Bundle Adjustment

Yuji Oyamada [1,2]

[1] HVRL, Keio University

[2] Chair for Computer Aided Medical Procedure (CAMP), Technische Universität München

March 23, 2012

Bundle adjustment is an algorithm that estimates all unknown parameters using all available observation.

1 Problem statement

Suppose m cameras observe a static scene/objects at same time and n feature points are totally detected. Our goal is to estimate the 3D positions of the points and camera parameters.

The 3D position of j-th feature point is represented by a 3D vector as

$$\boldsymbol{X}_j = [X_j, Y_j, Z_j]^{\mathbf{T}}, \ \ j = 1, \ldots, n.$$

A feature point $\boldsymbol{X}_j$ observed by i-th camera is represented by a 2D vector as $\boldsymbol{x}_{ij} = [x_{ij}, y_{ij}]^{\mathbf{T}}$. With a 3×4 matrix $\mathtt{P}_i$, $\boldsymbol{x}_{ij}$ is described as

$$\begin{bmatrix} \boldsymbol{x}_{ij} \\ 1 \end{bmatrix} \propto \mathtt{P}_i \begin{bmatrix} \boldsymbol{X}_j \\ 1 \end{bmatrix}, \tag{1}$$

where

$$\mathtt{P}_i \equiv \mathtt{K}_i \left[\mathtt{R}_i | \mathbf{t}_i\right]. \tag{2}$$

Here, $\mathtt{K}_i, \mathtt{R}_i, \mathbf{t}_i$ denote the intrinsic parameter, the rotation matrix and the translation vector respectively. Figure 1 shows the camera geometry of target scene.

Bundle adjustment estimates unknown camera parameters and 3D positions of feature points under following assumption. Let a vector $\boldsymbol{p}_i$ pack all unknown parameters of i-th camera. If we have ground truth $\{\boldsymbol{p}_i\}$ and $\{\boldsymbol{X}_j\}$, re-projected feature point

$$\begin{aligned} \hat{\boldsymbol{x}}_{ij} &= \begin{bmatrix} \hat{x}_{ij} & \hat{y}_{ij} \end{bmatrix}^{\mathbf{T}} \\ &= \begin{bmatrix} \mathrm{comp}(x; \boldsymbol{p}_i, \boldsymbol{X}_j) & \mathrm{comp}(y; \boldsymbol{p}_i, \boldsymbol{X}_j) \end{bmatrix}^{\mathbf{T}} \\ &= \begin{bmatrix} \frac{(\mathtt{P}_i)_1 [\boldsymbol{X}_j^{\mathbf{T}} \ \ 1]^{\mathbf{T}}}{(\mathtt{P}_i)_3 [\boldsymbol{X}_j^{\mathbf{T}} \ \ 1]^{\mathbf{T}}} & \frac{(\mathtt{P}_i)_3 [\boldsymbol{X}_j^{\mathbf{T}} \ \ 1]^{\mathbf{T}}}{(\mathtt{P}_i)_3 [\boldsymbol{X}_j^{\mathbf{T}} \ \ 1]^{\mathbf{T}}} \end{bmatrix}^{\mathbf{T}} \end{aligned}$$

should be same on the i-th image. Here, $((\mathtt{P}_i)_k$ denotes k-th row vector of $\mathtt{P}_i$) and its observation $\boldsymbol{x}_{ij}$ Thus, the difference of observation and this re-projection $||\boldsymbol{x}_{ij} - \hat{\boldsymbol{x}}_{ij}||_2^2$ should be slight, not zero due to observation error. Figure 2 illustrates the geometrical relationship.

