在 SLAM 系统的时候，传感器数据在不断增加，例如，摄像头在不断地采集图像数据，因此，被优化的变量也在不断增加。当被优化的变量维度增加到一定程度时，势必出现在下一个传感器数据来临时，上一个传感器数据还没有处理完成的情况。因此需要限制状态变量的规模。考虑目标函数：

$$\mathbf{L}(\delta\mathbf{x}_p, \delta\mathbf{x}_m) = \Big[f(\mathbf{x}_p, \mathbf{x}_m) + \mathbf{J}_{\mathbf{x_p}}\delta\mathbf{x}_p + \mathbf{J}_{\mathbf{x_m}}\delta\mathbf{x_m}\Big]^\top \mathbf{W} \Big[f(\mathbf{x}_p, \mathbf{x}_m) + \mathbf{J}_{\mathbf{x_p}}\delta x_p + \mathbf{J}_{\mathbf{x_m}}\delta\mathbf{x_m}\Big] \tag{1}$$

令目标函数的一阶导为 0，可以得到正规方程：

$$\begin{bmatrix} \mathbf{\Lambda_p} & \mathbf{\Lambda_{pm}} \\ \mathbf{\Lambda_{pm}^{\mathit{T}}} & \mathbf{\Lambda_m} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x_p} \\ \delta\mathbf{x_m} \end{bmatrix} = \begin{bmatrix} \mathbf{b_p} \\ \mathbf{b_m} \end{bmatrix} = -\begin{bmatrix} \mathbf{J}_p^\top \mathbf{W} f \\ \mathbf{J}_m^\top \mathbf{W} f \end{bmatrix} \tag{2}$$

根据前文的描述，正规方程的求解可以用舒尔补加速：

$$\begin{bmatrix} \mathbf{\Lambda_p} - \mathbf{\Lambda_{pm}}\left(\mathbf{\Lambda_m}\right)^{-1}\mathbf{\Lambda_{pm}^T} & 0 \\ \mathbf{\Lambda_{pm}^{\mathit{T}}} & \mathbf{\Lambda_m} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x_p} \\ \delta\mathbf{x_m} \end{bmatrix} = \begin{bmatrix} \mathbf{b_p} - \mathbf{\Lambda_{pm}}\left(\mathbf{\Lambda_m}\right)^{-1}\mathbf{b_m} \\ \mathbf{b_m} \end{bmatrix} \tag{3}$$

不妨假设 $\mathbf{x}_m$ 为需要被边缘化的变量。变量在被边缘化之后，残差对于变量的雅可比不再变化，即线性化点不再改变，其对应的信息矩阵 Λ_m 不再变化。令 $\mathbf{\Lambda_{prior}} = -\mathbf{\Lambda_{pm}}\left(\mathbf{\Lambda_m}\right)^{-1}\mathbf{\Lambda_{pm}^T}$, $\mathbf{b}_{prior} = -\mathbf{\Lambda_{pm}}\left(\mathbf{\Lambda_m}\right)^{\mathbf{-1}}\mathbf{b_m}$，则正规方程变为：

$$(\mathbf{\Lambda_p} + \mathbf{\Lambda_{prior}})\delta\mathbf{x}_p = \mathbf{b_p} + \mathbf{b_{prior}} \tag{4}$$

并且有：

$$\begin{aligned}
\frac{\partial\mathbf{b_{prior}}}{\partial\mathbf{x_p}} &= \frac{\partial(-\mathbf{\Lambda_{pm}}\left(\mathbf{\Lambda_m}\right)^{\mathbf{-1}}\mathbf{b_m})}{\partial\mathbf{x_p}} &(5)\\
&= \frac{(-\mathbf{\Lambda_{pm}}\left(\mathbf{\Lambda_m}\right)^{\mathbf{-1}})\partial\mathbf{b_m}}{\partial\mathbf{x_p}} &(6)\\
&= \frac{(-\mathbf{\Lambda_{pm}}\left(\mathbf{\Lambda_m}\right)^{\mathbf{-1}})\mathbf{J_m}^\top\mathbf{W}\partial\mathbf{f}}{\partial\mathbf{x_p}} &(7)\\
&= (-\mathbf{\Lambda_{pm}}\left(\mathbf{\Lambda_m}\right)^{\mathbf{-1}})\mathbf{J_m^\top}\mathbf{W}\mathbf{J_p} &(8)\\
&= (-\mathbf{\Lambda_{pm}}\left(\mathbf{\Lambda_m}\right)^{\mathbf{-1}})\mathbf{\Lambda_{mp}} &(9)\\
&= \mathbf{\Lambda_{prior}} &(10)
\end{aligned}$$

