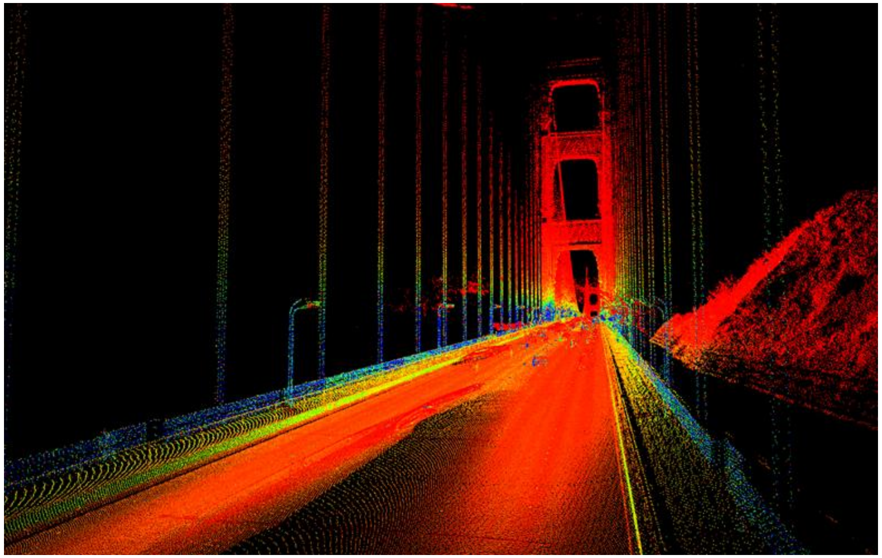
Analyzing Point Clouds
Bridge Inspection Project

S Krishna Savant

Introduction

- What?
- Why?
- How?



Motivation for the Project

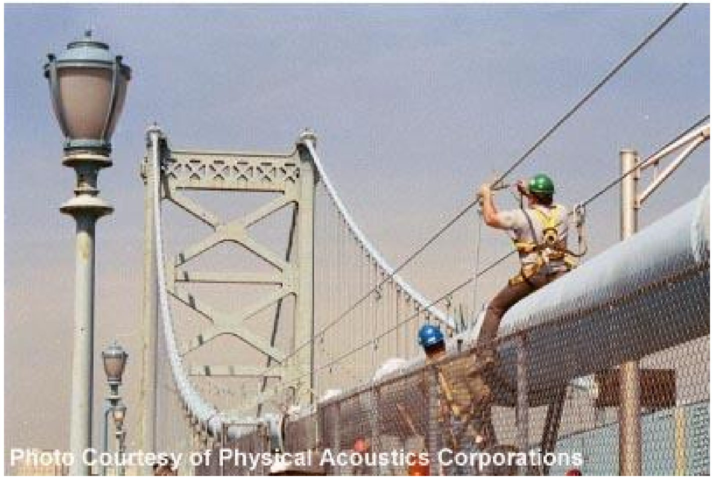
- Requirement of periodic inspection on a biennial basis
- Time-consuming manual inspection and insufficient manpower

State of the art technique?

Motivation for the Project

- Requirement of periodic inspection on a biennial basis
- Time-consuming manual inspection and insufficient manpower
- Inefficient data collection
- Paper-based storage

State of the art technique?



Goals of the project

- Automated collection of data
- Better visualization of the bridge
- Reliable storage of information over time

The big picture

- UAV scans bridge
  - Laser Range Scanner
  - GPS
  - Cameras
- Range Scans to Point Clouds
- Registering different Point Clouds
- Analysis of point clouds
  - Error evaluation
  - Coverage analysis
- Visualization of Point Clouds
  - Ability to look around the point cloud data
  - View images from a particular viewpoint

