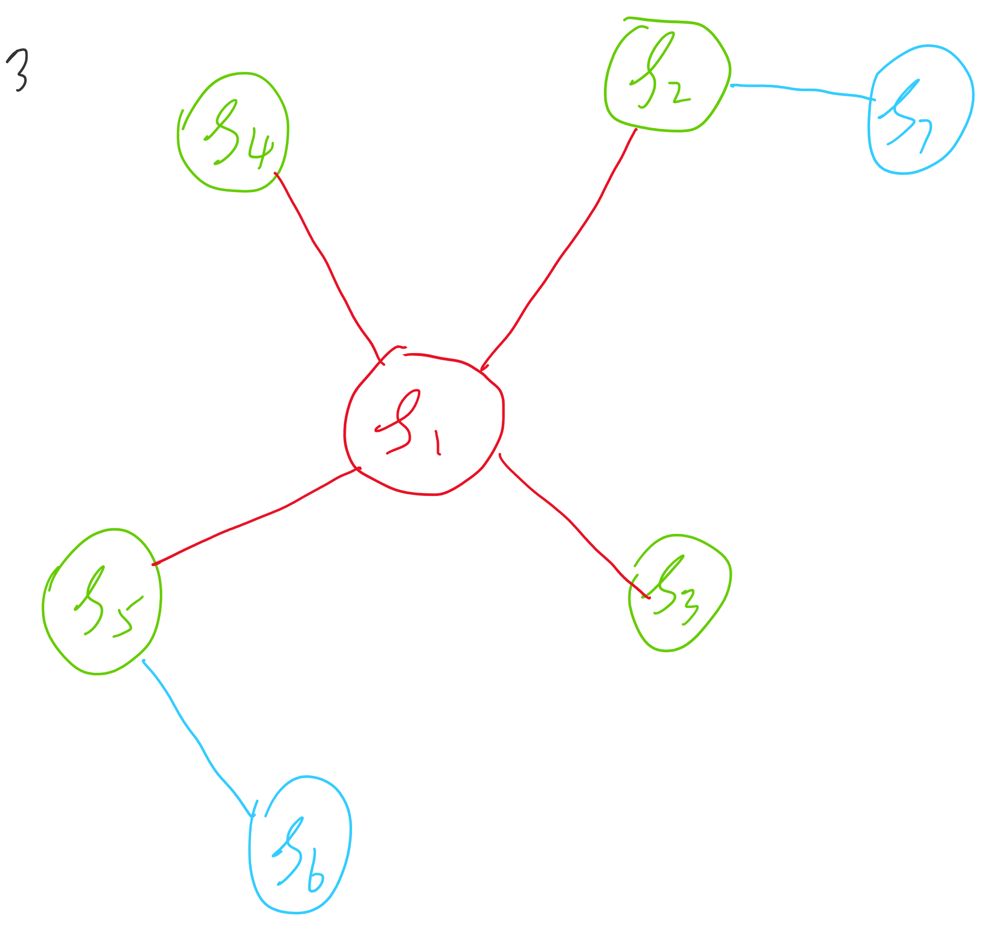
VIO from scratch 6-3

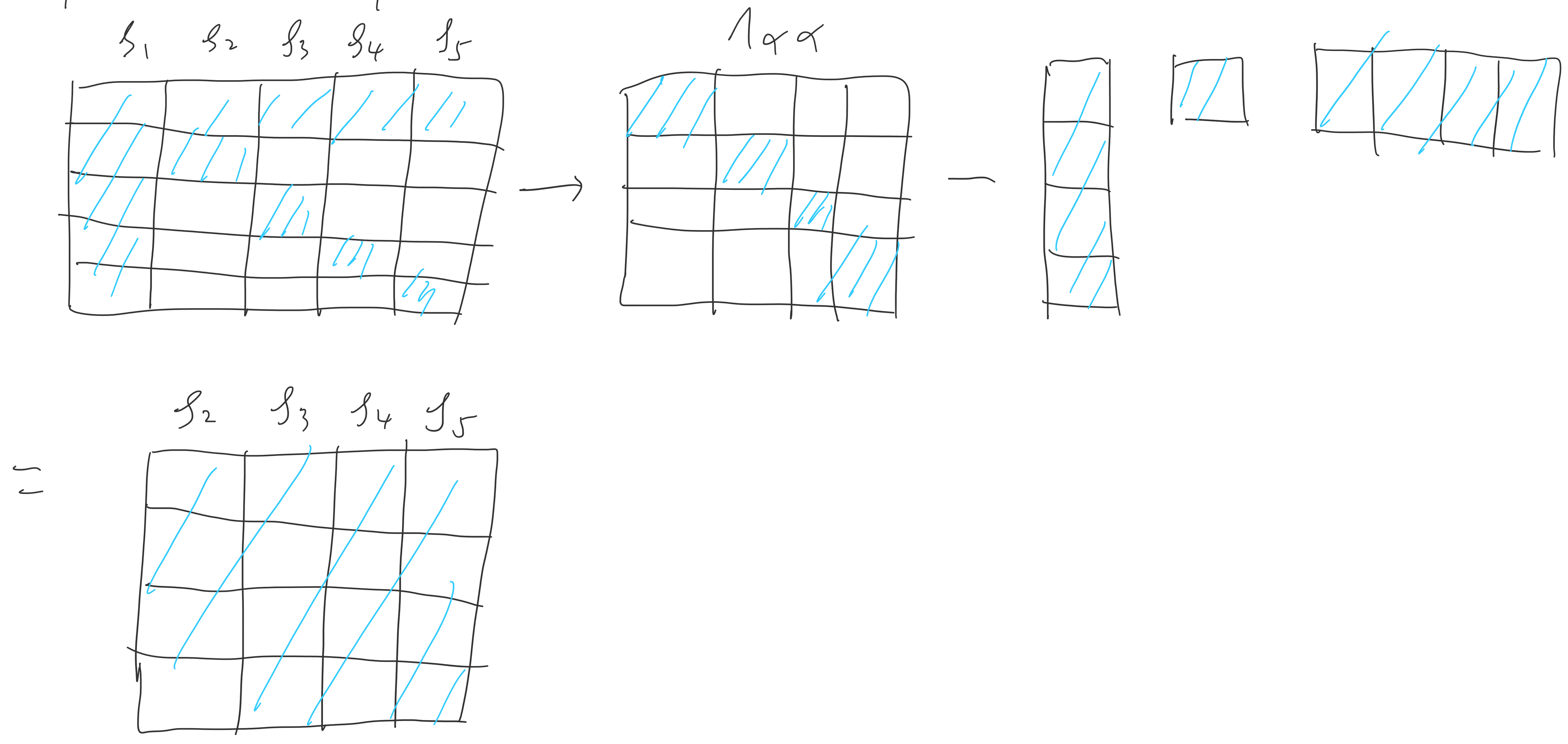
Thursday, April 9, 2020   10:28 AM

toy eg 3

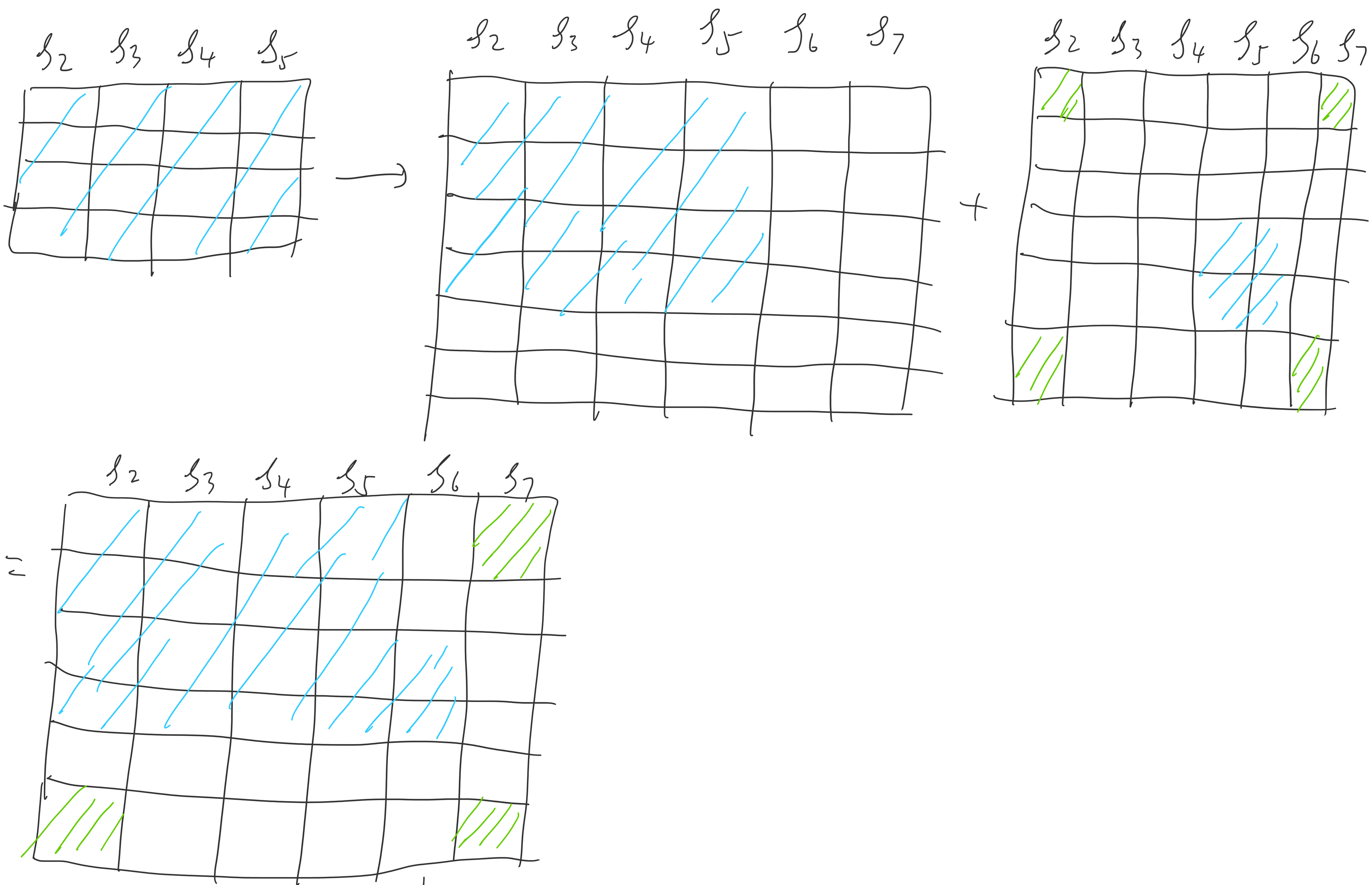


red: variables and constrains to be marginalized,
green: variables related to the one to be marginalized
blue: variables independent to the one to be marginalized.

Step 1: Construct prior



Step 2: prior + new measurements → new info. matrix



Compared with BA, we have added the prior.

issue:

$$\Lambda_p \delta x_p = -b_p$$

when x_p gets updated, how do we change b_p?

Note that Λ_p is fixed.

