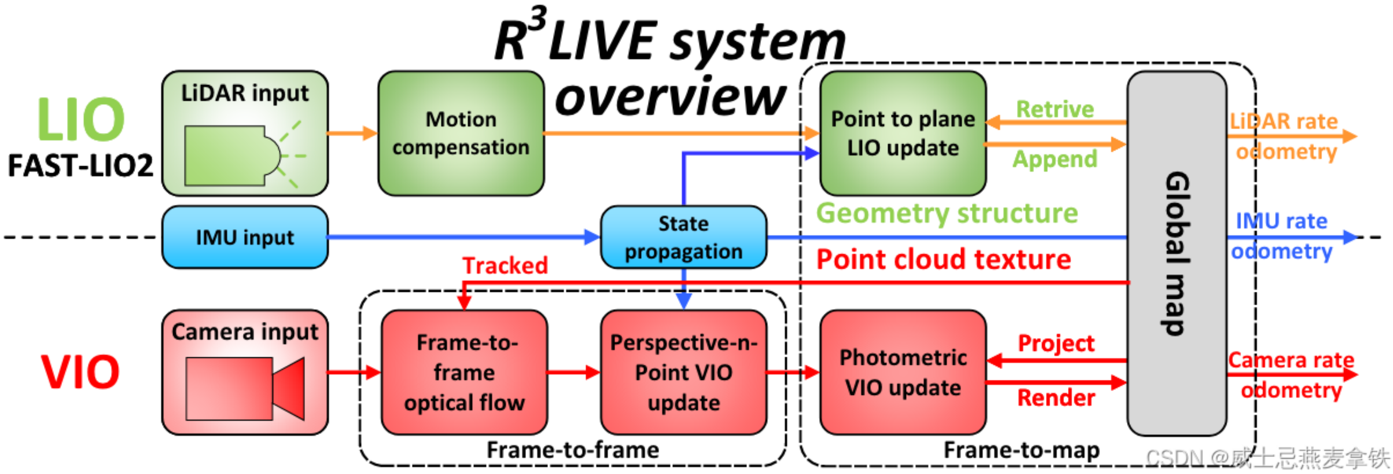


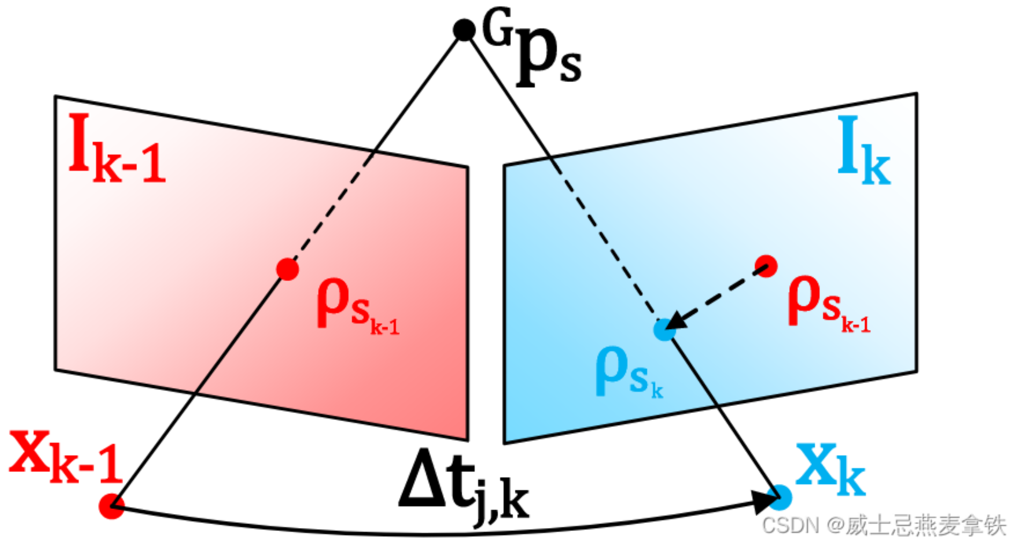
VIO子系统通过最小化帧到帧的PnP重投影误差和最小化帧到全局地图的光度误差来估计系统的状态，并渲染全局地图的纹理。 具体过程如下：

1. 在帧与帧之间，通过Lucas-Kanade光流法跟踪一定数量的地图点，并通过最小化这些点的PnP重投影误差来估计系统状态。系统估计通过ESIKF框架进行迭代求解，与高斯牛顿法（GN）等价。
2. 在帧与全局地图之间，将全局地图中被跟踪的点投影到当前图像帧中，然后最小化这些点的帧与地图之间的光度误差，从而更精确的估计系统状态。同样的，系统估计通过ESIKF框架进行迭代求解。
3. 在帧到全局地图的VIO更新之后，可以得到当前图像帧的精确位姿，然后通过对地图点在当前图像帧上对应的像素和相邻像素的RGB颜色进行线性插值，从而渲染地图点的颜色。