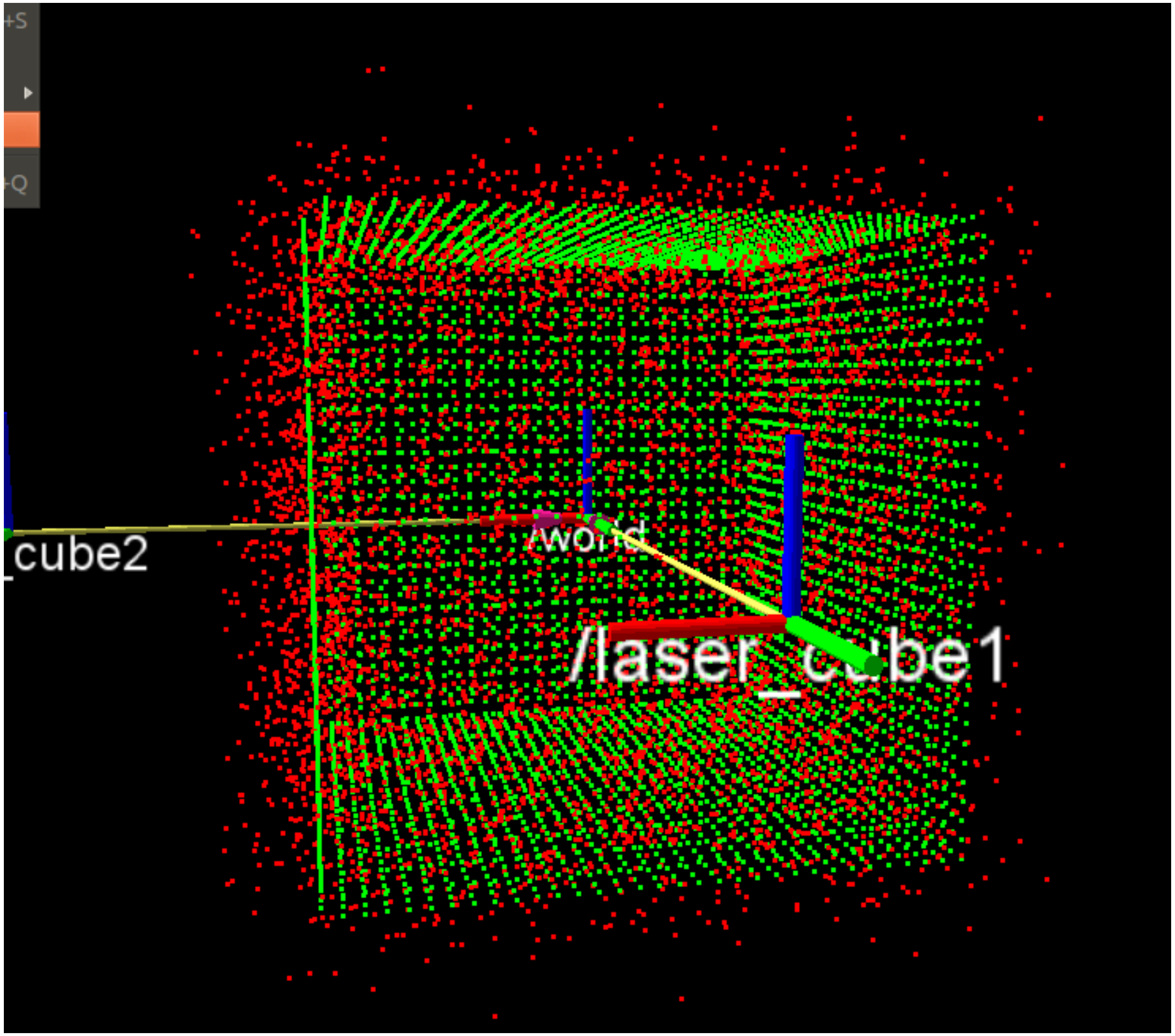
Analyzing Point Clouds :
Error evaluation

- Problem:
  - Given two point clouds representing same object define a **metric** which quantifies the similarity.
- We call this metric **error**
- Error is a function of the distance between the nearest neighbour correspondences
  - Lower the error, higher the chance of the point clouds being similar.

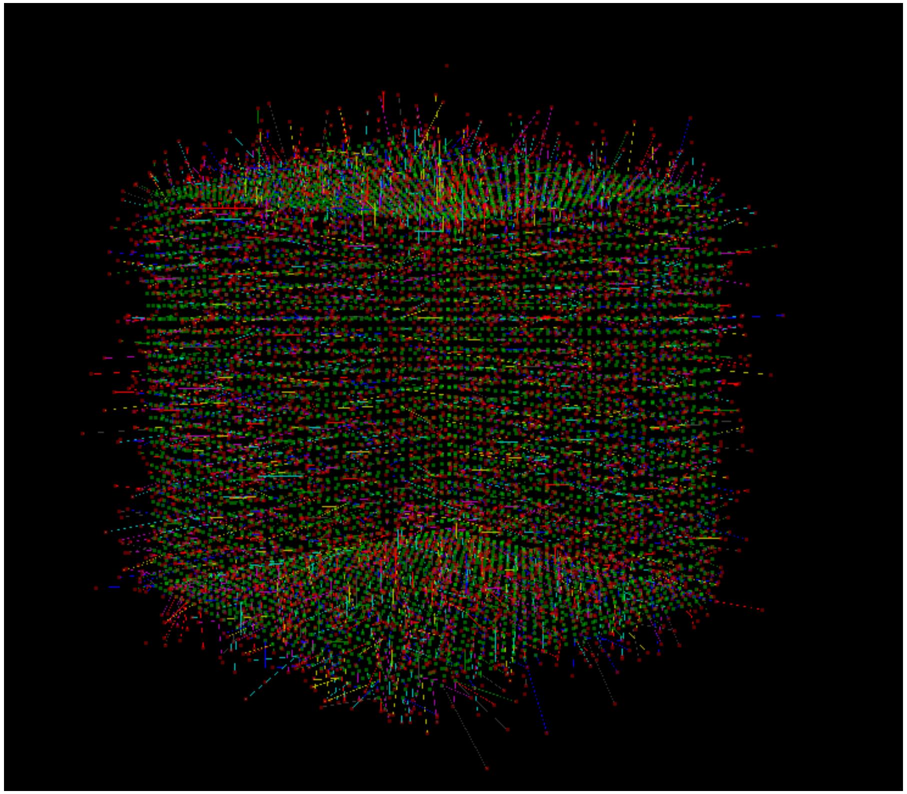
Analyzing Point Clouds :
Error evaluation

