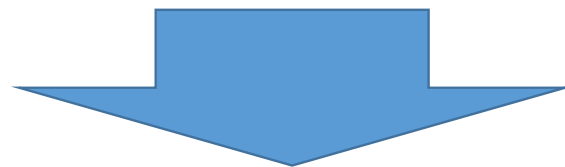


コンピュータグラフィックス(CG)で3次元立体データを表現



**3DデータがCGを描画するアプリやプログラム言語に依存してしまう**

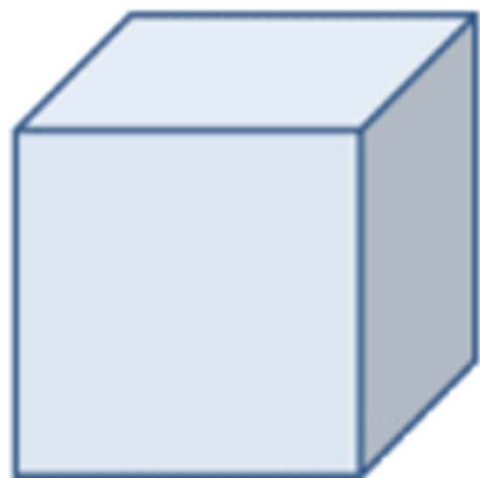
3D立体データは「点の座標だけ」のような単純な構造ではない

CG描画アプリやプラットフォームに依存しない3Dデータ表現は？

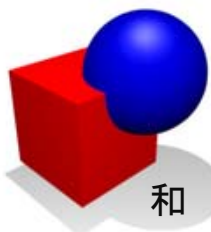
# 3次元形状のデータ表現方法

【参考文献】 <http://d-engineer.com/3dcad/3dmsyurui.html> および ウィキペディア

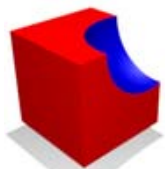
ソリッド



中身の詰まった立体形状を表現  
プリミティブ立体の論理演算  
(加減算など)でモデリングする  
CADでは最も良く用いられる



和

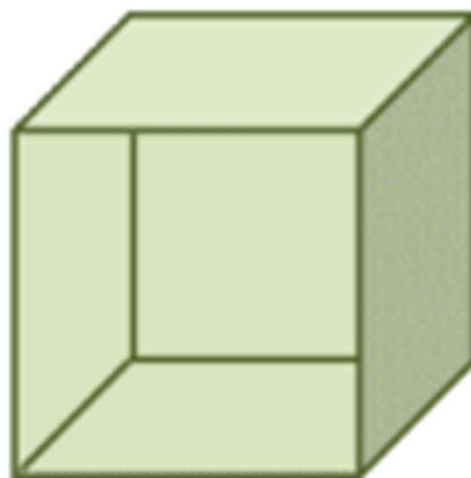


差



積

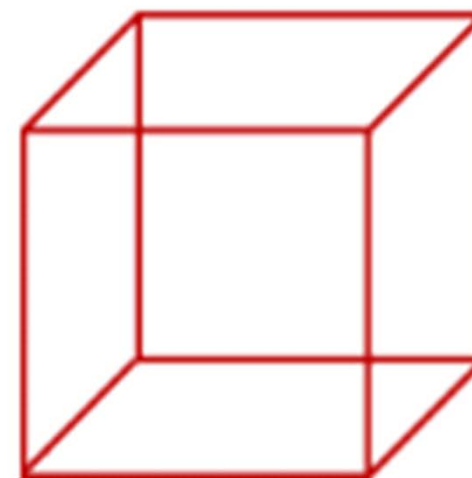
サーフェス



面を張って立体形状を表現する  
面には厚みが無く、一般に  
容積や重心などは計算できない  
CGで良く用いられる

板殻構造物である船舶用CADは  
これがメイン  
(板厚の情報は面の属性として保持)

