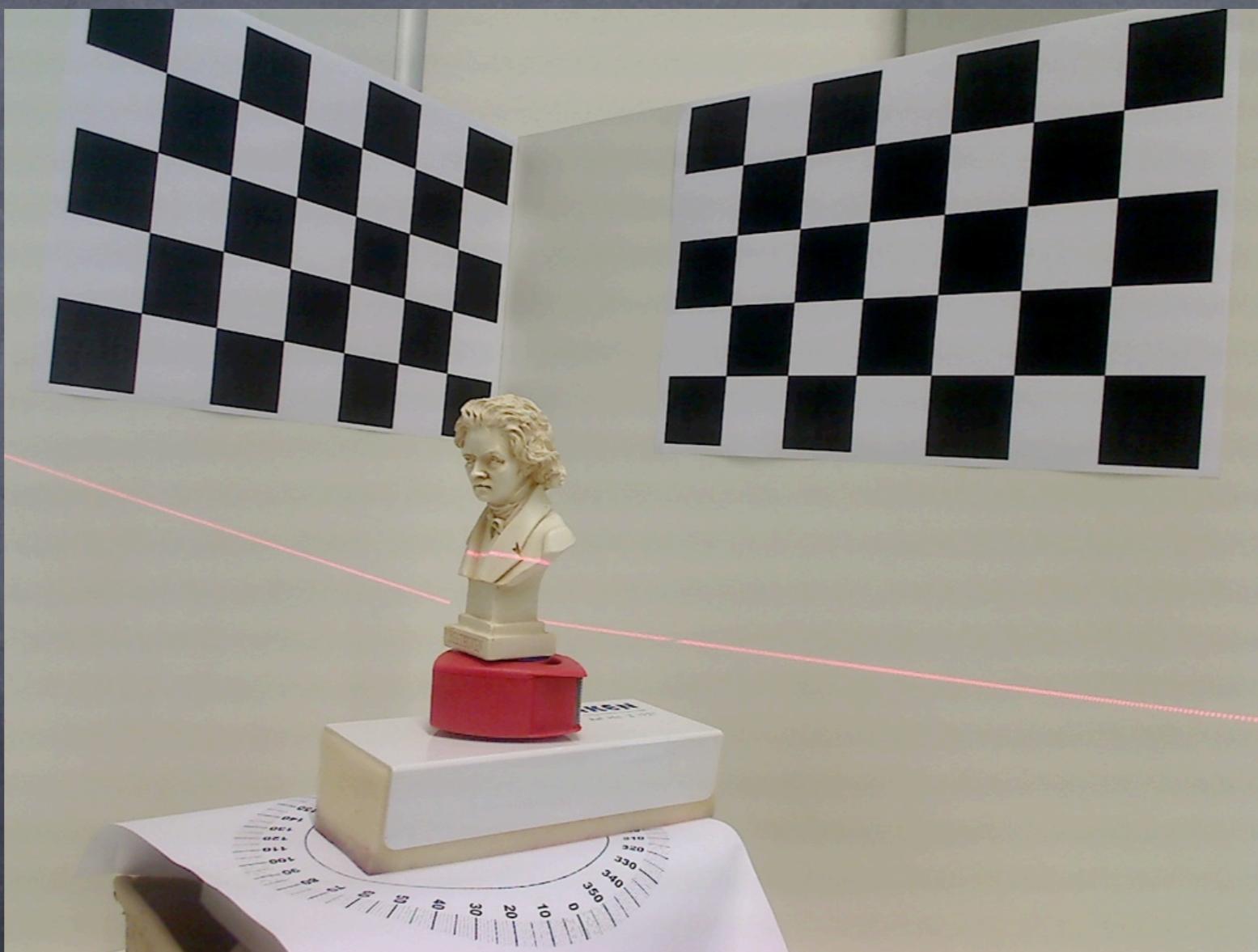


# 3D Laser Scanning



Vaibhav Bajpai

# Contents

- ⦿ Calibrate the Camera.
- ⦿ Search for the Laser Line (and object points)
- ⦿ Generate a Color 3D Point Cloud.
- ⦿ Register different scans using Slam6D/ICP.