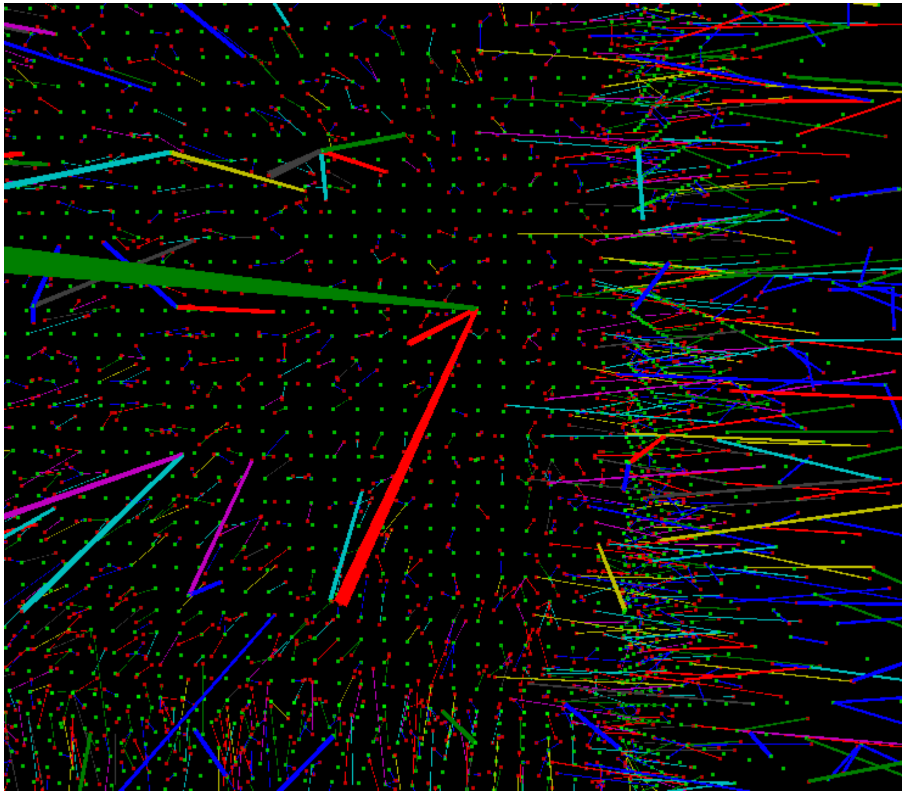
- Naive approach :
  - Assuming registered point clouds, find ***average euclidean distance*** from the nearest neighbor
- Issues?
  - If there are many uncovered points, leads to increase in the error metric
  - Maybe the point clouds are not registered correctly.
  - Prone to outlier noise

Analyzing Point Clouds :
Error evaluation (Naive approach)



Analyzing Point Clouds :
Error evaluation (Naive approach)





Analyzing Point Clouds

- Point clouds not aligned?
- Need to register them

- Problem:
  - Given two point clouds align them
  - Find the *best* possible transformation from one point cloud to another.
    - What does *best* mean?
    - Best in terms of minimal error between the point clouds

Registering Point Clouds

- Definition?
  Combining several point cloud datasets into a global consistent model

- How is it done?
  - Keypoints
  - Feature descriptors
  - Feature matching : Correspondences

Keypoints

- What are keypoints?
  - Also called interest points, they are stable, distinctive points which can be identified using well-defined detection criterion

