

# Dynamic fusion

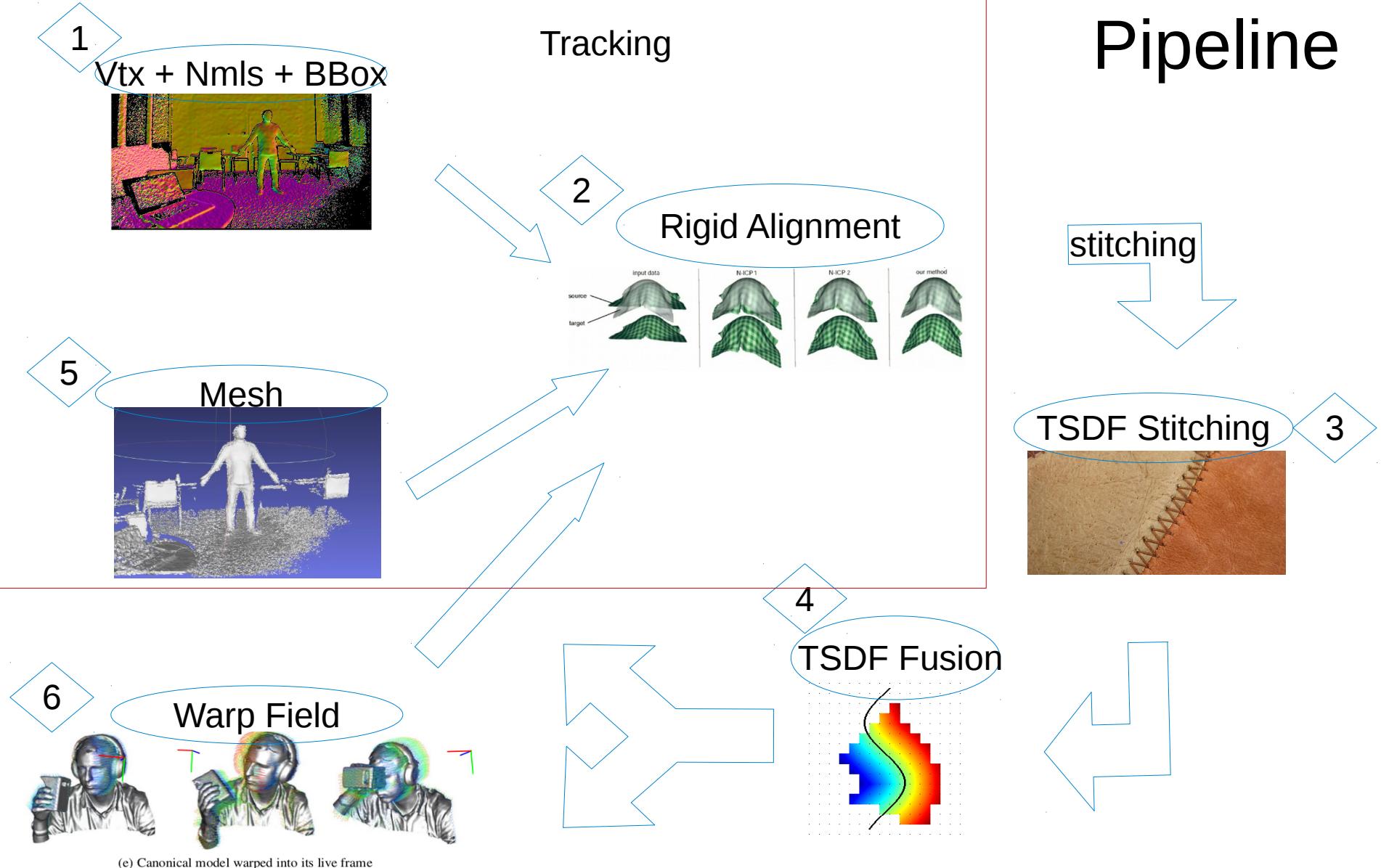


Internship Week 16  
Tracking Mesh and Fusion (IV)  
7<sup>th</sup> June 2017

# Last meeting

- Previously
  - Identify source
  - Mesh data checked
  - Normals correspondence
- Plan for today's meeting:
  - Identify source in tracking with Mesh
  - Mesh Tracking
  - Fusion