变量被边缘化后，根据变量被边缘化时的状态计算出来的 $\mathbf{\Lambda_{prior}}$ 不再改变，只需要对 $\mathbf{b_{prior}}$ 更新，即边缘化后，若 LM 算法某一次迭代计算出来的步长为 $\delta\mathbf{x_p}$, 则 $\mathbf{b_{prior}}$ 需要使用一阶泰勒近似更新为 $\mathbf{b_{prior}} + \mathbf{\Lambda_{prior}}\delta\mathbf{x_p}$，因此变量被边缘化的过程就是计算 $\mathbf{\Lambda_{prior}}$ 和 $\mathbf{b_{prior}}$ 的过程，即正规方程变为：

$$(\mathbf{\Lambda_p} + \mathbf{\Lambda_{prior}})\delta\mathbf{x}_r = \mathbf{b_p} + \mathbf{b_{prior}} + \mathbf{\Lambda_{prior}}\Delta\mathbf{x_r} \tag{11}$$

上式等价于在边缘化后剩下的残差项的基础上，新加一个残差 $\frac{1}{2}(\bar{\mathbf{f}}_0+\bar{\mathbf{J}}_{prior}\mathbf{x}_r)^\top(\bar{\mathbf{f}}_0+\bar{\mathbf{J}}_{prior}\mathbf{x}_r)$, 其中 $\bar{\mathbf{J}}_{prior}^\top\bar{\mathbf{J}}_{prior}=-\mathbf{\Lambda}_{prior}$，考虑 $\mathbf{\Lambda}_{prior}$ 是一个是对称矩阵，可以对 $\mathbf{\Lambda}_{prior}$ 进行特征值分解得到 ${\mathbf{J}_{prior}^{-}}^\top\mathbf{J}_{prior}^{-}=\mathbf{USU}^\top$，于是 $\bar{\mathbf{J}}_{prior}=\mathbf{S}^{\frac{1}{2}}\mathbf{U}^\top$。

