

機械学習によるガン細胞の位置・形状 の判定手法

高水裕一 梅村雅之 矢島秀伸 安部牧人 佐藤大樹
星詳子²

2) 浜松医科大学光先端医学教育研究センター

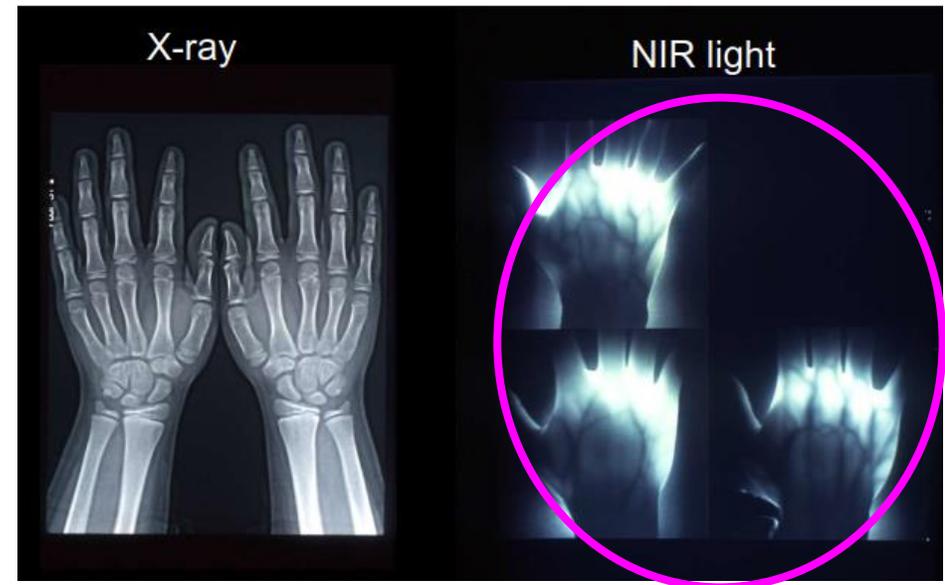
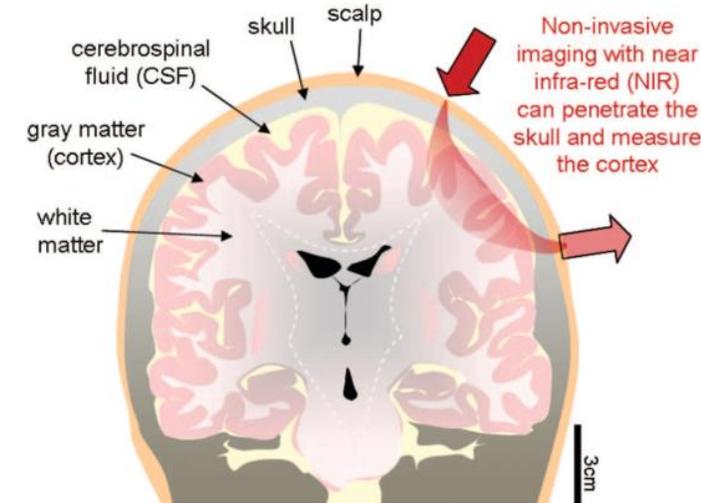
計算メディカル全体会議
2021/6/7

光バイオイメージング

- 近赤外線を体内に入射させることで、生体内の情報を**イメージング**する
- **ターゲットとなる症例**
- ガン リウマチ 脳内出血 循環障害
- **とくにガン細胞位置の検出を目指す**
食道がん 大腸がん

ガン細胞は、通常の細胞より血液量が高く、
数倍の吸収係数

- *Zero radiation Exposure* ゼロ被爆,
- *Low cost for Bedside test* 安価テスト
- *Best tool for Infants and Newborn baby*
乳幼児 新生児

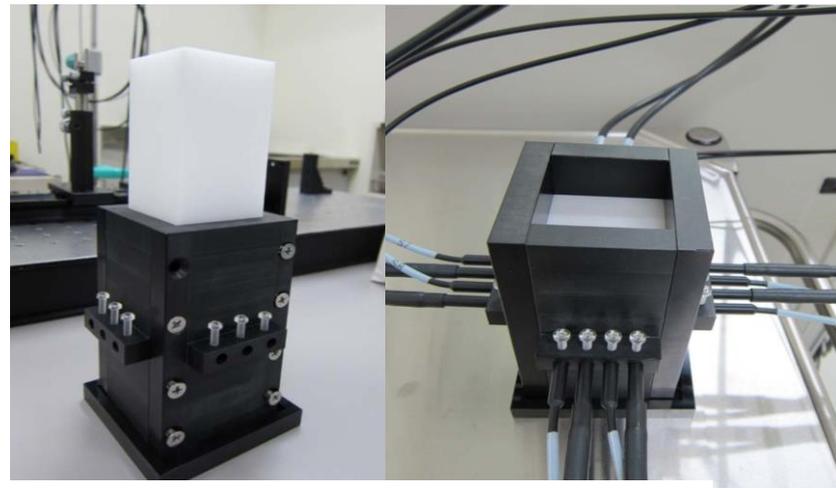


光計測モデル

生体内の光学特性値を模した物質：

ファントム

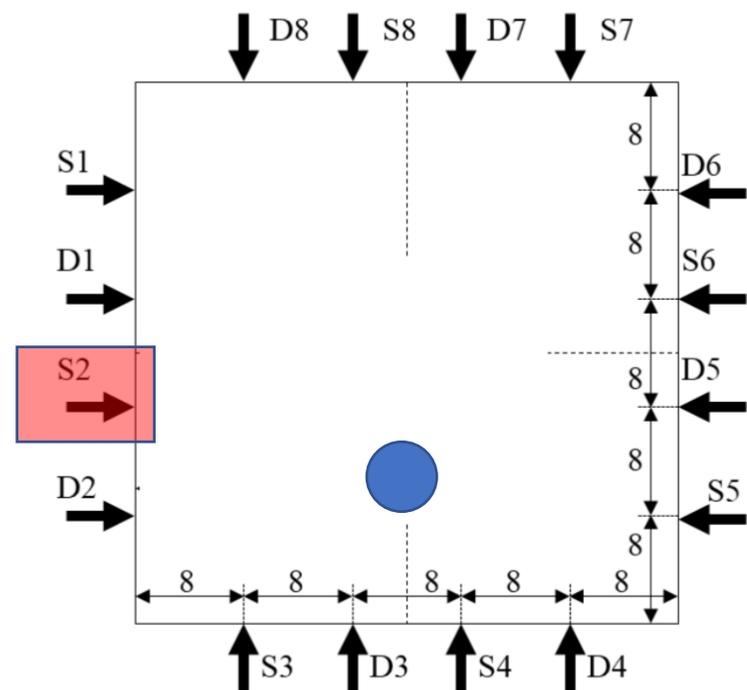
を用いた光計測モデルを作成
星先生(浜松医科大)



S:光源位置 D:検出器位置
まずS2から光を照射する場合を考える

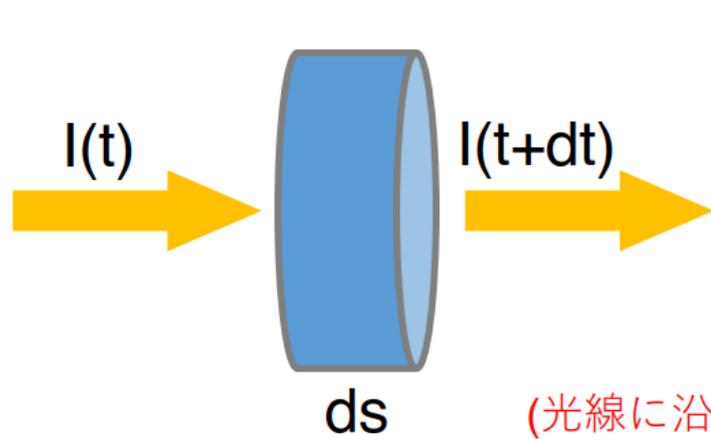
S2からの光をD1~D8の8ヶ所で検出

計算サイズ：4cm × 4cm
生体： 吸収係数 $\mu_a=0.21/cm$ 散乱係数 $\mu_s=22.45/cm$
吸収体(ガン)： 直径5mm
吸収係数 $\mu_a=0.63/cm$ 散乱係数 $\mu_s=22.63/cm$



これを模した設定を2次元シミュレーションで計算する

輻射輸送方程式



I : intensity (輻射強度)
ある時間にある場所である
方向からくる光の強さ

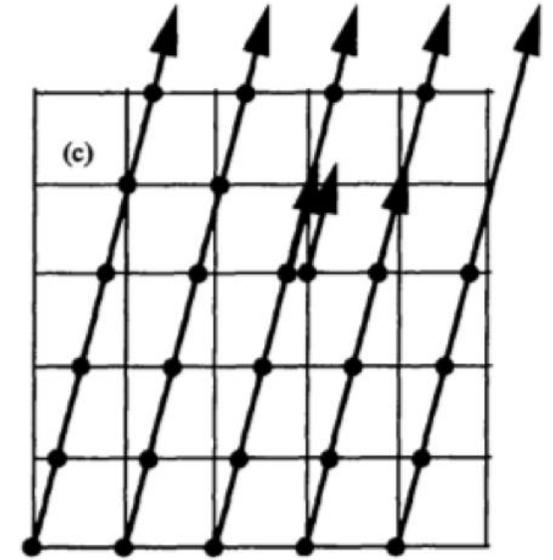
$$\frac{1}{c} \frac{\partial I}{\partial t} + \frac{\partial I}{\partial s} = -\chi I + \eta$$

減光 増光

(光線に沿った光の変化
~ラグランジュ微分)

$$\eta = \epsilon + \int \phi(\Omega, \Omega') I d\Omega'$$

- ART法 (Authentic Radiative Transfer) :
- 輻射用格子を設け、その流線上で輻射輸送計算した値を近傍の実格子に補間して与える
- 高精度で計算時間も短い



ART法の概念図

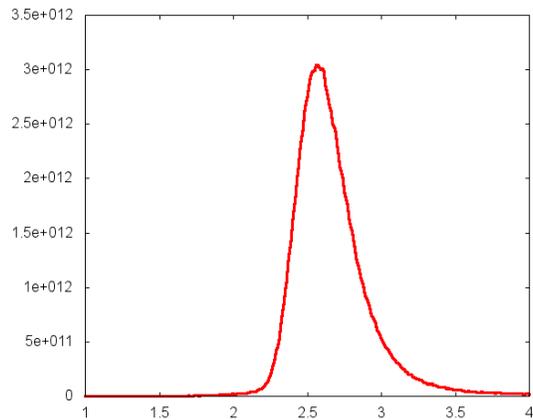
宇宙天文分野での輻射
輸送に用いられている
手法

輻射輸送方程式の形式解 :

$$I(\tau) = I(0)e^{-\tau} + \int_0^\tau e^{-(\tau-\tau')} \mathbf{S}(\tau') d\tau'$$

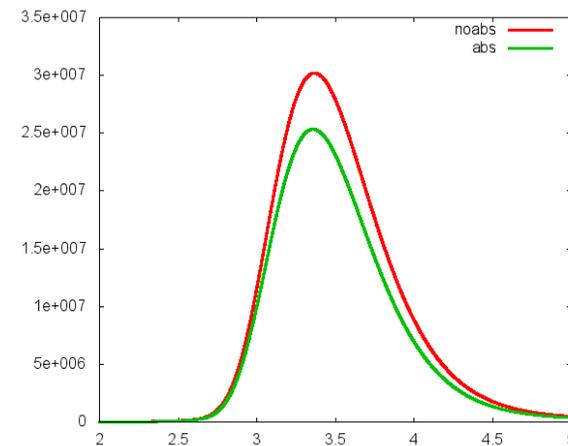
コード開発 : 矢島、安部(筑波大学)