ワイヤースケーム



線分によって針金細工のように  
立体形状を表現  
立体の頂点や稜線の情報を持つが  
面や体積などの情報を持たない

ポイントクラウド(点群)  
レーザスキャナ等による  
3D観測データ

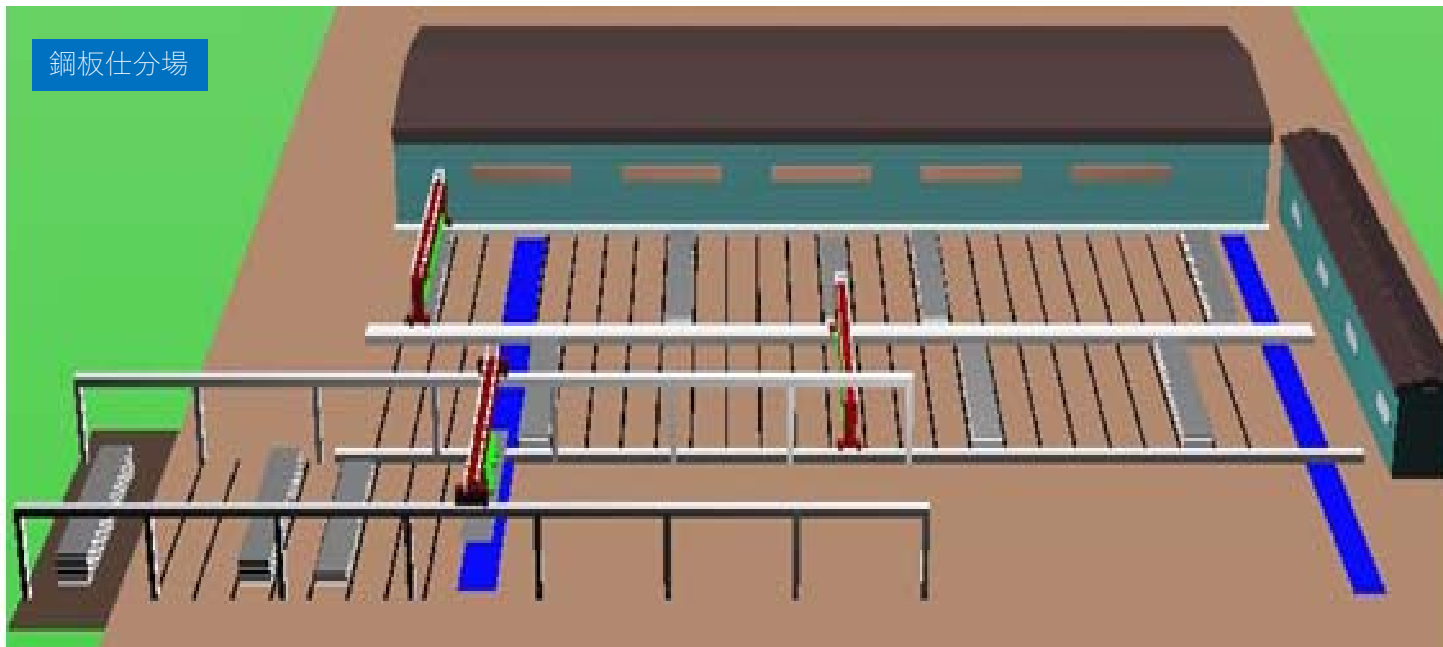


# 仮想現実モデル記述言語: VRML (Virtual Reality Modeling Language)

- ・コンピュータ・グラフィクスで3次元の立体を表現するためのファイルフォーマット
- ・アニメーションも表現できる
- ・テキストファイルにて作成 (拡張子はwrl)
- ・WWWで利用されることを前提、1994年にVRML1.0がWeb3Dコンソーシアムにより策定  
その後VRML97 (ISO/IEC14772-1)に改良された
- ・VRMLの表現能力の限界などから、次世代としてXMLベースのX3Dが登場

VRMLファイルの例  
マークアップ言語ではない

```
#VRML V1.0 ascii
Separator
{
ShapeHints
{
vertexOrdering CLOCKWISE
shapeType UNKNOWN_SHAPE_TYPE
faceType UNKNOWN_FACE_TYPE
creaseAngle 0.0
}
DEF BackgroundColor Info
{
string "0.000000 0.000000 0.000000"
}
DEF Default_View PerspectiveCamera
{
position 8150.783003 3924.765539 -5466.294460
orientation -0.590284 -0.769274 -0.244504 0.987861
focalDistance 6628.960341
heightAngle 0.049209
}
DEF view_1 PerspectiveCamera
{
```

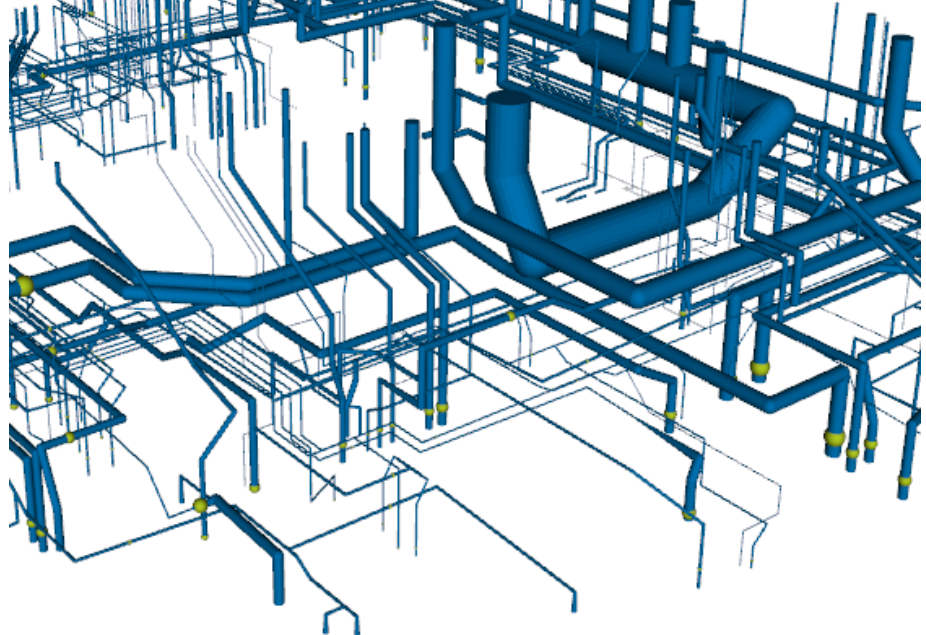
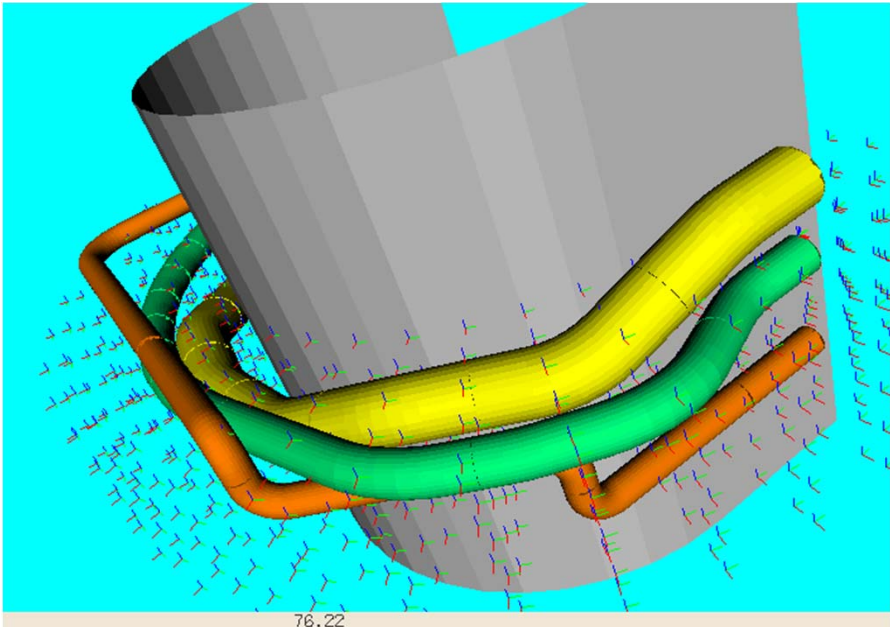


