1   Definitions

Φ bijective mapping from minimal coordinates to tangent space.
Perturbation around estimate $\bar{\mathbf{x}}$: $\delta \mathbf{x} = \mathbf{x} \boxplus \bar{\mathbf{x}}^{-1}$

$$\delta \mathbf{x} = \exp(\Phi(\delta \boldsymbol{\chi})) =: \mathrm{Exp}(\boldsymbol{\chi})$$
$$\delta \boldsymbol{\chi} = \Phi^{-1}(\log(\delta \mathbf{x})) =: \mathrm{Log}(\delta \mathbf{x})$$

Parameter Block $\leftrightarrow$ State
Residual Block $\leftrightarrow$ Error term, measurement...

2   The Marginalization Error Term

The steps to marginalize out a certain state in the code is:

1. Add all related residual blocks to the marginalization error term.

2. Marginalize the parameter blocks.

3. Update error computation before next optimization.

2.1   Adding a residual block

Firstly all the states / parameter blocks related to this residual block are found. If this state is not yet related to the marginalization error term, the Hessian $\mathbf{H}$ and the right hand side vector $\mathbf{b}$ is resized accordingly. The values of $\mathbf{H}$ and $\mathbf{b}$ are shuffled around to make sure the sparsity pattern remains the same. Landmark states are added at the end and the others are added in between the 'dense' part and the 'sparse/landmark' part. See Figure 1 for a visualisation. The new areas are set to zero. The current estimate of the parameters are saved as linearization point.

The residual $\mathbf{r}$ and the (minimal) jacobians J_i of the error term with respect to the parameters i are calculated. Some magic is done with the Jacobians if the loss function is not quadratic.

$\mathbf{H}$ and $\mathbf{b}$ are updated at the right location:

$$\mathbf{H}.\texttt{block}(idx(i), idx(i), dim(i), dim(i)) \mathrel{+}= J_i^T J_i \qquad \forall i \qquad (2.1)$$
$$\mathbf{H}.\texttt{block}(idx(i), idx(j), dim(i), dim(j)) \mathrel{+}= J_i^T J_j \qquad \forall j < i \qquad (2.2)$$
$$\mathbf{H}.\texttt{block}(idx(j), idx(i), dim(j), dim(i)) \mathrel{+}= J_j^T J_i \qquad \forall j < i \qquad (2.3)$$
$$\mathbf{b}.\texttt{segment}(idx(i), dim(i)) \mathrel{-}= J_i^T \mathbf{r} \qquad \forall i \qquad (2.4)$$

2.2   Marginalization

We want to marginalize out a few parameters. First, the diagonal of the whole Hessian $\mathbf{H}$ is damped:

$$p_i = \sqrt{\mathbf{H}_{ii}} \qquad (2.5)$$
$$\mathbf{H} \leftarrow \mathrm{diag}(\mathbf{p})^{-1} \mathbf{H} \mathrm{diag}(\mathbf{p})^{-1} \qquad (2.6)$$
$$\mathbf{b} \leftarrow \mathrm{diag}(\mathbf{p})^{-1} \mathbf{b} \qquad (2.7)$$
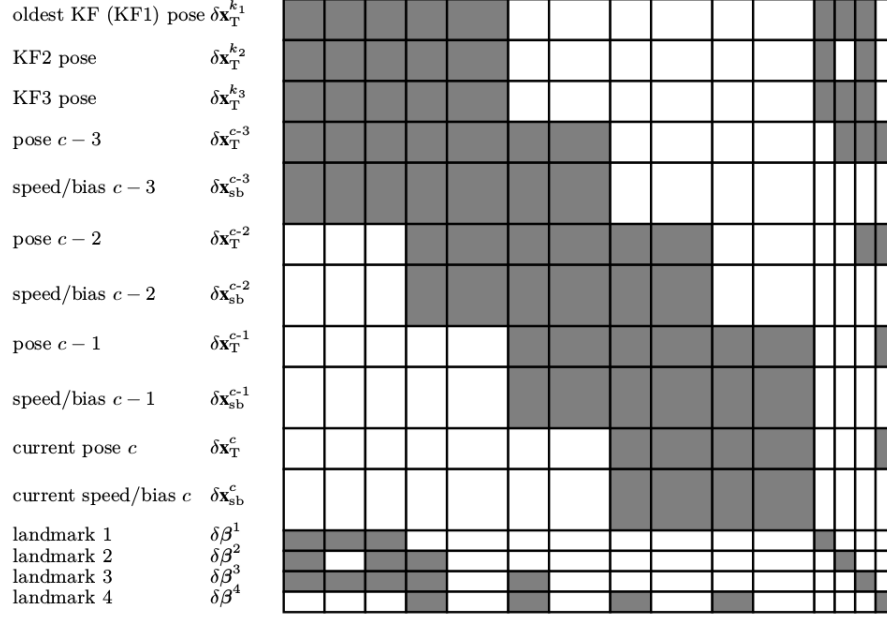
Figure 1: Sparsity pattern of Hessian. [1]