入射波

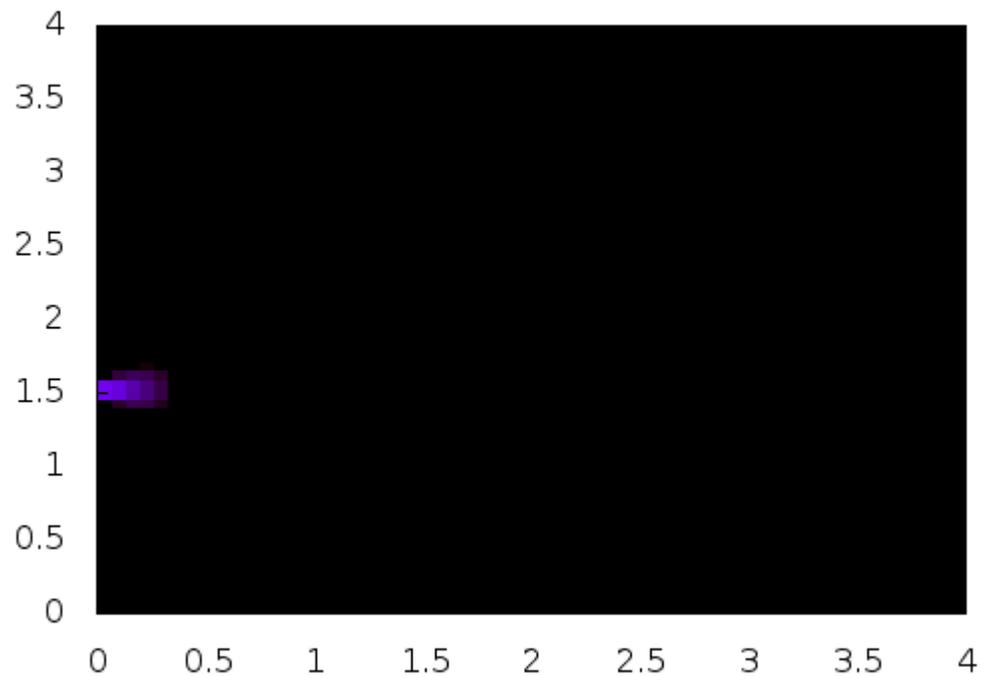


検出波

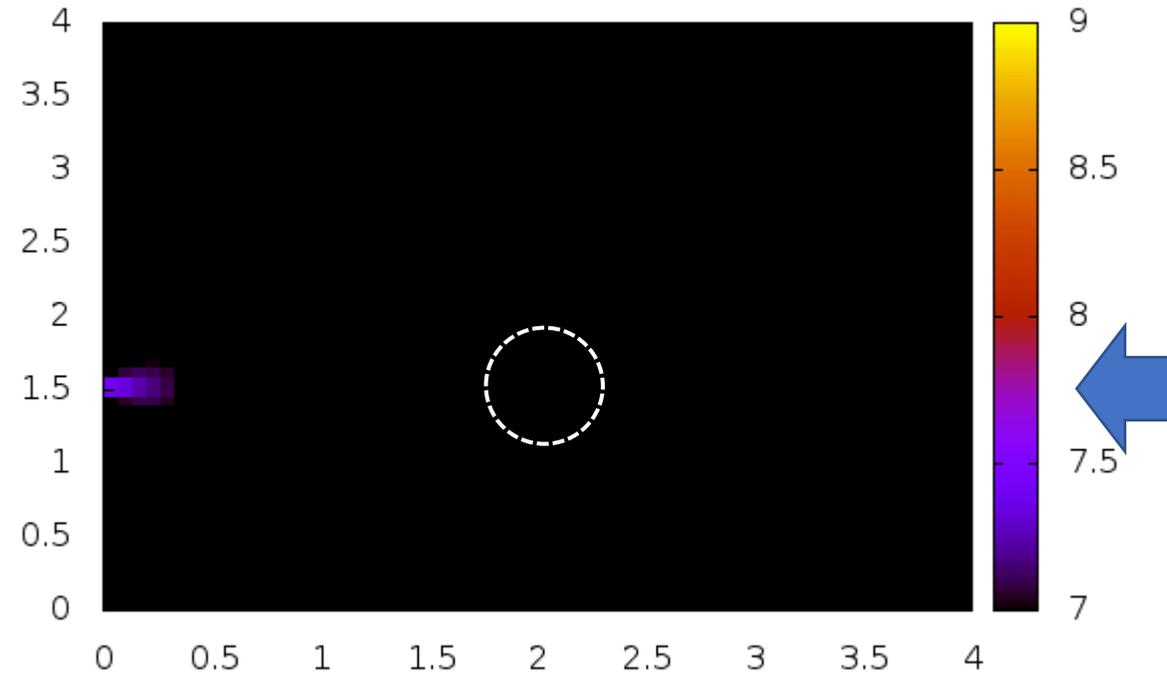
10^{-4}



吸収体ありなし
でわずかの差
吸収率 =
あり/なし - 1
で定量化



吸収体なし
生体内：散乱/吸収 = 100



吸収体 = ガンあり
生体内の**3倍**の吸収

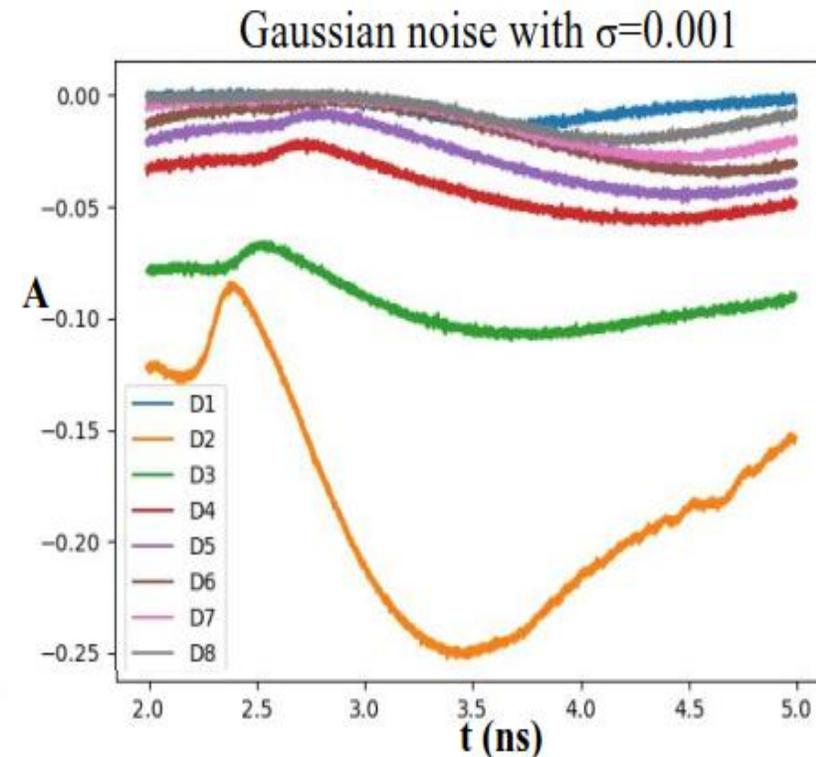
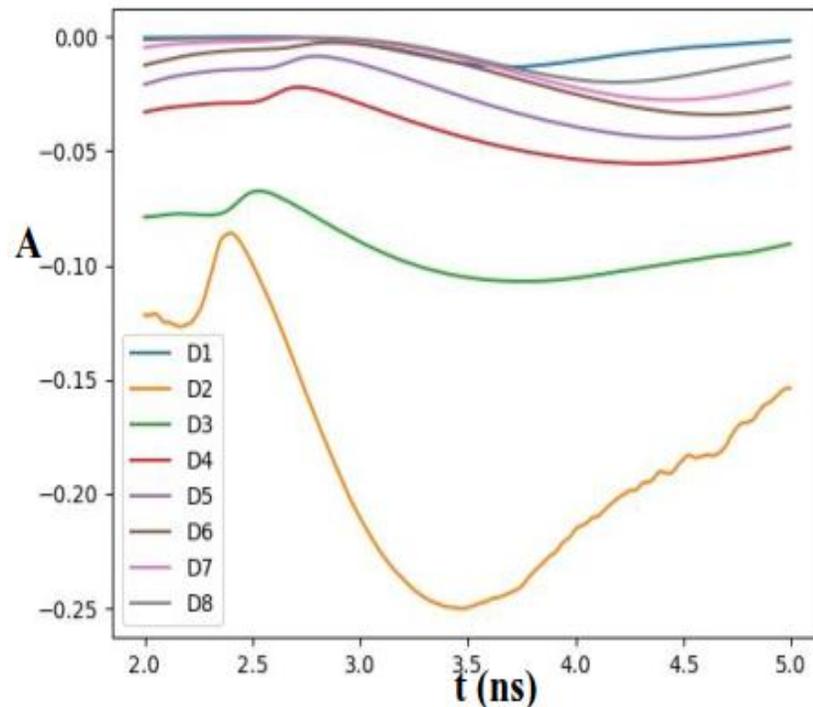
機械学習で用いるデータ：8次元の時系列データ

光伝播シミュレーション

$$A: \text{吸収率} = \frac{I_{abs}}{I_{noabs}} - 1$$

8つの検出器での時系列
データを分類する

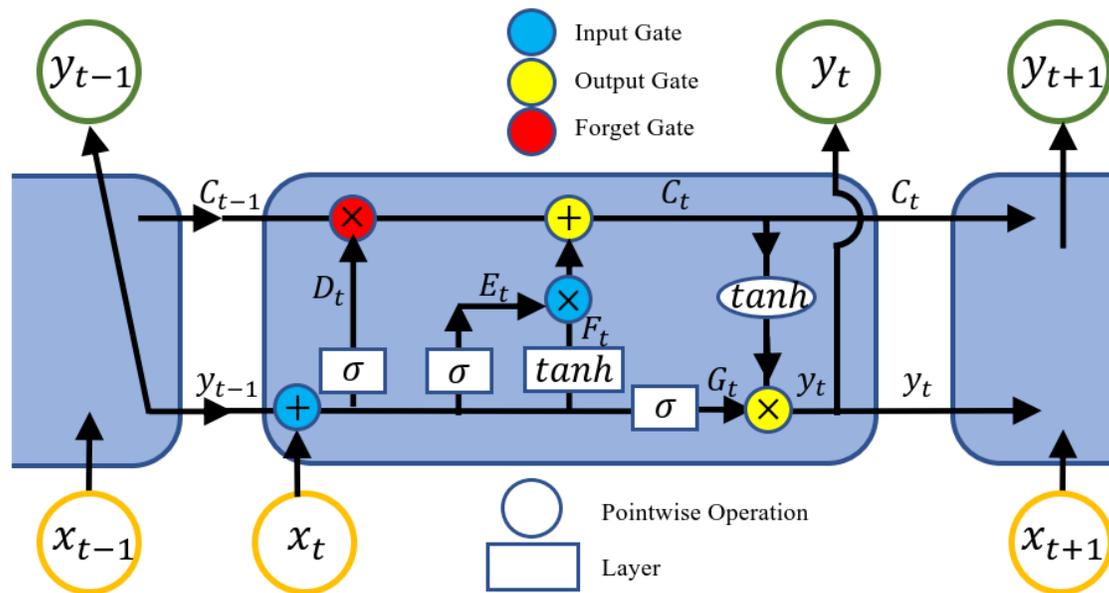
= 教師あり機械学習



深層学習：LSTMの構造

出力データ $Y_t=8$ 次元ベクトル

入力データ
(8次元、timestep)を各timestepに分割したもの
 $X_t=8$ 次元ベクトル



LSTM2層のあと、全結合層にいて、出力は分類数の次元をもったベクトル

Fig. 3

重みW Uは行列
バイアスb

-1, 1). The explicit calculation can be shown in the following.

Tanh層は、-1から1に数値の圧縮 これを成分ごとに計算して F_t も8次元

$$D_t = \sigma(W_D \cdot y_{t-1} + U_D \cdot x_t + b_D),$$

$$E_t = \sigma(W_E \cdot y_{t-1} + U_E \cdot x_t + b_E),$$

$$F_t = \tanh(W_F \cdot y_{t-1} + U_F \cdot x_t + b_F),$$

$$G_t = \sigma(W_G \cdot y_{t-1} + U_G \cdot x_t + b_G),$$

$$C_t = E_t \times F_t + D_t \times C_{t-1},$$

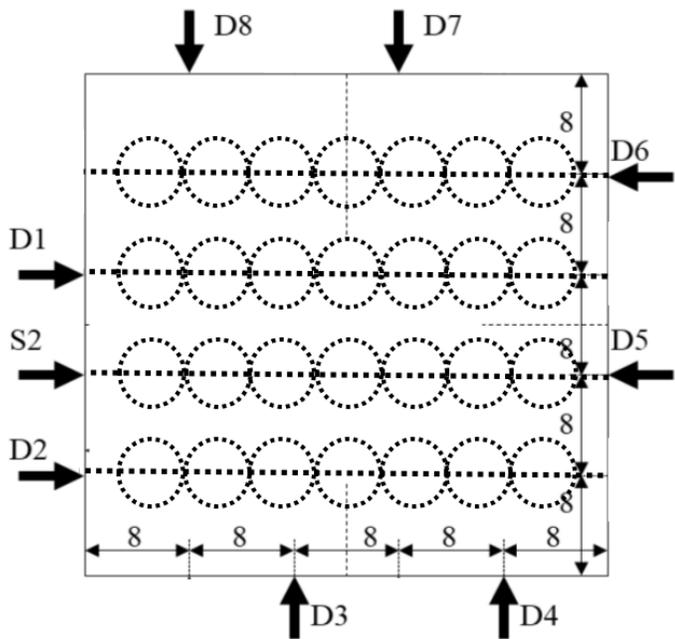
$$y_t = G_t \times \tanh(C_t),$$

Sigmoid層は、0か1の出力 これを成分ごとに計算して D_t も8次元

ベクトルの積は、アダマール積成分ごとに積を取る

2次元検出モデル

吸収係数はファントムと同じ、**直径を半分にした吸収体を領域内に敷き詰める** 5mm → **2.5mm**



	6	10	14	18	22	26	30	34	38	42	46	50	54	58
54	11	23	35	47	59	71	83	95	107	119	131	143	155	167
50	10	22	34	46	58	70	82	94	106	118	130	142	154	166
46	9	21	33	45	57	69	81	93	105	117	129	141	153	165
42	8	20	32	44	56	68	80	92	104	116	128	140	152	164
38	7	19	31	43	55	67	79	91	103	115	127	139	151	163
34	6	18	30	42	54	66	78	90	102	114	126	138	150	162
30	5	17	29	41	53	65	77	89	101	113	125	137	149	161
26	4	16	28	40	52	64	76	88	100	112	124	136	148	160
22	3	15	27	39	51	63	75	87	99	111	123	135	147	159
18	2	14	26	38	50	62	74	86	98	110	122	134	146	158
14	1	13	25	37	49	61	73	85	97	109	121	133	145	157
10	0	12	24	36	48	60	72	84	96	108	120	132	144	156