# 仮想現実モデル記述言語: X3D (Extensible 3D) VRMLの後継

- ・ VRML97規格を包含した国際標準規格 ISO/IEC-19775-1
- ・ マルチメディア・Javascriptなどとの結合、HTMLへの埋め込み等に対応
- ・ XMLベースのテキストファイルにて作成 (拡張子はx3d) 3Dファイル変換のための中間フォーマットとしても期待
- ・ 無料のX3Dビューワが利用可能
- ・ VRML同様、CGとして表示することだけを目的としているので現実に存在できない形状もとりうる

3Dのオブジェクトをクリックすると関連付けられたHTML文書がブラウザで表示

```
PipeN = 2
LS011 1 40.0 4.6 1
KeiroN = 1 TenQt = 4
11350.0 -1035.0 8700.0 40.0
11350.0 -787.0 8700.0
10925.0 -787.0 8677.73
10836.73 -855.27 8671.19 40.0
```



# X3Dによる立体記述手順

背景色や視点を設定 Sceneタグ



(1) プリミティブ立体(球/Box/円柱/円錐/文字/押し出し図形/ポリゴン図形)  
について寸法や形状を決めて生成  
Sphere/Box/Cylinder/Cone/Text/Extrusion/ElevationGrid/IndexedFaceSet要素



(2) 生成した立体に対し表面の質感を設定(色/反射度合/発光度合/透明度/テクスチャー)  
Appearanceタグ中のMaterial要素



(3) 生成した立体を配置(平行移動/拡大縮小/回転)  
Transformタグ: translation/scale/rotation 属性

必要な個数分繰り返し



必要に応じ、複数の立体をまとめて配置  
Transformタグ: translation/scale/rotation 属性

# X3Dの記述例

XML宣言

DOCTYPE宣言

```
<?xml version="1.0"?>  
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.2//EN" "http://www.web3d.org/specifications/x3d-3.2.dtd">
```

移動・回転・  
拡大縮小

プリミティブ立体 (Box) を生成して  
色付け、平行移動

```
<!-- Box -->  
<Transform translation="1000.0 1000.0 2100.0">  
  <Shape  
    <Box size="900.0 900.0 1200.0"/>  
    <Appearance  
      <Material diffuseColor='0.5 0.5 0.5' emissiveColor="" transparency='0'/>  
    </Appearance>  
  </Shape>  
</Transform>
```

プリミティブ立体 (球) を生成して  
色付け、平行移動

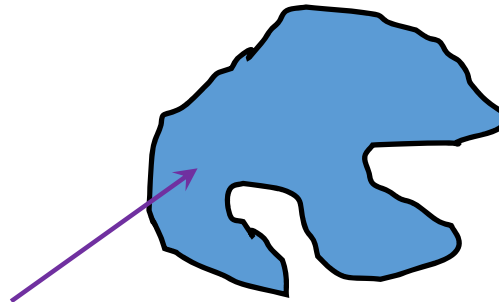
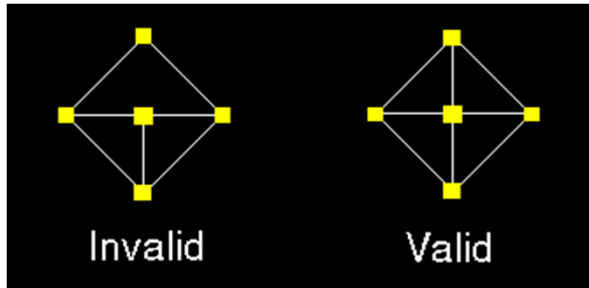
```
<!-- Sphere -->  
<Transform translation=" 2000 1000 500 ">  
  <Shape  
    <Sphere radius ="500.0" />  
    <Appearance  
      <Material diffuseColor='1 0 0' emissiveColor="" transparency='0'/>  
    </Appearance>  
  </Shape>  
</Transform>
```

```
<X3D>  
<Scene>  
  <Background skyColor="0 1 1" />  
  <Transform translation=' -500 -500 -500' scale = ' 1 1 1 ' >  
    <!-- Box -->  
    <Transform translation="1000.0 1000.0 2100.0">  
      <Shape  
        <Box size="900.0 900.0 1200.0"/>  
        <Appearance  
          <Material diffuseColor='0.5 0.5 0.5' emissiveColor="" transparency='0'/>  
        </Appearance>  
      </Shape>  
    </Transform>  
    <!-- Sphere -->  
    <Transform translation=" 2000 1000 500 ">  
      <Shape  
        <Sphere radius ="500.0" />  
        <Appearance  
          <Material diffuseColor='1 0 0' emissiveColor="" transparency='0'/>  
        </Appearance>  
      </Shape>  
    </Transform>  
  </Transform>  
</Scene>  
</X3D>
```