# Pipeline



# Progress

- Tracking with the Mesh
  - 1) Transform Mesh with current Pose
  - 2) Projection Vtx of Mesh in current depth frame
  - 3) Compare Vtx
  - 4) Compare Nmls
  - 5) Add correspondence to matrix
  - 6) ICP

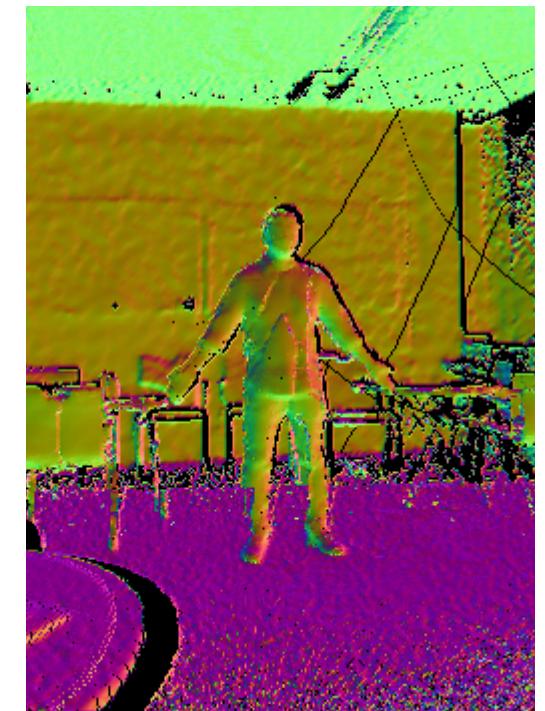
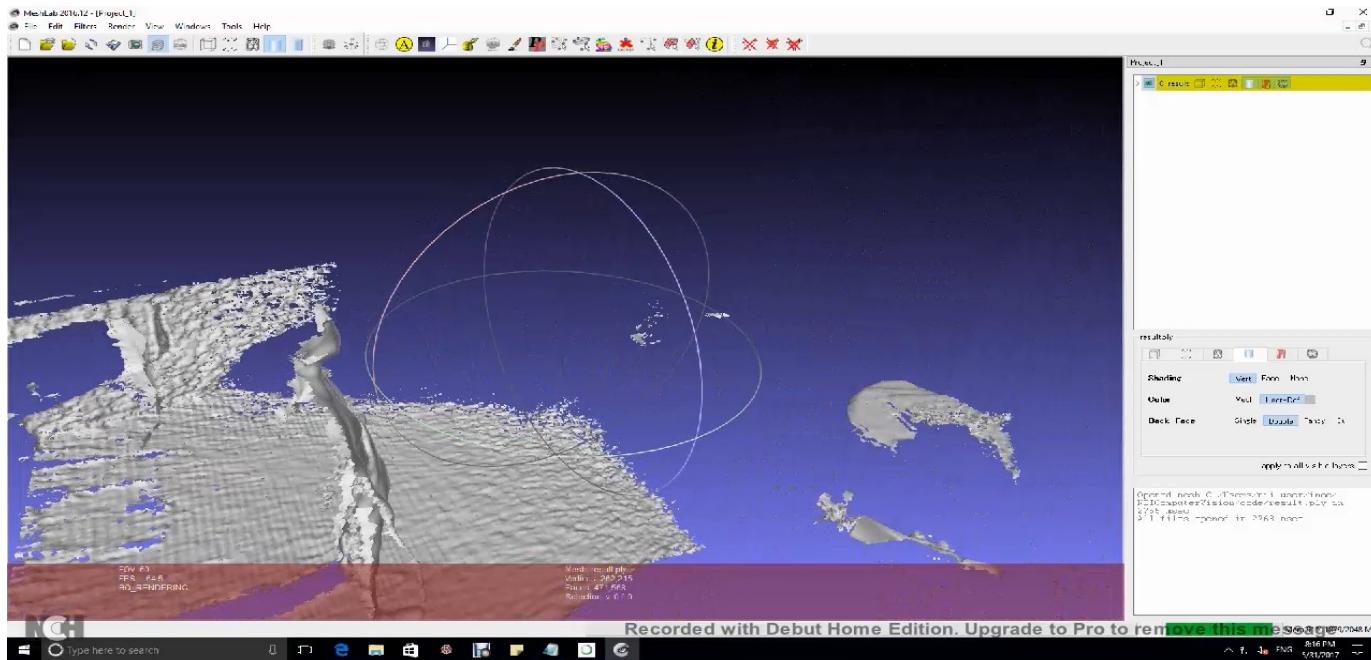
# Progress

- Normal correspondence:
  - If condition block normals transform in first iteration
  - Mesh 9232 → 32885 nmls and Final 28619
  - F2F without Mesh: about 60000
- Bad Pose ?

