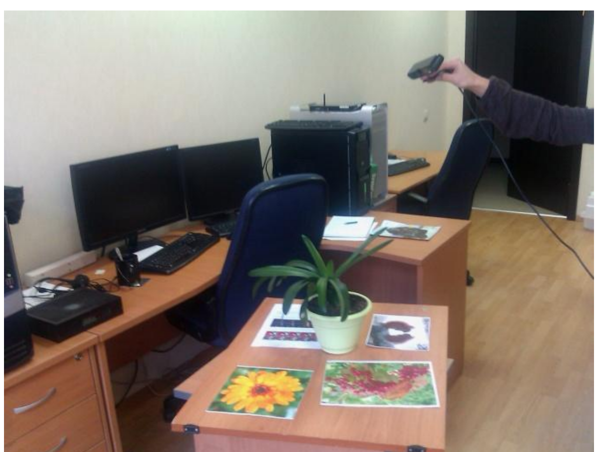
Tabletop Object Scanning with an RGB-D Sensor

Maria Dimashova, Ilya Lysenkov,
Vincent Rabaud and Victor Eruhimov

The goal

- To create a complete pipeline which scans accurate 3D models from RGB-D data and can be carried out both by a person and a robot



Motivation

- General reasons:
  - 3D models are required for robotics and perception tasks
  - accuracy and ease of use is important
- Particular reason:
  - Previous capture setups of Willow Garage:
    - table with visual markers
    - table with an arbitrary texture but there is the training stage
  - Reduce restrictions on the use

Scanning setup

- Two ways:
  - in-hand scanning
  - tabletop
- In-hand scanning does not meet the goal
  - problems with object segmentation and mis-registration of symmetric objects

Scanning setup

- Object lying on a flat surface (table)
  - Automatic object segmentation
- Texture on the table
  - RGB-D odometry works in case of dominant plane
- One turn around the table
  - a drift compensation
  - it is more reliable to determine a simple structure of poses graph

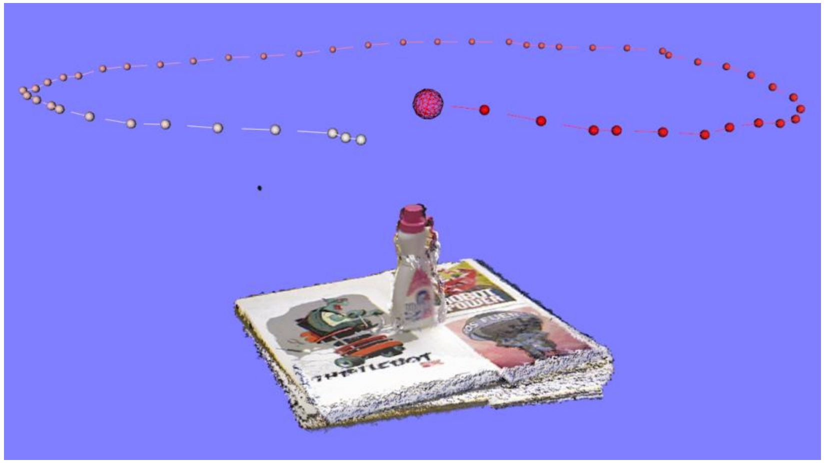
Online stage

- “Frame-to-frame” dense RGB-D odometry (Steinbrucker E. at al.) and detection of one loop closure

- For each frame:
  - plane and object segmentation
  - odometry estimation and outliers filtering
  - is it a keyframe?
  - is it loop closure?

Online stage

- Output:
  $\{x_n^{(0)},\ n = 1, \ldots, N\}$ - initial camera poses estimated by accumulating consecutive frame-to-frame odometry transformations $\{z_n,\ n = 1, \ldots, N-1\}$ and z_N - loop closure

