Aitor Aldoma and Markus Vincze

Pose Alignment for 3D Models and Partial View Point Clouds Based on Stable Planes
Motivation

- Efficient method to align 3D models and 2.5D/3D.

3D

(R,t,s)
Motivation

- Efficient method to align 3D models and 2.5D/3D.

Partial views

Segmentation (plane extraction) → Object retrieval → \((R, t, s)\)
Motivation

- Efficient method to align 3D models and 2.5D/3D.

Easier for a robot to manipulate the object if 3D mesh available and aligned with the real object.

Partial views

Segmentation (plane extraction) → Object retrieval → (R, t, s)
Related work

- **3D alignment:**
  - **CPCA**, global optimization methods based on object symmetries
  - ... not usable for 2.5D alignment

- **2.5D alignment:**
  - **ICP**: needs a good initial approximation
  - **Object recognition + pose estimation**: usually fail when training data source is different than the detection data source
    - VFH, CAP descriptors
  - **Local features based methods**:
    - hard to find features that work across different type of objects
    - globally similar objects might not have enough local features
    - have troubles dealing with noise
Stable planes

Why?:

- Similar objects share stable planes that give a partial alignment.
- Segmentation prior: we can use the dominant plane extracted in the segmentation step as stable plane for the partial views.
- Efficiency reasons:
  - From 7DoF to $N \times 4$DoF.
Stable planes

Why?:
- Similar objects share stable planes that give a partial alignment.
- Segmentation prior: we can use the dominant plane extracted in the segmentation step as the stable plane for the partial views.
- Efficiency reasons:
  - From 7DoF to $N \times 4$DoF.

![Diagram of stable planes and objects]
Stable planes

Why?:
- Similar objects share stable planes that give a partial alignment.
- Segmentation prior: we can use the dominant plane extracted in the segmentation step as stable plane for the partial views.
- Efficiency reasons:
  - From 7DoF to \( N \times 4\text{DoF} \).
Alignment overview

Stable planes

Select planes hypothesis

Place models on the planes and align them

Project points on the plane

Align 2D (rotation, scale, dx, dy)

Evaluate alignment
Alignment overview

Stable planes

Select planes hypothesis

Next hypothesis...

Place models on the planes and align them

Project points on the plane

Align 2D (rotation, scale, dx, dy)

Evaluate alignment

Indepedent from each other. Runs in parallel.

Do we need to align this hypothesis?
2D alignment (4DoF)

Projections
2D alignment (4DoF)

Projections

dx, dy

...
2D alignment (4DoF)

- Why scale and translation?
  - Object recognition might be scale invariant.
  - CoM of a partial view is translated from model CoM.

![Projections diagram](image)

**Dx*Dy**

Cross-correlations in log-polar space

Done in parallel on the GPU
Different alignment hypotheses.

Which is the final alignment?

We need a similarity metric:
- Number of inliers: count the number of points in target that have a neighbor in the query within a certain distance.
- Euclidean distance between target and query.
- Projection on XY, XZ, YZ plane and 2D-shape metrics.
- ...any pose and scale dependent metric is valid.

Number of inliers!
3D alignment examples
Stereo & Kinect Data

Stereo

Kinect
Conclusions

- Pose alignment for 3D models and partial views.
- Efficiently computes 7-DOF transformations by taking advantage of parallelization (multi-threaded & GPU) and the stable plane prior.
- Works among different classes of objects.

Thanks!