# Progress

- Inverse result in mesh tracking

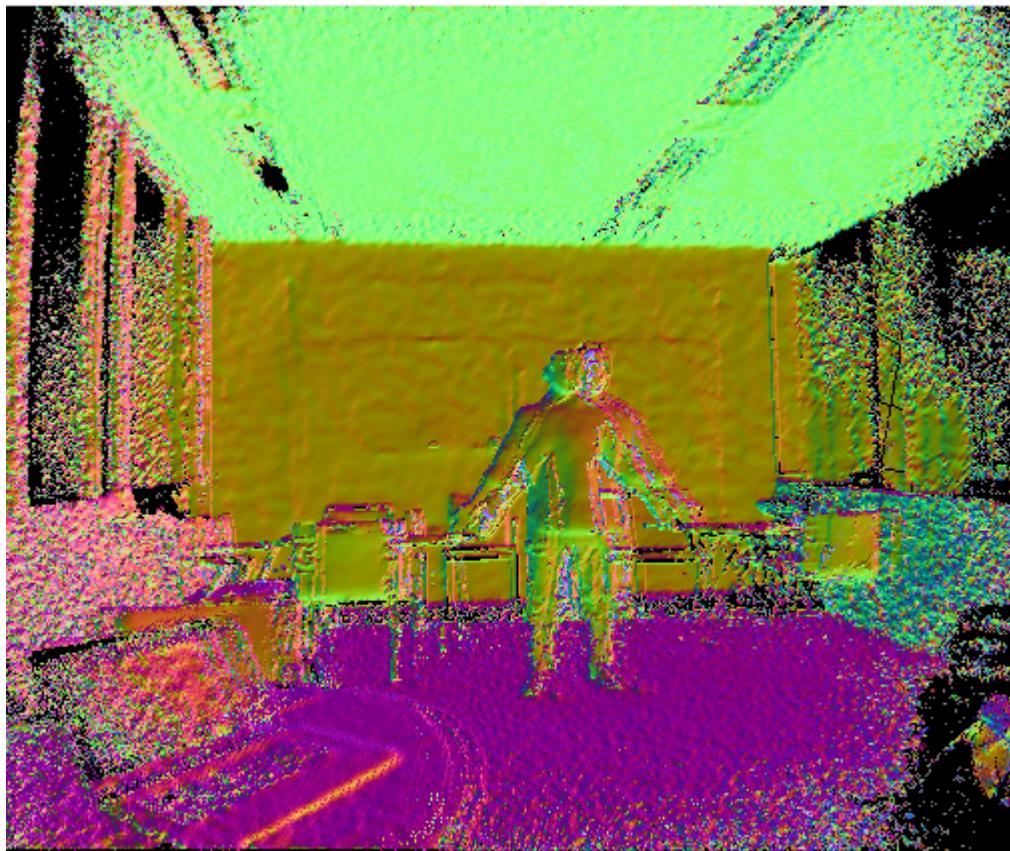
Fusion of 5 images



# Progress

- Inverse result + double alignment in mesh tracking

Fusion of 5 images



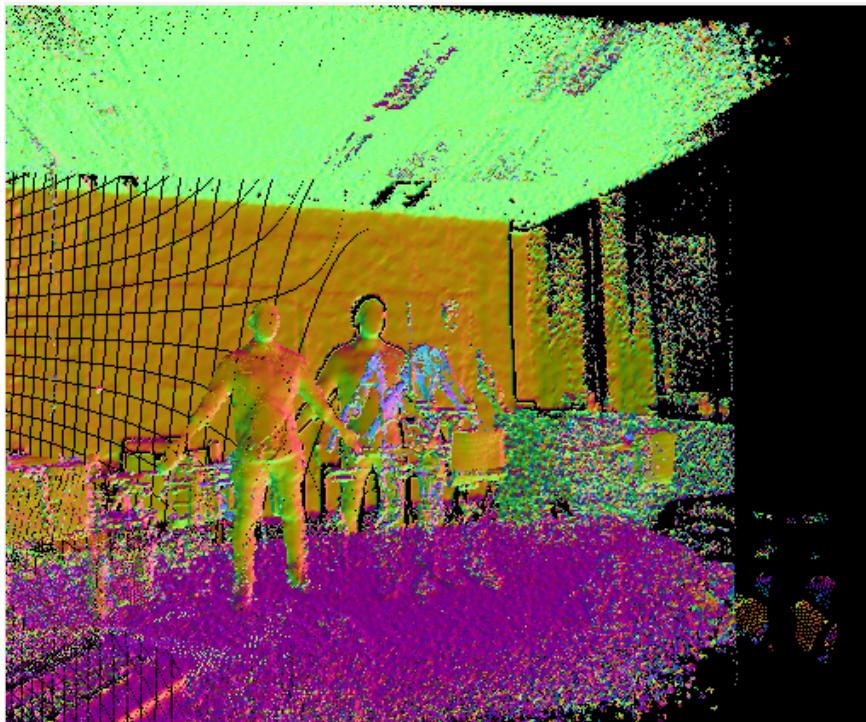
I thought there were a difference between the mesh and the depth map (RGBD.Vtx, RGBD.Nmls). Therefore I tried to align the two.

Result : worse

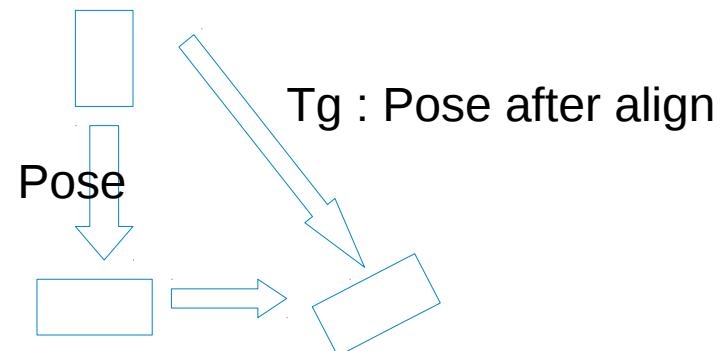
Conclusion : Align the depth map to the mesh is not the issue

# Progress

- Realizing why self.Pose (not IncPose) in mesh tracking:
  - My thoughts : Mesh also pose updated → incremental pose.