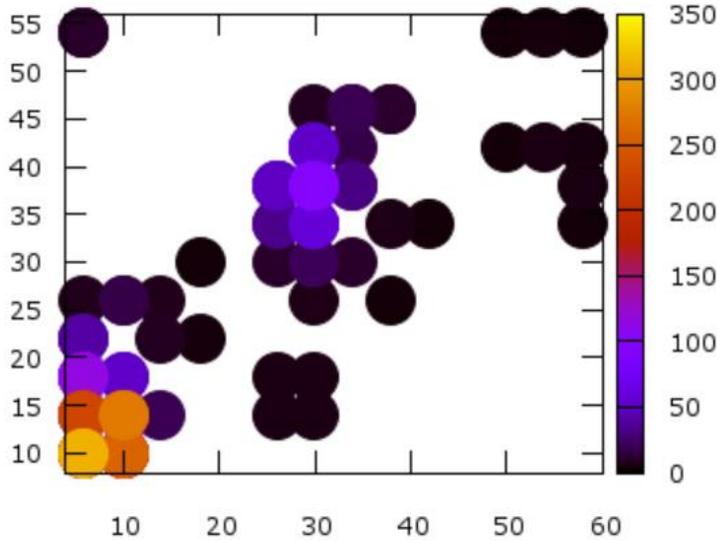
検出位置：28か所

ガンの大きさ、形状、個数など、より任意性の高い設定における検出を目指す

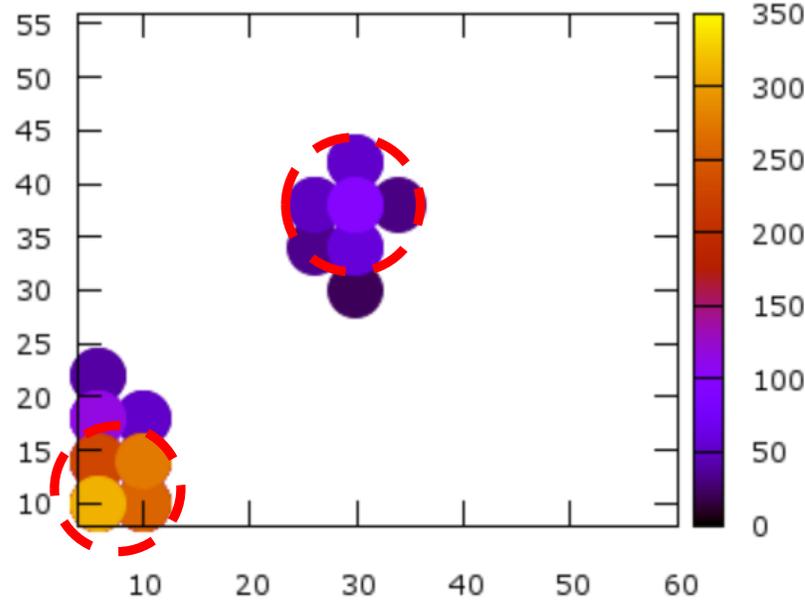
検出位置：**168**か所

入射8か所からの分類モデルを用いて、高得点領域を表示する

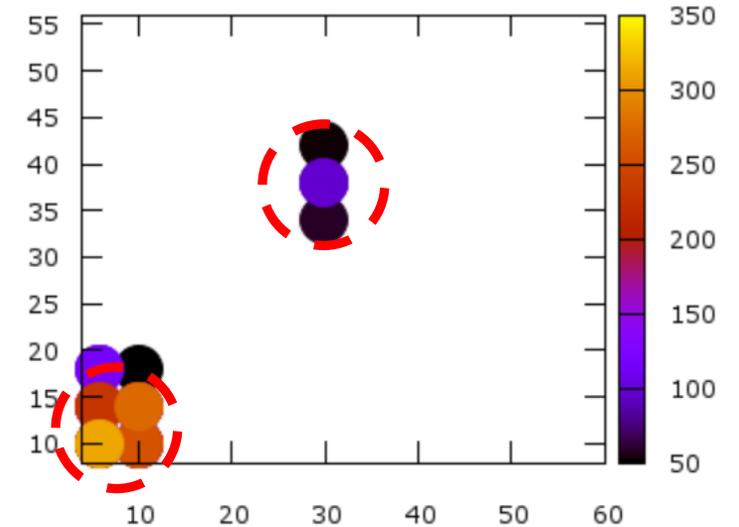
全得点表示



20点未満非表示

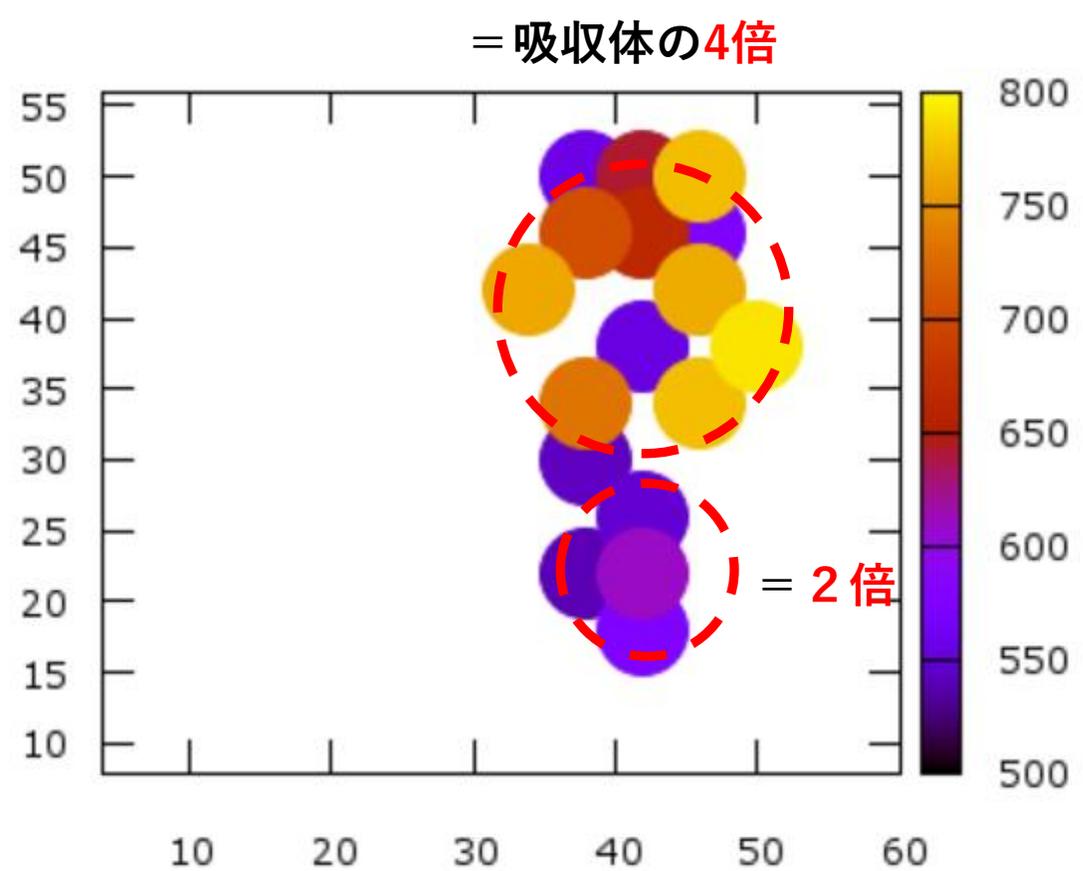
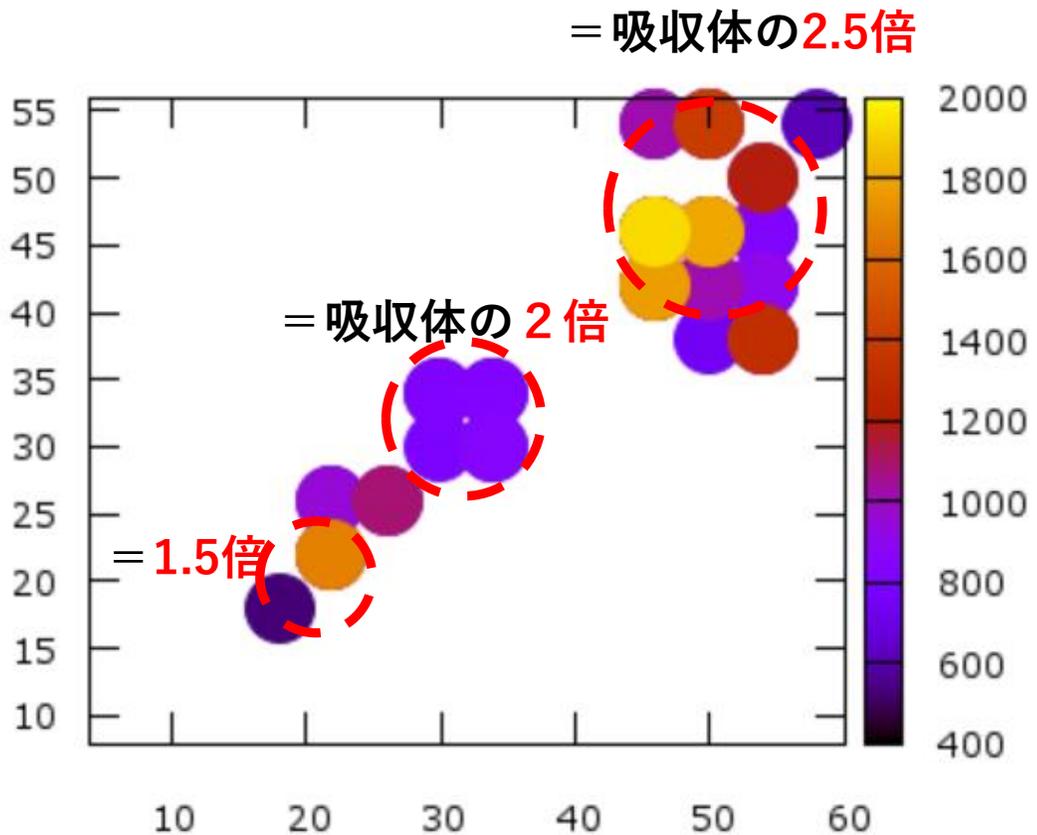


50点未満非表示



点数の下限値を調整していくことで、真のガンの位置がクリアになり、大きさもわかる

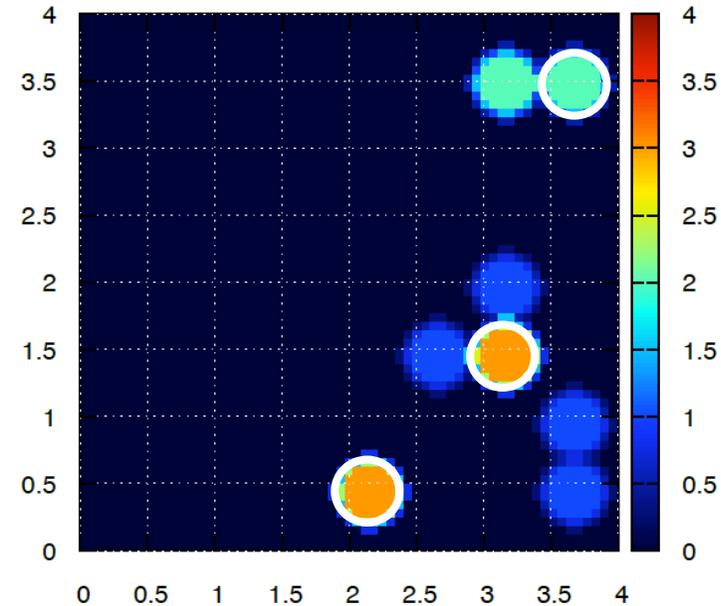
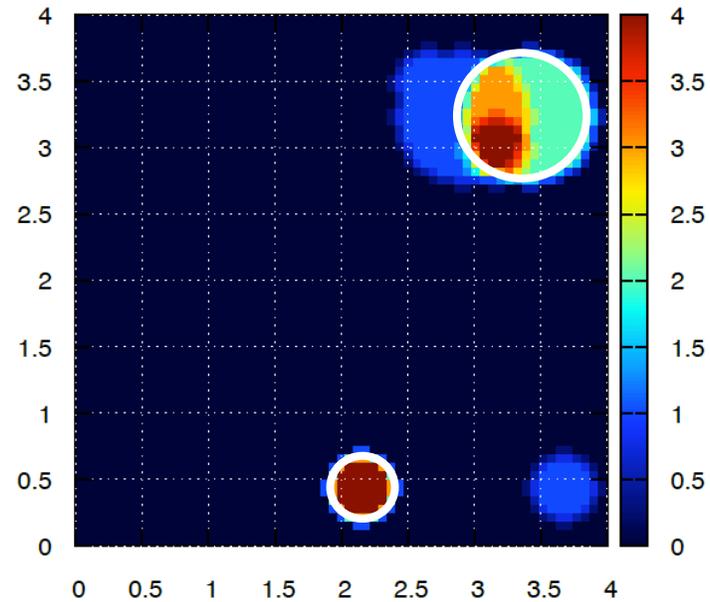
異なる大きさの複数体