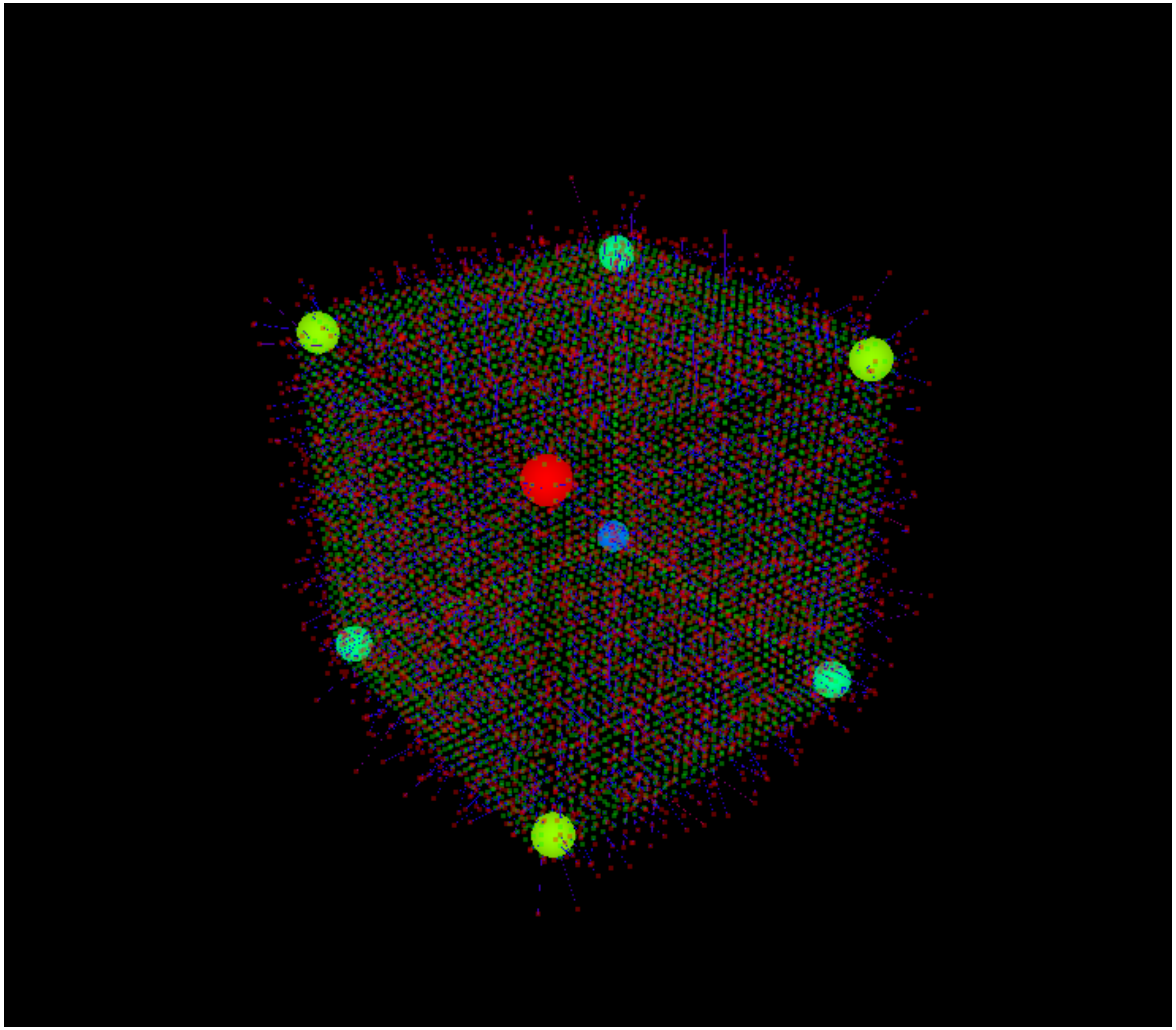
- Why keypoints?
  - Not feasible to find matches for all the points
  - Sparseness : Much smaller in number
  - Using all points to find correspondences is error-prone
  - An efficient way to reasonably represent huge point clouds

Keypoints :
Harris Keypoints

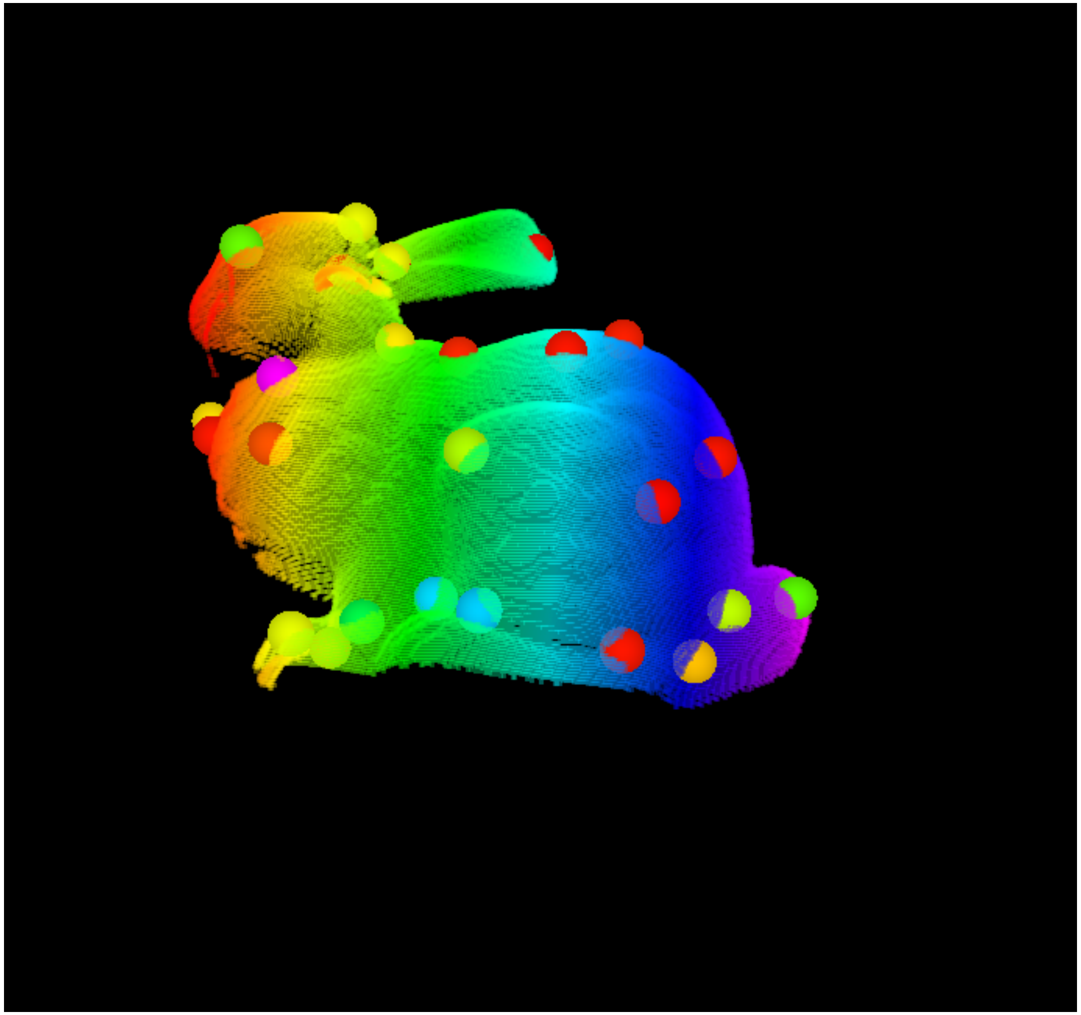
- Examples of Keypoints
  - NARF Keypoints (Normal Aligned Radial Feature)
  - SIFT Keypoints (Scale Invariant Feature Transform)
  - Harris 3D Keypoints

