Кратко о Bundle Adjustment

Сергей Кривохатский

СПбГУ, Современное программирование

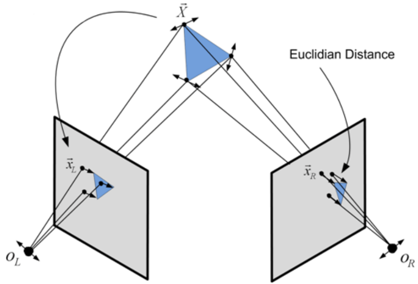
21 октября 2022 г.

Bundle Adjustment I

Оптимизируем позиции камеры и координаты 3D-точек

$$\sum_j \sum_{i \in V_j} \left\| \pi(r_j, t_j, X_i) - x_{ij} \right\|^2 \to \min$$

- r_j, t_j — позиция камеры в кадре j
- X_i — 3D-позиция i-й точки
- V_j — номера видимых в кадре j точек
- x_{ij} — 2D-позиция i-й точки в кадре j
- π — функция проекции



Bundle Adjustment II

Изменим обозначения. Нужно найти минимум суммы квадратов

$$\arg\min_{p} u(p)^\mathsf{T} u(p)$$

Оптимизируемые параметры запишем в один вектор

$$\begin{aligned} p &= \begin{pmatrix} p_1 & p_2 & \ldots & p_{6n+3m} \end{pmatrix}^\mathsf{T} = \\ &= \begin{pmatrix} r_1^\mathsf{T} & t_1^\mathsf{T} & \ldots & r_n^\mathsf{T} & t_n^\mathsf{T} & X_1^\mathsf{T} & \ldots X_m^\mathsf{T} \end{pmatrix}^\mathsf{T} \end{aligned}$$

где n — число кадров, m — число 3D-точек

Bundle Adjustment III

Невязки запишем сначала для первой 3D-точки (в порядке возрастания номера кадра, на которой одна видна), потом для второй и т. д.

$$u(p) = \begin{pmatrix} u_1 & u_2 & \dots & u_k \end{pmatrix}^\mathsf{T} =$$

$$= \begin{pmatrix} \pi_p(X_1) - x_{1i_{11}} \\ \pi_p(X_1) - x_{1i_{12}} \\ \dots \\ \pi_p(X_2) - x_{2i_{21}} \\ \pi_p(X_2) - x_{2i_{22}} \\ \dots \end{pmatrix}$$

Алгоритм Левенберга — Марквардта

Мы находимся в точке p. Чтобы найти следующую позицию $p + \delta$, нужно решить систему уравнений

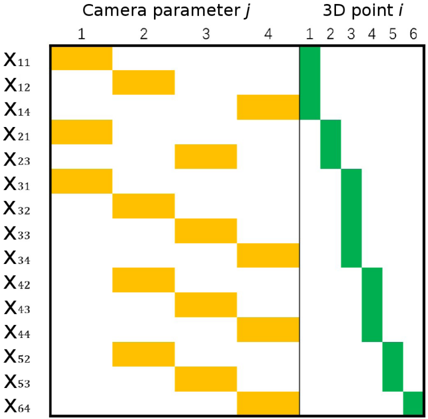
$$(J^\intercal J + \lambda \operatorname{diag} J^\intercal J)\delta = -J^\intercal u(p)$$

где J — матрица Якоби функции u в точке p

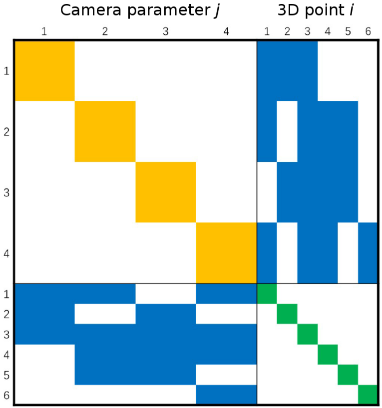
$$J = \begin{pmatrix} \dfrac{\partial u(p)}{\partial p_1} & \cdots & \dfrac{\partial u(p)}{\partial p_{6n}} & \cdots & \dfrac{\partial u(p)}{\partial p_{6n+3m}} \end{pmatrix} = \begin{pmatrix} J_c & J_x \end{pmatrix}$$

