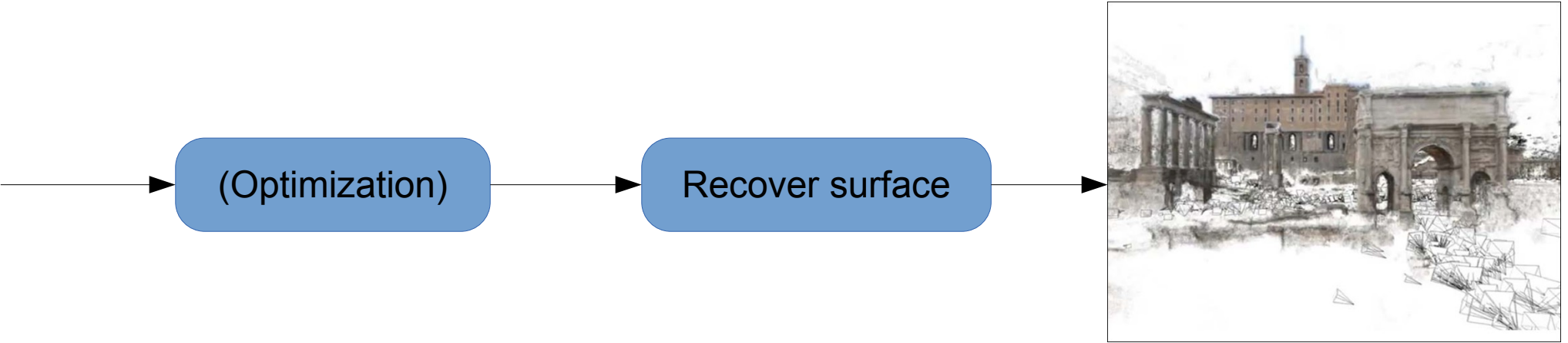
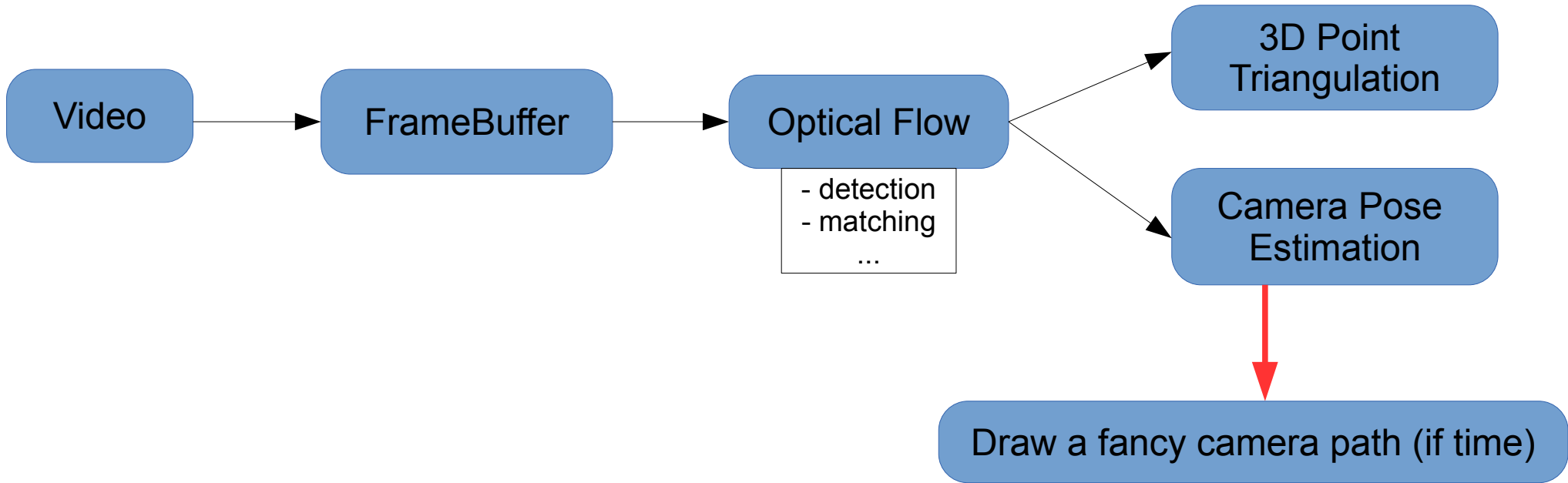


