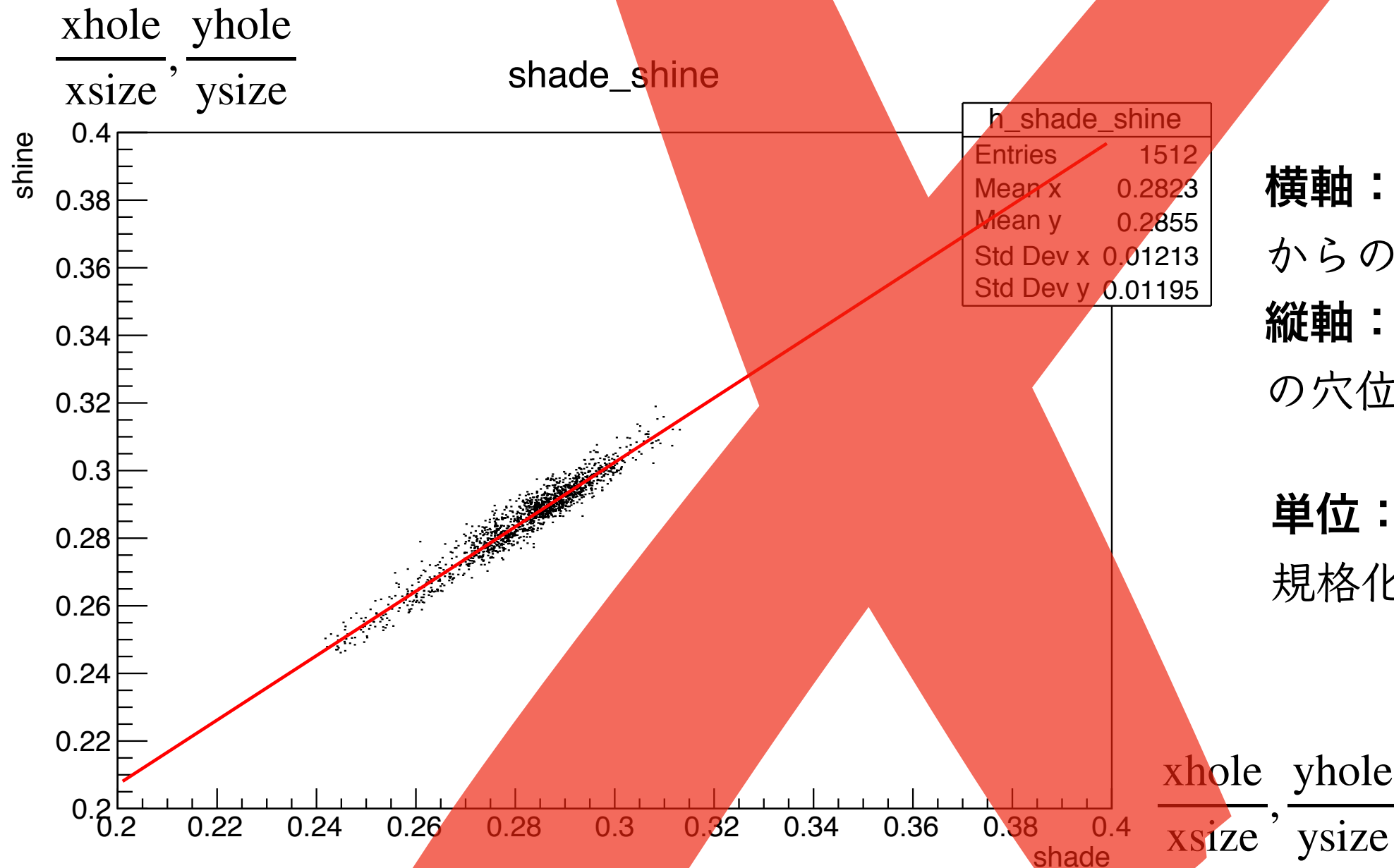


- ※xhole は光の当たらないところ、yhole は光の当たるところを基準として補正してしまったので、この絵ではMean が異なっています。
- 以前も確認した、yholeの分布に2つ山が確認される問題が現れた。

(右図：以前報告したslide0917.pdfより。このときとはxhole とyhole の定義が反対になっていることに注意)



相関、補正関数



横軸：光の当たらない方
からの穴位置の測定値

縦軸：光の当たる方から
の穴位置測定値

単位：キューブの幅で
規格化した穴の位置

相関を直線でフィット

Chi2 : 3333.5

NDf : 215

p0 : 0.0166 +/- 0.0047

p1 : 0.9529 +/- 0.0166

より意味があると考えられる補正を考え、実行したので、
次ページより、その報告を行う。