- Harris Keypoints
  - Corner detection

Harris Keypoints : Examples
Cube Point Cloud

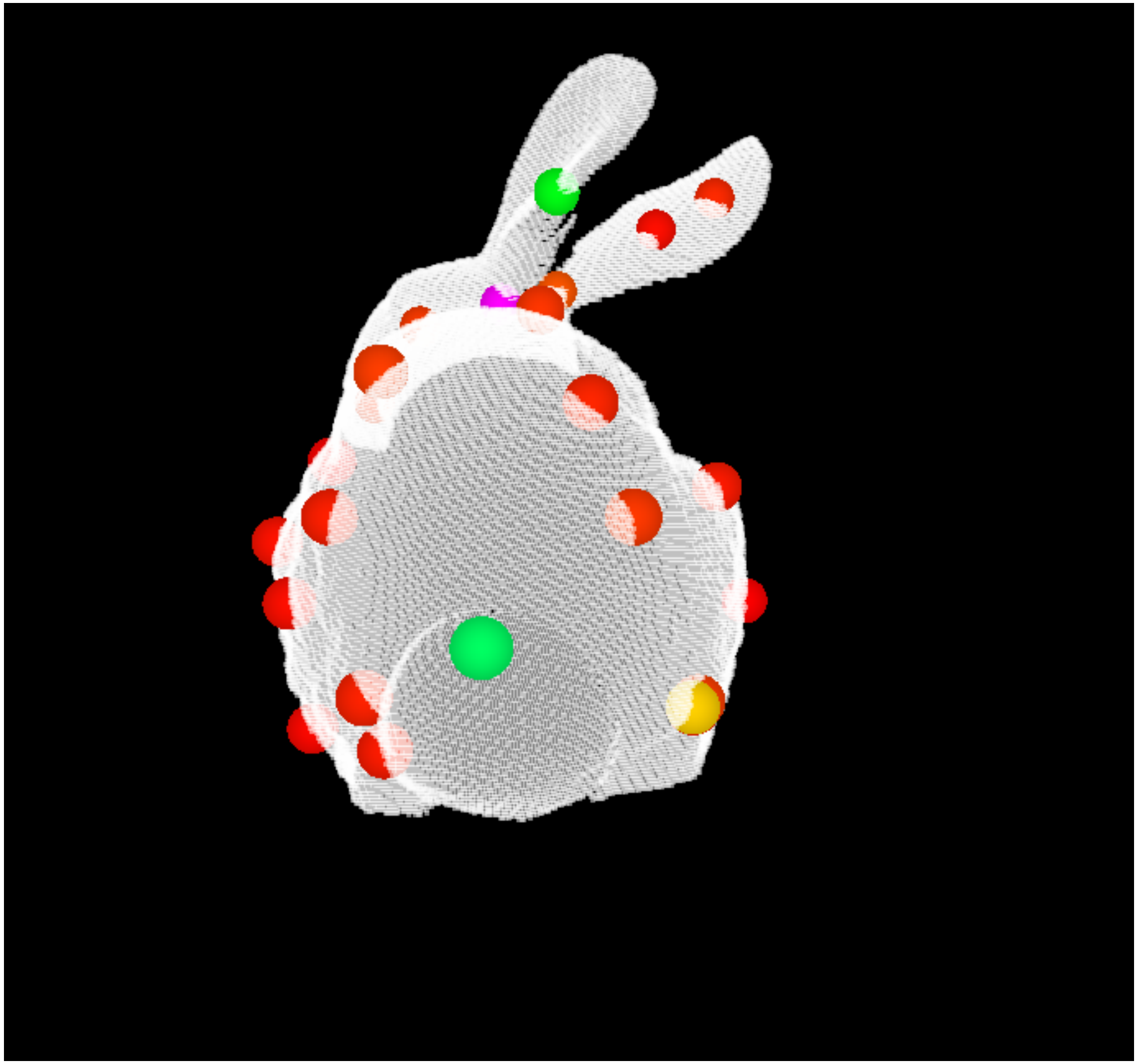


Harris Keypoints : Examples
Stanford Bunny (View 1)



Harris Keypoints : Examples

Stanford Bunny (View 2)



Feature Descriptors

- Features :
  Representations at a certain 3D point or position in space, which describe geometrical patterns based on the information available around the point