X3D本体

# STL (Stereo lithography)

- 3D CADソフト用ファイルフォーマットの1つ  
3Dプリンタで造形する入力ファイル形式として一般的
- 3角形 (facet) の集合として立体を表現 (サーフェスモデル)  
ただし下図の右側のように隣接する3角形の頂点を共有しない表現はNG

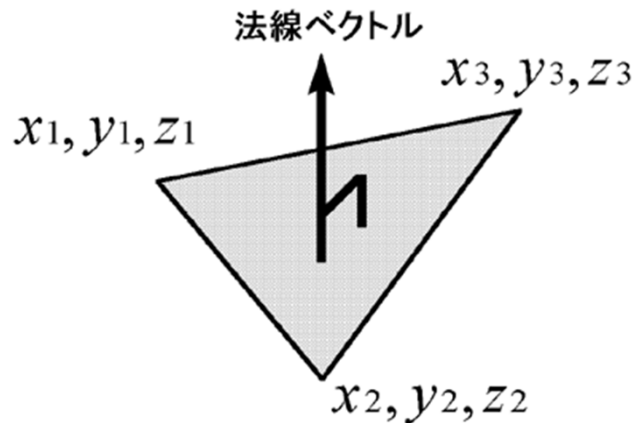


STLファイルの構成  
(アスキー形式はテキストファイル)

```
solid
...
facet normal 0.00 0.00 1.00
  outer loop
    vertex 2.00 2.00 0.00
    vertex -1.00 1.00 0.00
    vertex 0.00 -1.00 0.00
  endloop
endfacet
...
endsolid
```

- 実在可能な立体としては**3角形で囲まれた領域が閉じている**必要がある

- 1つ1つの3角形はそれを表現する  
頂点の並びによって面の裏表を表す  
(反時計方向に回転する頂点の並びが表)



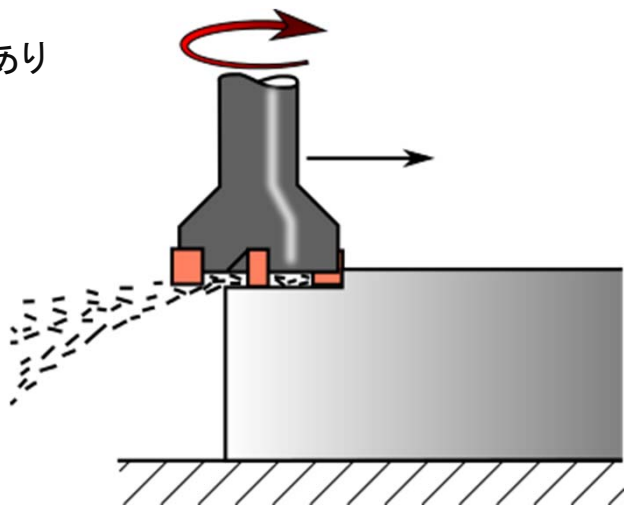
- 基本的にポリゴン形状だけの表現だが、  
facet毎に色の指定をする場合がある

アスキー形式では  
ファイル容量が大きくなりすぎるため  
バイナリ形式もある

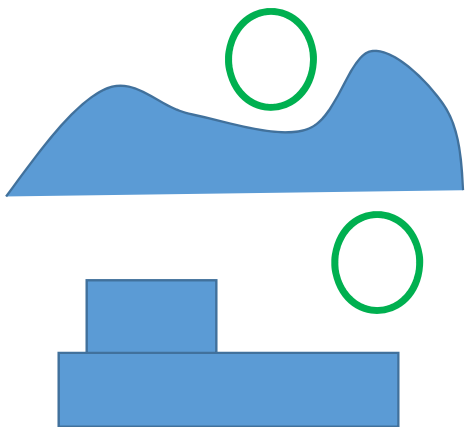
# フライス加工による3D立体の造形

「エンドミル」と呼ばれる刃物を回転させて材料を削り造形

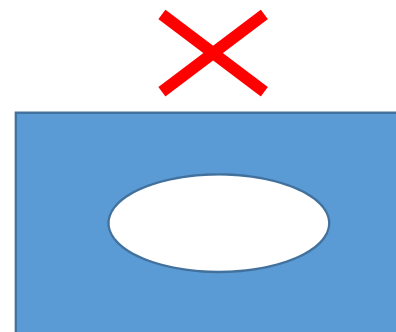
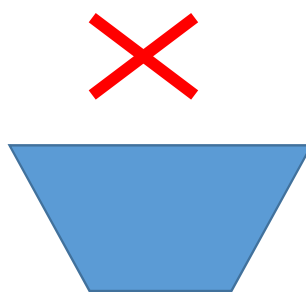
- × 造形可能な形状に制約あり
- × 大量の削りかす発生



「上に凸」な立体なら造形可能だが...