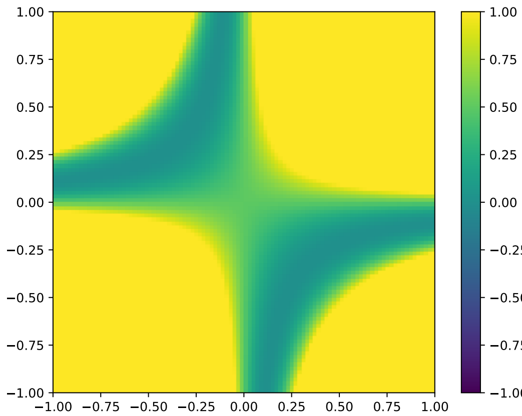

图 1　原始目标函数 $f(x,y)=\frac{1}{2}(xy-1)^2$

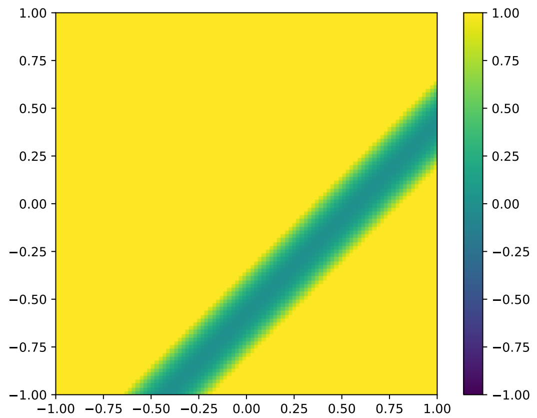

图 2　在 $(2,2)$ 处线性化后的目标函数

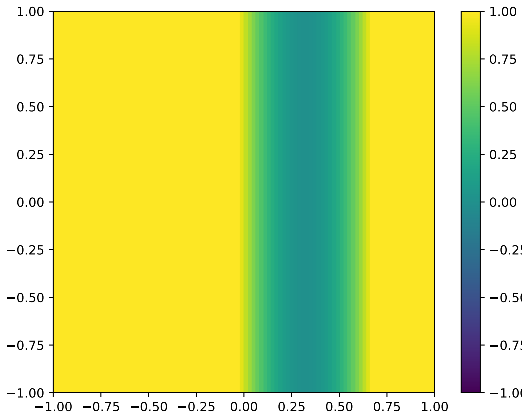

图 3　在 $(1.1,1.1)$ 处边缘化后的目标函数

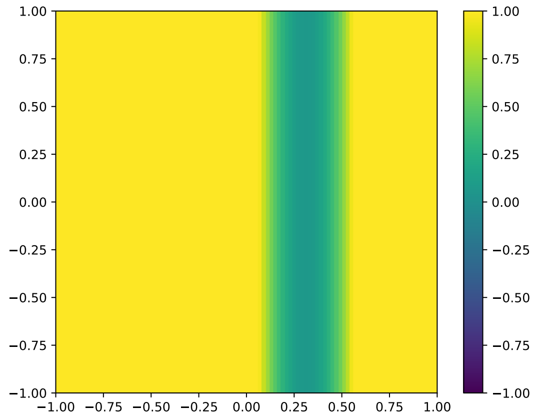

图 4　在 $(1.5,1.5)$ 处边缘化后的目标函数

为了更进一步观察边缘化对优化的影响，以非线性函数 $\frac{1}{2}f(x,y)=(xy-1)^2$ 为例，我们在图1中画出了原始目标函数的图像，其中横坐标为 x，纵坐标为 y，函数值大于 1 的函数值统一为黄色，显然该目标函数的解空间为 $xy=1$，对于该目标函数而言，变量 x 和 y 都是不可观的。在图2中画出了 $f(x,y)=\frac{1}{2}(xy-1)^2$ 在 $(2,2)$ 处线性化后的图像，线性化后目标函数变为 $\frac{1}{2}(f(2,2)+2(x-2)+2(y-2))^2$, 注意此时变量 x 和变量 y 依然是不可观测的。在图3中画出了目标函数在 $(1.1,1.1)$ 处将变量 y 边缘化后的图像，此时目标函数变为 $f(x,y)=\frac{1}{2}(1.1x-1)^2+\frac{1}{2}(f(1.1,1.1)+1.1(x-1.1))^2$，在图4中画出了目标函数在 $(1.5,1.5)$ 处将变量 y 边缘化后的图像，此时目标函数变为 $f(x,y)=\frac{1}{2}(1.5x-1)^2+\frac{1}{2}(f(1.5,1.5)+1.5(x-1.5))^2$，并且无论是在 $(1.1,1.1)$ 处边缘化，还是 $(1.5,1.5)$ 处边缘化，目标函数关于 x 的解都是唯一的，也就是 x 的可观性发生了变化，本来它是不可观的，经过边缘化后变得可观了, 具体来说，是信息矩阵的秩发生了变化[1]。在某些情况下，变量的可观性的变化对最终的优化结果有负面影响，

若固定线性化点以及雅可比后变量的可观性不会发生变化，因此在[2,3] 等论文中，采用了 FEJ(First Estimate Jacobian) 策略[4]。但 FEJ 对精度的影响不一定是正面的，例如在 Vins-Mono[5] 对 VIO 的实现中，作者声称即便不采用 FEJ 策略，在不可观的自由度上也不会有明显的漂移，反而采用 FEJ 策略后，系统的精度会有一定的下降。