If we fix b_p as well, then residuals wrt x_p are not changing. ∴ x_p can't be optimized.

So we have to use Taylor expansion:

$$b_p' = b_p + \frac{\partial b_p}{\partial x_p}\delta x_p$$

$$= b_p + \frac{\partial\left(-J^T\Sigma^{-1}r\right)}{\partial x_p}\delta x_p$$

$$= b_p - \Lambda_p \delta x_p$$

$$\frac{\partial -J^T\Sigma^{-1}r}{\partial x_p} = -J^T\Sigma^{-1}\frac{\partial r}{\partial x_p}$$

$$= -J^T\Sigma^{-1}J$$

$$= \Lambda_p$$

Sliding window in VINS-mono:

two way marginalization:

1. If the previous image is a key frame, marginalize the oldest frame
2. If the previous image is not a key frame, then discard all visual observations of the previous image, only keep the imu pre-integration.

In VINS-Mono, there's an estimator.cpp:

The most important function is optimization.

```
ceres::Problem problem;
ceres::LossFunction *loss_function;
```
define problem

...

`AddParameterBlock` — add vertices

...

`MarginalizationFactor` — add prior to the problem

...

`imu_factor` — add imu residual

`ProjectionFactor` — add reprojection residual

...

`ceres::DOGLEG` — optimize using DOGLEG.

`ceres::Solve`

...

`double2vector();` — record the state of first pose before and after the optimization, and use this transformation and apply to all poses in window.