「逆テーパ」や中空な形状は造形できない



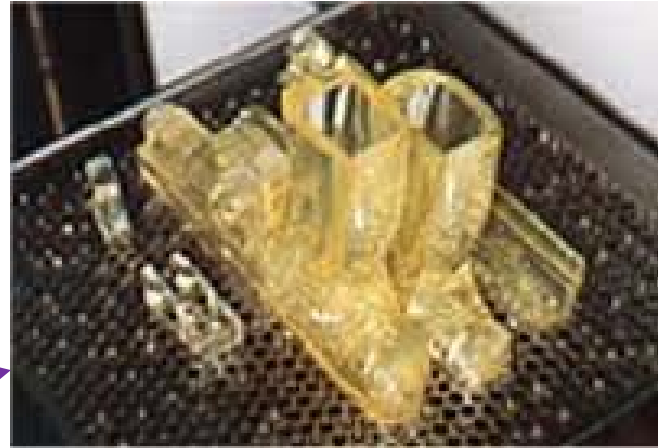


# 3Dプリンタによる3D立体の造形

立体データをスライスした断面形状を積層して造形していく  
造形できる形状の制約は小さいが、材料に制約がある

## 光造形法

光を照射すると硬化する樹脂を積層 光硬化樹脂は高価  
60万円台から数千万円台まで**精度**によっていろいろ



樹脂造形サンプル写真(粉末固着方式)

## 粉末法

粉末をレーザーで焼結したり接着剤で固めて積層 一般に脆くなるが**着色が容易**



## 熱熔融積層法

熱可塑性樹脂を加熱して細いノズルから押し出して積層 材料はABSかPLA  
**積層方向に直交する方向の強度は大きい**  
10万円台の最も普及している3Dプリンタはこのタイプ



画像出典: 株式会社セイコーアイ・インフォテック

## シート積層法

カッティングプロッタで切り込みを入れた紙を積層  
金属シートを積層して超音波で圧着するなどもある

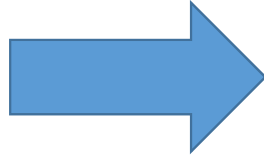
## インクジェット法

インクジェットプリンターでインクを積層

私見だが、実験用ロボット試作に要する作業時間が  
今まで1ヶ月以上かかっていたものが  
3Dプリンタで1週間-3日程度にまで短縮

# 3Dプリンタ造形時のSTLデータのスライス作業について

STLはサーフェスモデル



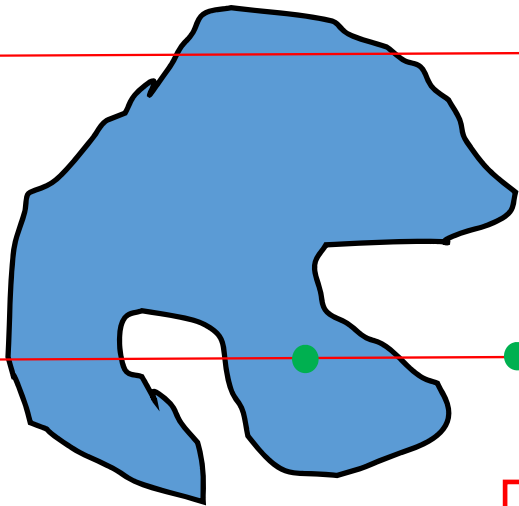
どの領域が物体の内側なのか  
についての情報を持っていない



「STLで表された立体は、3角形で囲まれた領域が閉じている」

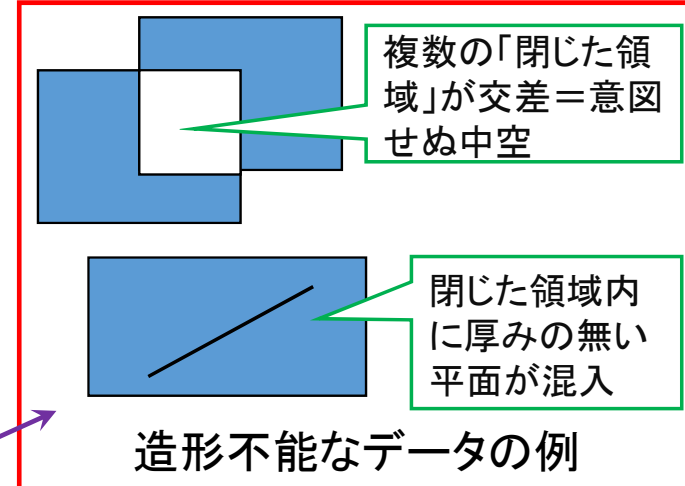


この性質を利用して  
物体内側を判定・造形



「閉じた領域」の境界  
と交差する直線の交  
点は必ず偶数個

任意の点からスタートした直線が、  
境界面と偶数回交差している場合、  
その点は領域外  
奇数回交差している場合は領域内



現実に存在できるかどうかを意識していないCGモデル (VRMLやX3D) では  
上記のSTLの条件を満たさず、単なる変換だけでは造形できないケースあり注意

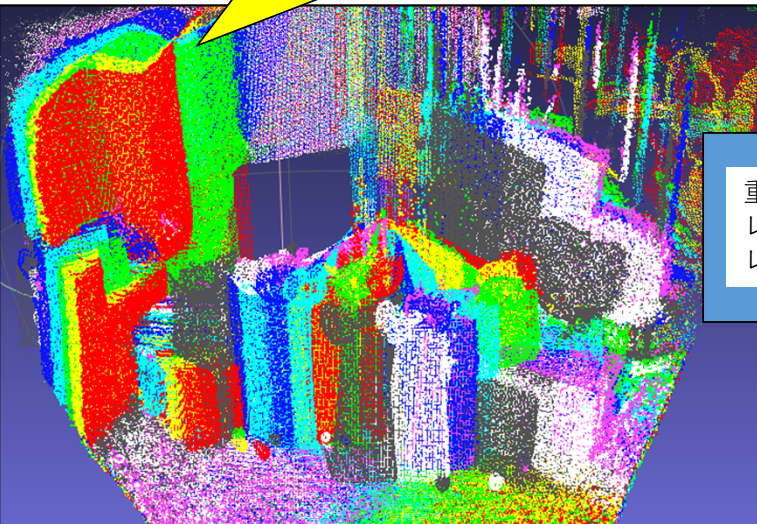
# ポイントクラウド(PCD: point cloud data)ファイルフォーマット