# Calibrate the Camera

- ⦿ Find Camera' Intrinsic Parameters
- ⦿ Find Camera' Extrinsic Parameters

# Extrinsics?! Intrinsics?!

points in image plane

points in the world  
coordinate system

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

camera intrinsics

camera extrinsics

# Calibrate the Camera Intrinsic Parameters



# Calibrate the Camera Extrinsic Parameters



# Calibrate the Camera

how does the code look? :P

```
vector<CvMat*> cameraParameters = camera->calibrate(imageList);
```

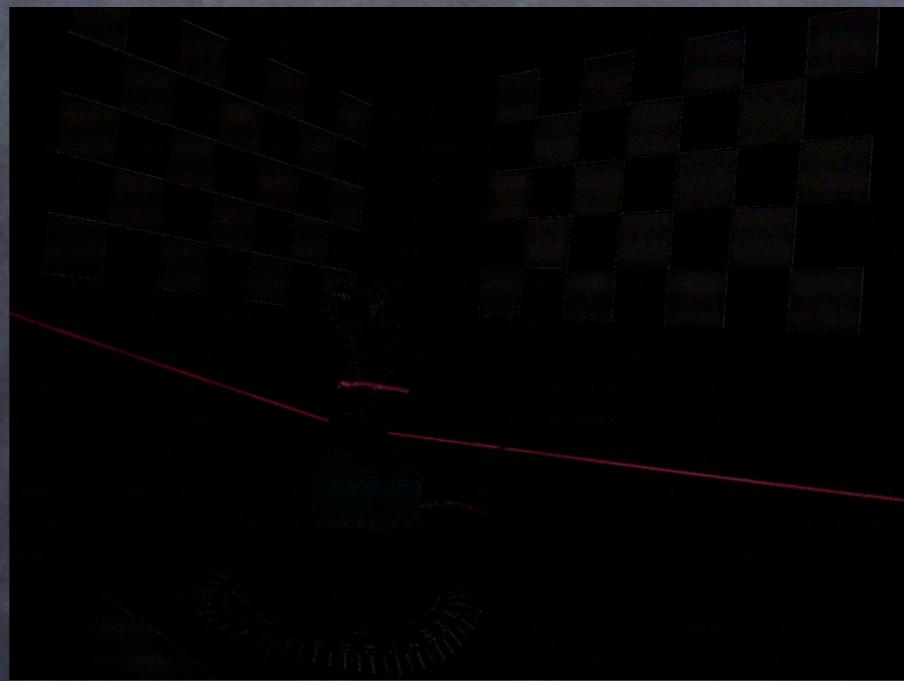
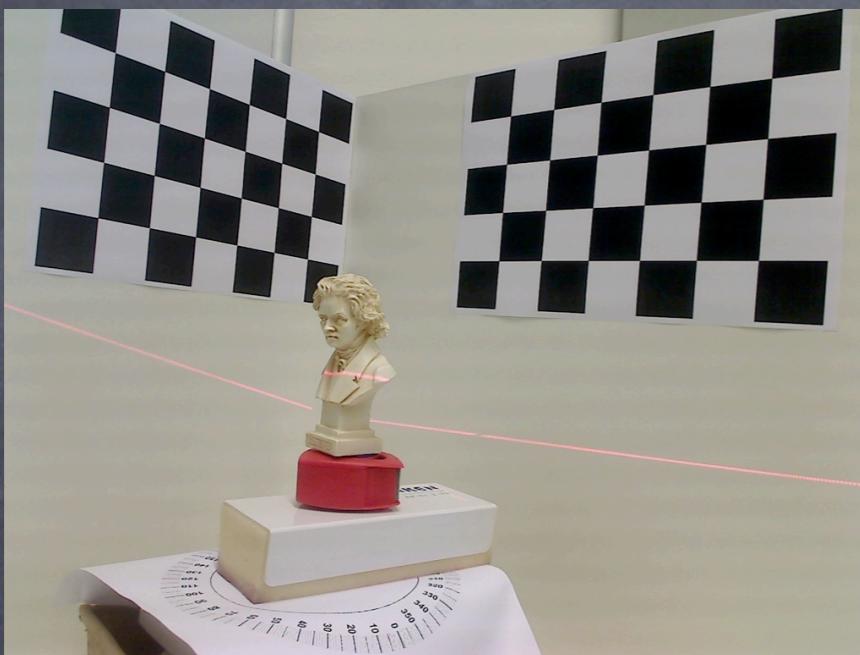
0	cameraMatrix
1	rotationVector (R1 and R2)
2	translationVector (T1 and T2)