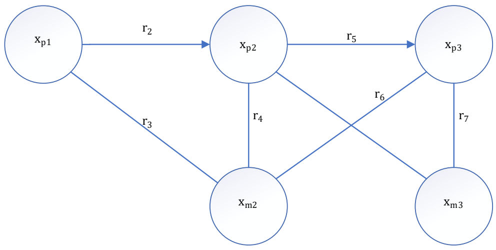

图 5　边缘化 $\mathbf{x}_{m1}$

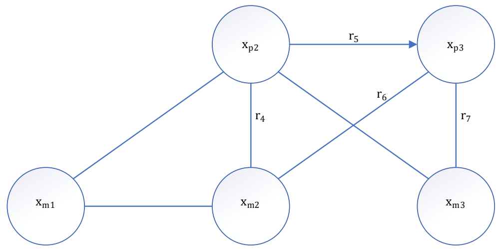

图 6　边缘化 $\mathbf{x}_{p1}$

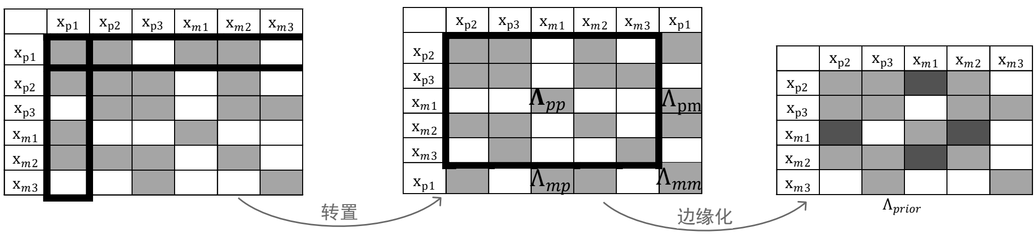

图 7　边缘化中信息矩阵的变化过程，在确定需要边缘化的变量之后，需要首先将它们移动到信息矩阵的右下角，然后使用舒尔补更新信息矩阵，得到的信息矩阵将作为先验信息参与到后面的优化中。

边缘化除了会改变变量的解空间，让本不可观的变量变得可观外，还会破坏信息矩阵的稀疏性，在图??中，我们给出了一个 VIO 例子，该例子中有 3 个相机，3 个特征点，相机 $\mathbf{x_{p1}}$ 可以观测到 1 个特征点，相机 $\mathbf{x_{p2}}$ 和 $\mathbf{x_{p3}}$ 都可以观测到 2 个特征点，在图5中画出了边缘化 $\mathbf{x_{m1}}$ 后图的变化，在图6中画出了边缘化 $\mathbf{x_{p1}}$ 后图的变化，从图上来看，边缘化一些变量后会让剩下的变量之间的连接更为稠密。在图7给出了 Λ_{prior} 的计算过程示意图，如前文所说，Λ_{prior} 会作为先验信息加到后续优化的信息矩阵中。显然，最初的信息矩阵 $\mathbf{x_{m1}}, \mathbf{x_{m2}}, \mathbf{x_{m3}}$ 对应的信息矩阵是对角矩阵，它是稀疏的，而 Λ_{prior} 中 $\mathbf{x_{m1}}, \mathbf{x_{m2}}, \mathbf{x_{m3}}$ 对应的信息矩阵却不是对角的，对它的逆的求解的求解的时间复杂度远大于对对角矩阵求逆的时间复杂度。也就是说边缘化会导致信息矩阵变得更为稠密。在实际操作中，为了维持信息矩阵的稀疏性，通常需要用一些策略丢弃掉一些信息，例如，可以只边缘化相机位姿，对于特征点的信息矩阵全部丢弃。

参考文献

[1] DONG-SI T C, MOURIKIS A I. Consistency analysis for sliding-window visual odometry[C]//2012 IEEE International Conference on Robotics and Automation. .[S.l.]: [s.n.] , 2012:5202–5209.

[2] DONG-SI T C, MOURIKIS A I. Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis[C]//2011 IEEE International Conference on Robotics and Automation. .[S.l.]: [s.n.] , 2011:5655–5662.

[3] LEUTENEGGER S, LYNEN S, BOSSE M, et al. Keyframe-based visual–inertial odometry using nonlinear optimization[J]. The International Journal of Robotics Research, 2015, 34(3):314–334.

[4] HUANG G P, MOURIKIS A I, ROUMELIOTIS S I. A first-estimates Jacobian EKF for improving SLAM consistency[C]//Experimental Robotics. .[S.l.]: [s.n.] , 2009:373–382.

[5] QIN T, LI P, SHEN S. Vins-mono: A robust and versatile monocular visual-inertial state estimator[J]. IEEE Transactions on Robotics, 2018, 34(4):1004–1020.