- ・レーザースキャナやキネクトのような深度計測デバイスで観測される3次元点群データの書式  
自動車の自動運転や自律ロボットの研究等で普及
- ・基本的に点のxyz座標の集合で表されるが、色を含む場合もある
- ・PCL(Point Cloud Library) <http://pointclouds.org/>  
に様々な点群処理を行うプログラムが無料にて利用可能

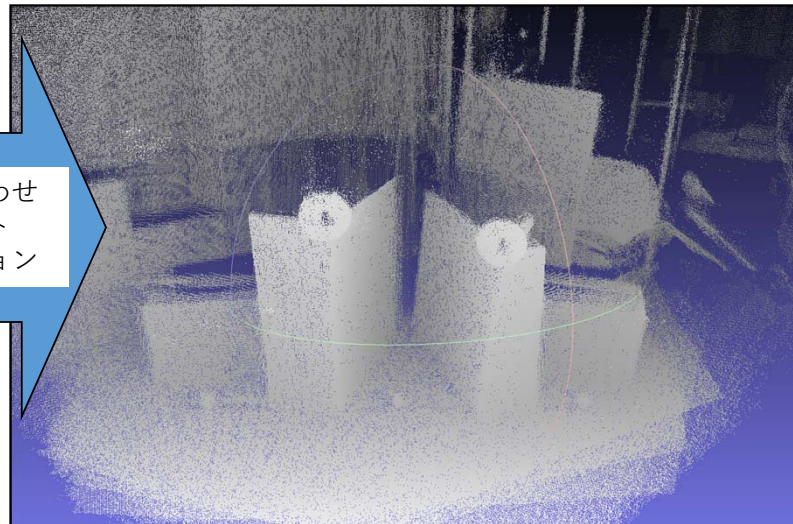
PCDファイルの例

レジストレーション  
平面検出  
etc..