Computer Vision for Robotics Lab
Final Project

~~Monocular Visual Odometry~~
~~Monocular Structure From Motion~~
Monocular Visual Odometry

Team #SWAG, WS 2013/14

Expectations



Data flow

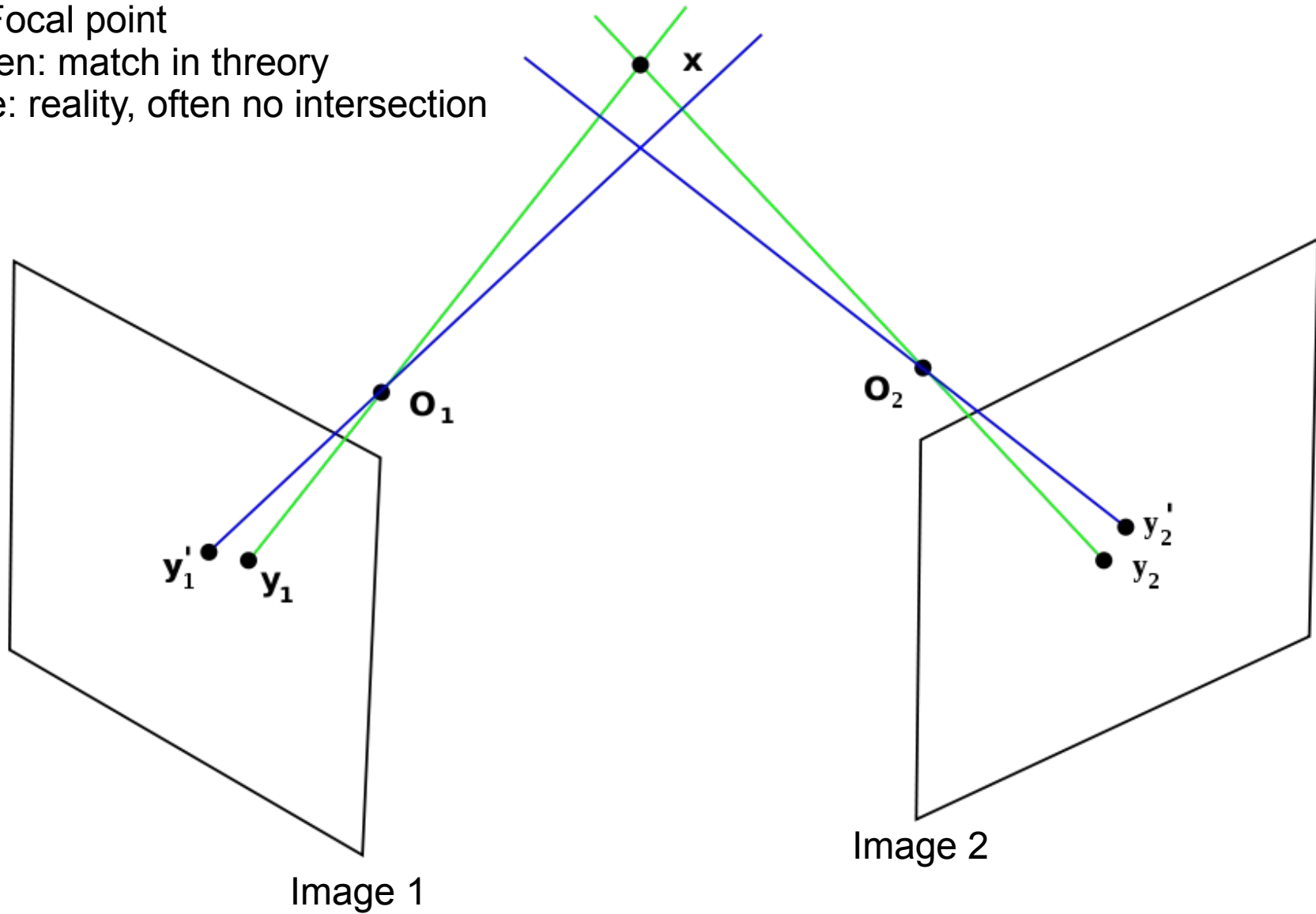
- Corner detection using GoodFeaturesToTrack, with FREAK descriptor
- Frame-by-frame matching
- Base triangulation using first pair of frames and keypoint matches, along with camera calibration
- Further triangulation of following frames' keypoints to 3D space
- Then ideally, refinement, bundle adjustment, loop closure, etc.