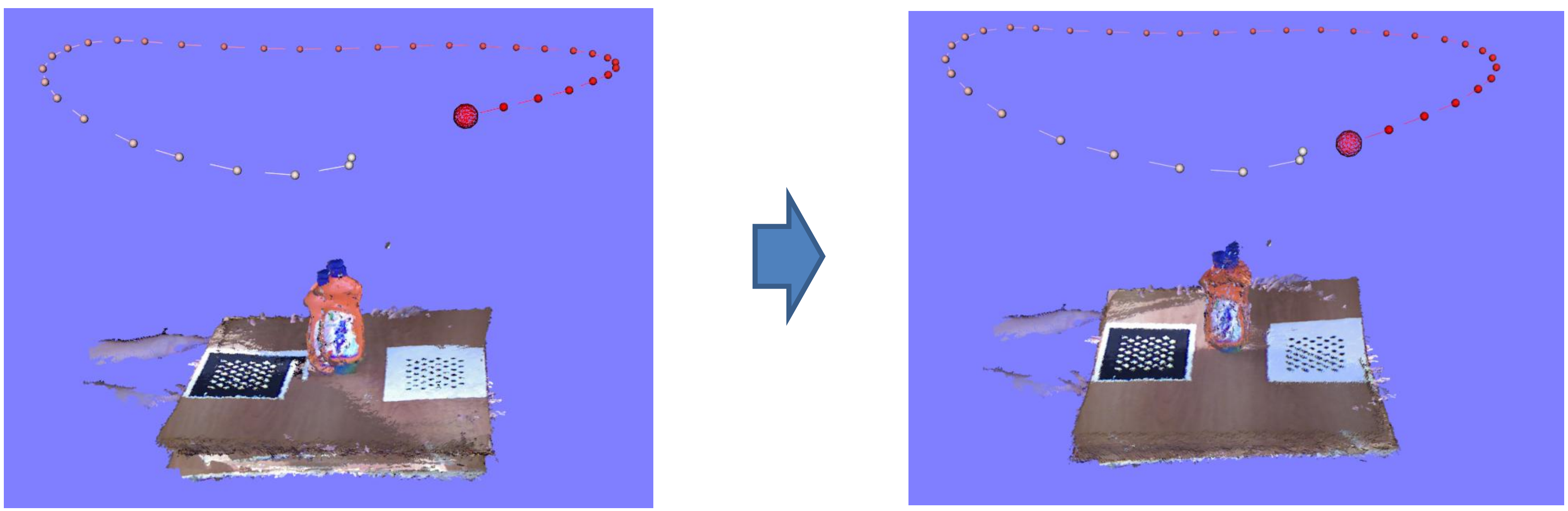


Offline stage

- First step
  - the drift compensation

$$F_{SE3}(x_{1:N}) = \sum_{n} (z_n \ominus y_n)^T \Omega_{SE3} (z_n \ominus y_n)$$

$$y_n = x_{n+1} \ominus x_n \ (n = 1, \ldots, N-1), \ y_N = x_N \ominus x_1$$



RGB-D odometry
(the circles grid is just a texture here!)

loop closure

Offline stage

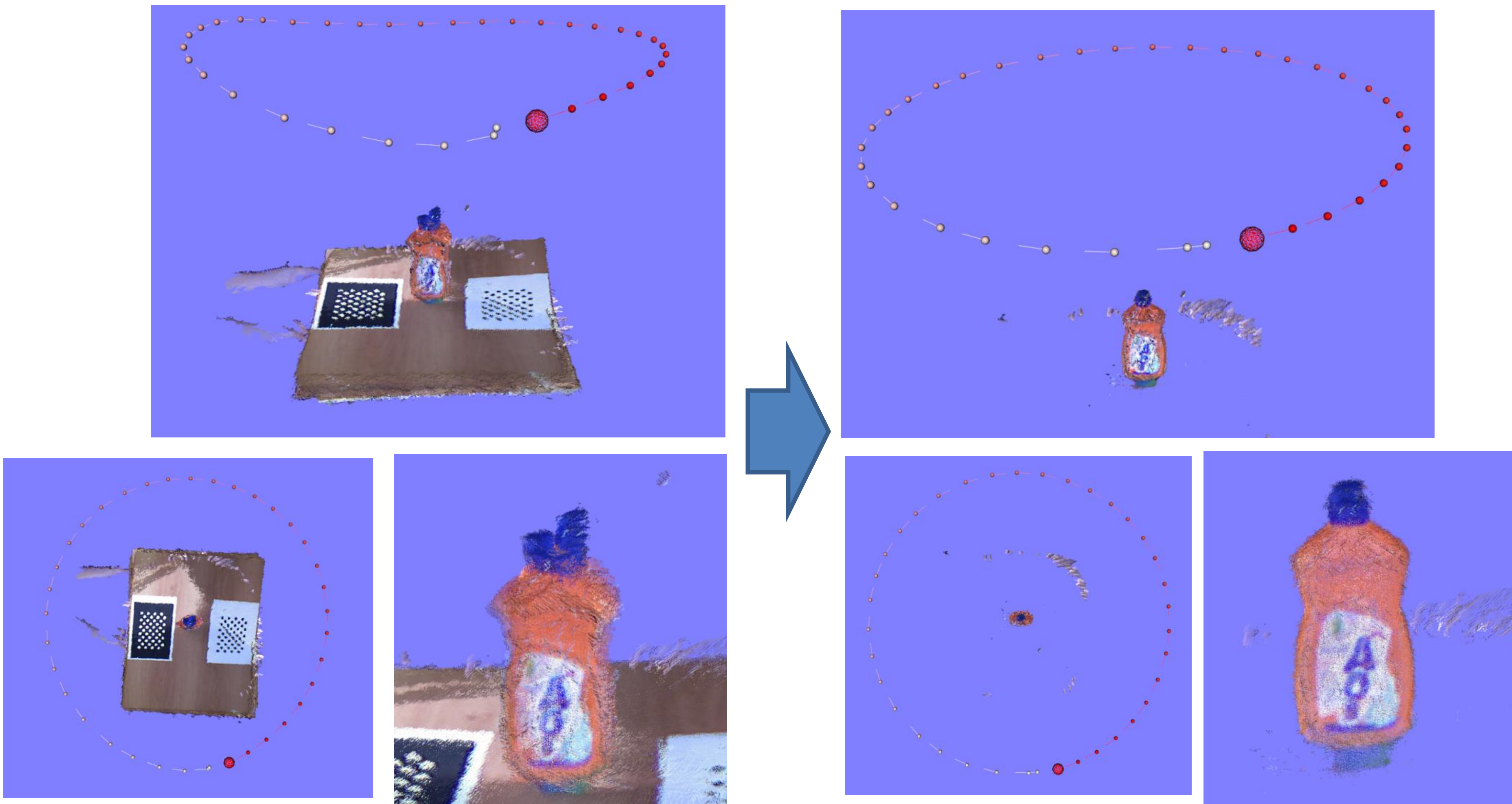
- Second step
  - add the term for coarse-to-fine global ICP and RGB-D optimization (rigid bodies)