キネクトを手に持って周囲を廻り、  
多方向から高速に形状計測

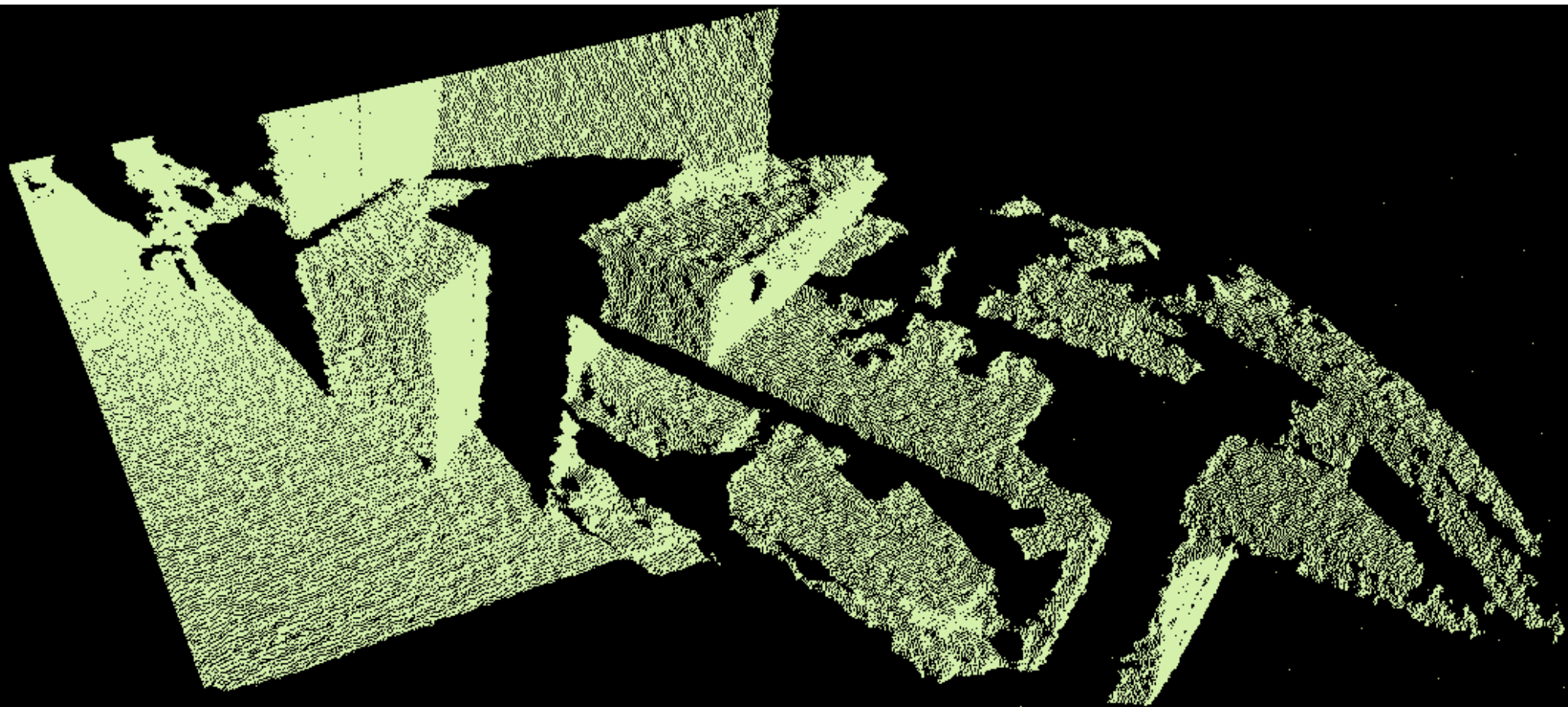


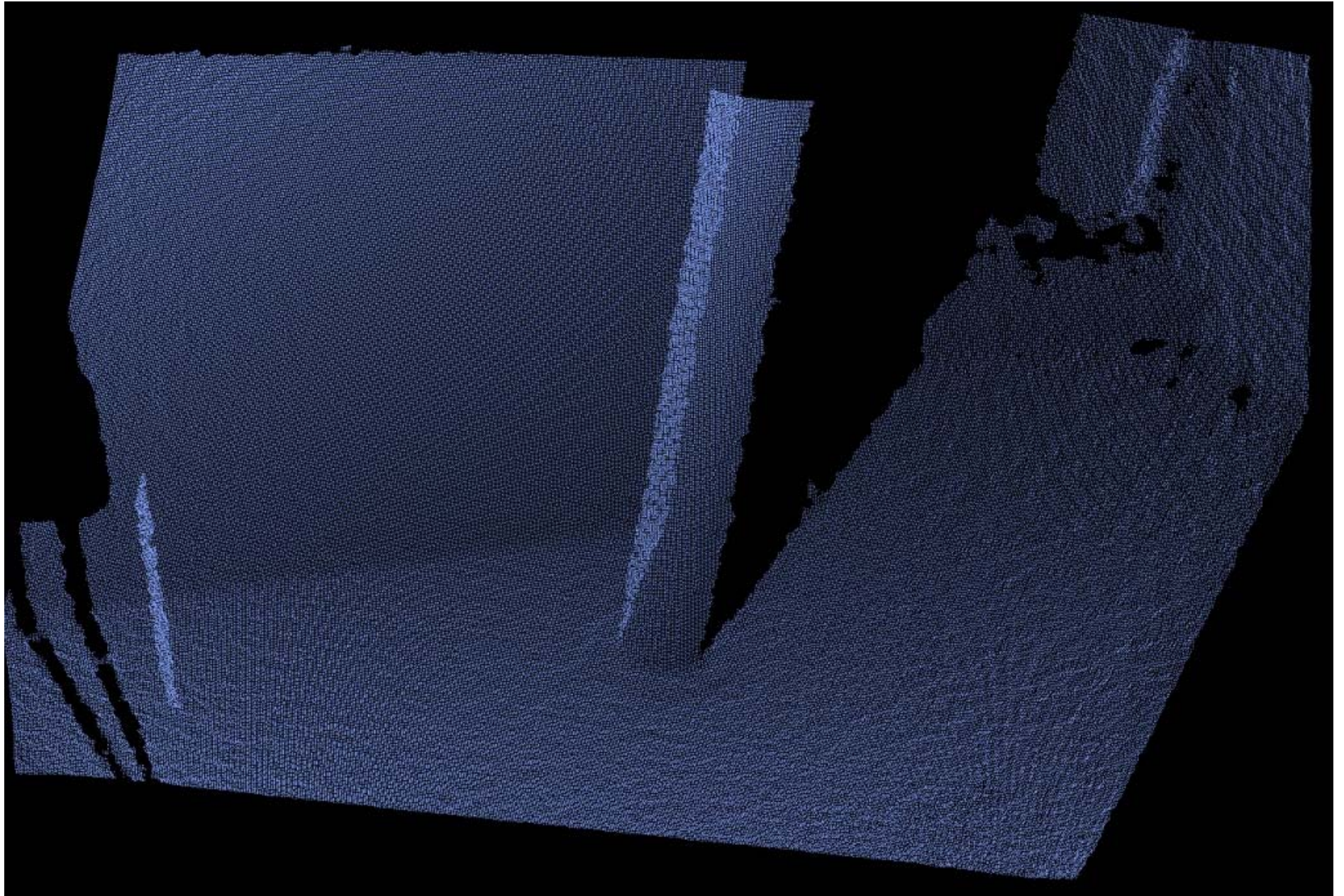
重ね合わせ  
レジスト  
レーション

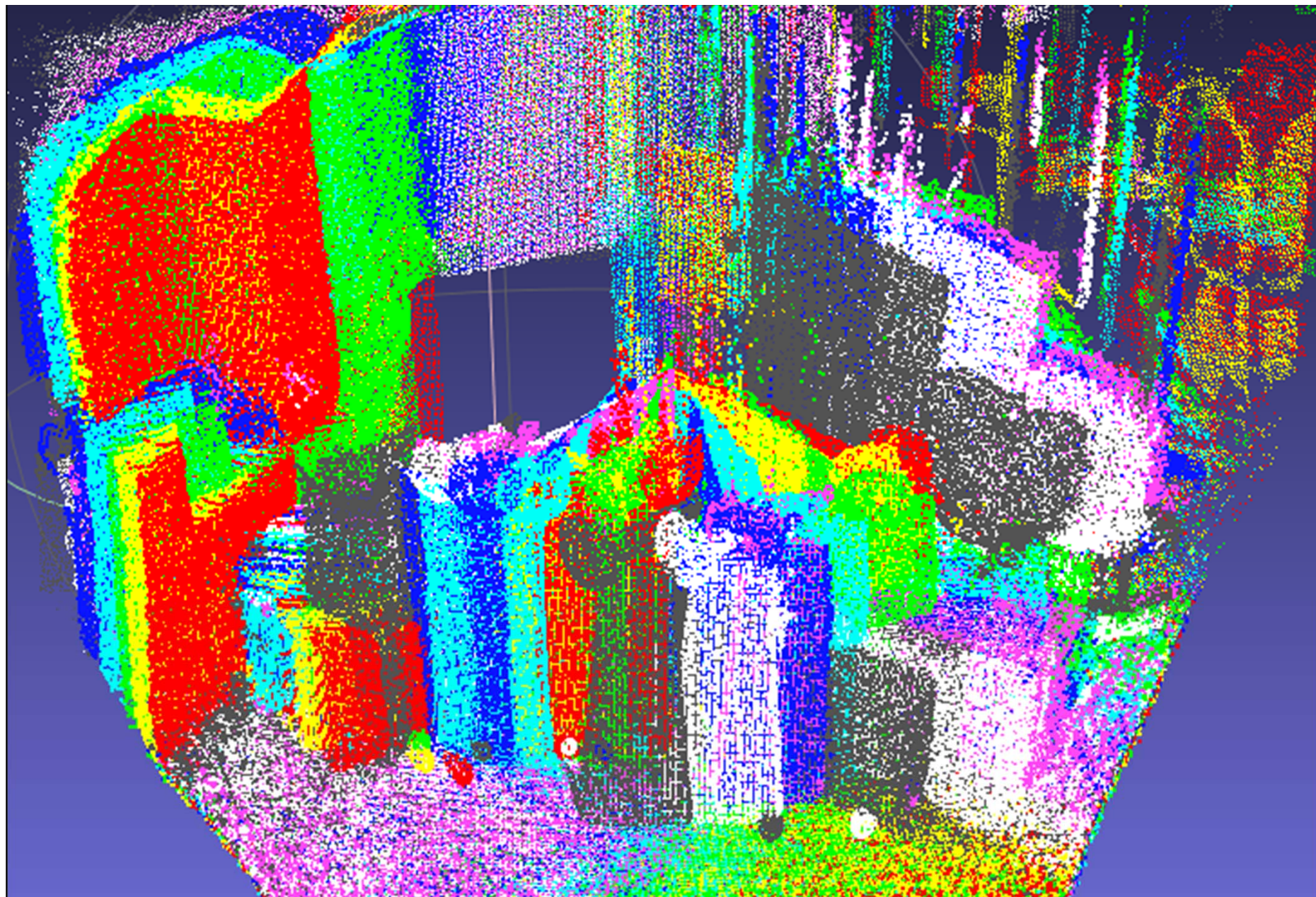


```
# .PCD v.7 - Point Cloud Data file format
VERSION .7
FIELDS x y z rgb
SIZE 4 4 4 4
TYPE F F F F
COUNT 1 1 1 1
WIDTH 213
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 213
DATA ascii
0.93773 0.33763 0 4.2108e+06
0.90805 0.35641 0 4.2108e+06
0.81915 0.32 0 4.2108e+06
0.97192 0.278 0 4.2108e+06
0.944 0.29474 0 4.2108e+06
0.98111 0.24247 0 4.2108e+06
0.93655 0.26143 0 4.2108e+06
0.91631 0.27442 0 4.2108e+06
0.81921 0.29315 0 4.2108e+06
0.90701 0.24109 0 4.2108e+06
0.83239 0.23398 0 4.2108e+06
```

点群データからCADデータを作成するのは大変難しいがARMONICOSという商用ソフトが存在する









