Openvins simulation feature measurements

Saturday, March 14, 2020 10:36 AM

In simulator.cpp, function Simulator::project_pointcloud,

```
Eigen::Matrix<double,3,3> R_ItoC = quat_to_Rot(camera_extrinsics.at(camid).block(0,0,4,1));
Eigen::Matrix<double,3,1> p_IinC = camera_extrinsics.at(camid).block(4,0,3,1);
Eigen::Matrix<double,8,1> cam_d = camera_intrinsics.at(camid);
```

This part reads the ${}^{C}R_{I}$, p_{I}^{C}, and gets K.

```
std::vector<std::pair<size_t, Eigen::VectorXf>> uvs;
```

The true measurements uvs is a vector that contains <size_t, Eigen::VectorXf> as elements. size_t is the feature id and Eigen::VectorXf is pixel measurements.

```
for(const auto &feat:feats){
    Eigen::Vector3d p_FinI = R_GtoI*(feat.second - p_IinG);
    Eigen::Vector3d p_FinC = R_ItoC * p_FinI + p_IinC;
```

Recall that in generate points, we have `featmap.insert({id_map, p_FinG});`

So feat.second means p_{FinG}, and we have

$$p_{FinI} = {}^{I}R_{W}(p_{FinG} - p_{IinG})$$
$$p_{FinC} = {}^{C}R_{I}\, p_{FinI} + p_{IinC}$$

```
Eigen::Vector2f uv_norm;
uv_norm << p_FinC(0)/p_FinC(2), p_FinC(1)/p_FinC(2);
```

uv_norm is $\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \triangleq \frac{1}{z} KP$ which are the projected pixels.

```
uvs.push_back({feat.first, uv_dist});
```

feat.first is the feature id, and uv_dist is the measurement.

In Simulator::get_next_cam function in src/sim/Simulator.cpp.

One input is `std::vector<std::vector<std::pair<size_t, Eigen::VectorXf>>> &feats`

This is a vector of vector, the elements are `std::pair<size_t, Eigen::VectorXf>`

feats contains the measurements of all camera frames. its size is $n \times m \times 2$, where n is frame number, m is measurement in each frame.