# Search for Laser Line

- ⦿ Step1: Take a Difference Image to find the Laser.
- ⦿ Step2: Smoothen the Difference Image.
- ⦿ Step3: Color Threshold to remove everything else.
- ⦿ Step4: Apply Hough Transform.
- ⦿ Step5: Wrap the Points and Return the results.

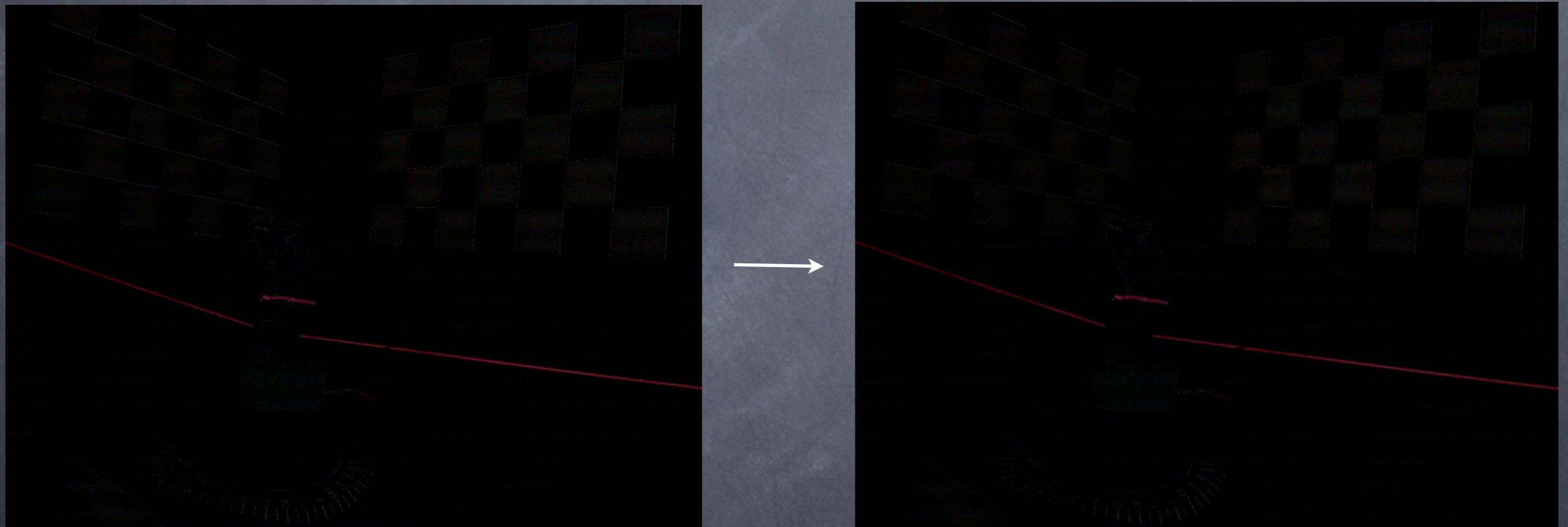
# Search for Laser Line

## Step1: Difference Image



# Search for Laser Line

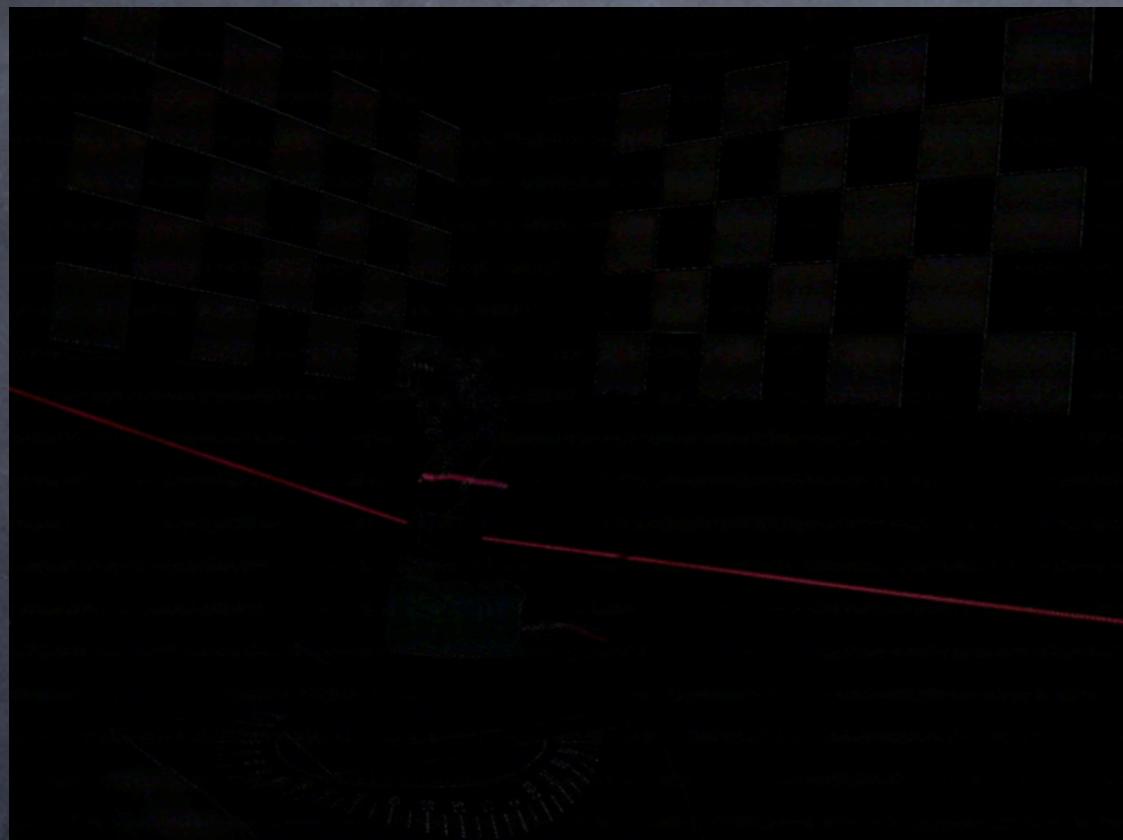
## Step2: Gaussian Smoothen



there is a difference; trust me!