Basics

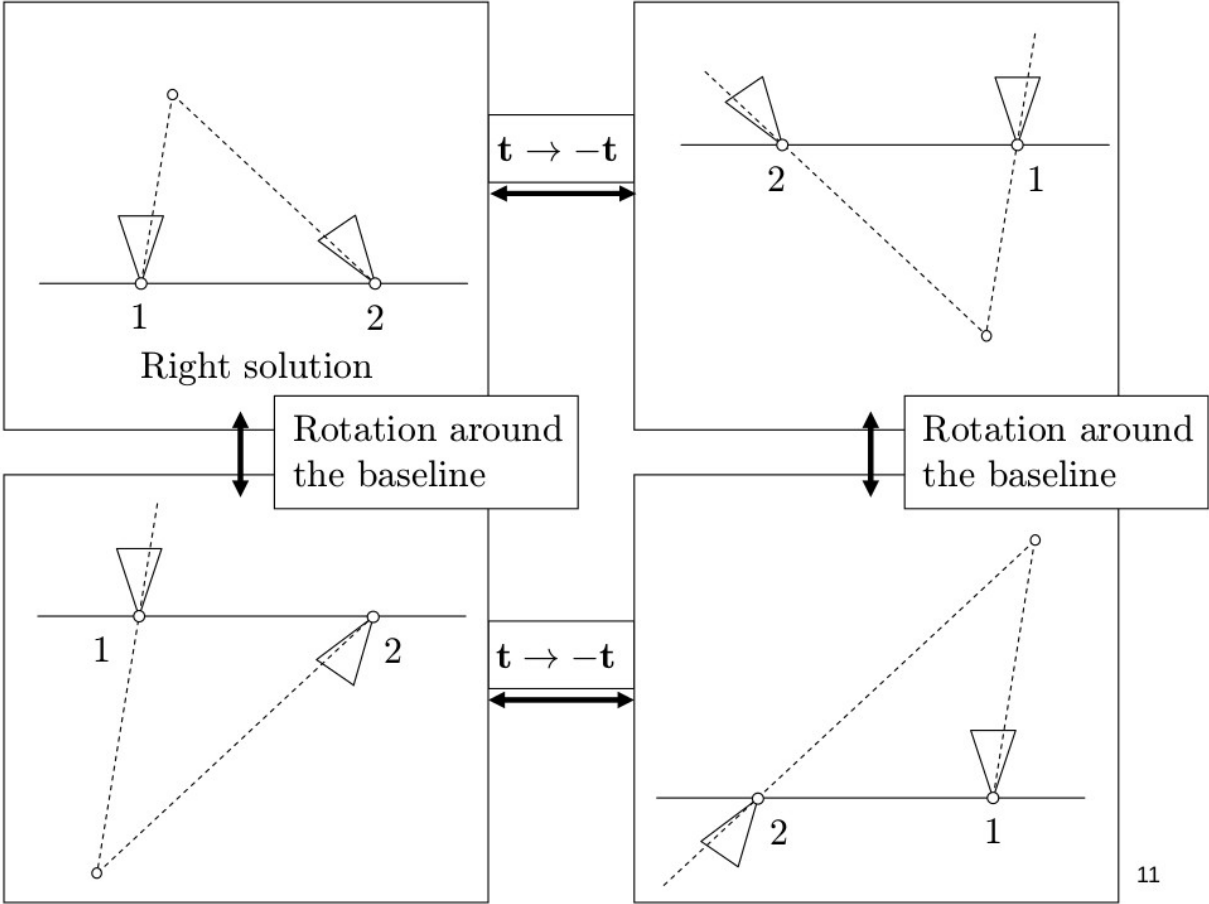
- How does this even work algorithmically?
 - No known translation as in Stereo, have to bootstrap
 - Lots of uncertainty (e.g. scale)
- Triangulation
 - 4 solutions, selecting the correct one
 - Edge case of all point being in front of camera, but in between both cameras in one plane
 - Couldn't be avoided, but can be detected and now being skipped