- Examples
  - SHOT (Signature of Histogram of Orientations)
  - ESFE (Ensemble of Shape Functions)
  - RIFT (Rotation Invariant Feature Transform)
  - PFH (Point Feature Histogram)

FPFH (Fast Point Feature Histogram)

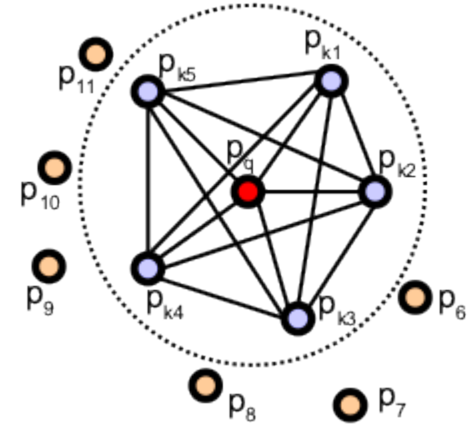
- *Step 1* :
  For each query point p_{query}, a set of t α, ϕ, θ between itself and its neighbors are computed - this will be called the Simplified Point Feature Histogram (SPFH);
- *Step 2* :
  For each point, its k neighbors are re-determined, and the neighboring SPFH values are used to weight the final histogram of p_{query} (FPFH)

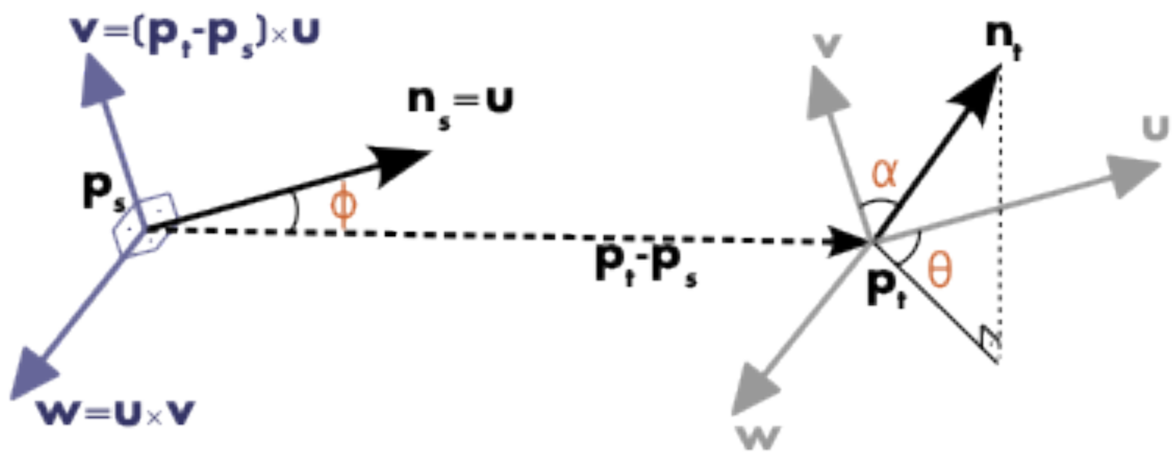
FPFH (Fast Point Feature Histogram)
Step 1 : SPFH

$$\alpha = \mathsf{v} \cdot \boldsymbol{n}_t$$

$$\phi = \mathsf{u} \cdot \frac{(\boldsymbol{p}_t - \boldsymbol{p}_s)}{d}$$

$$\theta = \arctan(\mathsf{w} \cdot \boldsymbol{n}_t, \mathsf{u} \cdot \boldsymbol{n}_t)$$