キレイに再現されるようになっている。
任意の形の複数体のガンが再現できる

多ラベル分類 大きさを加えた多ラベル分類を学習させた

光音響トモグラフィ(PAT)の機械学習による複数本の血管の位置と半径の同定

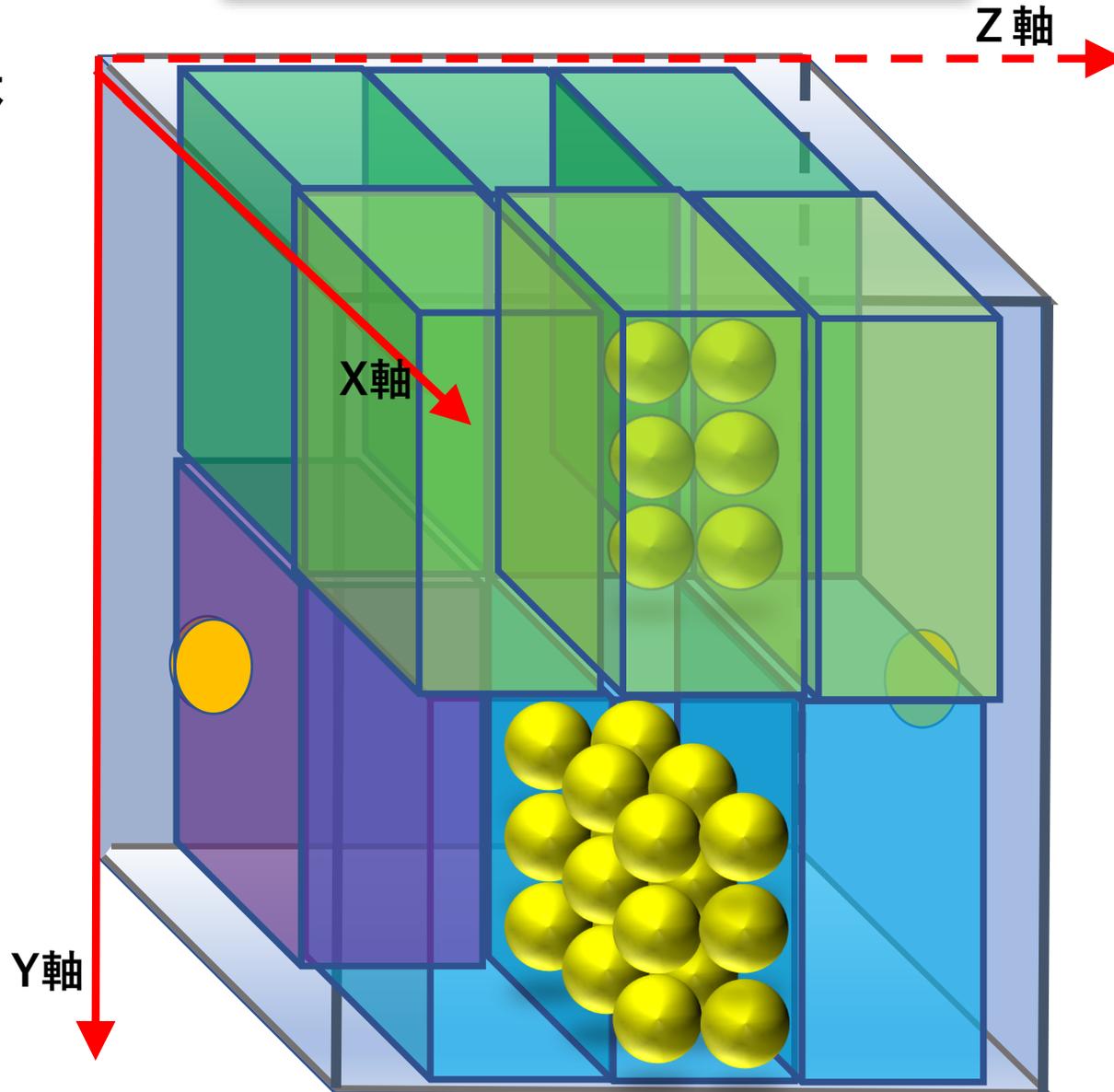


学生：佐藤大樹

白丸が血管の正しい位置。
血管が二本や三本ある時でも近しい位置もしくは、正確な位置を判定できている。

3次元検出モデル

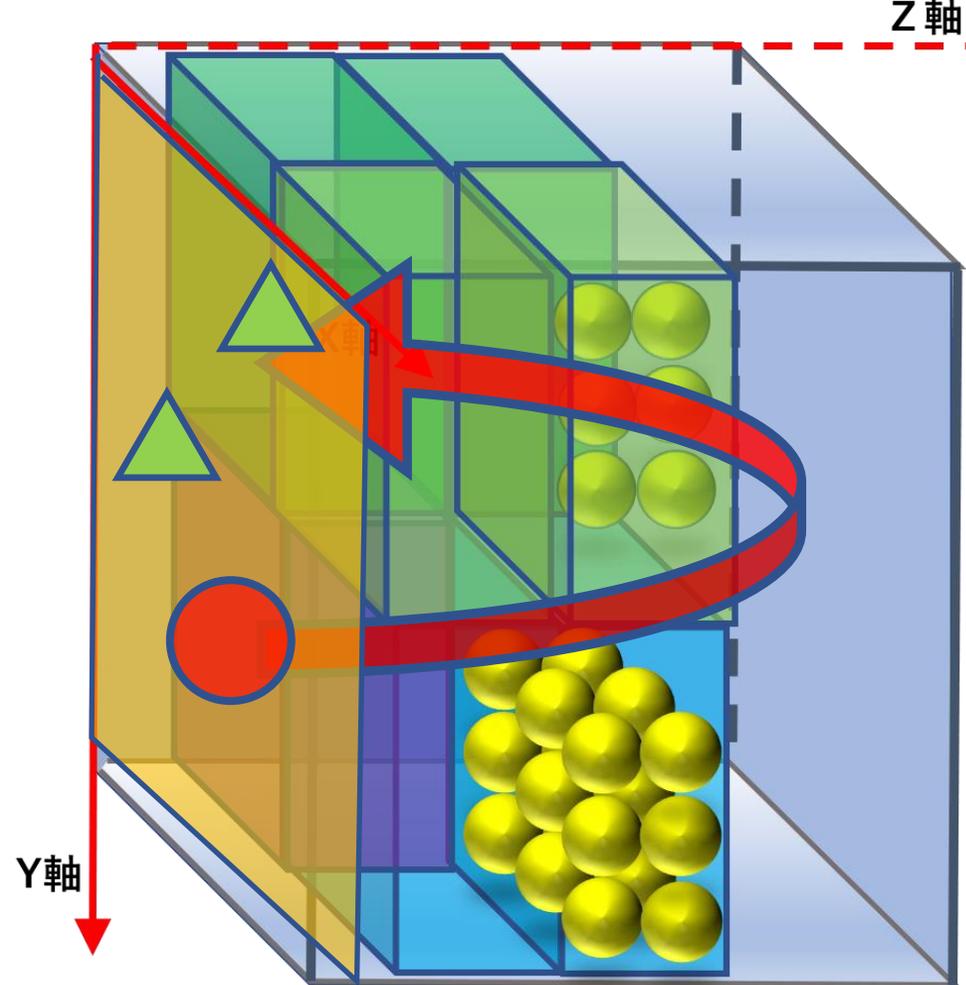
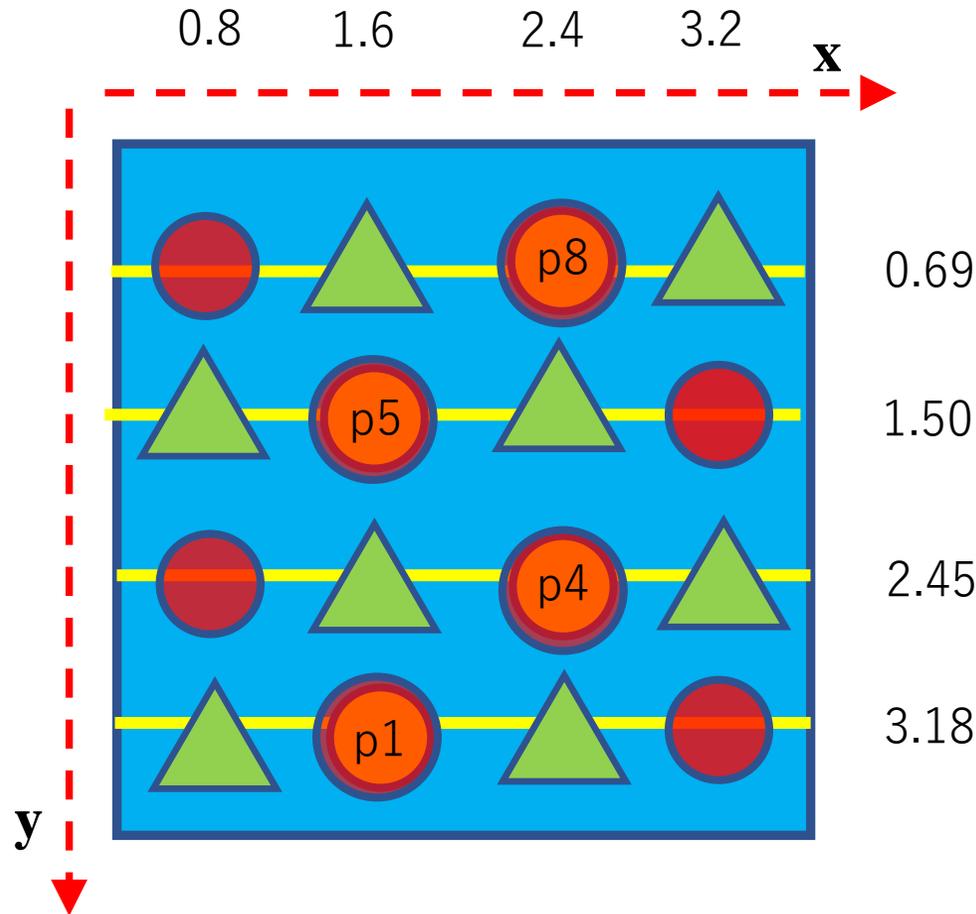
全**216個**の吸収体
(直径5mmの球体)



12グループ
各グループ 18個の球

=2段階 12分類×18分類

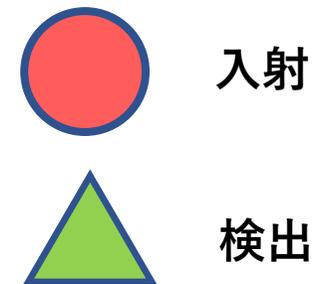
1面モデル



入射と検出がある1面
だけを用いる

面に垂直方向に伝播
して戻ってくる輻射
の情報

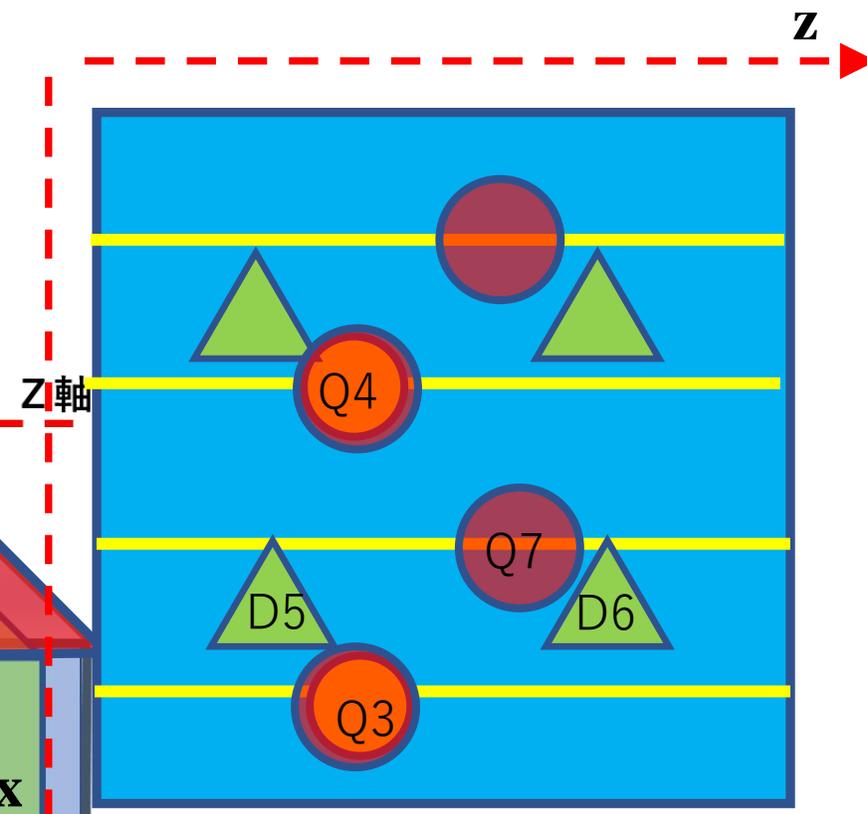
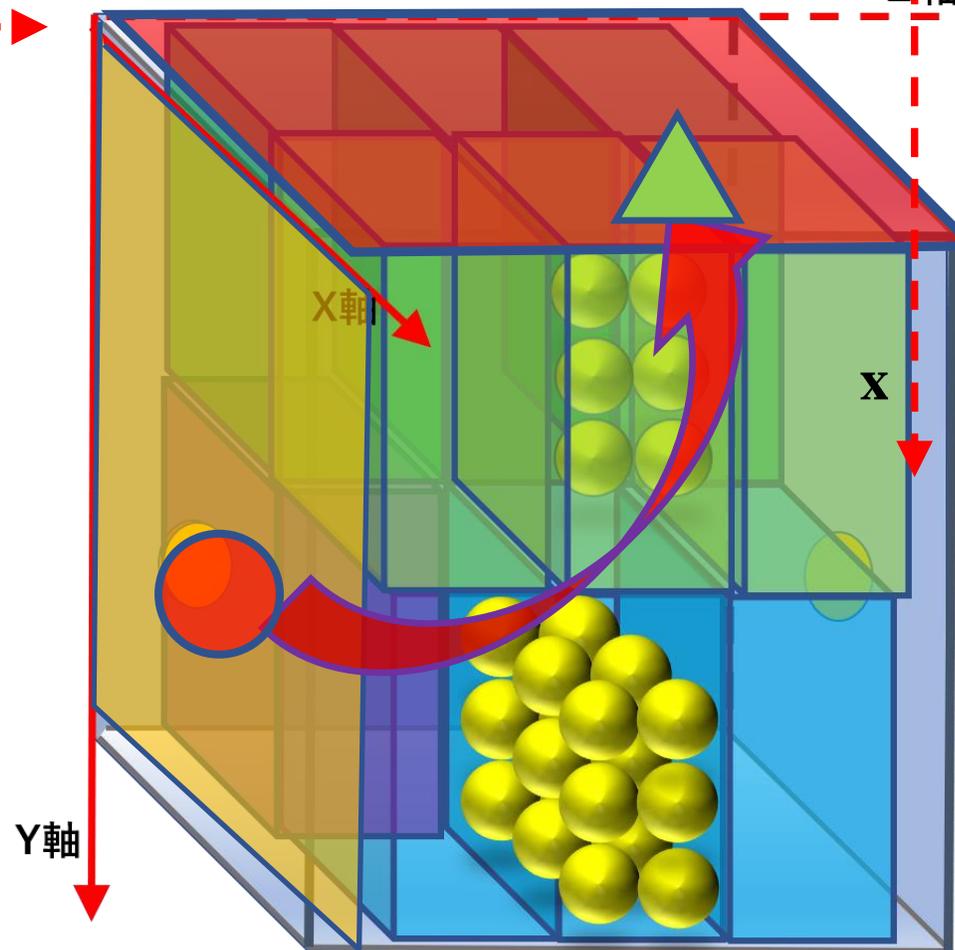
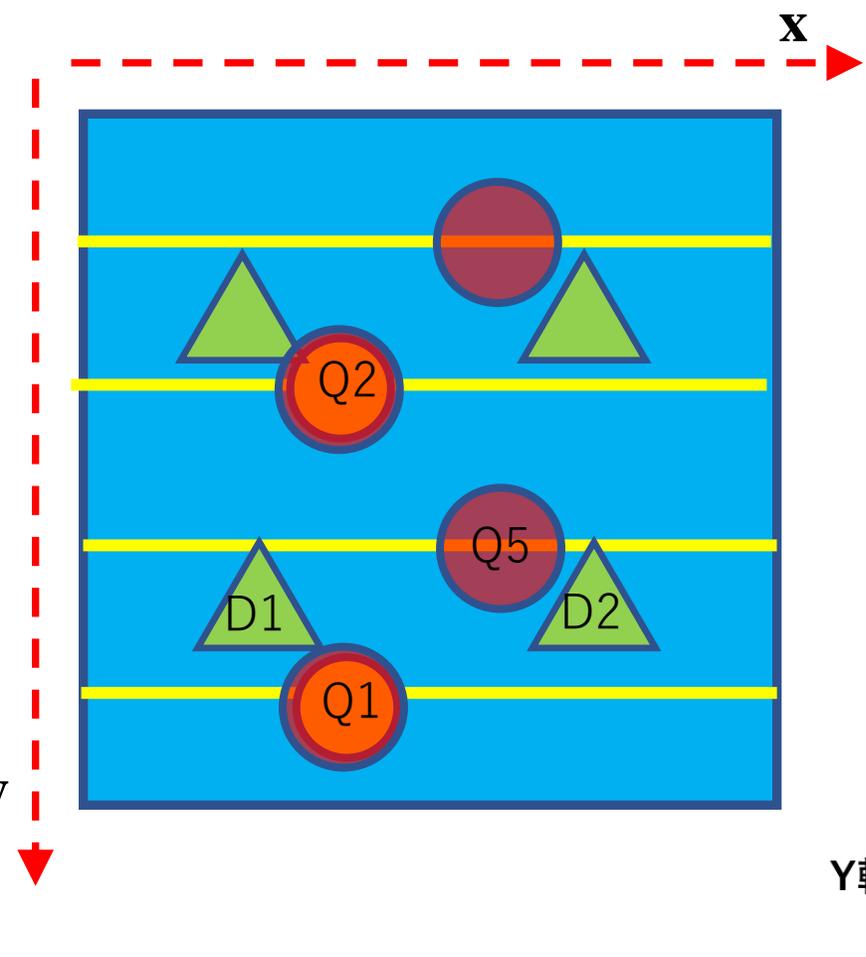
きちんと分類するた
めには、奥行きに限
界がある