The Hessian $\mathbf{H}$ is split into 3 parts. $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$. $\mathbf{U}$ is part of the Hessian that is kept, $\mathbf{V}$ is the part related to the soon to be marginalized states and $\mathbf{W}$ is the part that is related to both things. The rhs $\mathbf{b}$ is split into 'kept' parts $\mathbf{b}_a$ and 'to be marginalized' part $\mathbf{b}_b$. The preconditioner is also split into $\mathbf{p}_a$ and $\mathbf{p}_b$.

Now the Schur operation begins. For this the pseudo inverse of $\mathbf{V}$ is calculated $\mathbf{V}^+$

$$\mathbf{H} = \mathbf{U} - \mathbf{W}\mathbf{V}^+\mathbf{W}^T \tag{2.8}$$

$$\mathbf{b} = \mathbf{b}_a - \mathbf{W}\mathbf{V}^+\mathbf{b}_b \tag{2.9}$$

And now the Hessian $\mathbf{H}$ and rhs $\mathbf{b}$ are unscaled again.

$$\mathbf{H} \leftarrow \mathrm{diag}(\mathbf{p}_a)\mathbf{H}\mathrm{diag}(\mathbf{p}_a) \tag{2.10}$$

$$\mathbf{b} \leftarrow \mathrm{diag}(\mathbf{p}_a)\mathbf{b} \tag{2.11}$$

2.3 Update error computation

It is important that $\mathbf{H}$ stays positive semidefinit. Due to numeric noise this might not be the case. Therefore the eigenvalue decomposition (of the damped Hessian) $\mathbf{V}\mathbf{S}\mathbf{V}^T$ is calculated and all eigenvalues below a threshold are set to zero. The threshold is $t = \varepsilon \cdot \mathbf{H}.\mathtt{cols}() \cdot \max_i \lambda_i$.

Now certain values are calculated that are used in the error calculation: The Jacobian is 'extracted' from the decomposition.

$$\mathbf{J} = \left(\mathrm{diag}(\mathbf{p})V\sqrt{S}\right)^T \tag{2.12}$$

2

As well as the constant error $\mathbf{e}_0$:

$$(\mathbf{J}^T)^{-1} = \sqrt{S^{-1}} V^T \text{diag}(\mathbf{p})^{-1} \tag{2.13}$$

$$\mathbf{e}_0 = -(\mathbf{J}^T)^{-1} \mathbf{b} \tag{2.14}$$

2.4 Error calculation

We can approximate the right hand side as:

$$\mathbf{b} \approx \mathbf{b}_0 + \frac{\partial \mathbf{b}}{\partial \Delta \boldsymbol{\chi}} = \mathbf{b}_0 - \mathbf{H} \Delta \boldsymbol{\chi} \tag{2.15}$$

Or equivalently

$$\mathbf{e} \approx \mathbf{e}_0 + \mathbf{J} \Delta \boldsymbol{\chi} \tag{2.16}$$

Since

$$\mathbf{b} = -\mathbf{J}^T \mathbf{e} \tag{2.17}$$

$$\approx -\mathbf{J}^T \left(\mathbf{e}_0 + \left. \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \delta \boldsymbol{\chi}} \right|_{\mathbf{x}_0} \delta \boldsymbol{\chi} \right) \tag{2.18}$$

$$\approx -\mathbf{J}^T \left(\mathbf{e}_0 + \mathbf{J} \delta \boldsymbol{\chi} \right) \tag{2.19}$$

$$\approx \underbrace{-\mathbf{J}^T \mathbf{e}_0}_{\mathbf{b}_0} - \mathbf{H} \delta \boldsymbol{\chi} \tag{2.20}$$

$\Delta \boldsymbol{\chi}$ represents the state updates that occur after marginalization.

$$\Delta \boldsymbol{\chi} = \text{Log}(\bar{\mathbf{x}} \boxplus \mathbf{x}_0^{-1}) = \text{Log}(\bar{\mathbf{x}} \boxminus \mathbf{x}_0) \tag{2.21}$$

The error is:

$$\mathbf{e} = \mathbf{e}_0 + J \Delta \boldsymbol{\chi} \tag{2.22}$$

References

[1] S. Leutenegger, "Unmanned solar airplanes: Design and algorithms for efficient and robust autonomous operation," Ph.D. dissertation, 2014. [Online]. Available: http://hdl.handle.net/10044/1/18247

[2] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial slam using nonlinear optimization." [Online]. Available: http://hdl.handle.net/10044/1/23414

[3] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms.* Springer, 1999, pp. 298–372.