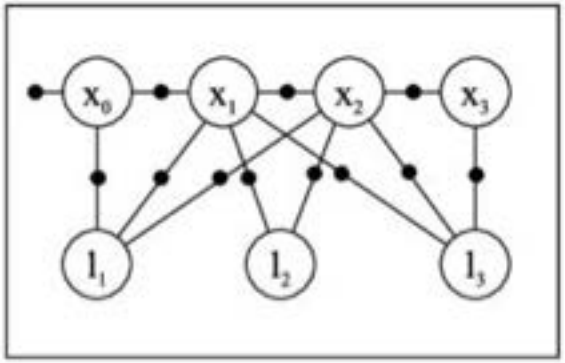


Lab 7: Optimization and GTSAM
VNAV 2020

Lab goals

- https://mit-spark.github.io/VNAV2020-handouts/lab7/

- Theoretical questions: dealing with Nonlinear leasts squares and optimization
  - Review linear algebra
  - Review concepts with Lie groups
  - Apply what MLE/MAP knowledge from lecture
- Team assignments: solving estimation problems using GTSAM (Georgia Tech Smoothing And Mapping)
  - https://smartech.gatech.edu/bitstream/handle/1853/45226/Factor%20Graphs%20and%20GTSAM%20A%20Hands-on%20Introduction%20GT-RIM-CP%26R-2012-002.pdf?sequence=1&isAllowed=y

GTSAM Preliminaries

- Use “factor graphs” to solve estimation problems

$$\underset{X}{\operatorname{argmax}}\, p(z_1, \ldots, z_n \mid X) p(X) \qquad X_{MAP} = \underset{X}{\operatorname{argmin}} \sum_i ||h_i(X_i) - z_i||^2_{\Sigma_i}$$

Assuming additive Gaussian noise

- A “factor” in the factor graph is a summand in the sum

$$||h_i(X_i) - z_i||^2_{\Sigma_i}$$

https://mit-spark.github.io/VNAV2020-handouts/lab7/exercises#gtsam-preliminaries

Anatomy of a Factor

y = x + noise

For each node (variable) we assign a key

Measurement

Noise definition

Interface

```
class UnaryFactor: public NoiseModelFactor1<Pose2> {
  double mx_, my_; ///< X and Y measurements

public:
  UnaryFactor(Key j, double x, double y, const SharedNoiseModel& model):
    NoiseModelFactor1<Pose2>(model, j), mx_(x), my_(y) {}

  Vector evaluateError(const Pose2& q,
                       boost::optional<Matrix&> H = boost::none) const
  {
    if (H) (*H) = Matrix_(2,3, 1.0,0.0,0.0, 0.0,1.0,0.0);
    return Vector_(2, q.x() - mx_, q.y() - my_);
  }
};
```

(q_x, q_y, q_θ),

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\|h(q) - m\|^2_\Sigma$$

Taken from GTSAM hands-on guide

Some factors that you will be using:

- Between factor (https://github.com/borglab/gtsam/blob/develop/gtsam/slam/BetweenFactor.h#L102)

Two nodes (here representing two poses) type gtsam::Key or gtsam::Symbol

Variable in SE3 (template as type gtsam::Pose3)

```
gtsam::BetweenFactor<Pose3>(key1, key2, measurement, noise)
```

$$\|log([T_t^{-1}T_{t+1}]^{-1}T^t_{t+1,meas})^\vee\|^2_\Sigma$$

Observed relative measurement of pose represented by key 2 in the frame of pose represented by key1. In this case between 2 poses, so of type gtsam::Pose3

Some factors that you will be using:

- Prior factor (https://github.com/borglab/gtsam/blob/develop/gtsam/nonlinear/PriorFactor.h#L94)

Variable in SE3 (template as type gtsam::Pose3)

```
gtsam::PriorFactor<Pose3>(key, pose, noise)
```

Observed pose

Single node here representing a single pose

$$||log(T_t^{-1} T_{t,meas})^{\vee}||^2_{\Sigma}$$

Some factors that you will be using:

- GenericProjectionFactor (https://github.com/borglab/gtsam/blob/develop/gtsam/slam/ProjectionFactor.h#L131)