In fact mesh done in the first Pose always transform current image in first Pose for TSDF.



Inc=?  
Inc =  $Tg^{-1} \cdot Pose$

# Progress

- Computation of `delta_transfo`:

Until now in the code:

```
delta_transfo = LA.inv(Exponential(delta_qsi))
```

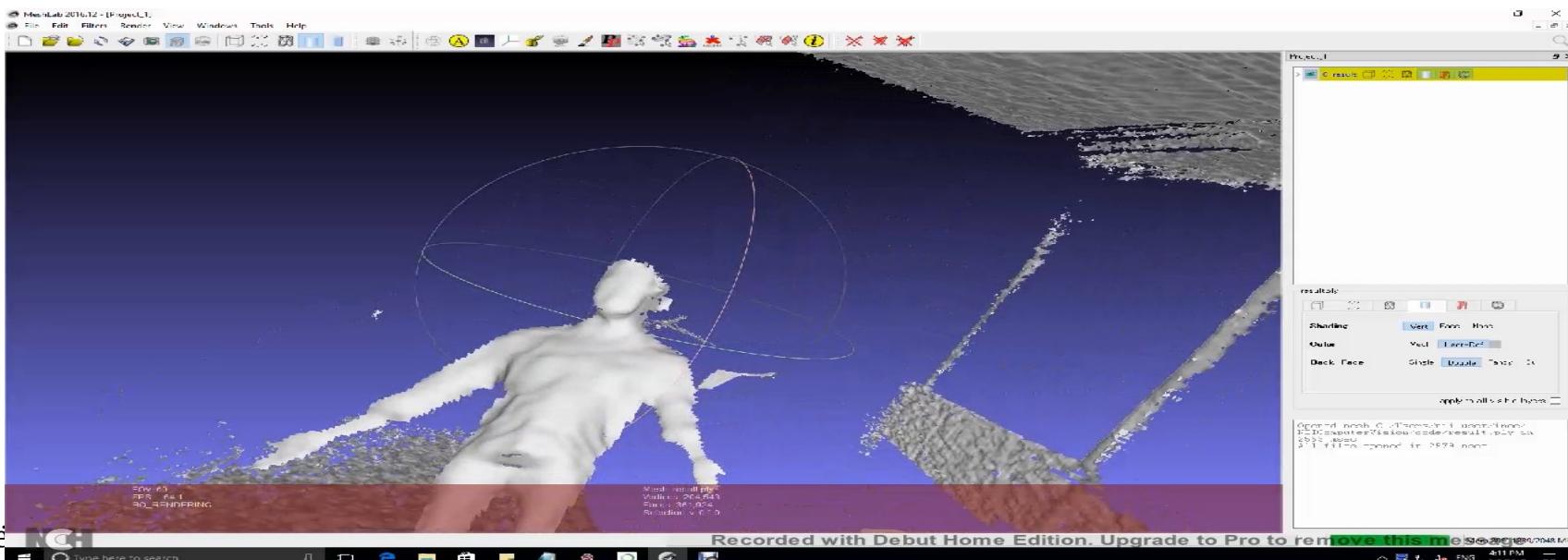
What to should be:

```
delta_transfo = Exponential(delta_qsi)
```

```
delta_transfo[0:3,0:3] = LA.inv(delta_transfo[0:3,0:3])
```

```
delta_transfo[0:3,3] = -np.dot(delta_transfo[0:3,0:3],delta_transfo[0:3,3])
```

```
delta_transfo[3,3] = 1.0      (missing at first)
```