# Search for Laser Line

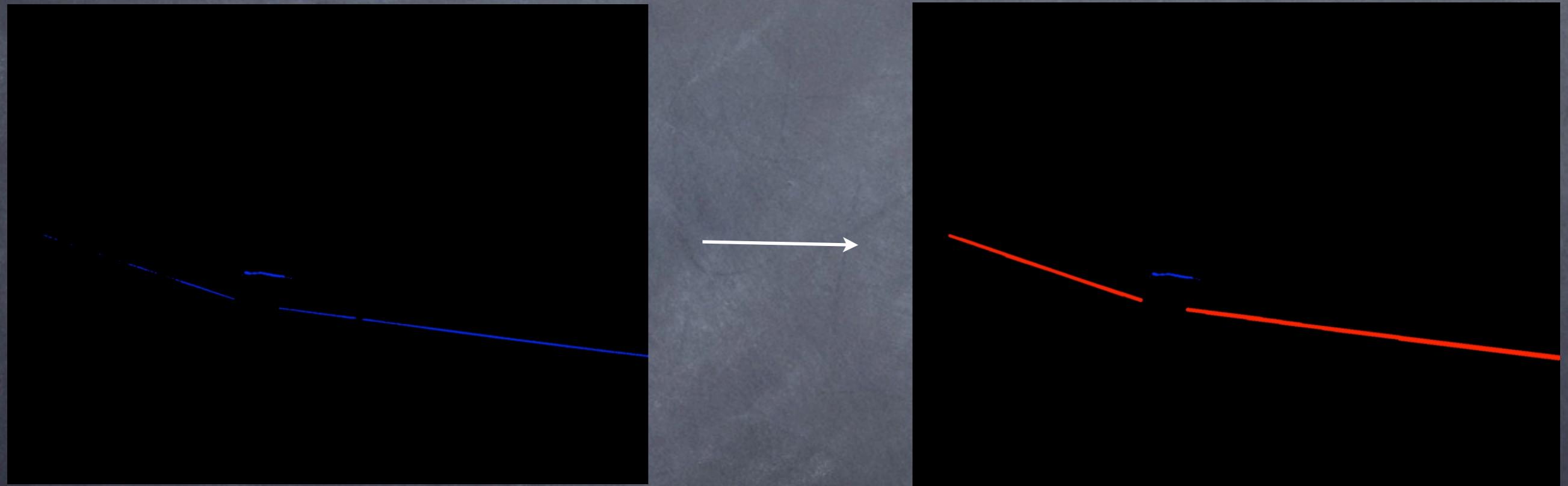
## Step3: Color Threshold



- \* split the image into RGB channels
- \* threshold along R channel.