1 帧到帧的VIO

假设我们在上一帧图像帧 I k − 1 \mathbf{I}_{k-1} Ik−1 中追踪了 m m m 个地图点，这 m m m 个地图点记为 P = { P 1 , … , P m } \mathcal{P}=\left\{\mathbf{P}_{1}, \ldots, \mathbf{P}_{m}\right\} P = {P1, …, Pm}，它们在图像帧 I k − 1 \mathbf{I}_{k-1} Ik−1 中的投影坐标记为 { ρ 1 k − 1 , … , ρ m k − 1 } \left\{\boldsymbol{\rho}_{1_{k-1}}, \ldots, \boldsymbol{\rho}_{m_{k-1}}\right\} {ρ1k−1, …, ρmk−1}，使用 **Lucas-Kanade光流法** 可以找出这些地图点在当前图像帧 I k \mathbf{I}_{k} Ik 的投影坐标，记为 { ρ 1 k , … , ρ m k } \left\{\boldsymbol{\rho}_{1_{k}}, \ldots, \boldsymbol{\rho}_{m_{k}}\right\} {ρ1k, …, ρmk}。然后我们通过 **ESIKF** 来计算和优化这些地图点的重投影误差，从而估计出系统的状态。

1.1 PnP重投影误差



如上图所示，以第 s s s 个点 P s = [G p s T , c s T] T ∈ P \mathbf{P}_{s}=\left[{ }^{G} \mathbf{p}_{s}^{T}, \mathbf{c}_{s}^{T}\right]^{T} \in \mathcal{P} Ps = [GpsT, csT]T ∈ P 为例，其中前面的3维向量 G p s T = [G p s x , G p s y , G p s z] { }^{G} \mathbf{p}_{s}^{T}= \left[{ }^{G} \mathbf{p_s}_{x},{ }^{G} \mathbf{p_s}_{y},{ }^{G} \mathbf{p_s}_{z}\right] GpsT = [Gpsx, Gpsy, Gpsz] 表示点 s s s 在世界坐标系下的三维坐标，后面的 c s T = [c s r , c s g , c s b] T \mathbf{c}_{s}^{T}=\left[\mathbf{c_s}_{r}, \mathbf{c_s}_{g}, \mathbf{c_s}_{b}\right]^{T} csT = [csr, csg, csb]T表示点 s s s 的RGB颜色。

为了计算点 s s s 的在图像帧 I k \mathbf{I}_{k} Ik 中的重投影误差，我们先要将点 s s s 转换到相机坐标系当中，即：

C p s = [C p x s , C p y s , C p z s] T = (G R ˇ I k ⋅ I R ˇ C k) T ⋅ G p s − I R ˇ C k T ⋅ I p ˇ C k − (G R ˇ I k ⋅ I R ˇ C k) T ⋅ G p ˇ I k (1) \begin{array}{r}{ }^{C} \mathbf{p}_{s}=\left[{ }^{C} \mathbf{p}_{x_{s}},{ }^{C} \mathbf{p}_{y_{s}},{ }^{C} \mathbf{p}_{z_{s}}\right]^{T}=\left({ }^{G} \check{\mathbf{R}}_{I_{k}} \cdot{ }^{I} \check{\mathbf{R}}_{C_{k}}\right)^{T} \cdot{ }^{G} \mathbf{p}_{s} -{ }^{I} \check{\mathbf{R}}_{C_{k}}^{T} \cdot{ }^{I} \check{\mathbf{p}}_{C_{k}}-\left({ }^{G} \check{\mathbf{R}}_{I_{k}} \cdot{ }^{I} \check{\mathbf{R}}_{C_{k}}\right)^{T} \cdot{ }^{G} \check{\mathbf{p}}_{I_{k}}\end{array}\tag{1}
Cps = [Cpxs, Cpys, Cpzs]T = (GRˇIk ⋅ IRˇCk)T ⋅ Gps − IRˇCkT ⋅ Ipˇ Ck − (GRˇIk ⋅ IRˇCk)T ⋅ GpˇIk(1)

这里要注意的是， C k p G ≠ − G p C k {}^{C_k}p_G \neq -{}^Gp_{C_k} CkpG ≠ −GpCk，而是 C k p G = − C k R G G p C k {}^{C_k}p_G = -{}^{C_k}R_{G}{}^Gp_{C_k} CkpG = −CkRGGpCk。

将点 s s s 的重投影误差记为 r (x ˇ k , ρ s k , G p s) \mathbf{r}\left(\check{\mathbf{x}}_{k}, \boldsymbol{\rho}_{s}^{k},{ }^{G} \mathbf{p}_{s}\right) r(xˇk, ρsk, Gps)，则有：

r (x ˇ k , ρ s k , G p s) = ρ s k − π (C p s , x ˇ k) (2) \mathbf{r}\left(\check{\mathbf{x}}_{k}, \boldsymbol{\rho}_{s_{k}},{ }^{G} \mathbf{p}_{s}\right)=\boldsymbol{\rho}_{s_{k}}-\boldsymbol{\pi}\left({ }^{C} \mathbf{p}_{s}, \check{\mathbf{x}}_{k}\right)\tag{2} r(xˇk, ρsk, Gps) = ρsk − π(Cps, xˇk)(2)