# Progress

- Test on Pose with mesh tracking with hand made transformation

Align former image to current image (hand made current image)

```
[[ 0.99935008 -0.02989301  0.02014532  0.00984999]
 [ 0.03009299  0.99950006 -0.0096977  0.02007494]
 [-0.01984535  0.01029763  0.99975003  0.01500005]
 [ 0.          0.          0.          1.          ]]
```

Doing the same but with mesh tracking

```
[[ 0.99277961  0.03579152 -0.1144887  0.01156546]
 [-0.03405555  0.9992739  0.01708358 -0.03543999]
 [ 0.11501702 -0.01306125  0.99327767 -0.03661365]
 [ 0.          0.          0.          1.          ]]
```

Should be the same if  
the alignment is well  
made

# Progress

- Test on Pose with mesh tracking

Align current image to former image

```
[[ 0.99277961 -0.03405555  0.11501702 -0.00847769]
 [ 0.03579152  0.9992739 -0.01306125  0.03452209]
 [-0.1144887  0.01708358  0.99327767  0.03829708]
 [ 0.          0.          0.          1.          ]]
```

Inverting it:

```
[[ 0.99277956  0.03579152 -0.1144887  0.01156546]
 [-0.03405555  0.99927394  0.01708358 -0.03543999]
 [ 0.11501701 -0.01306125  0.99327763 -0.03661365]
 [ 0.          0.          0.          0.          ]]
```

Align former image to current image:

```
[[ 0.99935007  0.03009299 -0.01984535 -0.01015002]
 [-0.02989301  0.99950004  0.01029763 -0.01992492]
 [ 0.02014532 -0.0096977  0.99975002 -0.01500005]
 [ 0.          0.          0.          1.          ]]
```

Should be the same if  
the alignment is  
invertible

# Progress

- Test on Pose with frame to frame tracking

Align current image (to list) to former image

```
[[ 0.99935007 -0.02989301  0.02014532  0.00984999]
 [ 0.03009299  0.99950004 -0.0096977  0.02007494]
 [-0.01984535  0.01029763  0.99975002  0.01500005]
 [ 0.          0.          0.          1.        ]]
```

Align current image to mesh (to list)

```
[[ 0.99277961  0.03579152 -0.1144887  0.01156546]
 [-0.03405555  0.9992739  0.01708358 -0.03543999]
 [ 0.11501702 -0.01306125  0.99327767 -0.03661365]
 [ 0.          0.          0.          1.        ]]
```

Remind : invert Align current image to former image

```
[[ 0.99277956,  0.03579152, -0.1144887,  0.01156546],
 [-0.03405555,  0.99927394,  0.01708358, -0.03543999],
 [ 0.11501701, -0.01306125,  0.99327763, -0.03661365],
 [ 0.          ,  0.          ,  0.          ,  1.        ]]
```