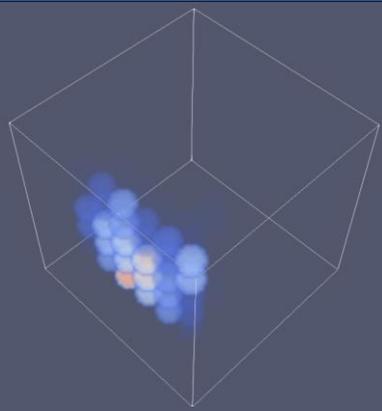
入射と検出がある **2面**を用いる

いずれも **面に垂直方向**に伝播していく
輻射の情報

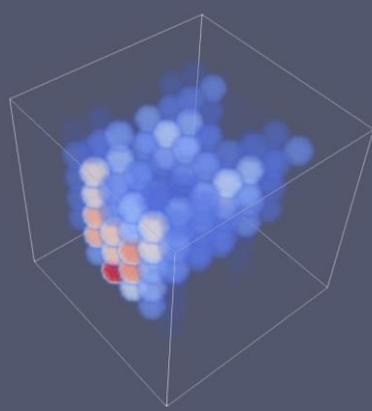
2面モデル



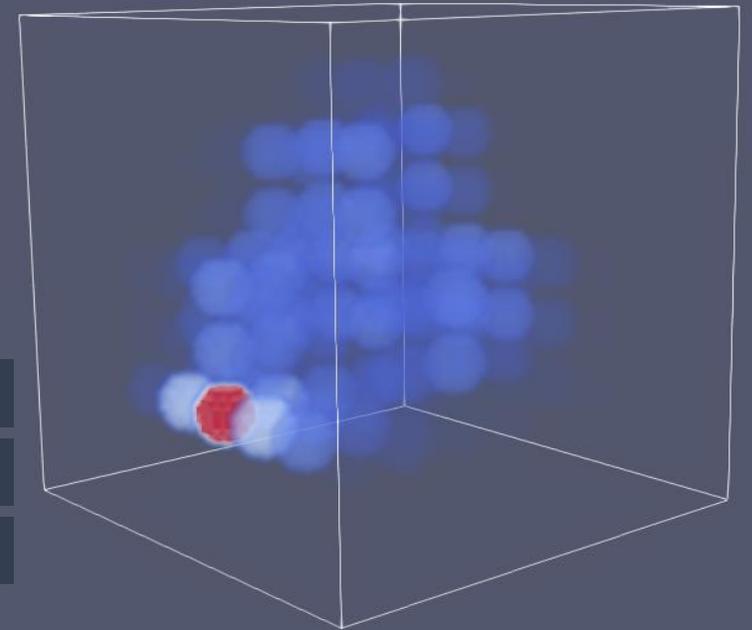
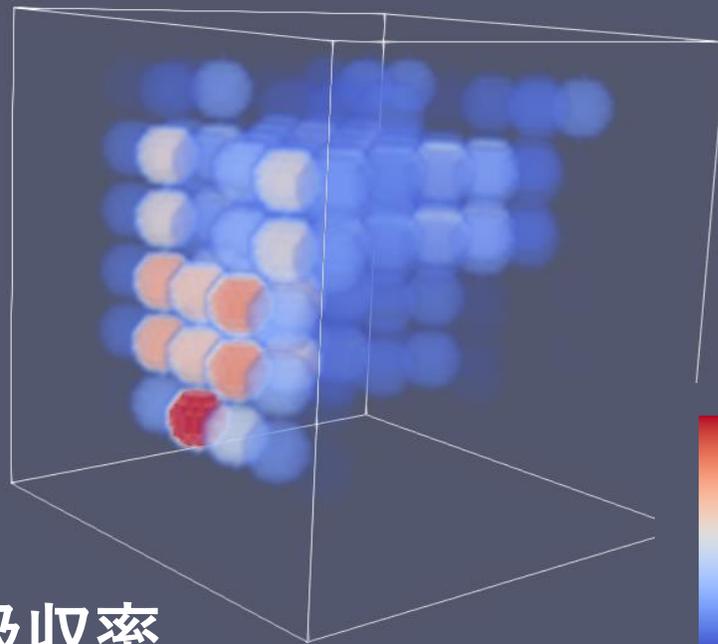
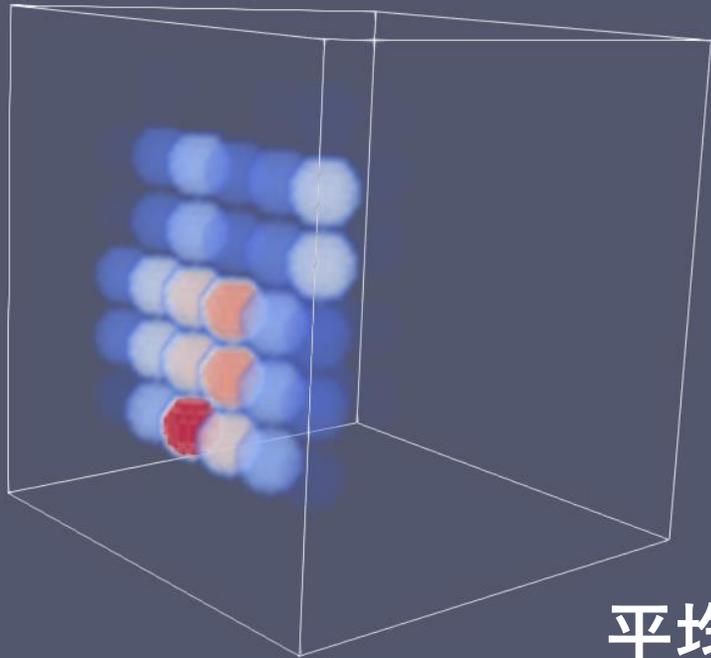
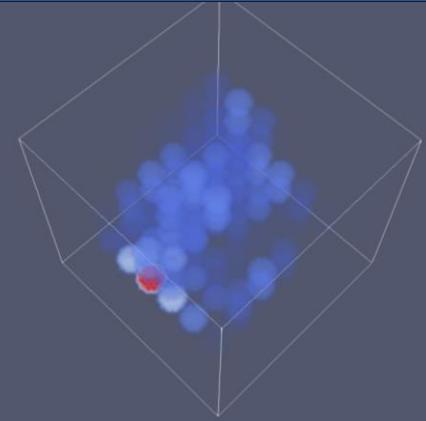
-  入射
-  検出



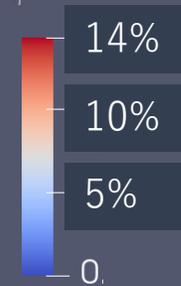
1面では
全体をカ
バーするの
が難しい



2面が多面
設定よりも、
もっとも全体
をよくカバー
している



平均吸収率



一面検出モデル

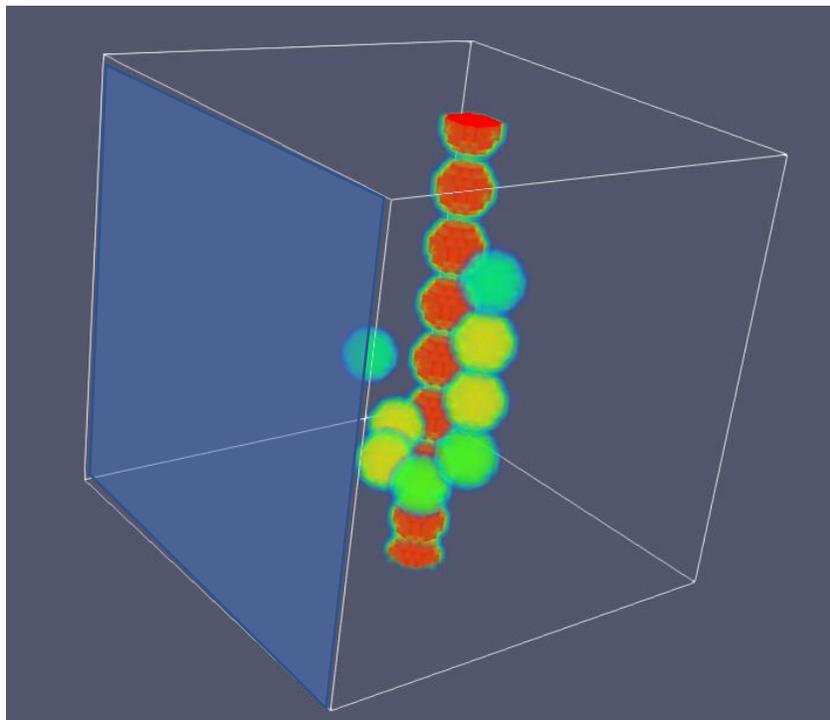
二面検出モデル

多面検出モデル

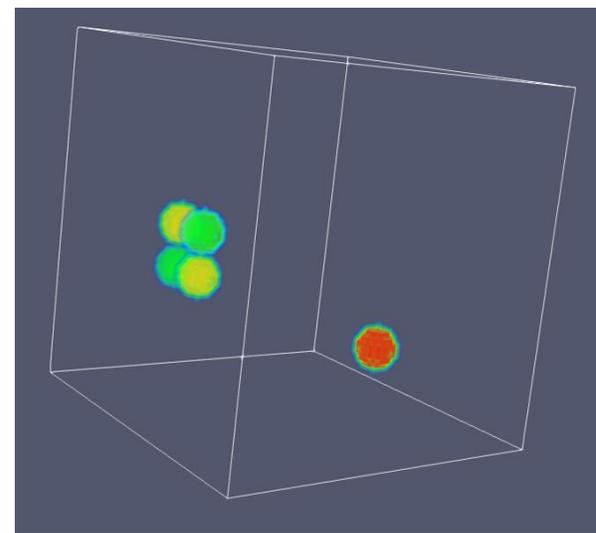
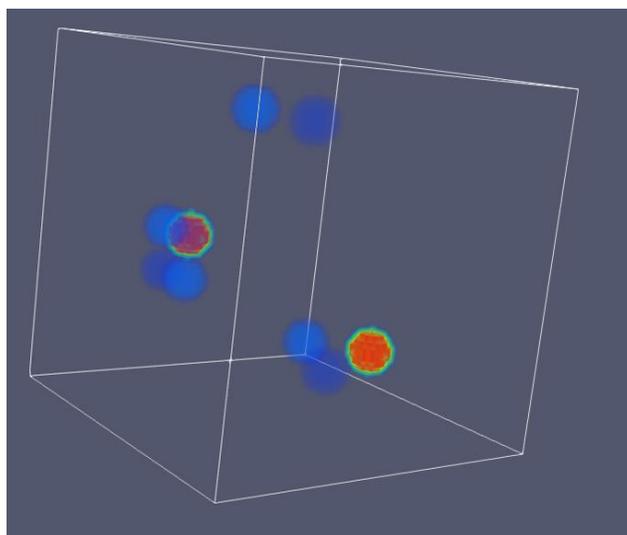
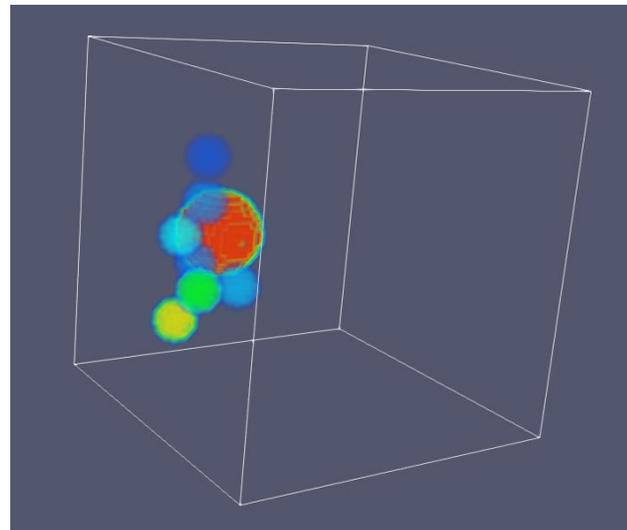
入射位置、検出位置の数は固定して比較

1面モデル

柱方向によく反応しているが、
位置がずれている



大きい1体も位置がずれている



FPGAに向けた学習の改良

堀江先生にもらったコードをもとに、ノイズ付加で水増しするやり方

入力データ (8次元、timestep)



ここにガウシアンノイズ層を加えてノイズを付加する

少ないサンプルからノイズ付加を用いて、内部で増やす
FPGAのメモリ問題が解消できるため

28か所を28分類する機械学習

28データをあらかじめ用意して、そこにノイズを付加してbatch=1でEpoch=40で、1study行う。
それをstudy=100回まで繰り返し学習
=実質、28×100のデータを学習したことに相当

1データ : 60kb

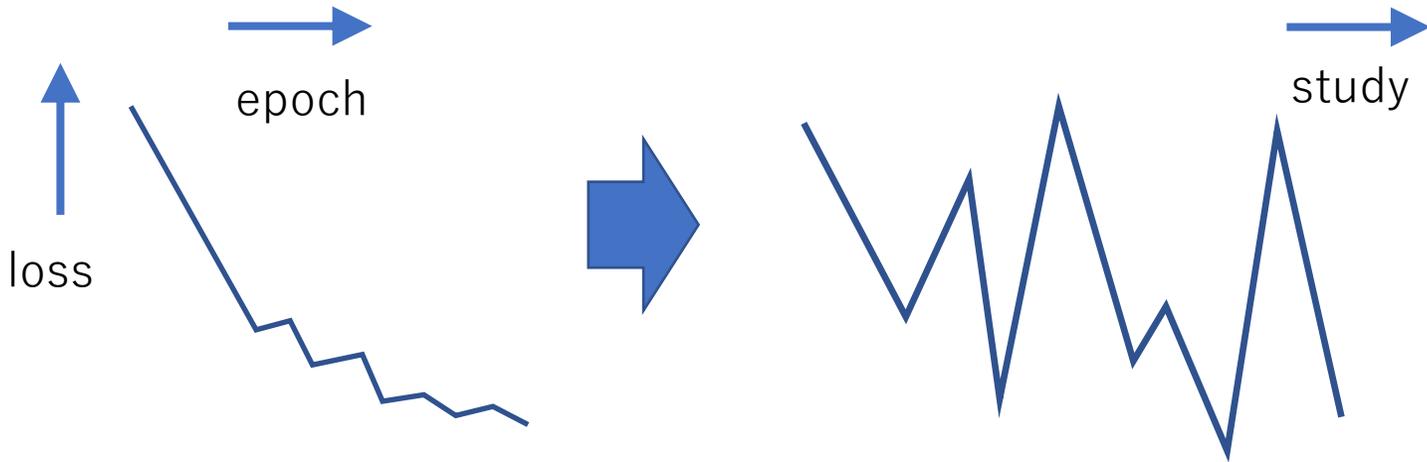
LSTM層①

出力データ

LSTM層②

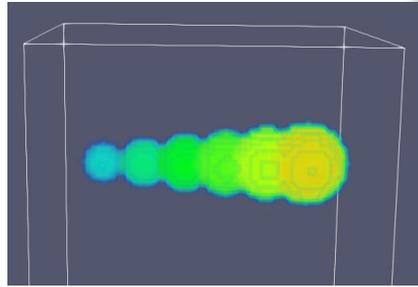
全結合層

出力データ (分類数)



1体

半径2倍

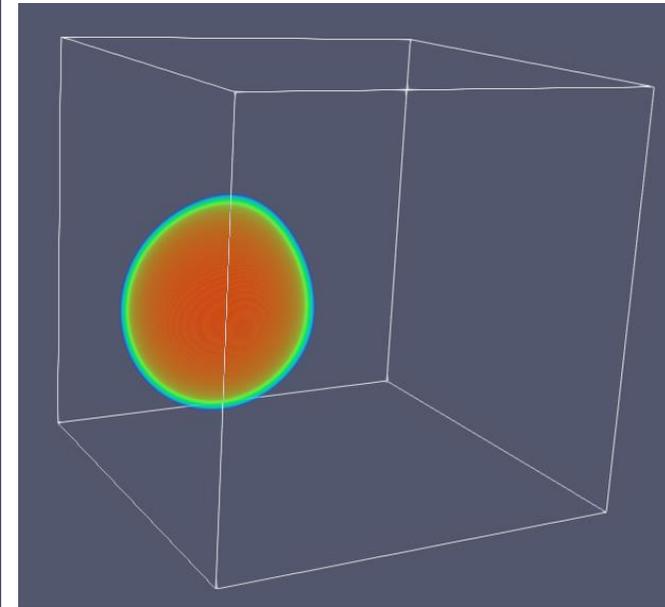
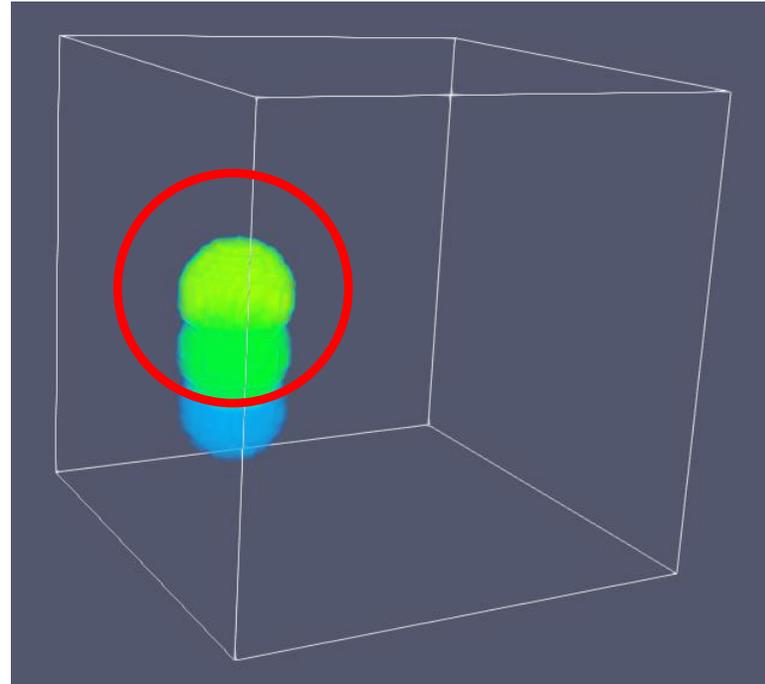
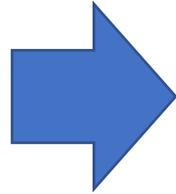
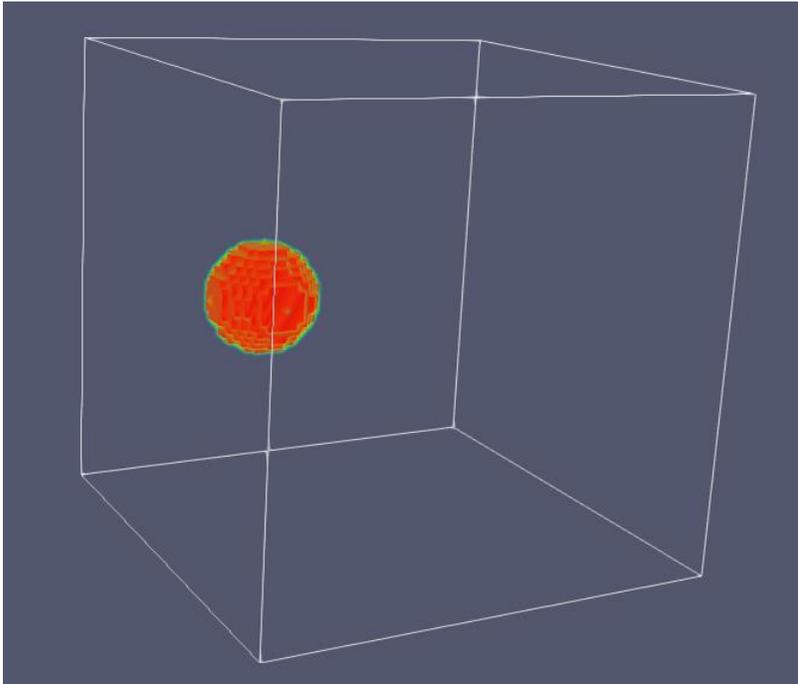