$$F_{SE3}(x_{1:N}) + F_{ICP}(x_{1:N}) + F_{RGBD}(x_{1:N})$$

  - loop closure allows to find correspondences using fast projective algorithm
  - optimize the function: at first, for a table and an object and then for the object only

Offline stage

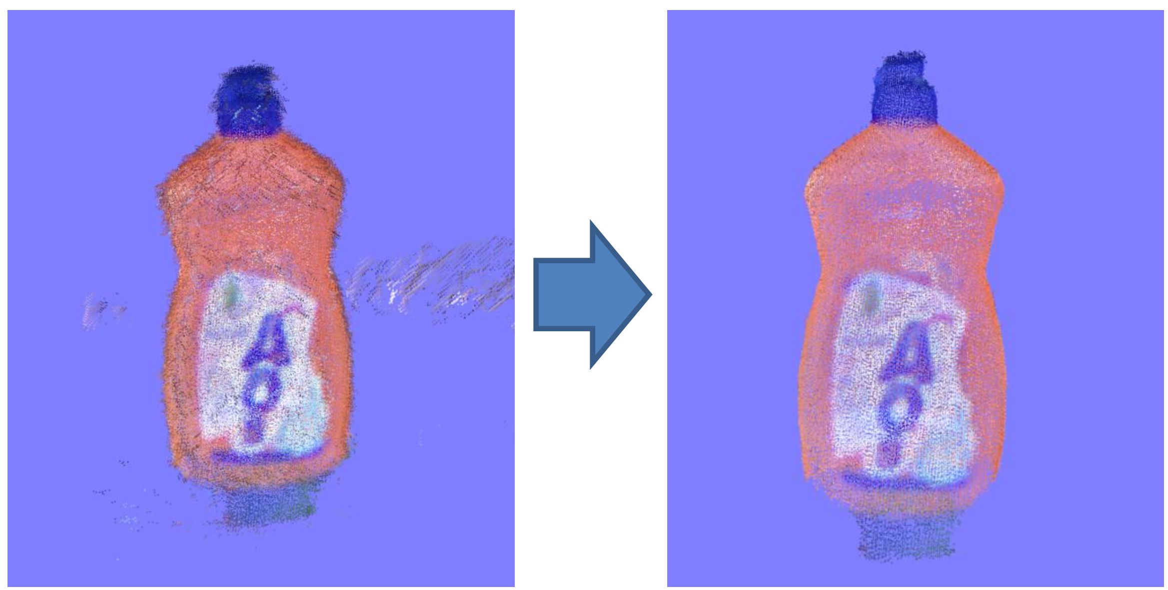
- Second step



Offline stage

- Third step
  - add term to refine model points (nonrigid bodies)

$$F_{SE3}(x_{1:N}) + F_{ICP}(x_{1:N}) + F_{RGBD}(x_{1:N}) + F_{model}(x_{1:N}, M)$$



Results

- Tested on tree capturing scenarios
- Gives accurate 3D models (visually) for a base of 42objects




- Convenient enough for usage by a person and also makes a robot closer to autonomous acquisition of a 3D model