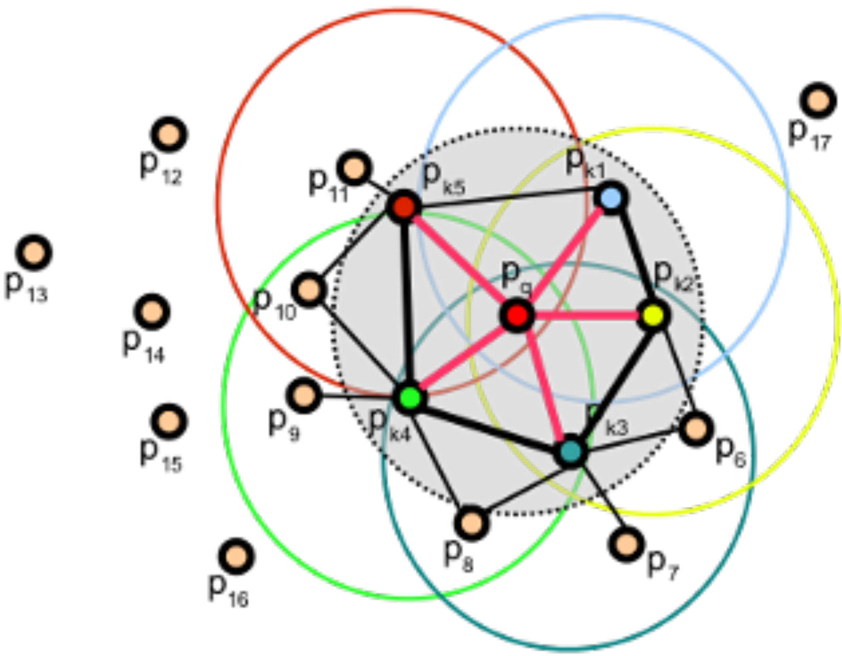
$$\mathsf{u} = \boldsymbol{n}_s$$

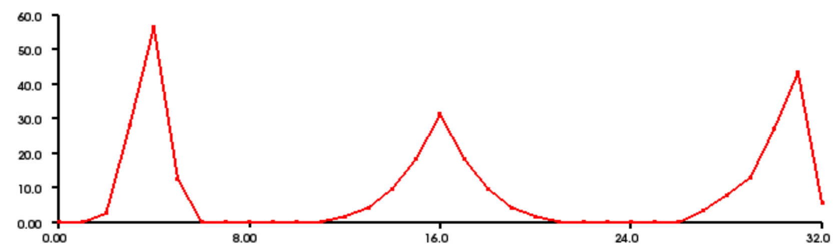
$$\mathsf{v} = \mathsf{u} \times \frac{(\boldsymbol{p}_t - \boldsymbol{p}_s)}{\|\boldsymbol{p}_t - \boldsymbol{p}_s\|_2}$$

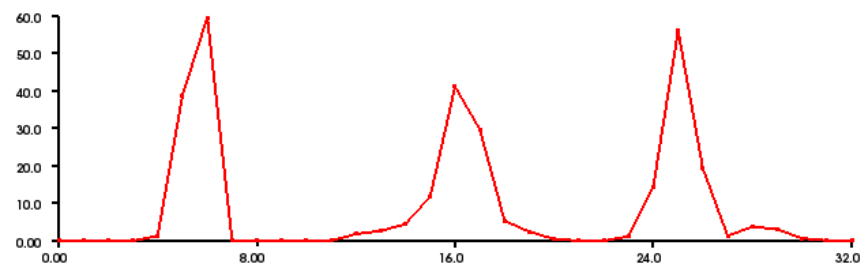
$$\mathsf{w} = \mathsf{u} \times \mathsf{v}$$

FPFH (Fast Point Feature Histogram)
Step 2

$$FPFH(\boldsymbol{p}_q) = SPFH(\boldsymbol{p}_q) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k}\cdot SPFH(\boldsymbol{p}_k)$$







How fast is FPFH?

- Theoretical complexity :
  - PFH $O(nk^2)$
  - FPFH $O(nk)$
- Setup:
  - ~40000 points per cloud
  - Finding FPFH description at harris keypoints ~25 in each cloud
  - Using OpenMP (shared memory parallel processing): 8 cores of ~2.33 GHz each
- Takes ~ 1 min
  - Isn't fast enough for real time processing

Feature Matching

- Match features/keypoints from one point cloud with those from other

- Examples
    - SAC (Sample and Consensus)
    - ICP (Iterative Closest Point)

RANSAC : General Algorithm

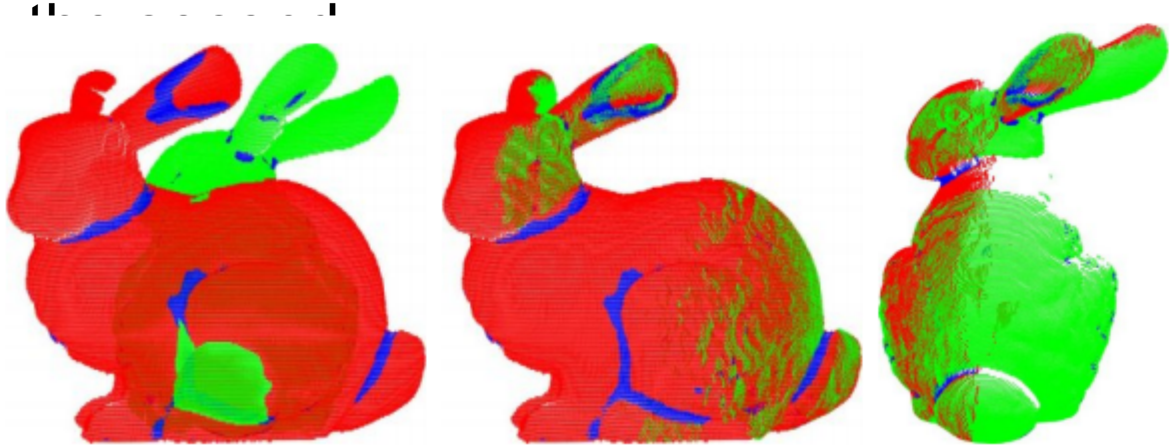
- Select randomly the minimum number of points required to determine the model parameters.
- Solve for the parameters of the model.
- Determine how many points from the set of all points fit with a predefined tolerance .