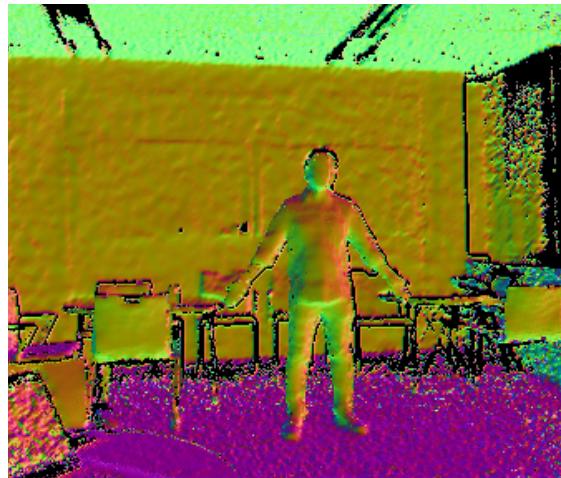
Similar to frame to frame tracking

# Progress

- **Changing mesh tracking:**

Transform current image instead of mesh

- 1) Transform **current image with inverse of current Pose**
- 2) Projection Vtx of Mesh in current depth frame
- 3) Compare Vtx
- 4) Compare Nmls
- 5) Add correspondence to matrix
- 6) ICP



self.Pose: with depth image transform into lists.

```
[[ 0.99935007 -0.02989301  0.02014532  0.00984999]
 [ 0.03009299  0.99950004 -0.0096977  0.02007494]
 [-0.01984535  0.01029763  0.99975002  0.01500005]
 [ 0.          0.          0.          1.        ]]
```

With Mesh

```
[[ 0.99934942 -0.02991215  0.02014929  0.010002 ]
 [ 0.03011265  0.9994992 -0.0097222  0.02017882]
 [-0.01984838  0.01032262  0.99974972  0.0150707]
 [ 0.          0.          0.          1.        ]]
```

# Progress

- Test mesh tracking in fusion:

Fuse 2 Images : Comparing F2F and MT2 in Fusion

Pose : MT2 fusion (Input : current image, mesh lists)

```
[ [ 9.99988794e-01 -3.19557916e-03 -3.49320681e-03 8.27364856e-04]
[ 3.20624560e-03 9.99990225e-01 3.05214128e-03 1.19936708e-02]
[ 3.48341931e-03 -3.06330714e-03 9.99989212e-01 -1.11433035e-02]
[ 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

Pose : F2Ffusion (input = current image and former image)

```
[ [ 9.99980605e-01 -4.78758656e-03 -3.98349311e-03 -5.87553262e-04]
[ 4.81498672e-03 9.99964620e-01 6.89750587e-03 6.10438334e-04]
[ 3.95032977e-03 -6.91655256e-03 9.99968278e-01 -6.23648955e-03]
[ 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

When the transformation is made by hand the transform matrices with both method are almost identical but as we can see here when I try on two successive images the transform matrices are different. Here I use marching cubes.

# Progress

- Test mesh tracking in fusion:

Fuse 2 Images : Comparing F2F and MT2 in Fusion

Pose : MT2 fusion (Input : current image, former image made into lists)

```
[[ 0.99988264  0.00341173  0.01493695 -0.03305718]
 [-0.00334553  0.9999845   -0.00445503 -0.0044524 ]
 [-0.01495191  0.00440454  0.99987853  0.0011979 ]
 [ 0.           0.           0.           1.          ]]
```