# Search for Laser Line

## Step4: Hough Transform



minimum threshold for accumulator plane to report a line : 70

minimum length of the hough line: 700

separation between collinear points to NOT join them in a line: 600

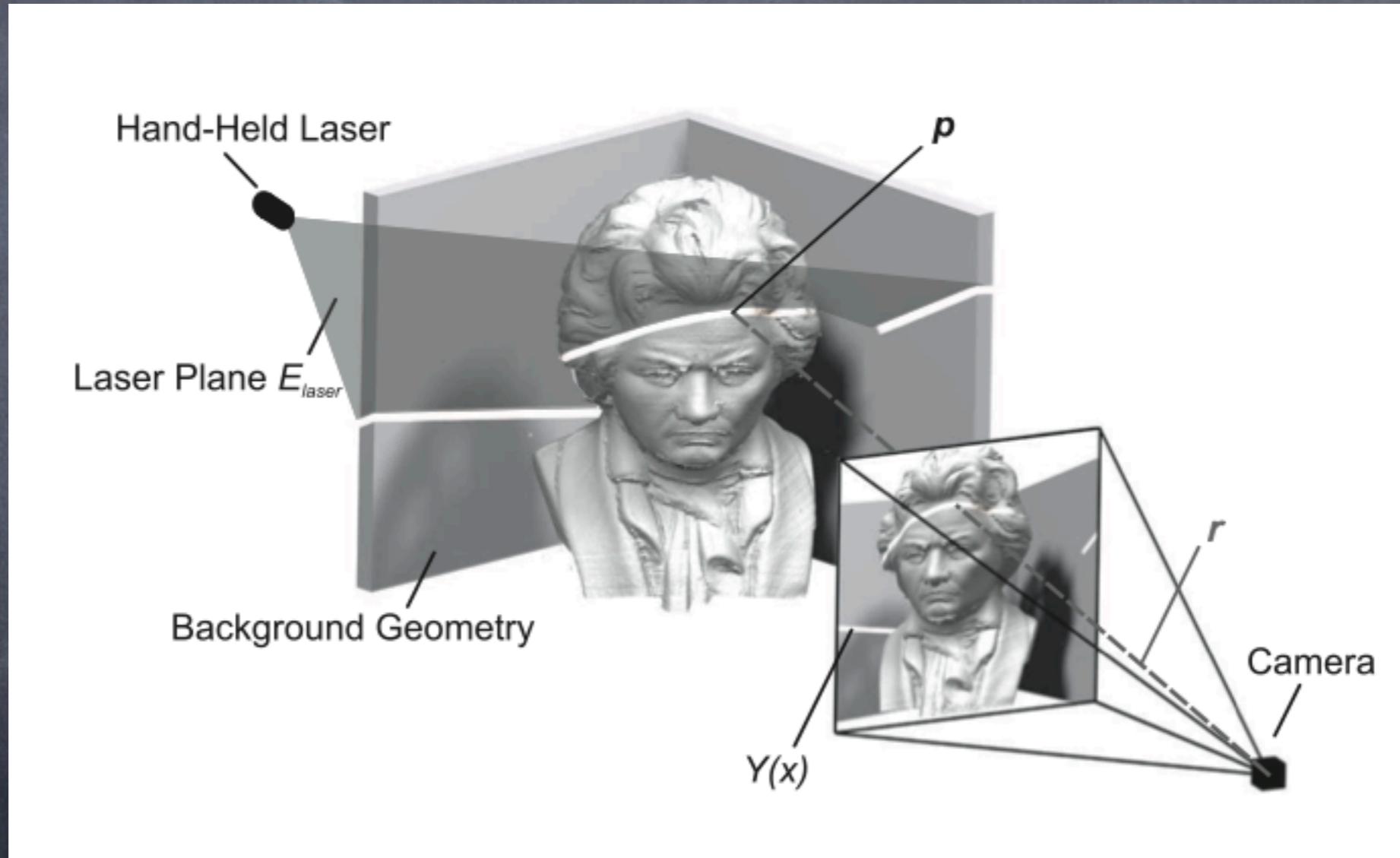
# Search for Laser Line

how does the code look? :P

```
vector < vector <CvPoint> > pointWrapper = scanner->findLaser(image);
```

0	Left Laser Points
1	Object Points
2	Right Laser Points

# Generate a Point Cloud



# Generate a Point Cloud

- Step1: Transform the right laser points to left coordinate system.
- Step2: Get the 3D laser points using camera extrinsics.
- Step3: Get the Laser Plane Equation using 3 Laser Points (2 from the left;1 from the right)
- Step4: Get the 3D Object Point by intersecting the Laser Plane with the Light Ray.
- Step5: Save the 3D Object Points with their color values using reference Image.

# Generate a Point Cloud

How does the code look? :P

```
vector <Point3DRGB*> pointCloudVector = pointCloud->generate(cameraParameters,  
pointWrapper);
```