Basics

O: Focal point
Green: match in theory
Blue: reality, often no intersection



Four solutions



Problems Faced

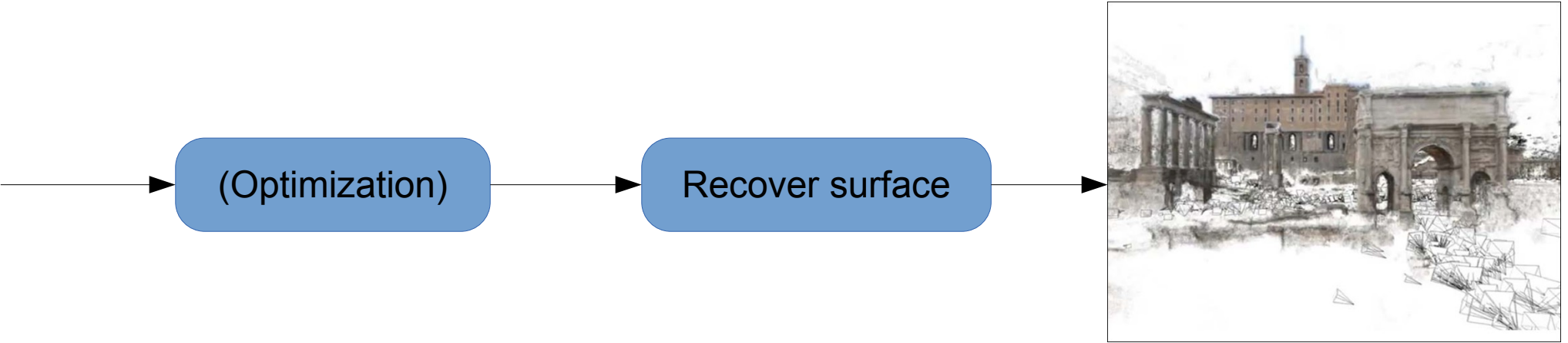
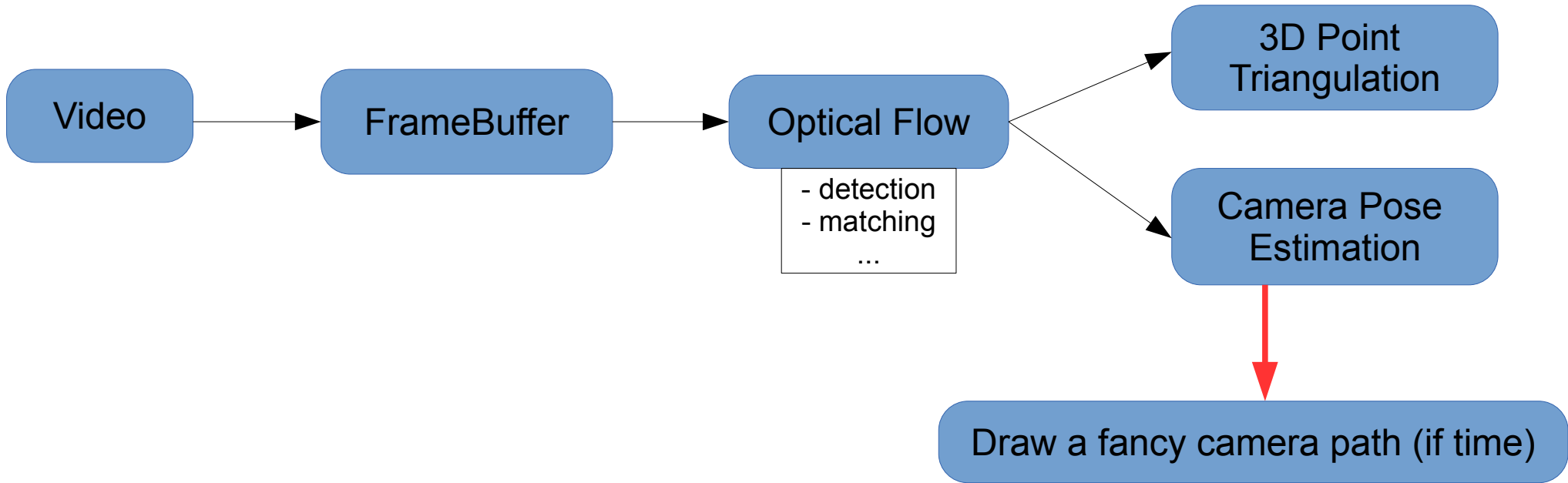
- More ROS
 - Pass parameters but still be able to debug
 - Deal with compressed video streams
 - Calibration (pretty simple finally)
 - RVIZ
- Tons of math
- General ROS and OpenCV stuff
 - Dealing with `cv::Mat` datatype not always trivial (also, greyscale image but colored pointcloud, different cameras, etc.)