Camera calibration encoded with K of type gtsam::Cal3_S2

GTSAM camera calibration type

```
gtsam::GenericProjectionFactor<Pose3, Point3, Cal3_S2>(measurement, noise, pose_key, landmark_key, K)
```

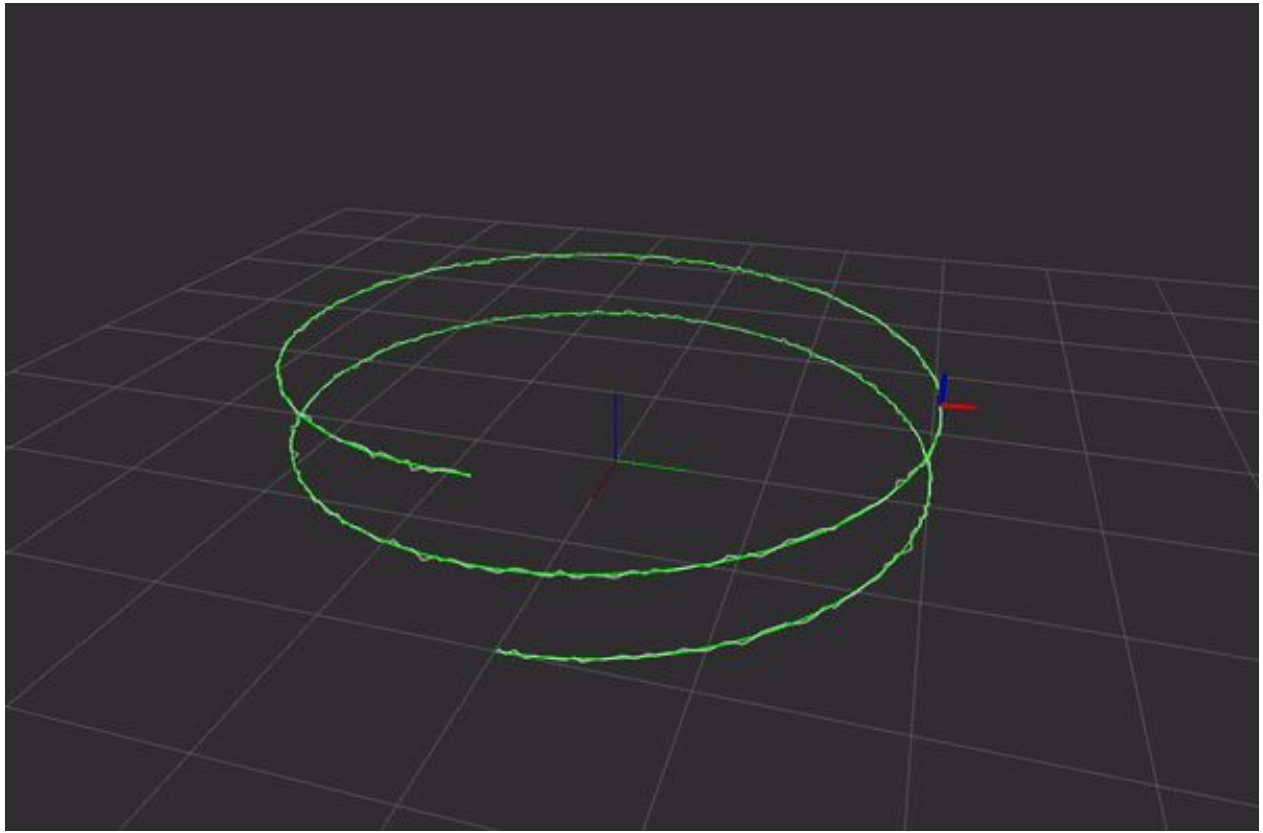
keypoint measurement in pixel coordinates