其中， x ˇ k \check{\mathbf{x}}_{k} xˇk 是每一次ESIKF迭代的当前状态， π (C p s , x ˇ k) ∈ R 2 \boldsymbol{\pi}\left({ }^{C} \mathbf{p}_{s}, \check{\mathbf{x}}_{k}\right) \in \mathbb{R}^{2} π(Cps, xˇk) ∈ R2 是相机投影模型，可以通过下式计算：

$$\boldsymbol{\pi}\left({}^{C}\mathbf{p}_{s}, \check{\mathbf{x}}_{k}\right)=\left[\check{f}_{x_{k}} \frac{{}^{C}\mathbf{p}_{x_{s}}}{{}^{C}\mathbf{p}_{z_{s}}}+\check{c}_{x_{k}}, \check{f}_{y_{k}} \frac{{}^{C}\mathbf{p}_{y_{s}}}{{}^{C}\mathbf{p}_{z_{s}}}+\check{c}_{y_{k}}\right]^{T}+\frac{\check{t}_{C_{k}}}{\Delta t_{k-1, k}}\left(\boldsymbol{\rho}_{s_{k}}-\boldsymbol{\rho}_{s_{k-1}}\right)\tag{3}$$

其中，$\Delta t_{k-1, k}$ 是上一帧图像帧 $\mathbf{I}_{k-1}$ 和当前帧图像帧 $\mathbf{I}_{k}$ 之间的时间间隔。上式等式右边第一个式子是针孔相机投影模型，第二个式子是在线时间校正因子。

误差公式 (2) 中的测量噪声包含两个部分：一个是 ${}^{G}\mathbf{p}_{s}$ 中的地图点位置误差，另一个是 $\boldsymbol{\rho}_{s_{k}}$ 中的像素跟踪误差。

$${}^{G}\mathbf{p}_{s}={}^{G}\mathbf{p}_{s}^{\mathrm{gt}}+\mathbf{n}_{\mathbf{p}_{s}}, \mathbf{n}_{\mathbf{p}_{s}} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{n}_{\mathbf{p}_{s}}}\right)\tag{4}$$

$$\boldsymbol{\rho}_{s_{k}}=\boldsymbol{\rho}_{s_{k}}^{\mathrm{gt}}+\mathbf{n}_{\boldsymbol{\rho}_{s_{k}}}, \mathbf{n}_{\rho_{s_{k}}} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{n}_{\rho_{s_{k}}}}\right)\tag{5}$$

其中，${}^{G}\mathbf{p}_{s}^{\mathrm{gt}}$ 和 $\rho_{s_{k}}^{\mathrm{gt}}$ 分别为为 ${}^{G}\mathbf{p}_{s}$ 和 $\boldsymbol{\rho}_{s_{k}}$ 的真值。

然后，我们得到零残差真值 $\mathbf{r}\left(\mathbf{x}_{k}, \boldsymbol{\rho}_{s_{k}}^{\mathrm{gt}}, {}^G\mathbf{p}_{s}^{\mathrm{gt}}\right)$ 的一阶泰勒展开为：

$$\mathbf{0}=\mathbf{r}\left(\mathbf{x}_{k}, \boldsymbol{\rho}_{s_{k}}^{\mathrm{gt}}, {}^{G}\mathbf{p}_{s}^{\mathrm{gt}}\right) \approx \mathbf{r}\left(\check{\mathbf{x}}_{k}, \boldsymbol{\rho}_{s_{k}}, {}^{G}\mathbf{p}_{s}\right)+\mathbf{H}_{s}^{r} \delta \check{\mathbf{x}}_{k}+\boldsymbol{\alpha}_{s}\tag{6}$$

其中：$\boldsymbol{\alpha}_{s} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}_{s}}\right)$，并且有：

$$\mathbf{H}_{s}^{r}=\left.\frac{\partial \mathbf{r}_{c}\left(\check{\mathbf{x}}_{k} \boxplus \delta \check{\mathbf{x}}_{k}, \boldsymbol{\rho}_{s_{k}}, {}^{G}\mathbf{p}_{s}\right)}{\partial \delta \check{\mathbf{x}}_{k}}\right|_{\delta \check{\mathbf{x}}_{k}=\mathbf{0}}\tag{7}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}_{s}}=\boldsymbol{\Sigma}_{\mathbf{n}_{\rho_{s_{k}}}}+\mathbf{F}_{\mathbf{p}_{s}}^{r} \boldsymbol{\Sigma}_{\mathbf{p}_{s}} \mathbf{F}_{\mathbf{p}_{s}}^{r T}\tag{8}$$

$$\mathbf{F}_{\mathbf{p}_{s}}^{r}=\frac{\partial \mathbf{r}\left(\check{\mathbf{x}}_{k}, \boldsymbol{\rho}_{s_{k}}, {}^{G}\mathbf{p}_{s}\right)}{\partial \mathbf{p}_{s}}\tag{9}$$