## まとめ

- (1) 3D形状の表現方法: (1) ワイヤーステイクモデル (2)サーフェスモデル (3)ソリッドモデル  
(0) ポイントクラウド(点群)
- (2) 仮想現実モデル記述言語: **VRML**(古い)、**X3D**(XMLベースでVRMLの後継)  
コンピュータに表示することだけを意図しているので現実には作れない形状かも
- (3) **STL**: CAD用ファイルフォーマットの1つ/3Dプリンタで造形  
立体表面を多数の3角形で包み込んで形状表現  
領域は閉じていなければならない
- (4) 3D立体の造形方法: フライス加工/3Dプリンタ
- (5) ポイントクラウド: **PCD**

## レポート課題:

自分の興味のある立体をX3Dで記述し、X3Dビューワで表示して確認せよ。(BS Contactを推奨)  
またその立体のどのような点が興味深いのか1行でも良いので説明を添付せよ。

### 【提出方法】

上記の説明をメール本文に記述し、上記のx3dファイルを直接メールに添付して、  
[kimura@nams.kyushu-u.ac.jp](mailto:kimura@nams.kyushu-u.ac.jp) 宛てに「第6回船舶海洋情報学レポート」とのタイトルで送信せよ。

下記の講義ページのURLにアップロードされているX3Dのテンプレートを利用せよ

<http://sysplan.nams.kyushu-u.ac.jp/gen/edu/NavalInformationProcessing/2018/index.html>