◆　**RGB-Dセンサー**



**３次元の色と形**を計測する装置．得られるデータは**色付きの３次元点群データ**．



図．RGB-Dセンサーの１つ．Xtion Pro の外観（amazon.co.jp より転載）

Kinectもそうです．

**◆ ３次元点群データ**

　　３次元の点がたくさん記述されたデータ．

　　実は，それぞれの**点**は，

　　　・**座標 (x, y, z) と色情報 (R, G, B) を属性**として持つ場合

　　　・**座標 (x, y, z) とラベル番号を属性**として持つ場合

などいろいろありえる．

　　　　（要点は）

**1つのオブジェクトの表面**　＝　**多数の点の集まり**（３次元点群データ）

　　　　として扱う

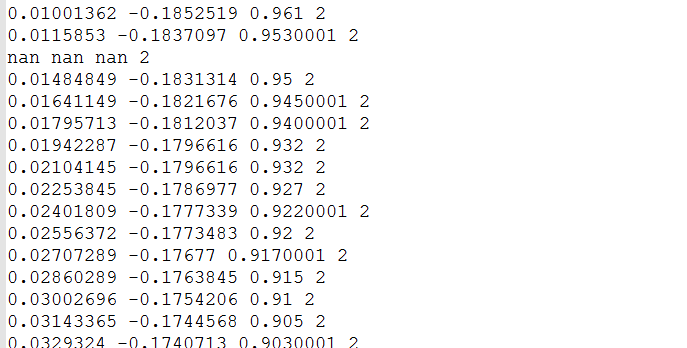
**◆ ３次元点群データのファイル形式の例**

アスキー形式

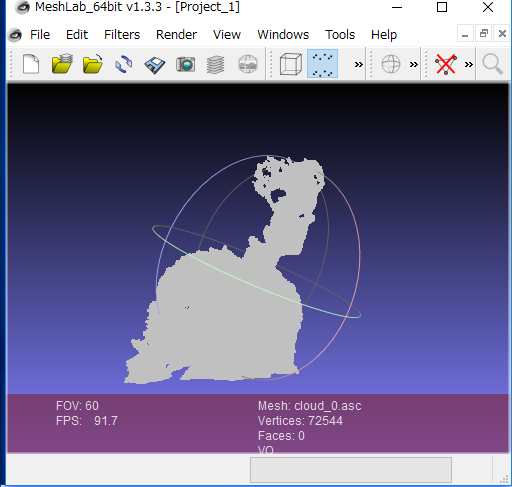
**１行で１つの点**．半角の空白文字でデータが区切られている．

**行の中には，x, y, z の値**．あわせて付加情報が付いてくることが多い．

前も使ったファイル cloud\_0.asc では，**ラベル番号**（「2」などの数値）が**付いている**．



図．3次元点群データファイルの例

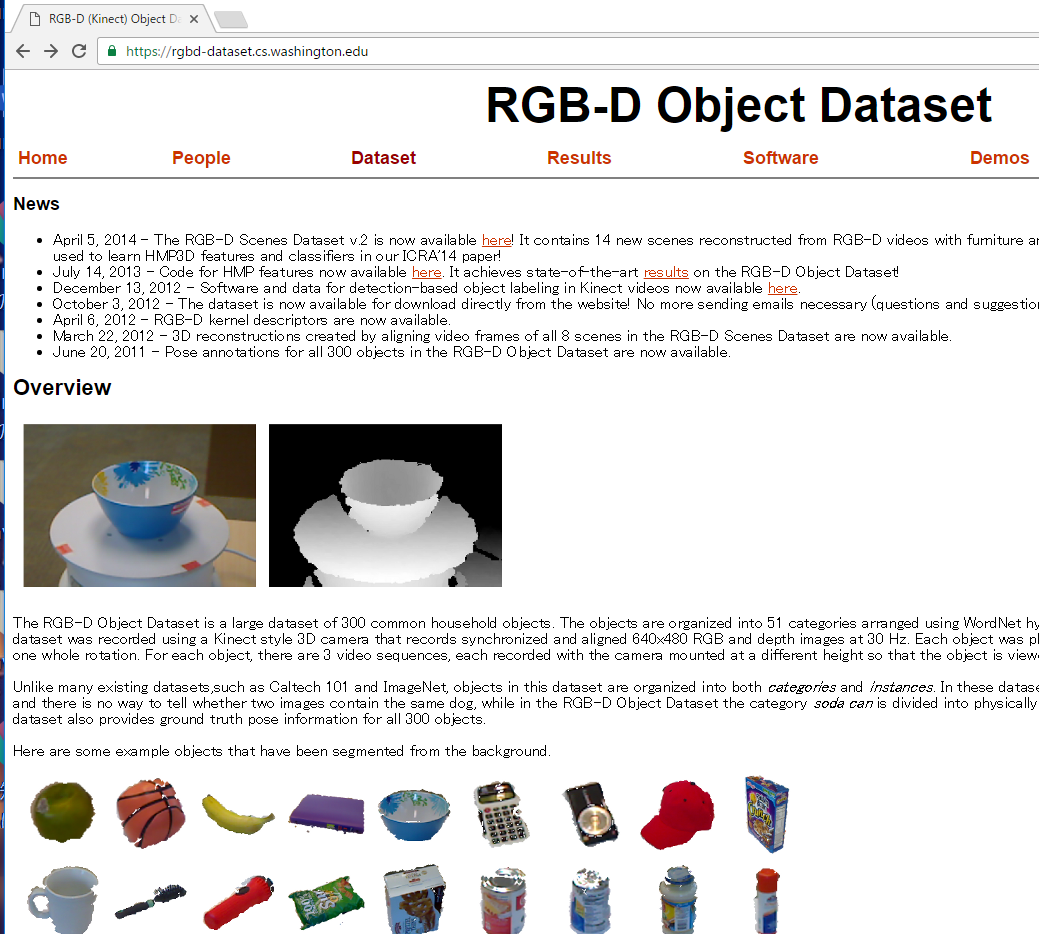


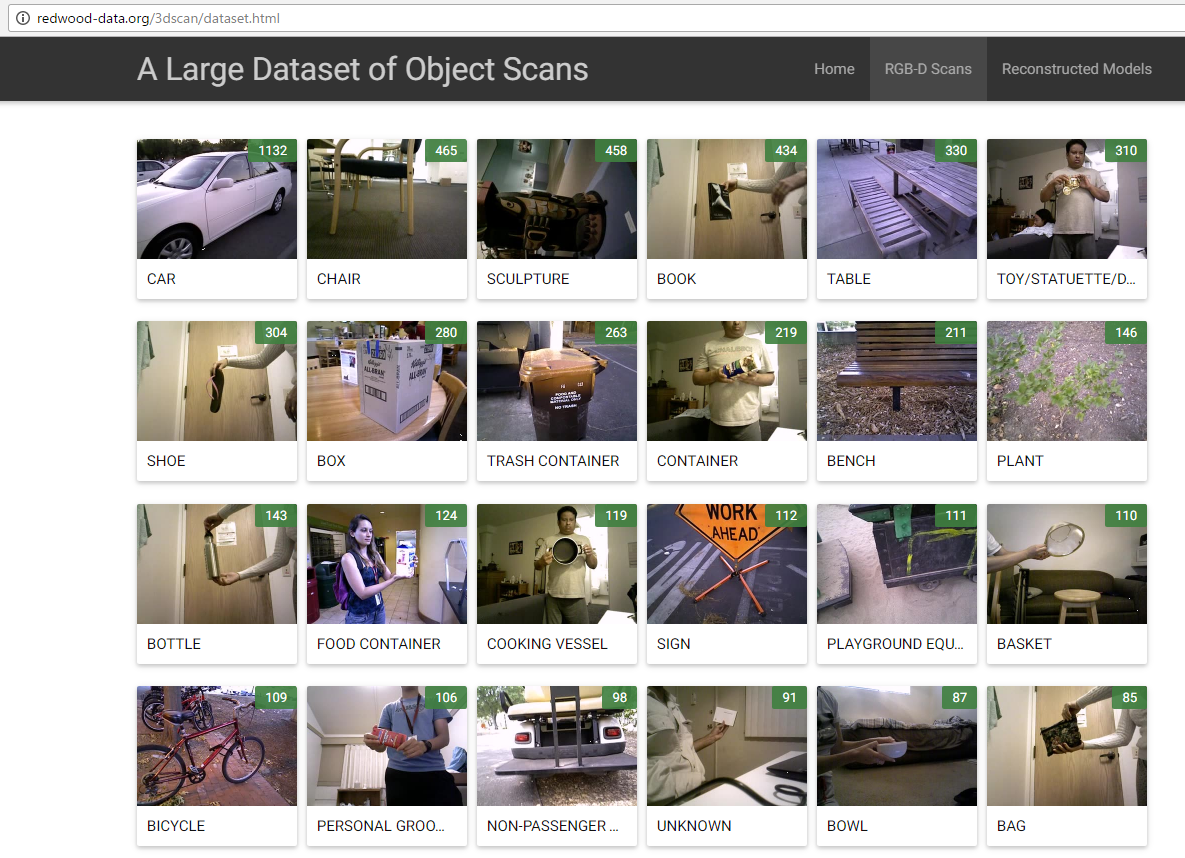
図．meshlab で表示