1.2 帧到帧的VIO ESIKF更新

公式（6）表示了 $\mathbf{x}_{k}$ 的观测分布，它可以可以与IMU传播的先验分布相结合，从而得到误差状态 $\delta \check{\mathbf{x}}_{k}$ 的最大后验估计（MAP）：

$$\min_{\delta \check{\mathbf{x}}_{k}}\left(\left\|\check{\mathbf{x}}_{k} \boxminus \hat{\mathbf{x}}_{k}+\boldsymbol{\mathcal{H}} \delta \check{\mathbf{x}}_{k}\right\|_{\boldsymbol{\Sigma}_{\delta \hat{\mathbf{x}}_{k}}}^{2}+\sum_{s=1}^{m}\left\|\mathbf{r}\left(\check{\mathbf{x}}_{k}, \boldsymbol{\rho}_{s_{k}}, {}^{G}\mathbf{p}_{s}\right)+\mathbf{H}_{s}^{r} \delta \check{\mathbf{x}}_{k}\right\|_{\boldsymbol{\Sigma}_{\boldsymbol{\alpha}_{s}}}^{2}\right)\tag{10}$$

其中 $\|\mathbf{x}\|_{\boldsymbol{\Sigma}}^{2}=\mathbf{x}^{T} \boldsymbol{\Sigma}^{-1} \mathbf{x}$ 表示马氏距离的平方，$\hat{\mathbf{x}}_{k}$ 是IMU传播的状态估计，$\Sigma_{\delta \hat{\mathbf{x}}_{k}}$ 是IMU传播状态的协方差，$\mathcal{H}$ 是从 $\hat{\mathbf{x}}_{k}$ 的切平面投影到 $\check{\mathbf{x}}_{k}$ 的切平面时的状态误差的雅可比矩阵。

其中：

$$\mathbf{H}=\left[\mathbf{H}_{1}^{r T}, \ldots, \mathbf{H}_{m}^{r T}\right]^{T}\tag{11}$$

$$\mathbf{R}=\operatorname{diag}\left(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}_{1}}, \ldots, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}_{m}}\right)\tag{12}$$

$$\check{\mathbf{z}}_{k}=\left[\mathbf{r}\left(\check{\mathbf{x}}_{k}, \boldsymbol{\rho}_{1_{k}}, {}^{G}\mathbf{p}_{1}\right) \ldots, \mathbf{r}\left(\check{\mathbf{x}}_{k}, \boldsymbol{\rho}_{m_{k}}, {}^{G}\mathbf{p}_{m}\right)\right]^{T}\tag{13}$$

$$\mathbf{P}=(\boldsymbol{H})^{-1} \boldsymbol{\Sigma}_{\delta \hat{\mathbf{x}}_{k}}(\boldsymbol{H})^{-T}\tag{14}$$

Kalman增益可以通过下式计算：

$$\mathbf{K}=\left(\mathbf{H}^{T} \mathbf{R}^{-1} \mathbf{H}+\mathbf{P}^{-1}\right)^{-1} \mathbf{H}^{T} \mathbf{R}^{-1}\tag{15}$$

从而，我们可以通过下式进行误差状态更新：

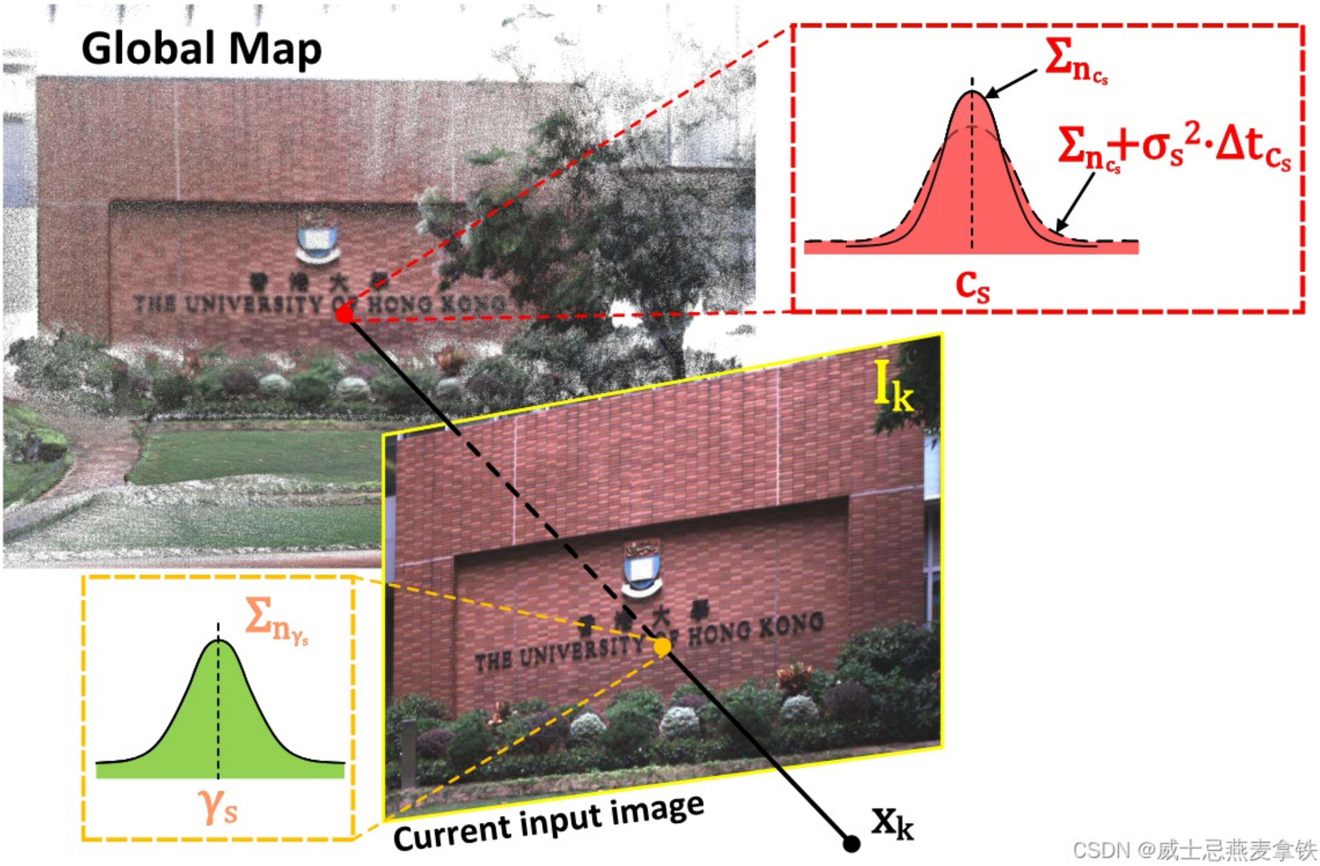
$$\check{\mathbf{x}}_{k}=\check{\mathbf{x}}_{k} \boxplus\left(-\mathbf{K}\check{\mathbf{z}}_{k}-(\mathbf{I}-\mathbf{K}\mathbf{H})(\mathcal{H})^{-1}\left(\check{\mathbf{x}}_{k} \boxminus \hat{\mathbf{x}}_{k}\right)\right)\tag{16}$$