Key representing camera pose

$$\|y_{k,t} - h_i(T_t^w, l_k^w)\|_\Sigma^2$$

Key representing landmark position

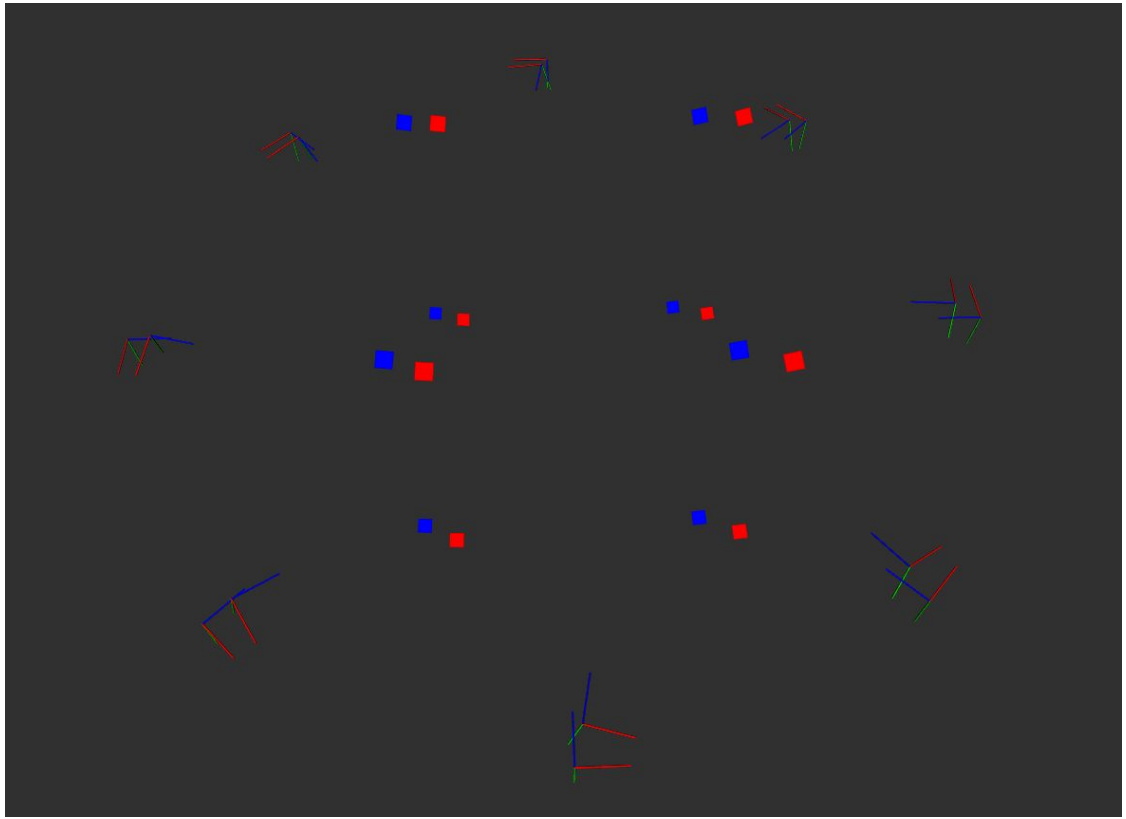
Motion estimation



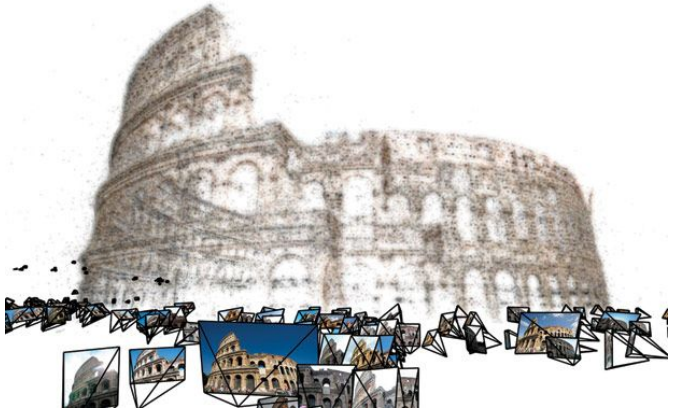
Finding optimal estimated trajectory with:

- Odometry measurements
- Motion capture measurements

(Simple) Bundle Adjustment



Optimizing landmark poses and camera poses with simulated 2D measurements of certain 3D landmarks in the world.



https://grail.cs.washington.edu/rome/

16.485 Visual Navigation for Autonomous Vehicles (VNAV)
Fall 2020