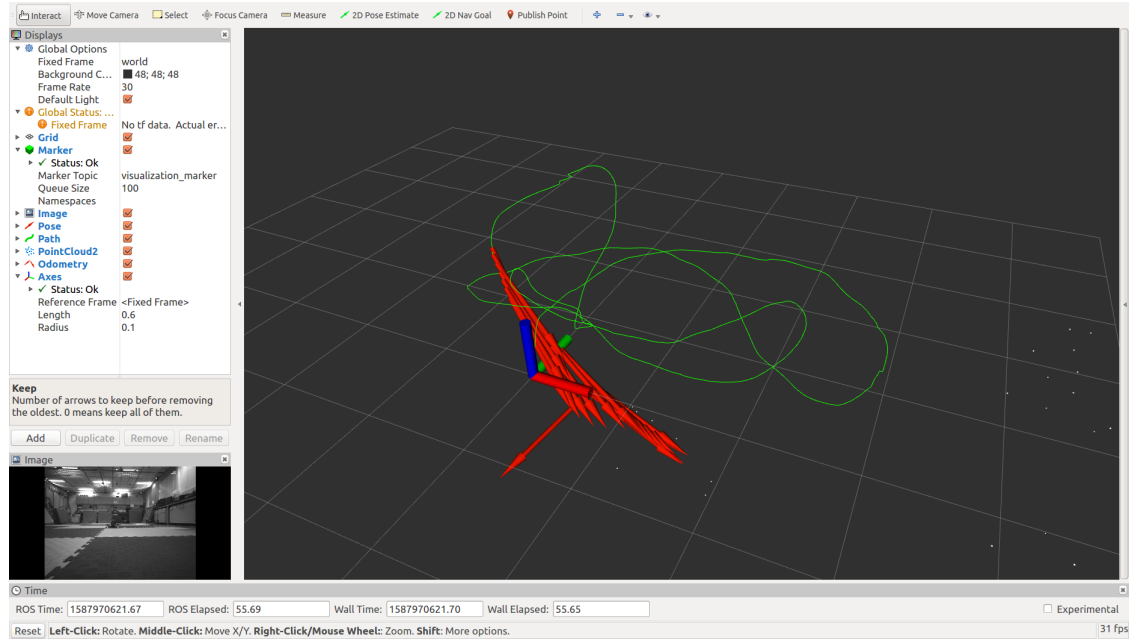


ELEC 5660 Project 2: Phase 2

Binqian JIANG

Figures

Same view angle as the hw guide file.



Description

About definition of transforms

There are 4 coordinate frames, body (inertial) b , left camera (frame) c , l , right camera r and world w , and also some variables defined in all kinds of coordinate frames. Here is the definition in my implementation. T_{ab} means transform from b to a .

Name	In code comments	Actual	Evidence
<code>w_R_i, w_t_i</code>	// r/t in world frame	T_{wc}	
<code>ric[2], tic[2]</code>	// r/t from body to camera	T_{bc}	My estimator.cpp , line 64.*
<code>Tlr</code>	// transform between left and right camera	T_{lr}	As above.
<code>R_21, t_21</code>	// R, t from the keyframe to the current frame	T_{21}	
<code>latest_P, latest_Q</code>	// latest transform R, t	T_{wb}	stereo_vo_node.cpp , published as path.
<code>latest_pointcloud</code>	// latest pointcloud of features in the world frame	In c frame	stereo_vo_node.cpp , line 106, $T_{wb}T_{bc}p_i$ before pub.

* In CV course, when talking about "Extrinsic parameters", it usually means "world-to-cam transformation", but here in the yaml file, it seems to be "cam-to-world/body transformation".

About implementation of each key component

Temporal Matching

LK optical flow is used to track corner points, and a PnP problem is solved to get current pose w.r.t. key frame.

3D reconstruction

LK optical flow is used to track left corner points in the right camera frame. Given the left-right calibration, a set of 3D points in current left camera frame is calculated using SVD.

As LK tracking may not track the same 3D points, causing inconsistent fundamental matrix between left-right and prev-curr images, so a ransac based outlier rejection is used for robust tracking.

About tracking and lost

There is a few cases where the estimation will drift unreasonably far away, possibly as a result of inadequate and wrong tracked points.

There is a `reset()` function in the file, it basically changes the key frame to the last frame and calls re-initialization. I think changing key frame and re-initializing method is helpless in case of lost. First, the criteria for choosing key frames is very loose, slight movement (5cm, 3 deg) will cause a key frame to change. Second, the VO usually get lost in case of radical motion, and in this case, we usually already set almost every frame as key frame. So when it get lost, we just set the prev frame, which is already a key frame, a key frame, which is meaningless. So, I think a feature based method for tracking lost may be more helpful, but that's another story.

Note

I hope the transforms in the code can be more clearly and consistently defined, e.g. `from left to right`, instead of `between left and right`.

Reference

Code partly adapted from VINS-mono.