迭代上式直到收敛（比如状态更新值小于给定的阈值）。**注意，该迭代卡尔曼滤波方法与GN高斯牛顿优化方法等价。**

2 帧到地图的VIO

2.1 帧到地图光度更新

在帧到帧的VIO更新之后，我们对系统状态 $\check{\mathbf{x}}_{k}$ 有了良好的估计，然后我们通过最小化跟踪点的光度误差来进行帧到地图的VIO更新，从而降低漂移。



如上图所示，以第 s s s 个点 P s = [G p s T , c s T] T ∈ P \mathbf{P}_{s}=\left[{ }^{G} \mathbf{p}_{s}^{T}, \mathbf{c}_{s}^{T}\right]^{T} \in \mathcal{P} Ps = [GpsT, csT]T ∈ P 为例，其帧到地图的光度误差 o (x ˇ k , G p s , C s) \mathbf{o}\left(\check{\mathbf{x}}_{k},{ }^{G} \mathbf{p}_{s}, \mathbf{C}_{s}\right) o(xˇk, Gps, Cs) 可以通过下式计算：

o (x ˇ k , G p s , c s) = c s − γ s (17) \mathbf{o}\left(\check{\mathbf{x}}_{k},{ }^{G} \mathbf{p}_{s}, \mathbf{c}_{s}\right)=\mathbf{c}_{s}-\gamma_{s}\tag{17} o(xˇk, Gps, cs) = cs − γs(17)

其中， c s \mathbf{c}_{s} cs 是保存在全局地图中的颜色， γ s \gamma_{s} γs 是当前图像帧 I k \mathbf{I}_{k} Ik 中观察到的颜色。为了得到观测到的颜色 γ s \gamma_{s} γs 及其协方差矩阵 Σ n γ s \Sigma_{\mathbf{n}_{\gamma_{s}}} Σnγs，我们预测点 s s s 在当前图像帧 I k \mathbf{I}_{k} Ik 上的坐标为 ρ ~ s k = π (C p s , x ˇ k) \tilde{\rho}_{s_{k}}=\pi\left({ }^{C} \mathbf{p}_{s}, \check{\mathbf{x}}_{k}\right) ρ~sk = π(Cps, xˇk)，然后线性插值计算得到相邻像素的RGB颜色。

我们同样考虑 γ s \gamma_{s} γs 和 c s \mathbf{c}_{s} cs 的测量误差：

γ s = γ s g t + n γ s , n γ s ∼ N (0 , Σ n γ s) (18) \gamma_{s}=\boldsymbol{\gamma}_{s}^{g t}+\mathbf{n}_{\boldsymbol{\gamma}_{s}}, \mathbf{n}_{\boldsymbol{\gamma}_{s}} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{n}_{\gamma_{s}}}\right)\tag{18} γs = γsgt + nγs, nγs ∼ N (0, Σnγs)(18)

c s = c s g t + n c s + η c s , n c s ∼ N (0 , Σ n c s) η c s ∼ N (0 , σ s 2 ⋅ Δ t c s) (19) \begin{gathered}\mathbf{c}_{s}=\mathbf{c}_{s}^{g t}+\mathbf{n}_{\mathbf{c}_{s}}+\boldsymbol{\eta}_{\mathbf{c}_{s}}, \\\mathbf{n}_{\mathbf{c}_{s}} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{n}_{\mathbf{c}_{s}}}\right) \\\boldsymbol{\eta}_{\mathbf{c}_{s}} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\sigma}_{s}^{2} \cdot \Delta t_{\mathbf{c}_{s}}\right)\end{gathered}\tag{19} cs = csgt + ncs + ηcs, ncs ∼ N (0, Σncs)ηcs ∼ N (0, σs2 ⋅ Δtcs)(19)