Demo (video)

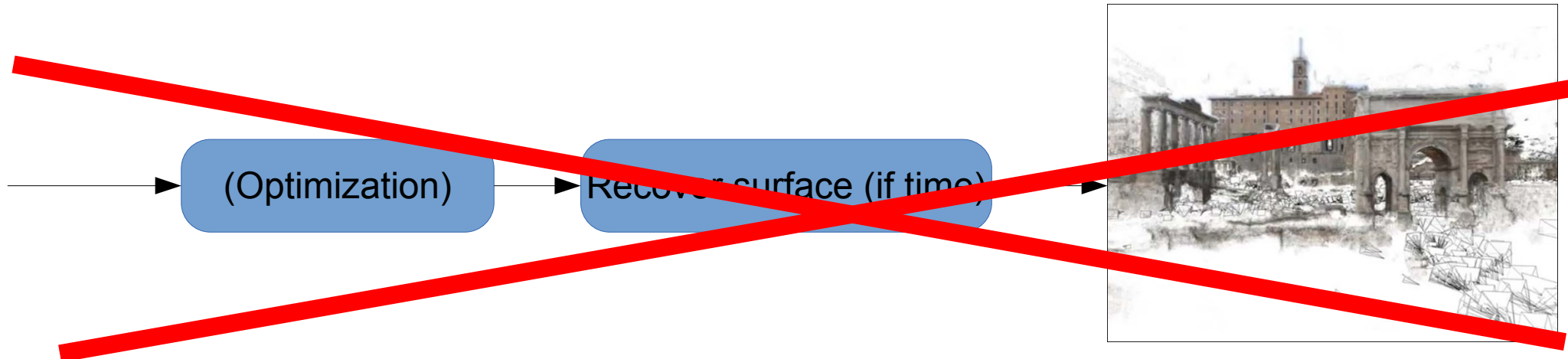
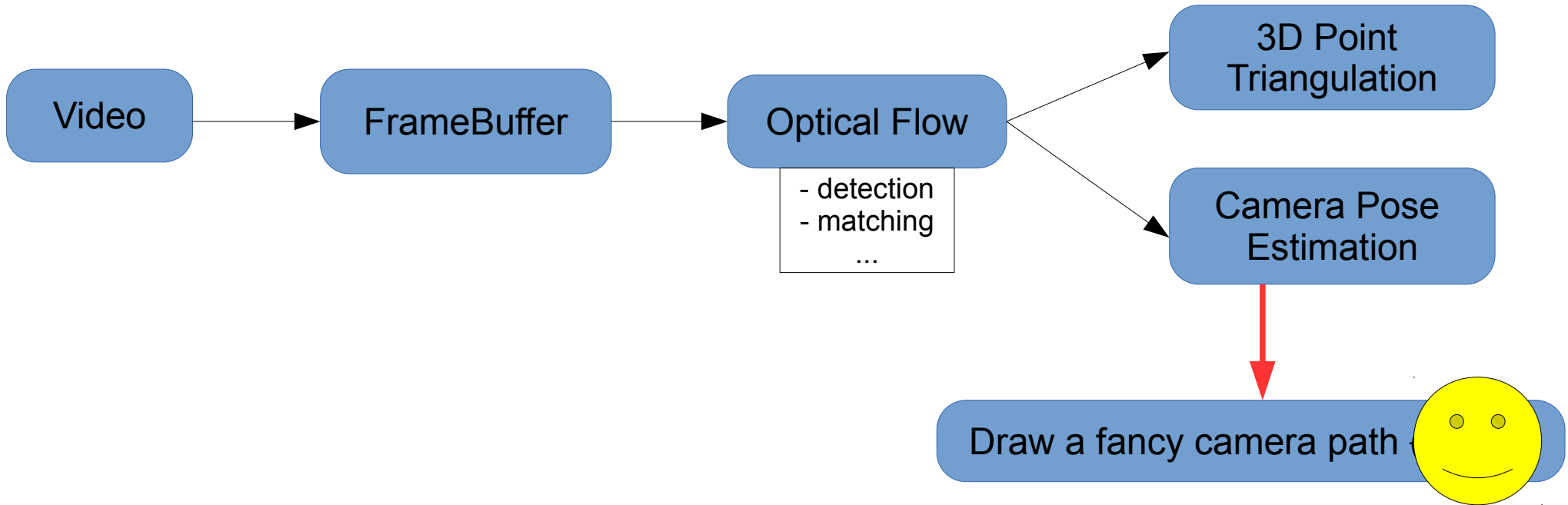
Remaining Problems