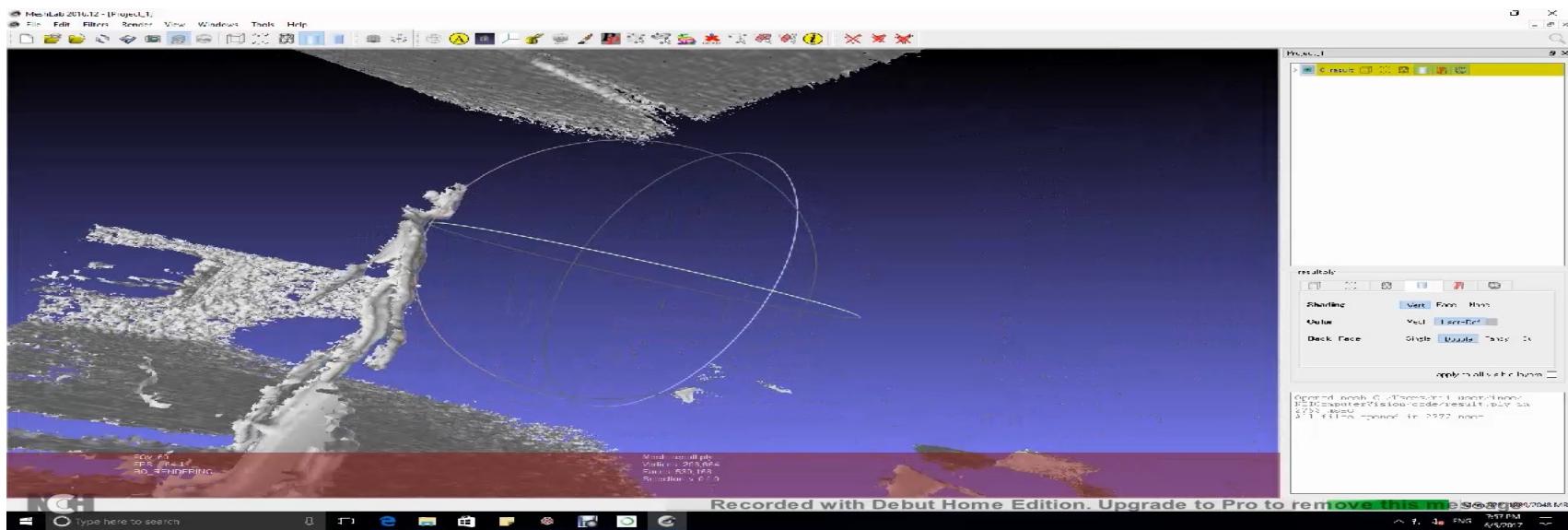
Two successive images with no marching cubes, the transform matrices are different

Pose : F2Ffusion (Input = current image and former image)

```
[[ 0.99986744  0.0033728   0.01592852 -0.00671804]
 [-0.00329238  0.99998172 -0.00507233 -0.0030414 ]
 [-0.01594533  0.00501921  0.99986027  0.00460481]
 [ 0.           0.           0.           1.          ]]
```

# Progress

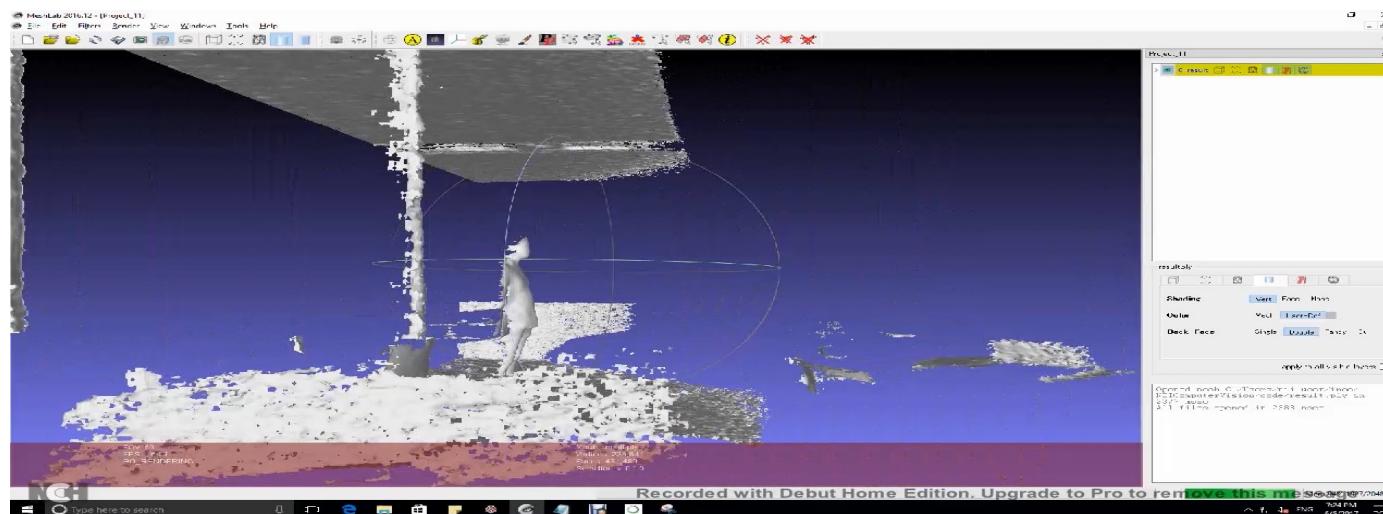
- Fusion with new mesh tracking:
  - Does work in test (image 15- 16) but not in fusion (Image 10-11, with 15-16 its works) → change threshold?