其中， γ s g t \gamma_{s}^{g t} γsgt 是 γ s \gamma_{s} γs 的真值， c s g t \mathbf{c}_{s}^{g t} csgt 是 c s \mathbf{c}_{s} cs 的真值， Δ t c s \Delta t_{\mathbf{c}_{s}} Δtcs 是当前帧和 P s \mathbf{P}_{s} Ps 上次渲染时的时间间隔。

在公式（19）中， c s \mathbf{c}_{s} cs 的测量噪声由两部分组成：上次渲染的估计误差 n c s \mathbf{n}_{\mathbf{c}_{s}} ncs 和脉冲随机游走过程噪声 η c s \boldsymbol{\eta}_{\mathbf{c}_{s}} ηcs（它解释了环境光照的变化）。

结合公式(17)(18)(19)，我们得到**残差零真值 o (x k , G p s g t , c s g t) \mathbf{o}\left(\mathbf{x}_{k},{ }^{G} \mathbf{p}_{s}^{g t}, \mathbf{c}_{s}^{g t}\right) o(xk, Gpsgt, csgt)** 的一阶泰勒展开：

0 = o (x k , G p s g t , c s g t) ≈ o (x ˇ k , G p s , c s) + H s o δ x ˇ k + β s (20) \mathbf{0}=\mathbf{o}\left(\mathbf{x}_{k},{ }^{G} \mathbf{p}_{s}^{g t}, \mathbf{c}_{s}^{g t}\right) \approx \mathbf{o}\left(\check{\mathbf{x}}_{k},{ }^{G} \mathbf{p}_{s}, \mathbf{c}_{s}\right)+\mathbf{H}_{s}^{o} \delta \check{\mathbf{x}}_{k}+\boldsymbol{\beta}_{s}\tag{20} 0 = o(xk, Gpsgt, csgt) ≈ o(xˇk, Gps, cs) + Hsoδxˇk + βs(20)

其中：

H s o = ∂ o (x ˇ k ⊞ δ x ˇ k , G p s , c s) ∂ δ x ˇ k ∣ δ x ˇ k = 0 (21) \mathbf{H}_{s}^{o}=\left.\frac{\partial \mathbf{o}\left(\check{\mathbf{x}}_{k} \boxplus \delta \check{\mathbf{x}}_{k},{ }^{G} \mathbf{p}_{s}, \mathbf{c}_{s}\right)}{\partial \delta \check{\mathbf{x}}_{k}}\right|_{\delta \check{\mathbf{x}}_{k}=\mathbf{0}}\tag{21} Hso = ∂δxˇk∂o(xˇk ⊞ δxˇk, Gps, cs) ∣ ∣ δxˇk=0(21)

Σ β s = Σ n c s + σ s 2 ⋅ Δ t c s + Σ n γ s + F p s o Σ p s F p s o T (22) \boldsymbol{\Sigma}_{\boldsymbol{\beta}_{s}}=\boldsymbol{\Sigma}_{\mathbf{n}_{\mathbf{c}_{s}}}+\boldsymbol{\sigma}_{s}^{2} \cdot \Delta t_{\mathbf{c}_{s}}+\boldsymbol{\Sigma}_{\mathbf{n}_{\gamma_{s}}}+\mathbf{F}_{\mathbf{p}_{s}}^{o} \boldsymbol{\Sigma}_{\mathbf{p}_{s}} \mathbf{F}_{\mathbf{p}_{s}}^{o T}\tag{22} Σβs = Σncs + σs2 ⋅ Δtcs + Σnγs + FpsoΣpsFpsoT(22)

F p s o = ∂ o (x ˇ k , G p s , c s) ∂ G p s (23) \mathbf{F}_{\mathbf{p}_{s}}^{o}=\frac{\partial \mathbf{o}\left(\check{\mathbf{x}}_{k},{ }^{G} \mathbf{p}_{s}, \mathbf{c}_{s}\right)}{\partial{ }^{G} \mathbf{p}_{s}}\tag{23} Fpso = ∂Gps∂o(xˇk, Gps, cs)(23)

2.2 帧到地图的VIO ESIKF更新

公式（20）构成了 δ x ˇ k \delta \check{\mathbf{x}}_{k} δxˇk 的另一个观测分布，它与IMU传播的先验分布相结合，得到了 δ x ˇ k \delta \check{\mathbf{x}}_{k} δxˇk 的最大后验(MAP)估计：

min⁡δ x ˇ k (∥ x ˇ k ⊟ x ^ k + H δ x ˇ k ∥ Σ δ x ^ k 2 + ∑ s = 1 m ∥ o (x ˇ k , G p s , c s) + H s o δ x ˇ k ∥ Σ β s 2) (24) \begin{aligned}\min _{\delta