2面モデル

多レベル分類

入射8か所の分類モデル
を用いた得点表示

形状を表示

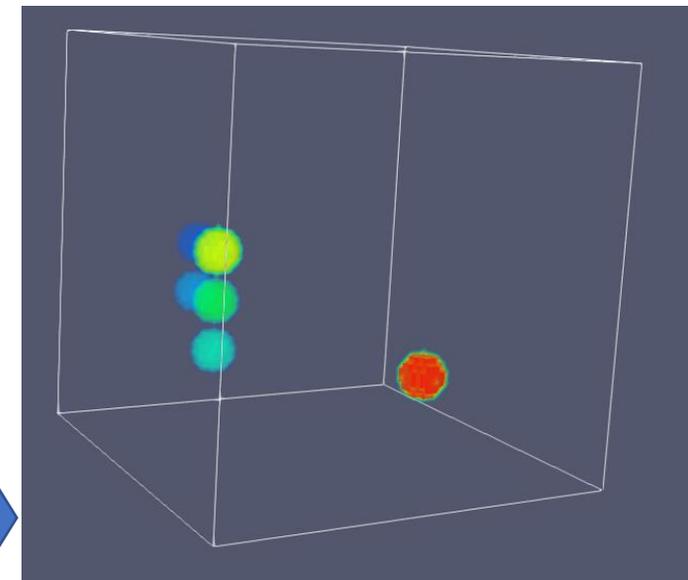
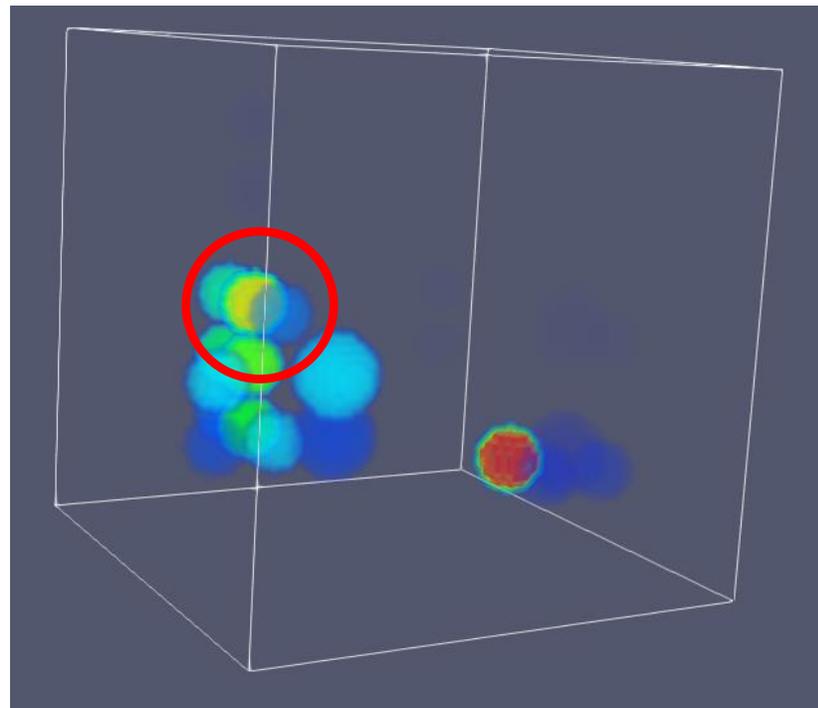
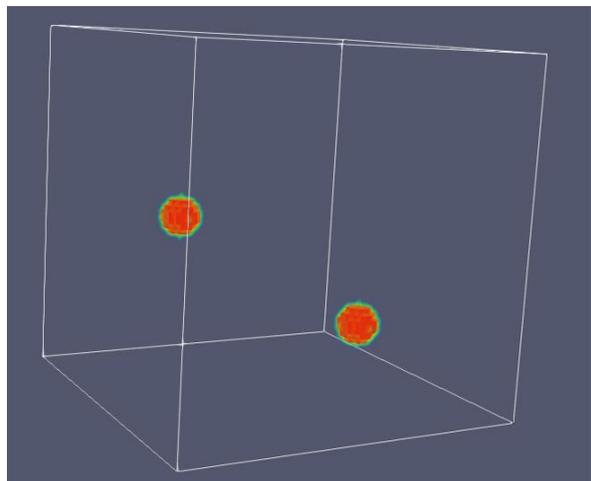


大きい一体、大きさ、位置とも
にかなりうまくいっている

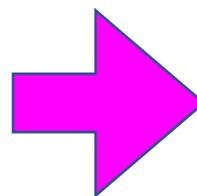
2体

2面モデル

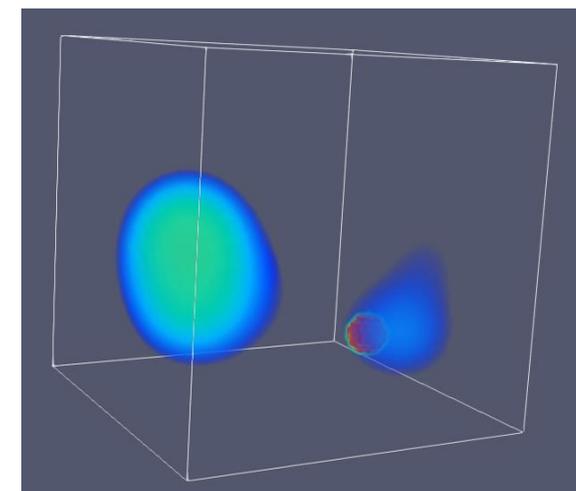
高得点を表示



2体ともきちんと反応。さらに1体は、
高得点として、大きさ、位置が正解



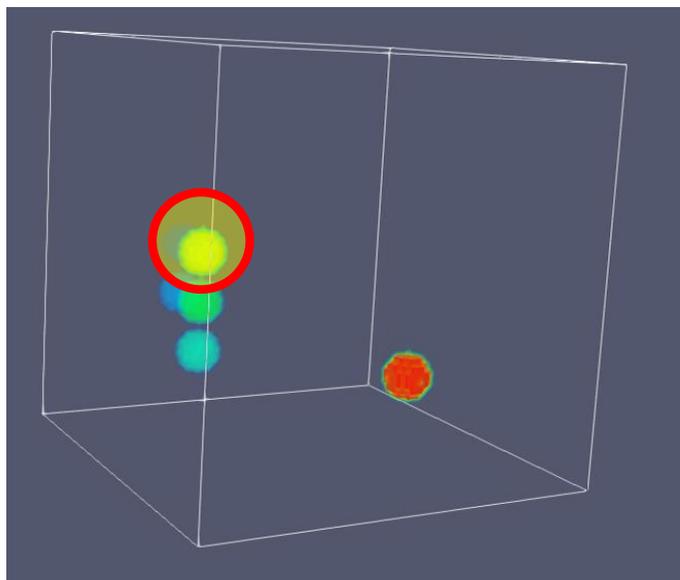
2次元で行ったように、高得点の1体を
データから取り除けば、もう1体の検
出もうまくいく



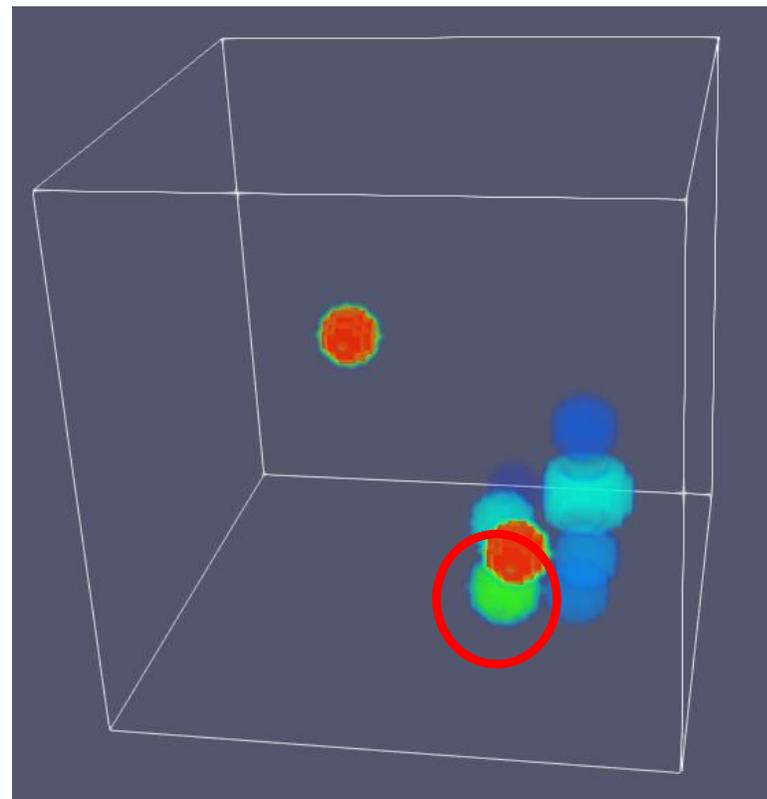
形状を表示

検出で確定した部分をデータから引
いて再度検出してみる

2体



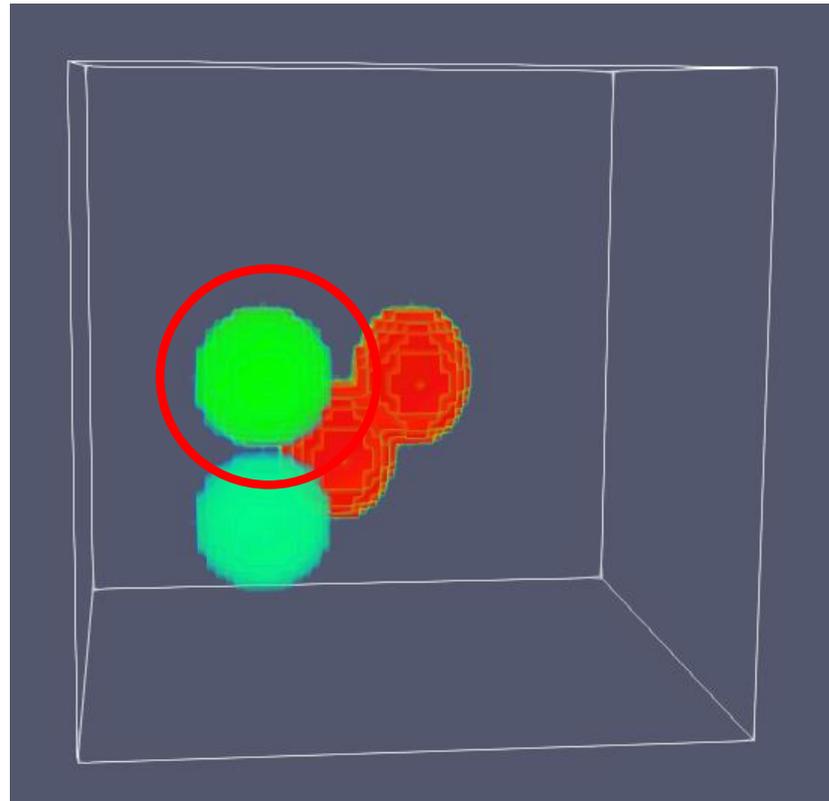
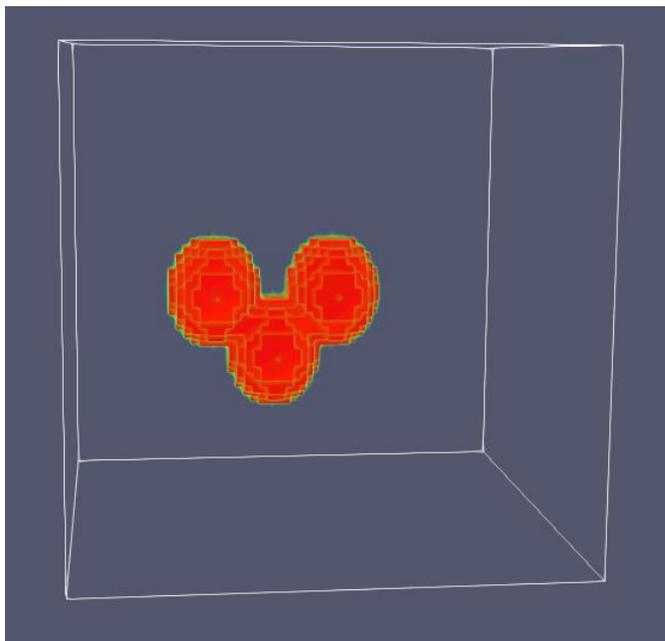
データを間引く



