

Robot Programming

ICP (exercise)

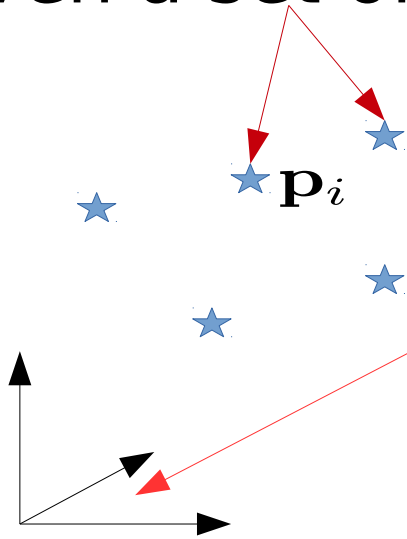
G. Grisetti

`grisetti}@diag.uniroma1.it`

Department of Computer, Control, and Management Engineering
Sapienza University of Rome

Example ICP Optimization in 3D

Given a set of points in the world frame



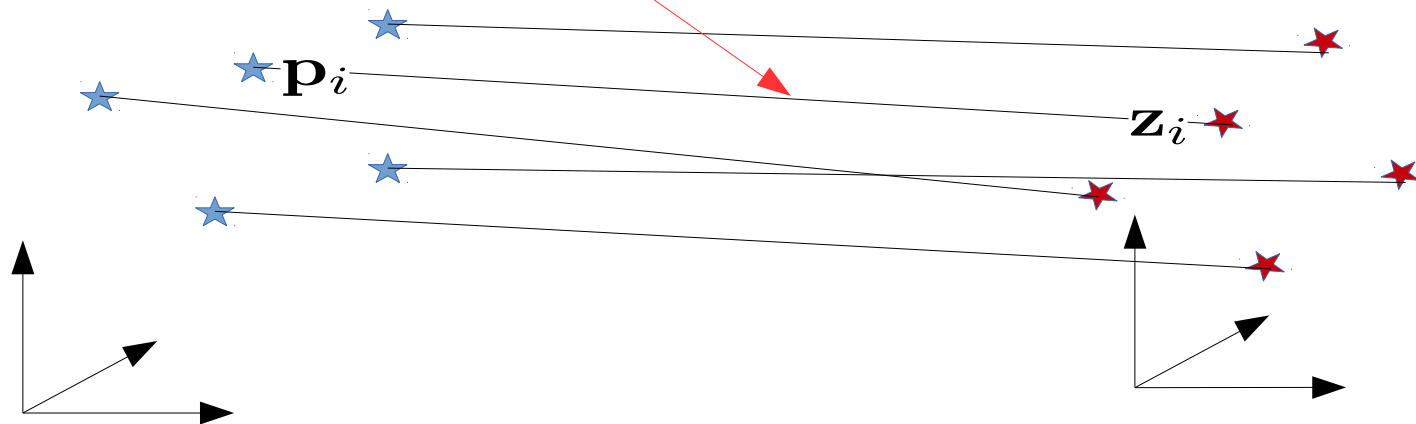
Example ICP Optimization in 3D

A set of 3D measurements in the robot frame



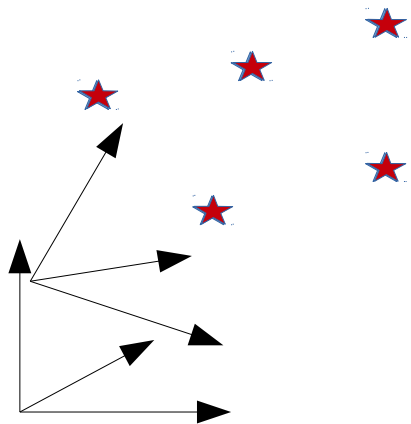
Example ICP Optimization in 3D

Roughly known correspondences



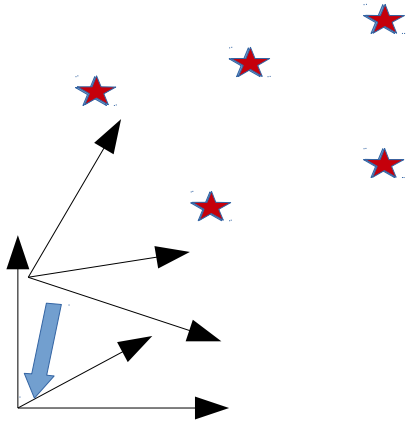
Example ICP Optimization in 3D

We want to find a transform that minimizes distance between corresponding points

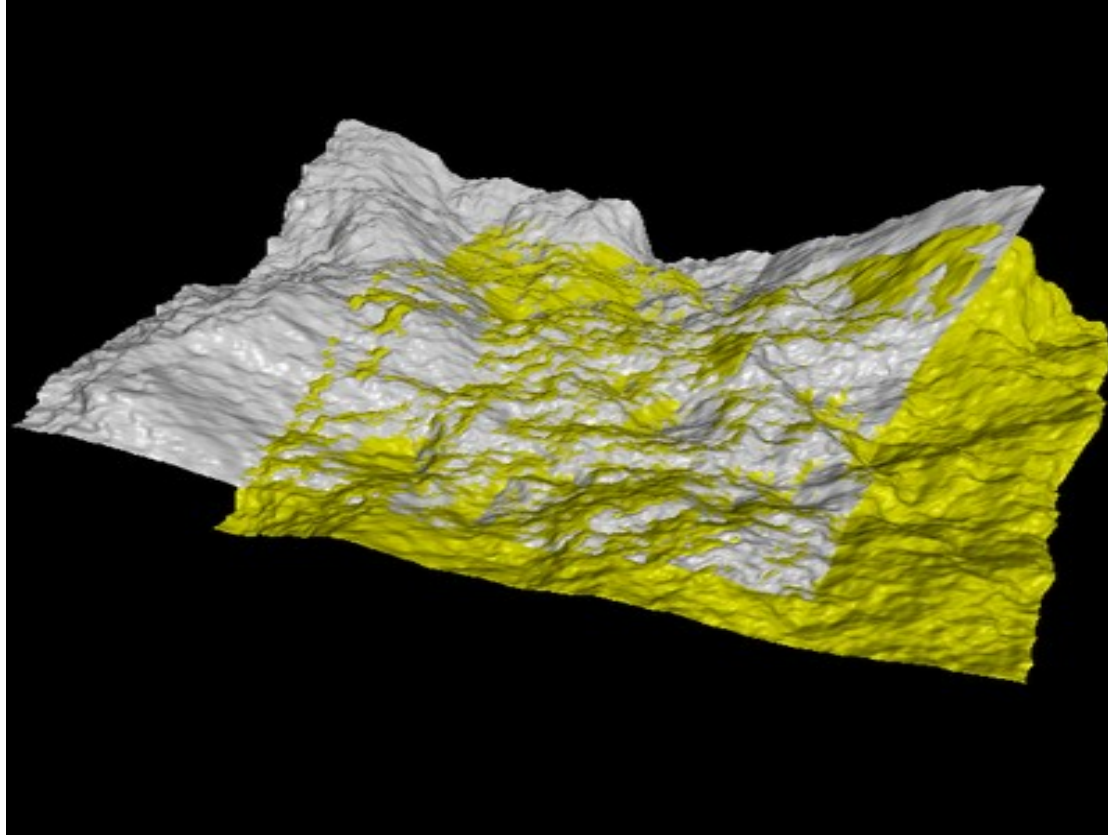


Example ICP Optimization in 3D

Such a transform will be the pose of world w.r.t. robot



Example



ICP: loop

Inputs:

- **X**: initial position of “moving” w.r.t reference
- $\{\mathbf{m}_i\}$: points in the moving set
- $\{\mathbf{f}_i\}$: points in the fixed set

Until convergence

- Determine pairs $\{<\mathbf{m}_i, \mathbf{f}_j>\}$ based on some similarity criterion, under the current transform **X**
- Calculate a new **X**, as the transform that applied to \mathbf{m}_i minimizes the distance between corresponding pairs

Our Naive Implementation

Optimization: given by me, but if you need to know more, follow Probabilistic Robotics

Correspondence Heuristic:

- For each \mathbf{m}_j , compute its position $\mathbf{m}'_j = \mathbf{X} * \mathbf{m}_j$, under the current transformation and look for the \mathbf{f}_j closest to \mathbf{m}'_j within a certain radius. If nothing found, no correspondence formed, otherwise return the pair $\langle \mathbf{f}_j, \mathbf{m}'_j \rangle$.

Your Tasks (1)

Complete the functions

```
Vector3f* bestMatchFast(const Vector3f& q,  
                        const float norm);  
Vector3f* bestMatchFull(const Vector3f& q,  
                        const float norm);
```

In the `kd_tree.h`. These functions have to return the point in the tree closest to **q**, within a distance of **norm**. If nothing found within norm, 0 is returned.

The first performs an approximated search,

The second a full search.

Validate the results with `kd_tree_test.h`

Your Tasks (2)

The class `ICP` contains an implementation of ICP. It initializes a tree on the fixed cloud.

In `x` the current transform is stored.

Complete the method `ICP::computeCorrespndences`, that fills the `_correspondences` vector with the pairs according to the heuristic of two slides ago.

Verify the result running `icp_test`.