\check{\mathbf{x}}_{k}}\left(\left\|\check{\mathbf{x}}_{k} \boxminus \hat{\mathbf{x}}_{k}+\mathcal{H} \delta \check{\mathbf{x}}_{k}\right\|_{\boldsymbol{\Sigma}_{\delta \hat{\mathbf{x}}_{k}}^{2}+\sum_{s=1}^{m}\left\|\mathbf{o}\left(\check{\mathbf{x}}_{k},{}^{G} \mathbf{p}_{s}, \mathbf{c}_{s}\right)+\mathbf{H}_{s}^{o} \delta \check{\mathbf{x}}_{k}\right\|_{\boldsymbol{\Sigma}_{\boldsymbol{\beta}_{s}}}^{2}\right)\end{aligned}\tag{24} δxˇkmin(∥xˇk⊟x^k+Hδxˇk∥Σδx^k2+s=1∑m∥ ∥o(xˇk,Gps,cs)+Hsoδxˇk∥ ∥Σβs2)(24)

其中：

H = [H 1 o T , … , H m o T] T (25) \mathbf{H}=\left[\mathbf{H}_{1}^{o T}, \ldots, \mathbf{H}_{m}^{o} T\right]^{T}\tag{25} H = [H1oT, … , HmoT]T(25)

R = diag(Σ β 1 , … , Σ β m) (26) \mathbf{R}=\operatorname{diag}\left(\boldsymbol{\Sigma}_{\boldsymbol{\beta}_{1}}, \ldots, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_{m}}\right)\tag{26} R = diag(Σβ1, … , Σβm)(26)

z ˇ k = [o (x ˇ k , G p 1 , c 1) … , o (x ˇ k , G p m , c m)] T (27) \check{\mathbf{z}}_{k}=\left[\mathbf{o}\left(\check{\mathbf{x}}_{k},{}^{G} \mathbf{p}_{1}, \mathbf{c}_{1}\right) \ldots, \mathbf{o}\left(\check{\mathbf{x}}_{k},{}^{G} \mathbf{p}_{m}, \mathbf{c}_{m}\right)\right]^{T}\tag{27} zˇk = [o(xˇk, Gp1, c1) … , o(xˇk, Gpm, cm)]T(27)

P = (H) − 1 Σ δ x ^ k (H) − T (28) \mathbf{P}=(\mathcal{H})^{-1} \boldsymbol{\Sigma}_{\delta \hat{\mathbf{x}}_{k}}(\mathcal{H})^{-T}\tag{28} P = (H)−1Σδx^k(H)−T(28)

然后，我们执行类似于公式（15）和（16）的状态更新：

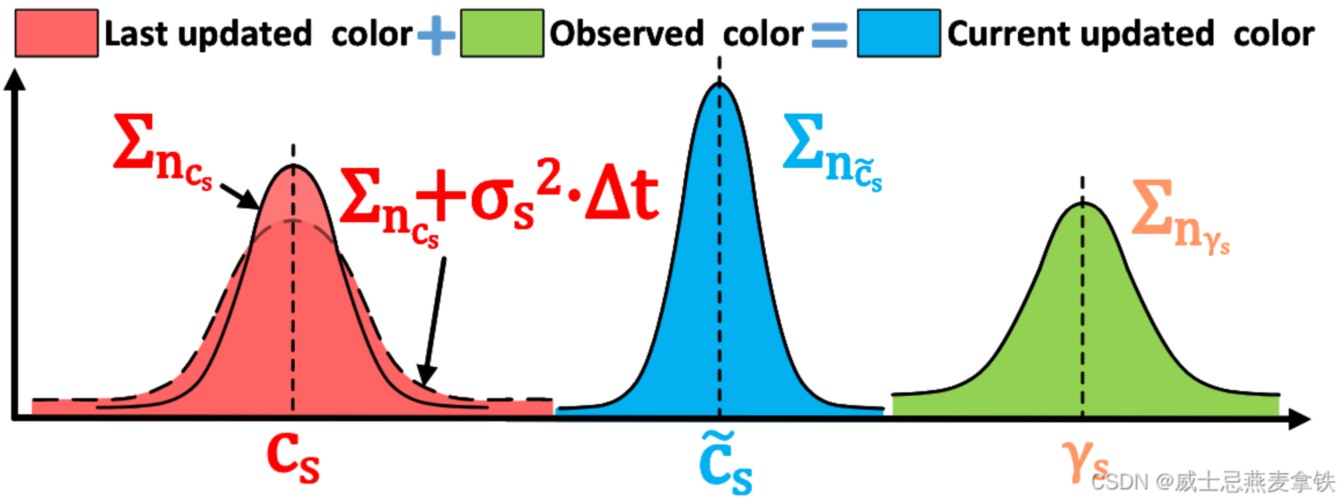
x ^ k = x ˇ k , Σ δ x ^ k = (I − K H) Σ δ x ˇ k (29) \hat{\mathbf{x}}_{k}=\check{\mathbf{x}}_{k}, \quad \boldsymbol{\Sigma}_{\delta \hat{\mathbf{x}}_{k}}=(\mathbf{I}-\mathbf{K} \mathbf{H}) \boldsymbol{\Sigma}_{\delta \check{\mathbf{x}}_{k}} \tag{29} x^k = xˇk, Σδx^k = (I − KH)Σδxˇk(29)