もう1体もきちんと検出できるよう
になる！

ミッキー型

2面モデル

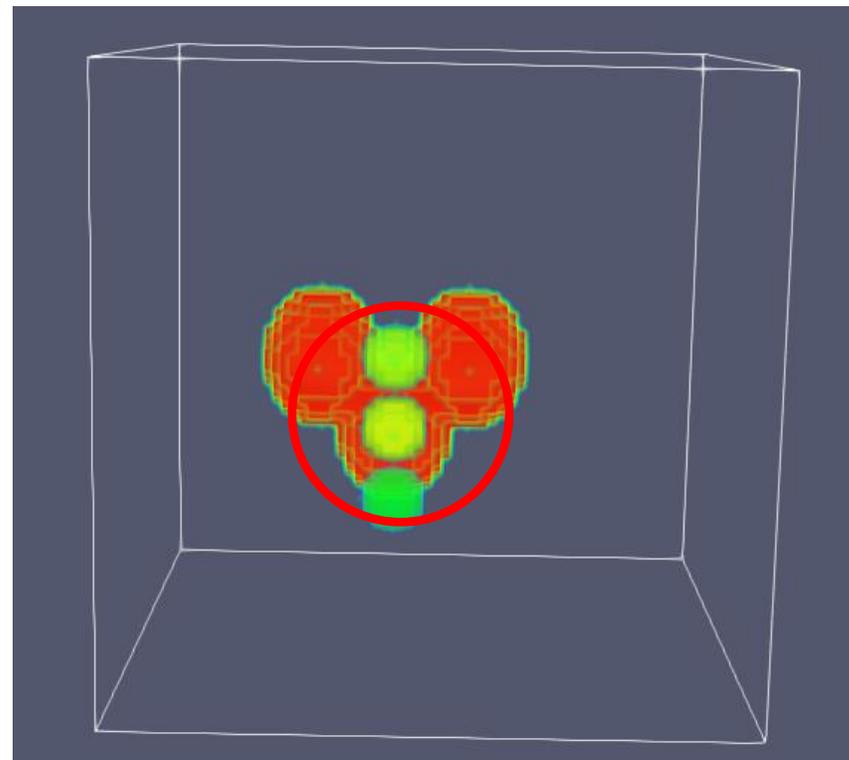
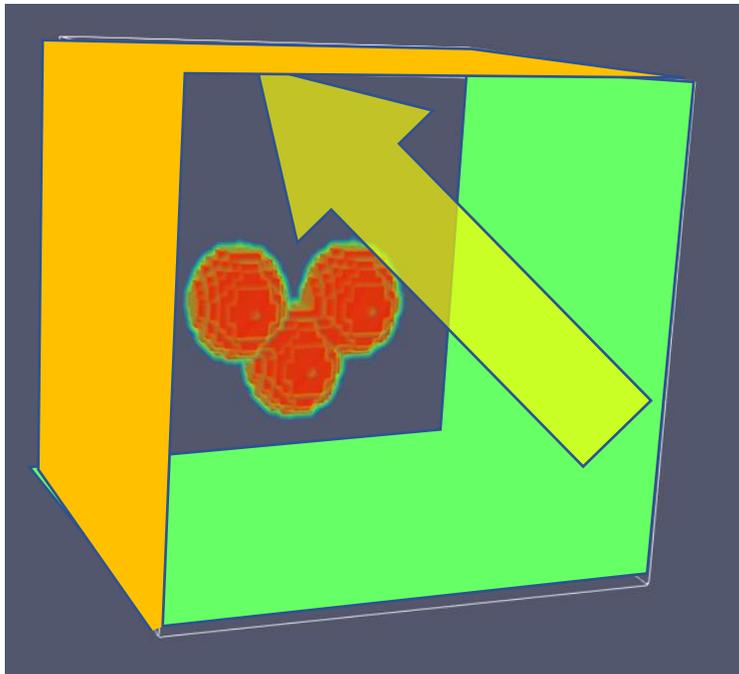


ミッキーの耳を、大きさも含めてきちんと、
とらえている

ミッキー型

別の2面モデルを加えてみる

入射面は変えるが、
検出面は同じ8か所

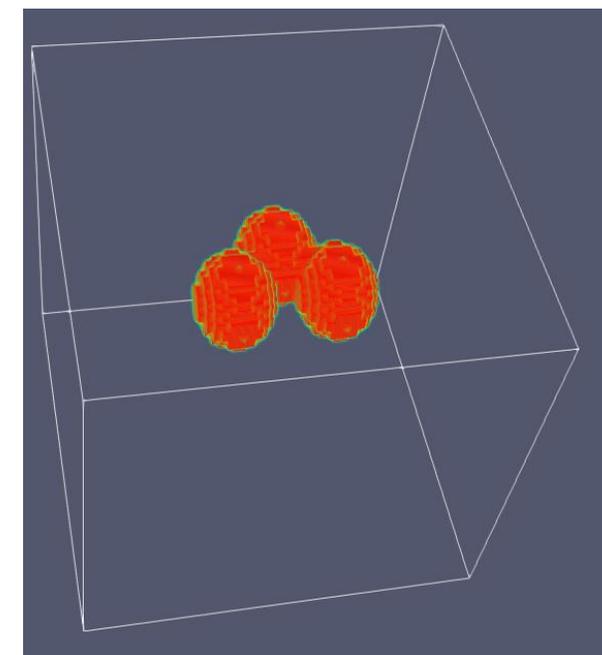
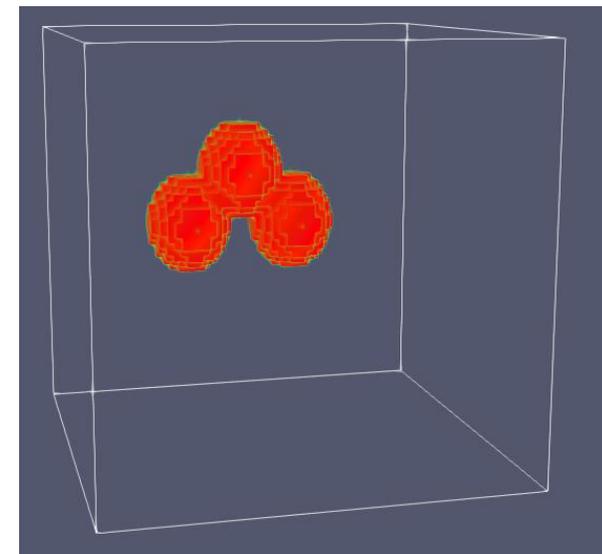
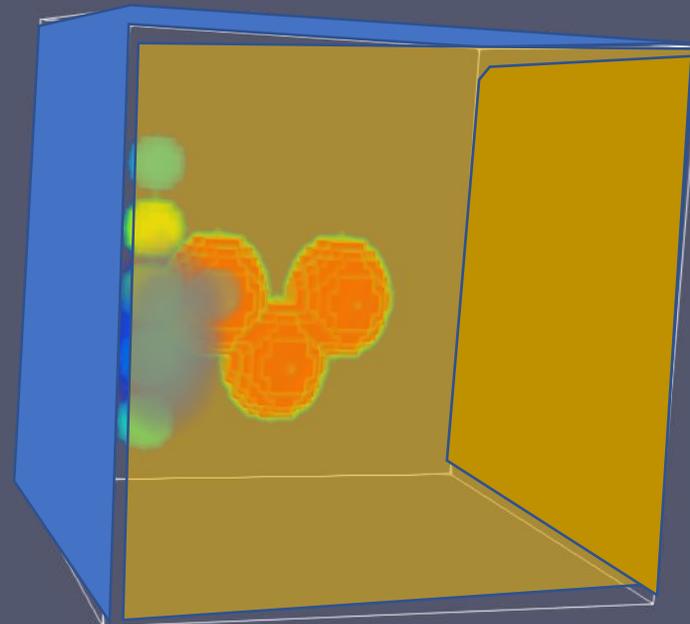
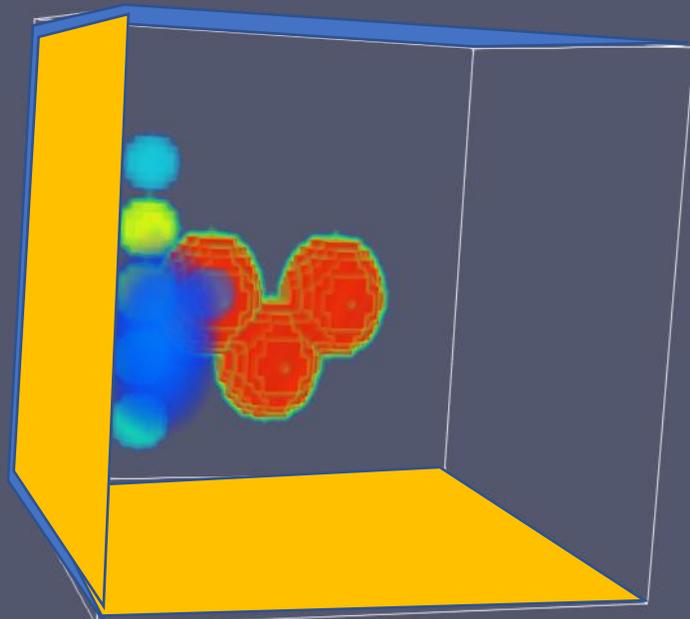