**◆　インターネットで公開されている RGB-D データセットの例**



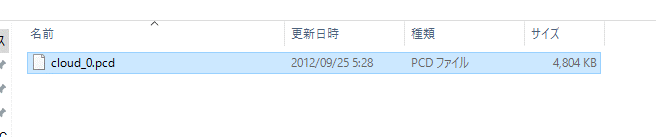
https://rgbd-dataset.cs.washington.edu/



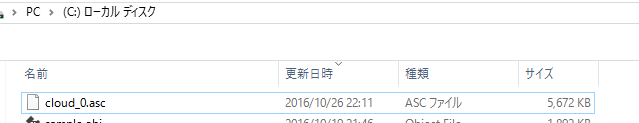
 http://redwood-data.org/3dscan/dataset.html

**３．3次元点群データをPythonで表示**



① 前も使った **cloud\_0.asc**というファイル名の**点群データファイル**を使う

② cloud\_0.asc を **C:\ の直下**にコピー



**③ 点群データを3次元プロット**

※　「pandas.read\_table("**C:/cloud\_0.asc**", sep="\s+", header=None)」は，**半角の空白文字で区切られたデータファイル**を**読み込む**ときの決まり文句

import pandas

**a** = pandas.read\_table("**C:/cloud\_0.asc**", sep="\s+", header=None)

a = a.dropna()

fig = plt.figure()

ax = Axes3D(fig)

ax.scatter3D(np.ravel(**a[0]**),np.ravel(**a[1]**),np.ravel(**a[2]**), s=1, depthshade=False)

plt.show()

**別ウインドウ**が開く．マウス操作でいろいろ回転ことは可能だが，処理が重いので，慌てずに操作する．