Multiply by 10 each threshold

# Progress

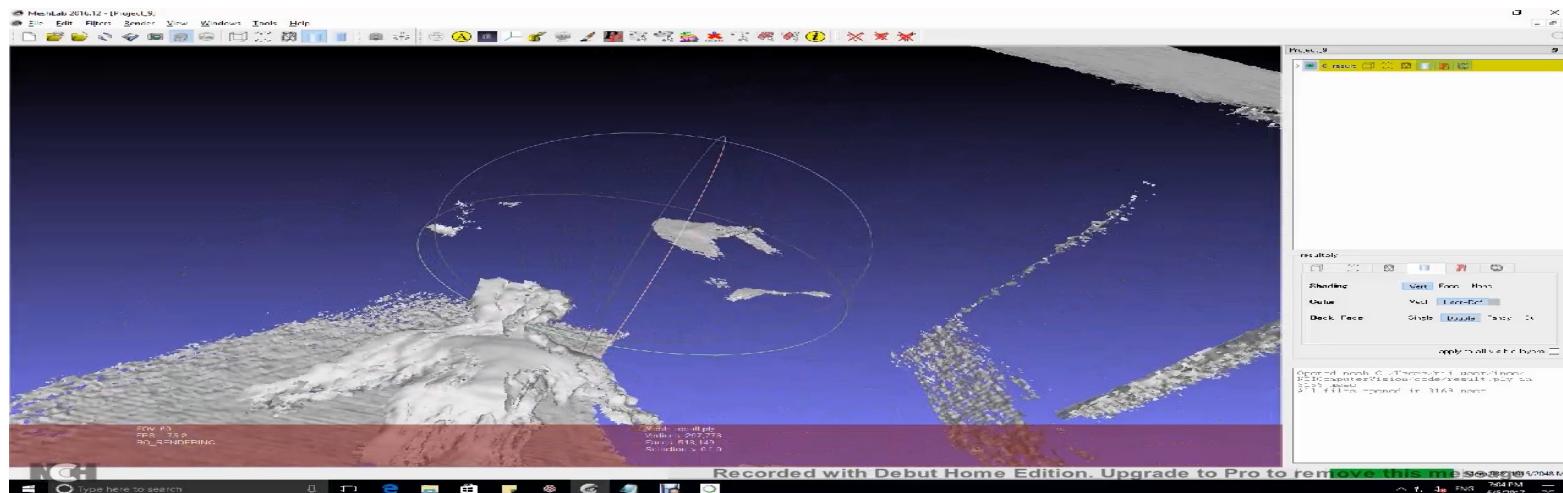
- Fusion with new mesh tracking:
  - Sequence 13-17 fusion OK with mesh track
  - Sequence 15-20 fusion not OK with mesh tracking



Sequence 13-17

# Progress

- Fusion with new mesh tracking:
  - The camera change direction in his movement in sequence 15-20



Sequence 15-20

# Progress

- Fusion with ray-casting:
  - Ray-casting function return an image of 0.
- Looking at values:
  - First some bad values but corrected after
- When bigger hand transform Jacobian = 0, comes from difference of Vertex.

# Progress

- Research papers:
  - Wei, T.-H., Lee, T.-Y. and Shen, H.-W. (2013), Evaluating Isosurfaces with Level-set-based Information Maps. Computer Graphics Forum, 32: 1–10. doi:10.1111/cgf.12087
  - Shungang Hua, Ang Xiong & Maodong Bai (2017) Matched part-pair blending for 3D shape creation, Computer-Aided Design and Applications, 14:2, 193-205, DOI: 10.1080/16864360.2016.1223431
  - Alhashim, Ibraheem. "Modeling and Correspondence of Topologically Complex 3D Shapes." arXiv preprint arXiv:1506.06855 (2015).

# Action plan

- Mesh tracking
- Papers research : find idea!
- Fusion for each segmented body part separately:
  - Coordinates change one by one
  - Fuse one by one

# Q&A

- Shall I do with Raytracing (Kinect Fusion)?
- How to choose threshold value?  
(grandeur of difference?)