注)この検出モデルは、大きさも
含めたマルチラベルではない

別の2面から検出することで、顔の部分が検出
できるようになる

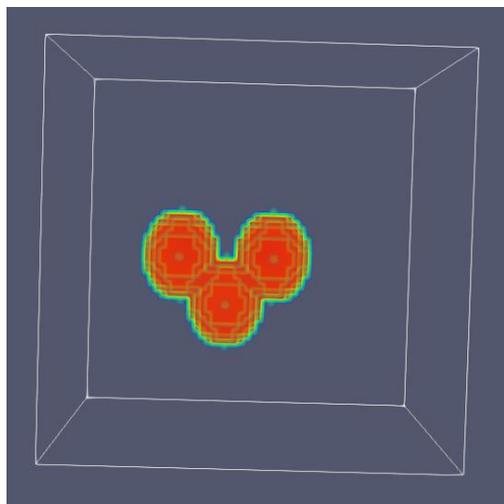
2面モデル

ミッキー型を
回転させる

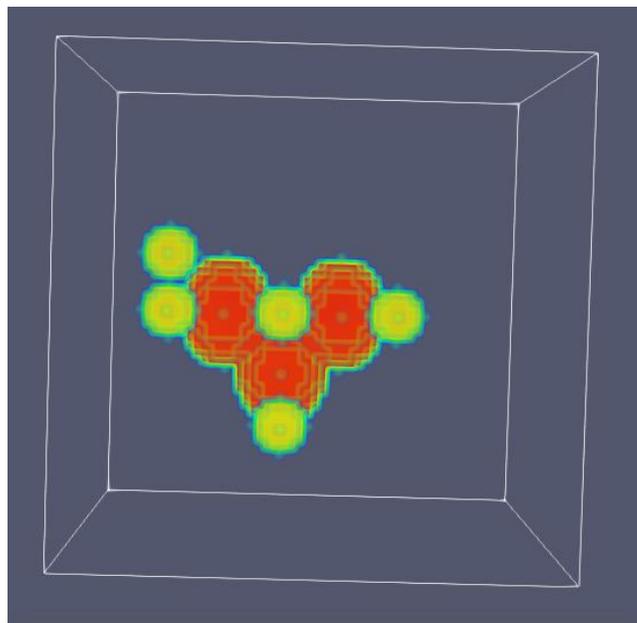


画像再構成

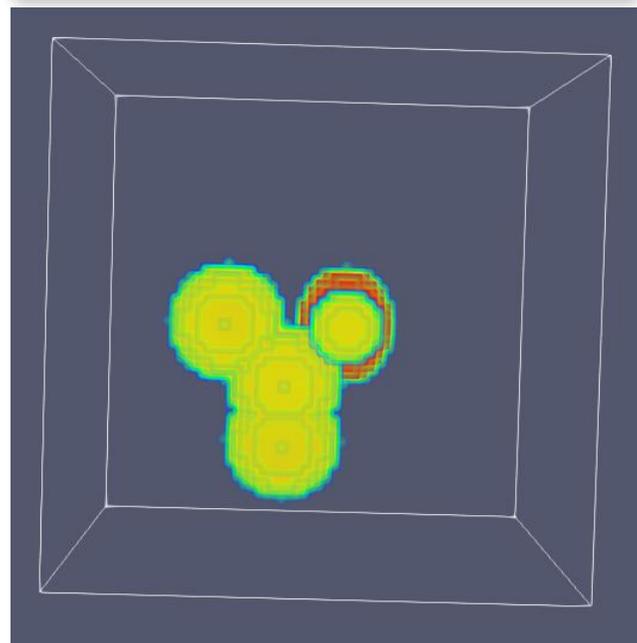
直径5mm分類



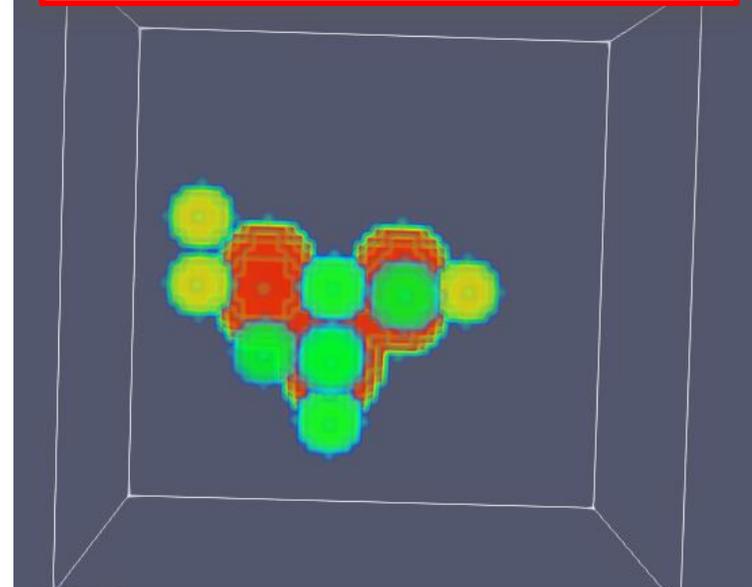
直径6段階分類



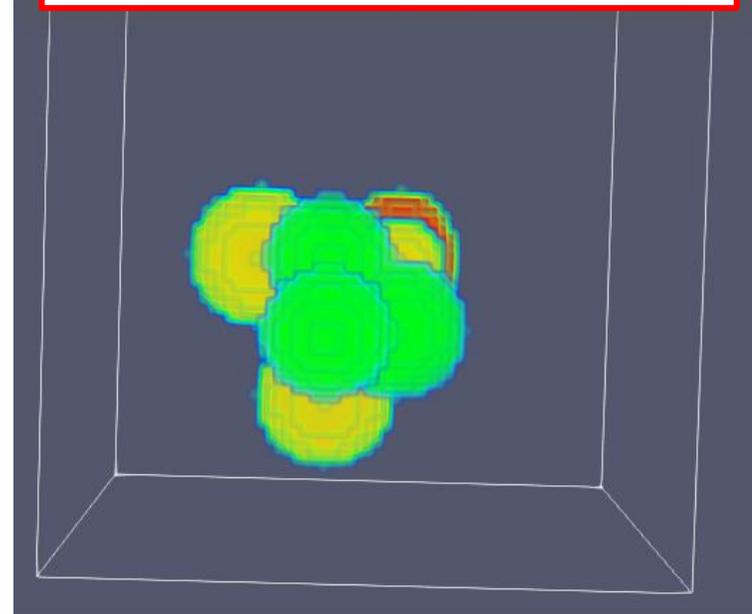
入射8か所×3回転



+別入射8×3回転

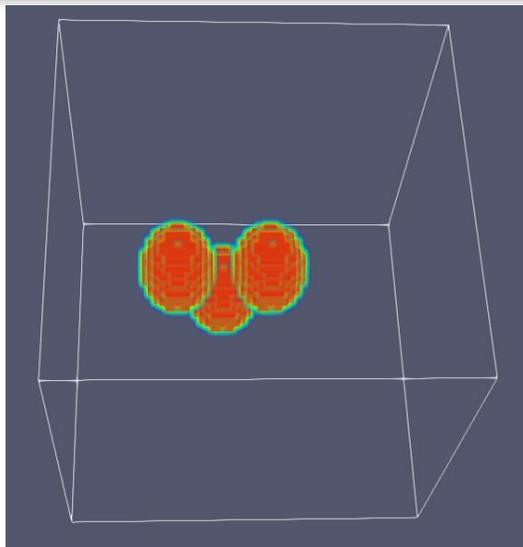


+差分8×3回転

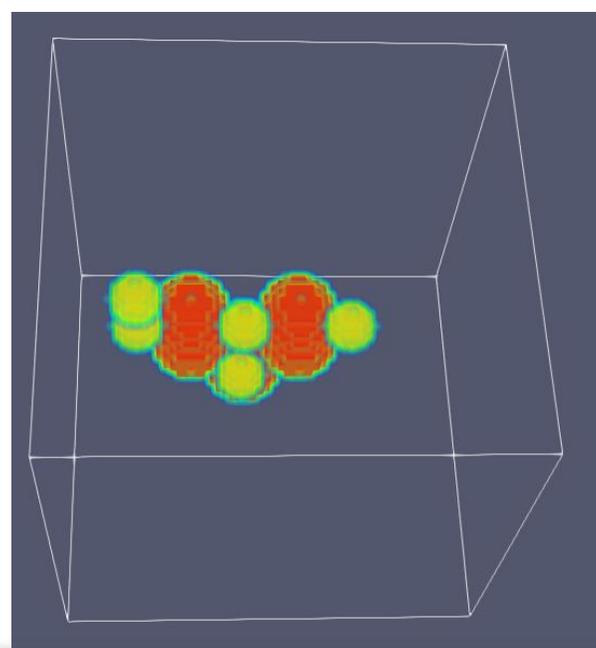


画像再構成

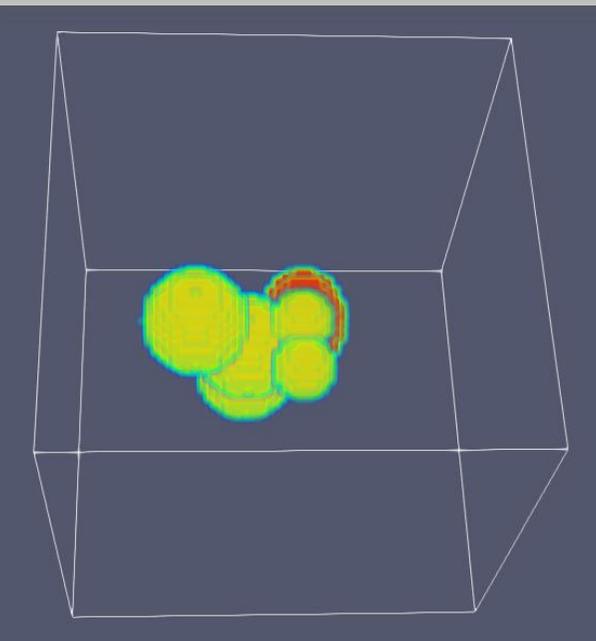
直径5mm分類



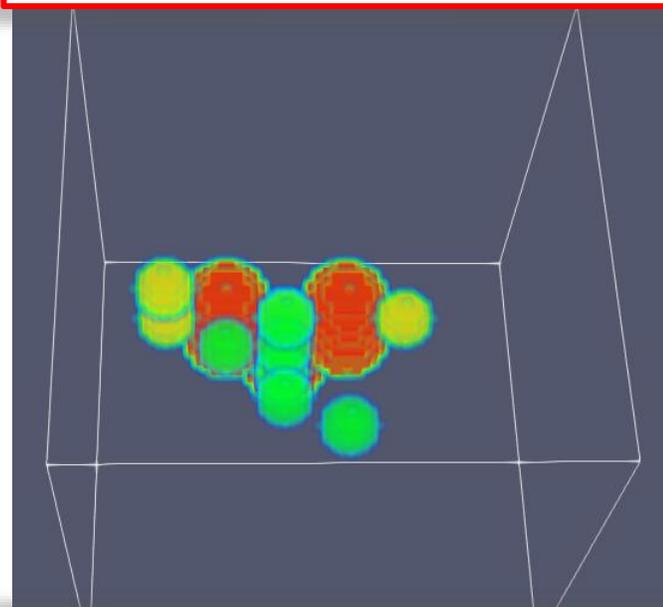
直径6段階分類



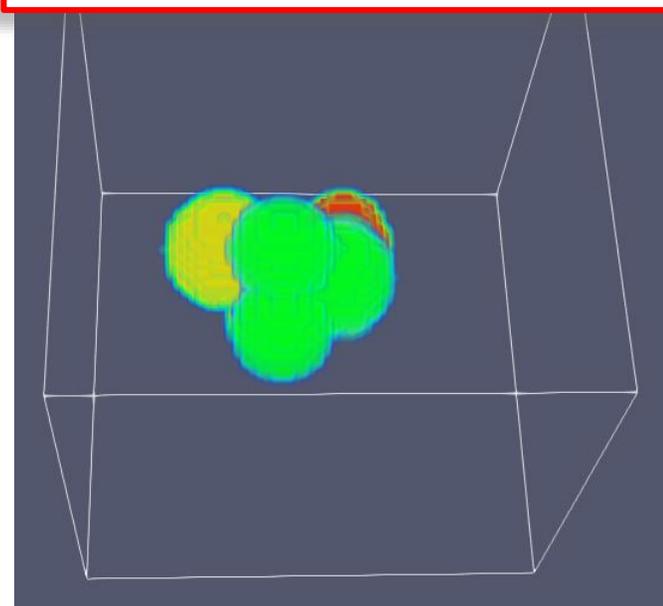
入射8か所×3回転



+別入射8×3回転



+差分8×3回転



まとめ

2次元：4cm平方 検出体 **直径2.5ミリ**で168か所 1つの入射に対して検出8か所 × 8入射ごとのモデルを重ね合わせた

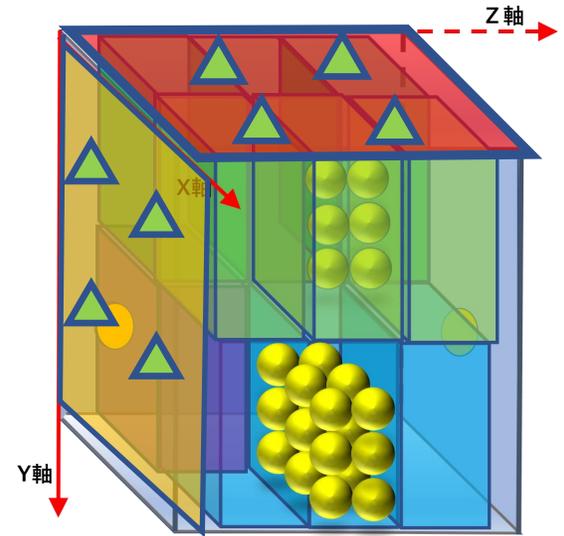
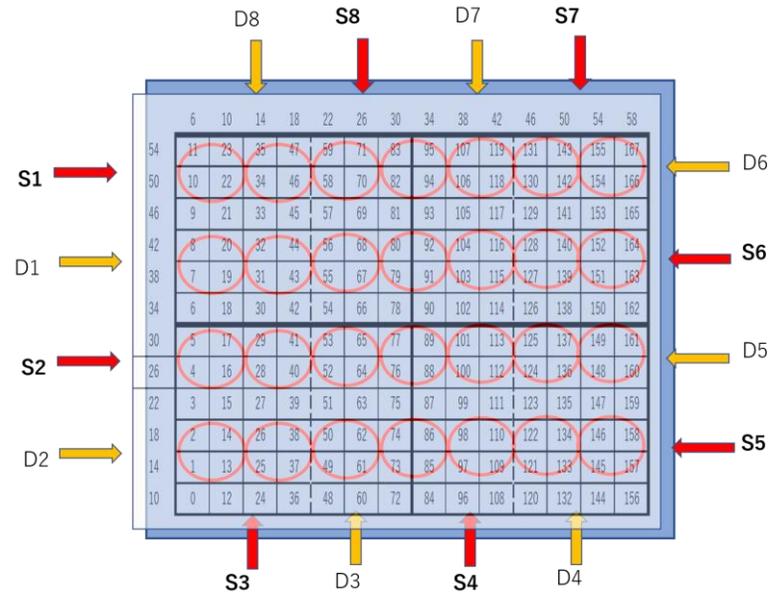
大きい吸収体 **直径1cm**に対して、検出体約16個で表現している

同様のことを3次元で行うと、64個必要 = 入射位置が**合計32か所**必要になる

入射位置の数 VS 再現に用いる検出体の数
1つの入射でせいぜい2-3個 VS r^3

2面を回転させて疑似的に入射位置を増やしてカバーしていた

検出体の**大きさを6段階考慮**した多ラベル分類モデル
直径5mm-1cmで216か所



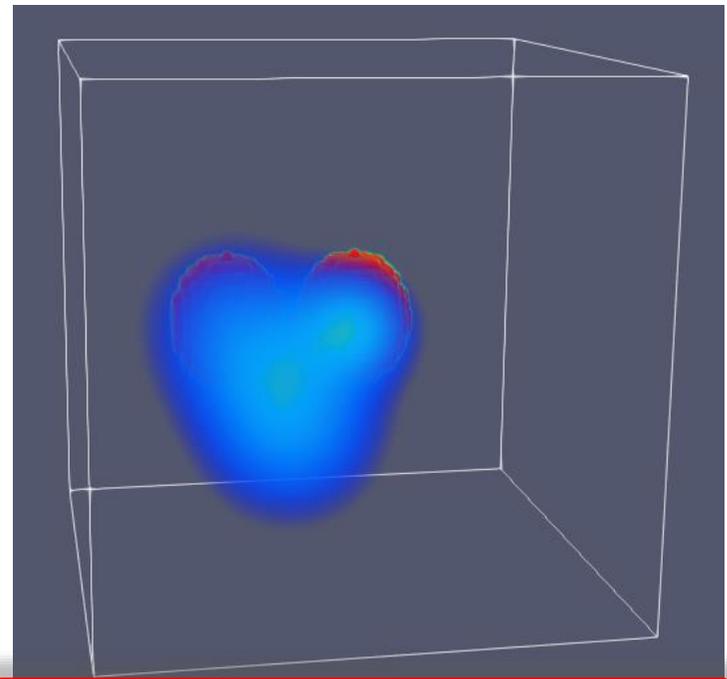
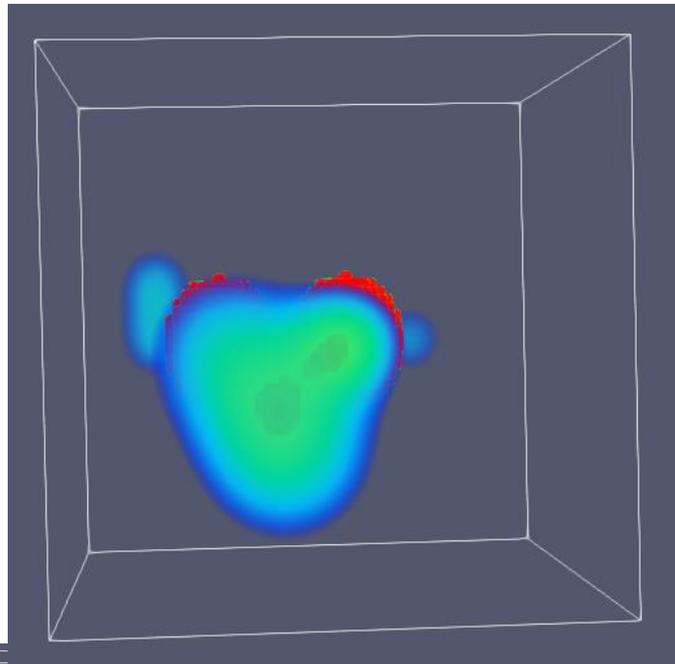
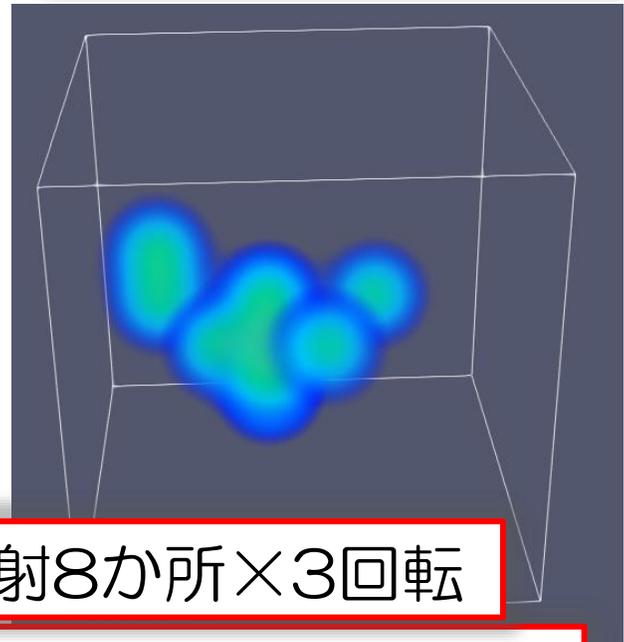
2面モデルの利点

対象に接する面を変えるだけでよい
学習モデルも1度作ったものを回転
させて用いるだけで済む

直径5mm分類

画像再構成
Gaussian Shape

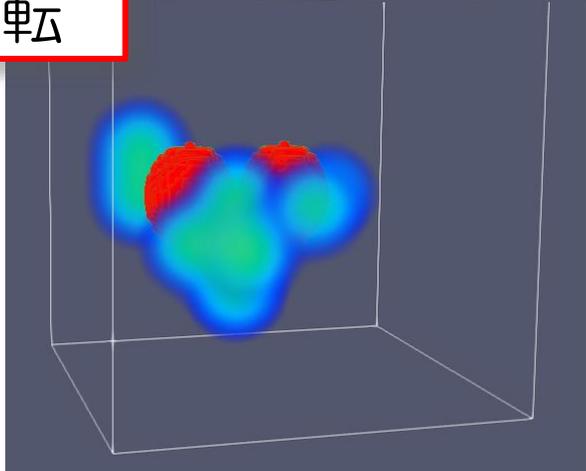
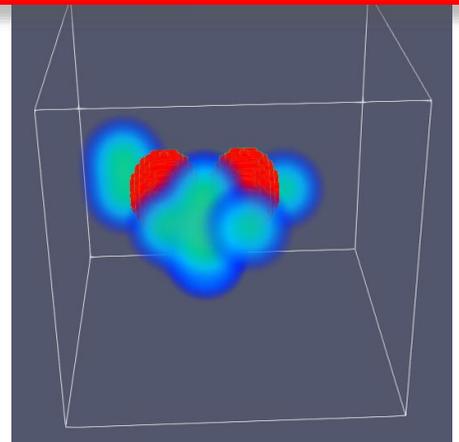
直径6段階分類



入射8か所×3回転

+別入射8×3回転

入射8か所×3回転のみ



小さい検出体のみは、比較的形の細かい情報がみれるが、大きい検出体で構成した画像は、全体にのっぺりとした形状になる。

FIN