Point3DRGB

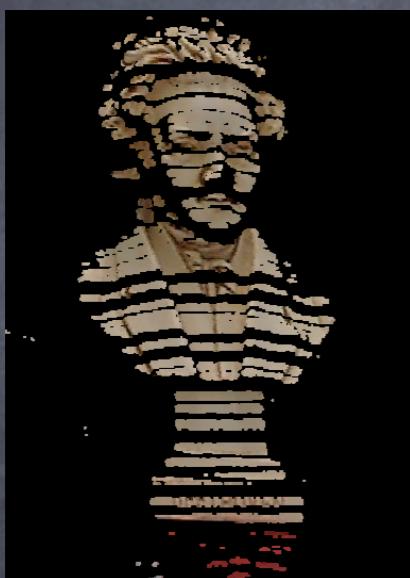
CvPoint3D32f

int RED

int GREEN

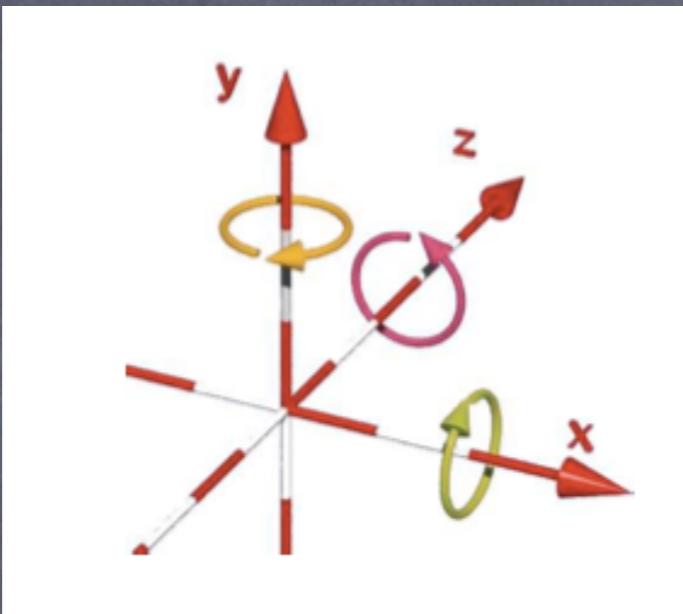
int BLUE

# Generate a Point Cloud Results?

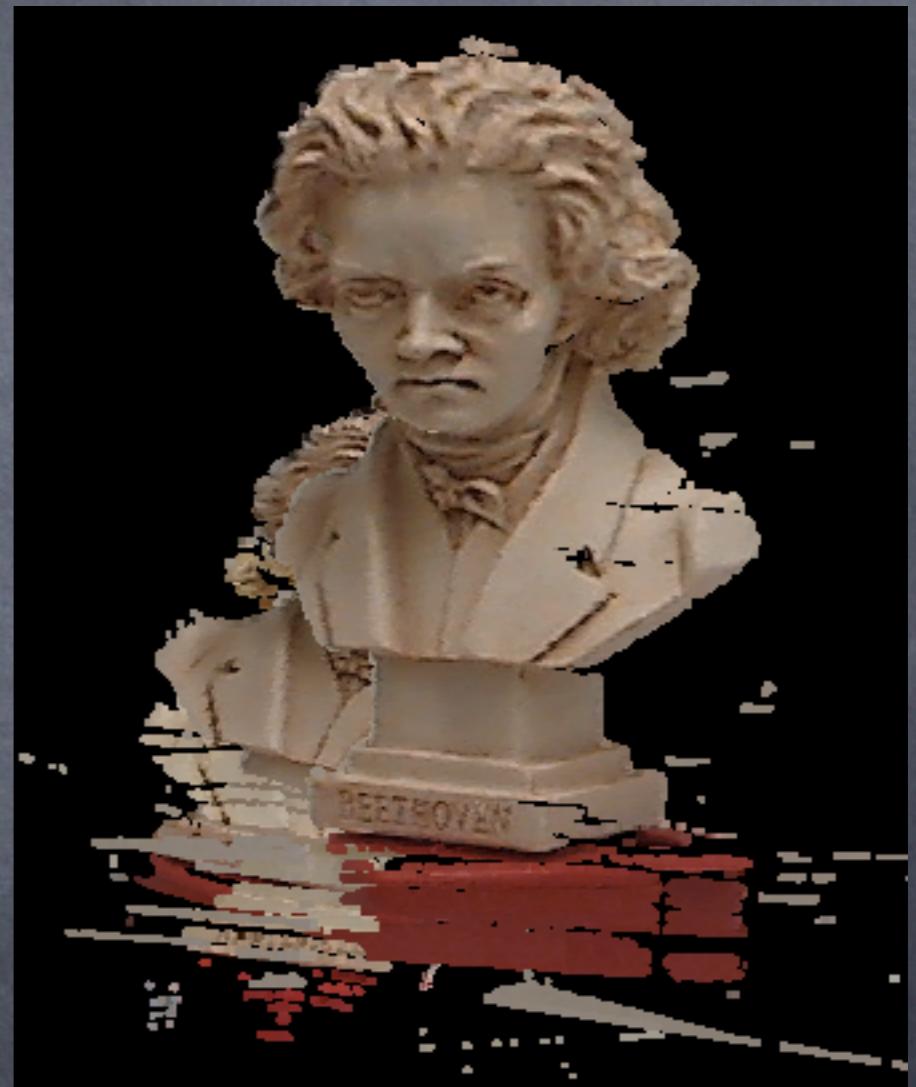


Register different scans using Slam6D/ICP

# Results?



made changes to the yaw in the pose files for each scan; to get this result



# Thank you!