Структура матриц I

$$\boldsymbol{J} = \begin{pmatrix} \boldsymbol{J}_c & \boldsymbol{J}_x \end{pmatrix}$$



$$\boldsymbol{J}^\top \boldsymbol{J} = \begin{pmatrix} \boldsymbol{J}_c^\top \boldsymbol{J}_c & \boldsymbol{J}_c^\top \boldsymbol{J}_x \\ \boldsymbol{J}_x^\top \boldsymbol{J}_c & \boldsymbol{J}_x^\top \boldsymbol{J}_x \end{pmatrix}$$



Структура матриц II

Т. о. матрицу $\boldsymbol{J}^\intercal \boldsymbol{J} + \lambda \operatorname{diag} \boldsymbol{J}^\intercal \boldsymbol{J}$ можно разбить на блоки

$$\boldsymbol{J}^\intercal \boldsymbol{J} + \lambda \operatorname{diag} \boldsymbol{J}^\intercal \boldsymbol{J} = \begin{pmatrix} \boldsymbol{U} & \boldsymbol{W} \\ \boldsymbol{W}^\intercal & \boldsymbol{V} \end{pmatrix}$$

где

$$\boldsymbol{U} = \begin{pmatrix} \boldsymbol{U}_1 & & \\ & \ddots & \\ & & \boldsymbol{U}_n \end{pmatrix} \qquad \boldsymbol{V} = \begin{pmatrix} \boldsymbol{V}_1 & & \\ & \ddots & \\ & & \boldsymbol{V}_m \end{pmatrix}$$

$\boldsymbol{U}_j$ — матрицы 6×6, а $\boldsymbol{V}_i$ — матрицы 3×3

Дополнение Шура I

Обозначим $\begin{pmatrix}\delta_c^\intercal & \delta_x^\intercal\end{pmatrix}^\intercal := \delta$ и $\begin{pmatrix}g_c^\intercal & g_x^\intercal\end{pmatrix}^\intercal := \boldsymbol{J}^\intercal u(p)$, получаем

$$\begin{pmatrix}\boldsymbol{U} & \boldsymbol{W}\\ \boldsymbol{W}^\intercal & \boldsymbol{V}\end{pmatrix}\begin{pmatrix}\delta_c\\ \delta_x\end{pmatrix} = -\begin{pmatrix}g_c\\ g_x\end{pmatrix}$$

Перемножим левую часть и посмотрим, что получится

$$\begin{cases}\boldsymbol{U}\delta_c + \boldsymbol{W}\delta_x = -g_c\\ \boldsymbol{W}^\intercal\delta_c + \boldsymbol{V}\delta_x = -g_x\end{cases}$$

Выразим δ_x из второго уравнения и подставим в первое

$$\begin{cases}(\boldsymbol{U} - \boldsymbol{W}\boldsymbol{V}^{-1}\boldsymbol{W}^\intercal)\delta_c = \boldsymbol{W}\boldsymbol{V}^{-1}g_x - g_c\\ \delta_x = \boldsymbol{V}^{-1}(-g_x - \boldsymbol{W}^\intercal\delta_c)\end{cases}$$

Дополнение Шура II

Итого вместо большой $(6n + 3m \times 6n + 3m)$ системы

$$\begin{pmatrix} \boldsymbol{U} & \boldsymbol{W} \\ \boldsymbol{W}^\intercal & \boldsymbol{V} \end{pmatrix} \begin{pmatrix} \delta_c \\ \delta_x \end{pmatrix} = - \begin{pmatrix} g_c \\ g_x \end{pmatrix}$$

решаем систему меньше $(6n \times 6n)$, находим δ_c, затем считаем δ_x

$$\begin{cases} (\boldsymbol{U} - \boldsymbol{W}\boldsymbol{V}^{-1}\boldsymbol{W}^\intercal)\delta_c = \boldsymbol{W}\boldsymbol{V}^{-1}g_x - g_c \\ \delta_x = \boldsymbol{V}^{-1}(-g_x - \boldsymbol{W}^\intercal\delta_c) \end{cases}$$

нужно лишь уметь обращать матрицу $\boldsymbol{V}$: для этого достаточно обратить m матриц 3×3

Как решать СЛАУ в Bundle Adjustment

Выбор метода зависит от размера задачи

- не более пары сотен кадров
  - сначала стоит применить дополнение Шура
  - получившуюся систему обычно решают с помощью разложения Холецкого
- больше нескольких сотен кадров
  - целесообразнее использовать итерационные методы
  - дополнение Шура применять не стоит, поскольку оно может дать плохо разреженную матрицу

Спасибо за внимание!