- High framerate is actually detrimental (bigger translation from frame to frame seems to perform better)
- Motion blur also problematic
- Can't see very much in the pointcloud
- Not as much optimization as we'd hoped for (no time)

Expectations



Expectations



System performance (kitti)

The screenshot displays the RViz interface for a KITTI dataset. The main view shows a point cloud with a camera trajectory and a grid. The left panel shows display settings for PointCloud2 and MarkerArray. The bottom panel shows time metrics: ROS Time: 1391386361.33, ROS Elapsed: 94.31, Wall Time: 1391386361.46, Wall Elapsed: 94.31. The interface also shows a 7 fps frame rate.

Displays Panel:

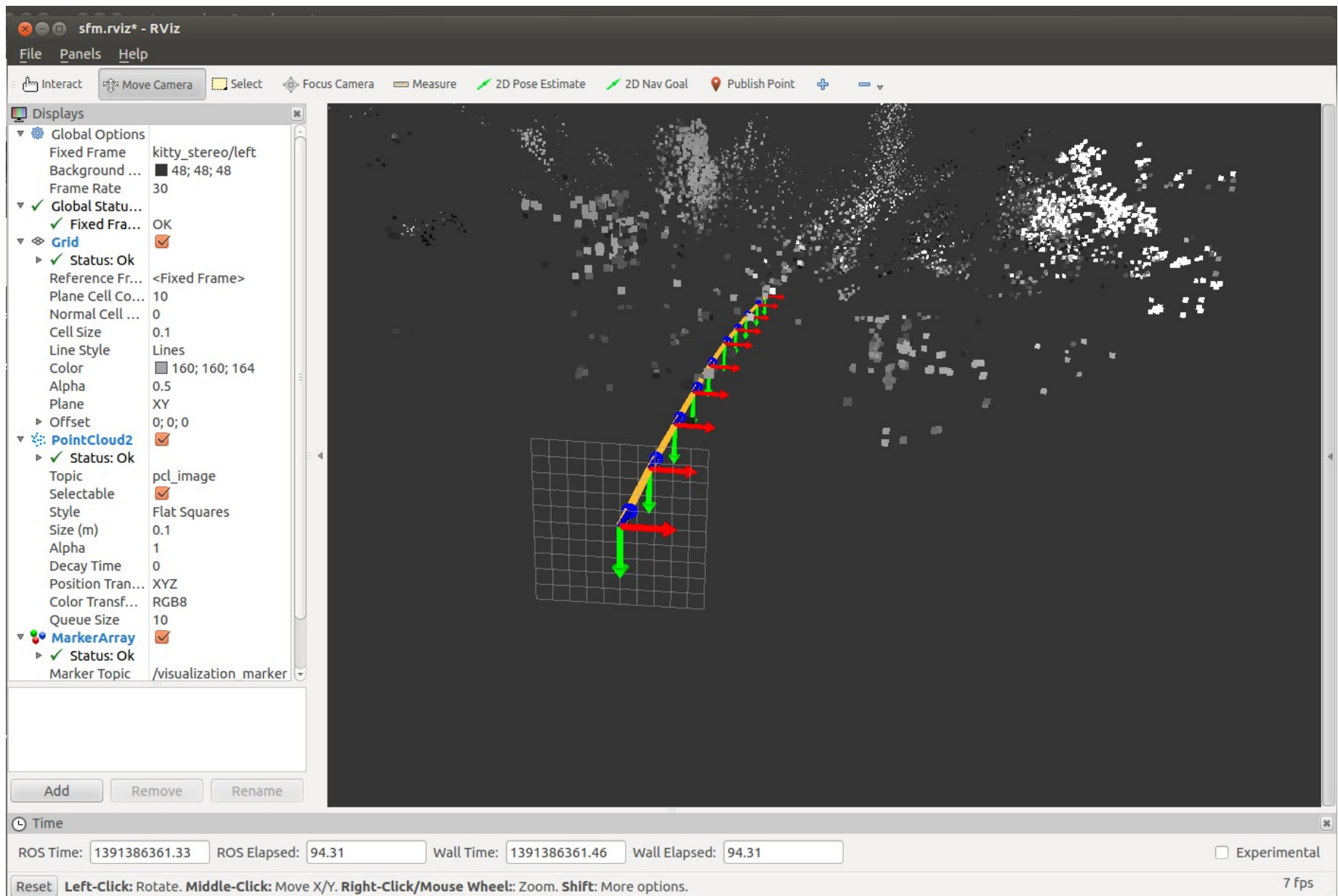
- Global Options
 - Fixed Frame: kitty_stereo/left
 - Background: 48; 48; 48
 - Frame Rate: 30
- Global Status: OK
- Grid
 - Status: OK
 - Reference Frame: <Fixed Frame>
 - Plane Cell Count: 10
 - Normal Cell Count: 0
 - Cell Size: 0.1
 - Line Style: Lines
 - Color: 160; 160; 164
 - Alpha: 0.5
 - Plane: XY
 - Offset: 0; 0; 0
- PointCloud2
 - Status: OK
 - Topic: pcl_image
 - Selectable:
 - Style: Flat Squares
 - Size (m): 0.1
 - Alpha: 1
 - Decay Time: 0
 - Position Transform: XYZ
 - Color Transform: RGB8
 - Queue Size: 10
- MarkerArray
 - Status: OK
 - Marker Topic: /visualization_marker

Time Panel:

ROS Time: 1391386361.33 ROS Elapsed: 94.31 Wall Time: 1391386361.46 Wall Elapsed: 94.31 Experimental

Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options. 7 fps

System performance (PSEye)



Expectations met?

- Odometry very good, especially in kitti dataset
- Manual camera: odometry also consistent with camera movement
- Point cloud OK, no details visible
- Detail and shapes would be improved by bundle adjustment

Outlook

- Meshing algorithms can recover basic structure quite fast
 - If pointcloud is good
 - Again, bundle adjustment would help
- Keyframe selection might also be improved upon

Lessons learned

- Monocular odometry / SFM / SLAM is pretty cool, but really hard
- Stereo allows for more stability and certainty (e.g. about things like scale)
- Kinect is basically cheating ;)
- ROS / OpenCV → stuff is always harder than you expect or hope

Monocular structure from motion