RANSAC : General Algorithm

- If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold τ , re-estimate the model parameters using all the identified inliers and terminate.
- Otherwise, repeat steps 1 through 4 (maximum of N? times)

SAC-IA (Sample and Consensus - Initial Alignment)

- Metric to choose the best alignment of the randomly chosen alignments?
    - Minimum Error in the point clouds
    - Minimum difference in the feature space
    - NN distance ratio : Ratio of the first nearest match to the second



ICP (Iterative Closest Point)

- SAC-IA gives a crude transformation
- Use that as initial guess for ICP to finetune the registration
- A form of gradient descent algorithm
- Need a relatively good starting point so as not to get stuck in the first local minima

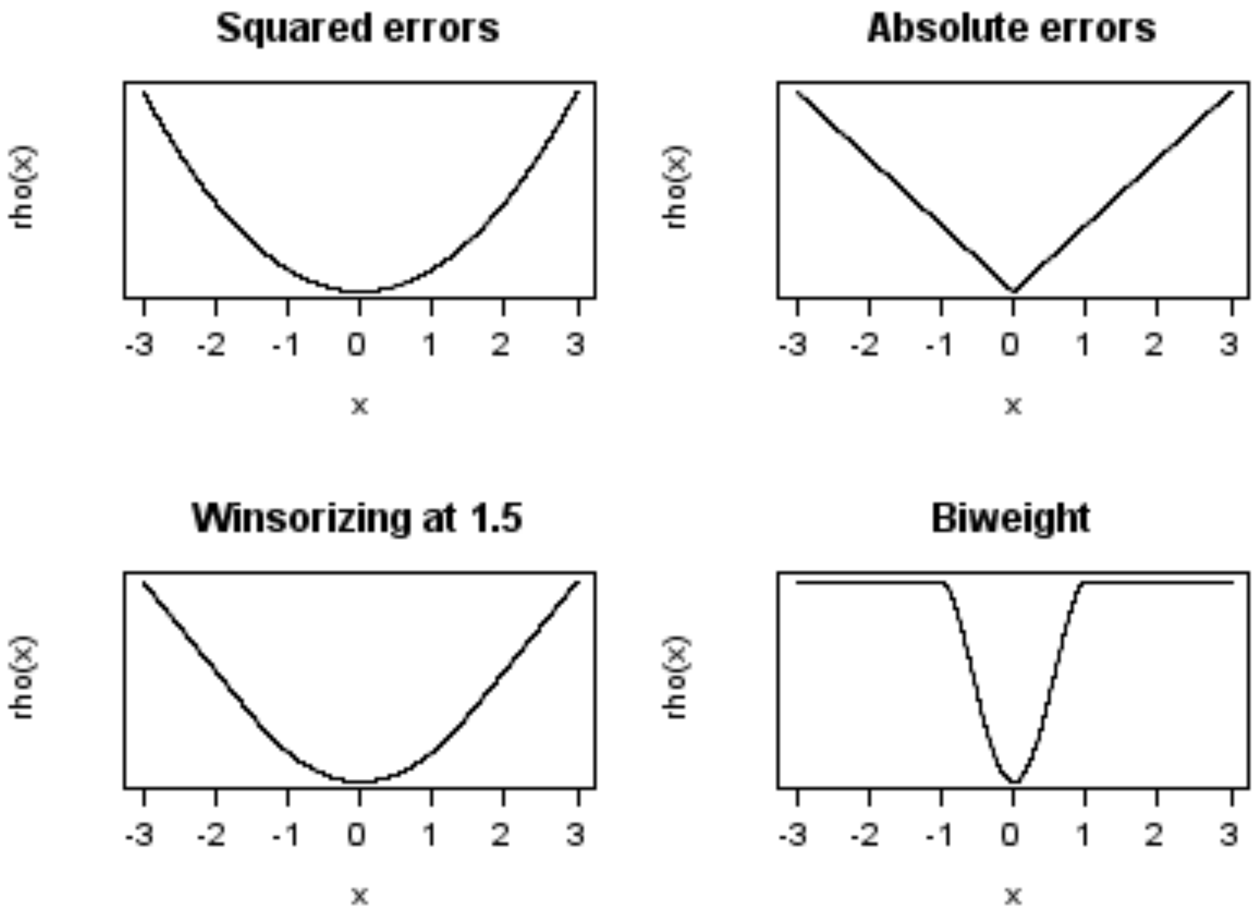
Error metrics

- Euclidean (Squared) distance
  - Suitable for gaussian error distribution
- Manhattan (Absolute) distance
- Huber metric

$$
\begin{aligned}
L_\delta(a) &= (1/2)a^2 && \text{for } |a| \le \delta, \\
L_\delta(a) &= \delta(|a| - \delta/2), && \text{otherwise.}
\end{aligned}
$$

  - Robust measure to outliers

Error metrics



Coverage of Point Clouds

- Coverage is percentage of the model that is 'covered' by the point cloud.

- Regions of point clouds may not have correspondences
- Interfere with the computation of error metrics

Coverage

- Basic approach :
  - Percentage of points which have correspondences within some maximum distance

- Issues?
  - Non uniform distribution of points
  - Defining the maximum correspondence distance

Coverage

- Different approaches :
  - Divide surface into cubes and assign area to each point in the cube inversely proportional to the density of points in that region
  - Uniformly sample the point cloud with limit on maximum density and find the percentage of points

- A complicated approach:
  - Surface reconstruction from point cloud using Voronoi diagram and Delaunay triangulation

Future work

- Implementing different coverage approaches
- Using a synthetic bridge model and simulating coverage
- Analyzing actual bridge data for error analysis and coverage

Thank You

EOP!