使用 ESIKF 对帧到地图的VIO 迭代更新直到收敛，然后使用收敛的状态估计值去：

1. 渲染地图的纹理
2. 更新当前跟踪点集合 P \mathcal{P} P ，给下一帧使用
3. 在下一帧LIO或者VIO更新中，作为IMU传播的起点

2.3 渲染全局地图的纹理

在帧到地图的VIO更新之后，我们得到了当前图像帧的精确位姿，然后我们执行渲染功能来更新地图点的颜色。



首先，我们在所有激活的体素中的检索所有点。假设总共有 n n n 个点，记为 ζ = { P 1 , … , P n } \zeta=\left\{\mathbf{P}_{1}, \ldots, \mathbf{P}_{n}\right\} ζ = {P1, … , Pn}。以第 s s s 个点 P s = [G p s T , c s T] T ∈ ζ \mathbf{P}_{s}=\left[{ }^{G} \mathbf{p}_{s}^{T}, \mathbf{c}_{s}^{T}\right]^{T} \in \zeta Ps = [GpsT, csT]T ∈ ζ 的颜色更新过程为例。如果点 P s \mathbf{P}_{s} Ps 落在当前图像帧 I k \mathbf{I}_{k} Ik 中，我们通过对当前图像帧 I k \mathbf{I}_{k} Ik 上相邻像素的RGB颜色值进行线性插值，得到其观测颜色 γ s \gamma_{s} γs 和协方差 Σ n γ s \boldsymbol{\Sigma}_{\mathbf{n}_{\gamma_{s}}} Σnγs。通过贝叶斯更新，图像上新观测到的点的颜色与记录在地图中的现有颜色值 c s \mathbf{c}_{s} cs 进行融合（如上图所示），得到更新后的颜色值和 c s \mathbf{c}_{s} cs 的协方差：

Σ n c ~ s = ((Σ n c s + σ s 2 ⋅ Δ t c s) − 1 + Σ n γ s − 1) − 1 (30) \boldsymbol{\Sigma}_{\mathbf{n}_{\tilde{\mathbf{c}}_{s}}}=\left(\left(\boldsymbol{\Sigma}_{\mathbf{n}_{\mathbf{c}_{s}}}+\boldsymbol{\sigma}_{s}^{2} \cdot \Delta t_{\mathbf{c}_{s}}\right)^{-1}+\boldsymbol{\Sigma}_{\mathbf{n}_{\boldsymbol{\gamma}_{s}}}^{-1}\right)^{-1}\tag{30} Σnc~s = ((Σncs + σs2 ⋅ Δtcs)−1 + Σnγs−1)−1(30)

c ~ s = ((Σ n c s + σ s 2 ⋅ Δ t c s) − 1 c s + Σ n γ s − 1 γ s) − 1 Σ n c ~ s (31) \tilde{\mathbf{c}}_{s}=\left(\left(\boldsymbol{\Sigma}_{\mathbf{n}_{\mathbf{c}_{s}}}+\boldsymbol{\sigma}_{s}^{2} \cdot \Delta t_{\mathbf{c}_{s}}\right)^{-1} \mathbf{c}_{s}+\boldsymbol{\Sigma}_{\mathbf{n}_{\gamma_{s}}}^{-1} \gamma_{s}\right)^{-1} \boldsymbol{\Sigma}_{\mathbf{n}_{\tilde{\mathbf{c}}_{s}}}\tag{31} c~s = ((Σncs + σs2 ⋅ Δtcs)−1 cs + Σnγs−1γs)−1 Σnc~s(31)

c s = c ~ s , Σ n c s = Σ n c ~ s (32) \mathbf{c}_{s}=\tilde{\mathbf{c}}_{s}, \quad \mathbf{\Sigma}_{\mathbf{n}_{\mathbf{c}_{s}}}=\boldsymbol{\Sigma}_{\mathbf{n}_{\tilde{\mathbf{c}}_{s}}}\tag{32} cs = c~s, Σncs = Σnc~s(32)

2.4 VIO子系统的跟踪点更新

在纹理渲染完成后，我们对所跟踪的点集 P \mathcal{P} P 进行更新：

- **删除点**：如果点集 P \mathcal{P} P 的点通过公式（2）计算出来的PnP重投影误差或者通过公式（17）计算出来的光度误差太大，我们会从跟踪点的集合中删除这些点，我们还会删除重投影之后不属于当前图像帧 I k \mathbf{I}_{k} Ik 的点。
- **添加点**：我们将 ζ \zeta ζ 中的每个点投影到当前图像帧 I k \mathbf{I}_{k} Ik 上，如果附近没有其它跟踪点（例如，半径50像素内），则将其添加到点集 P \mathcal{P} P 中。