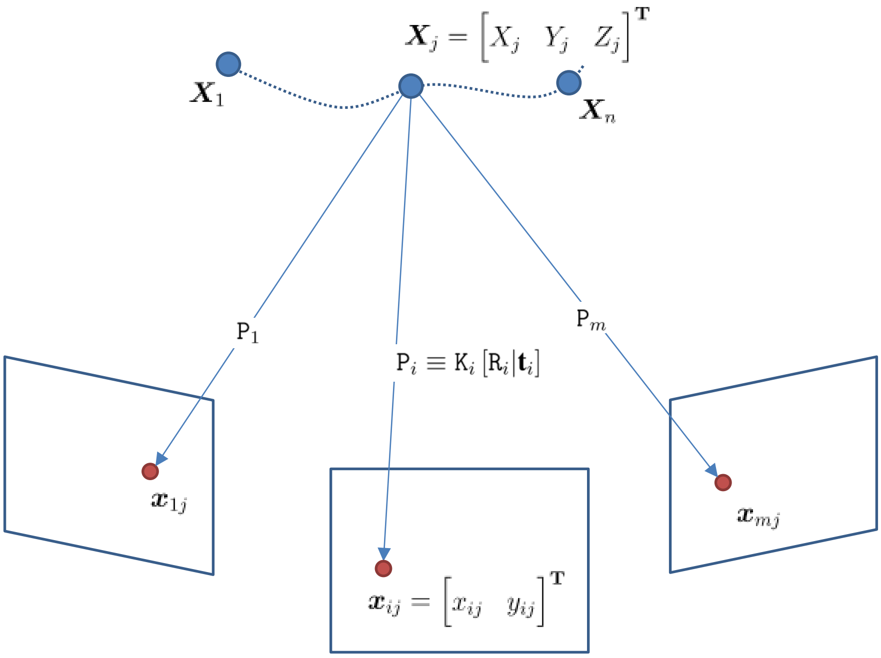

Figure 1: Camera Geometry

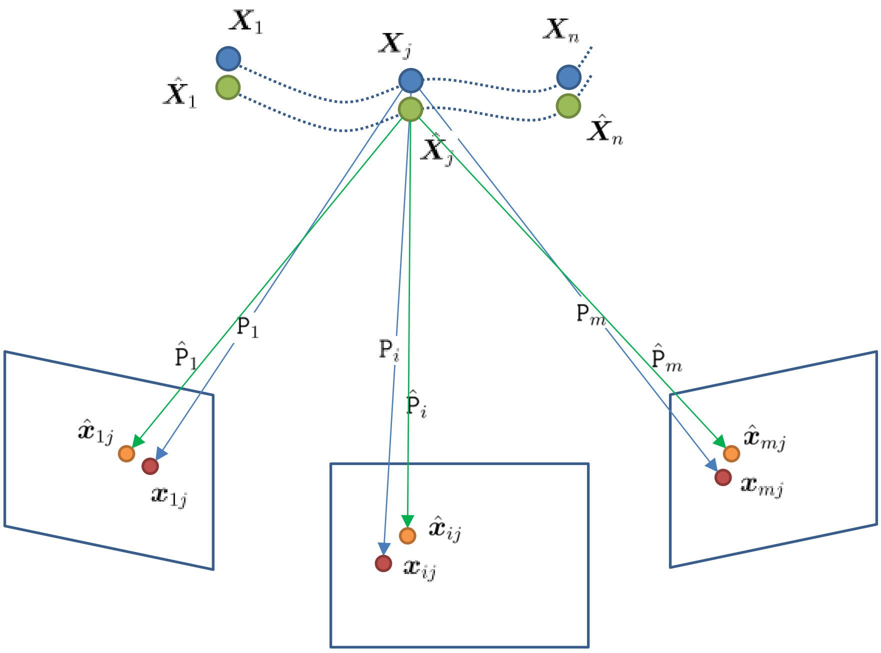

Figure 2: Re-projection: 2D point case.

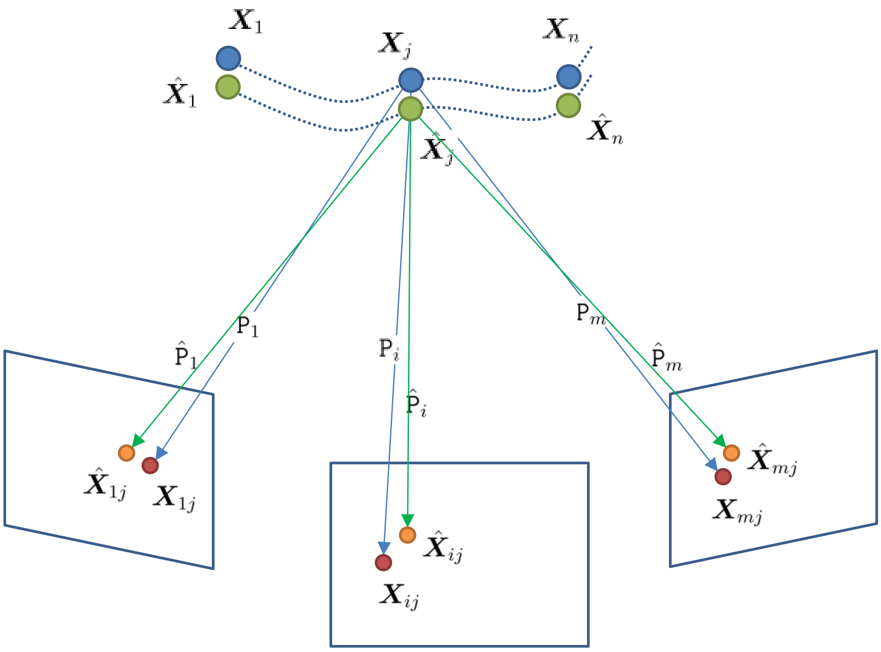

Figure 3: Re-projection: 3D point case.

Based on this assumption, Bundle adjustment estimates camera parameters and 3D position of features points by minimizing the following cost function:

$$
\begin{aligned}
E\left(\{\boldsymbol{p}_i\},\{\boldsymbol{X}_j\}\right) &= \frac{1}{2}\sum_{i,j}\left[(x_{ij}-\mathrm{comp}(x;\boldsymbol{p}_i,\boldsymbol{X}_j))^2+(y_{ij}-\mathrm{comp}(y;\boldsymbol{p}_i,\boldsymbol{X}_j))^2\right] \qquad (3)\\
&= \frac{1}{2}\sum_{i,j}\left[(x_{ij}-\hat{x}_{ij})^2+(y_{ij}-\hat{y}_{ij})^2\right]. \qquad (4)
\end{aligned}
$$

The cost function Eq. (3) is called re-projection error. This style simultaneous optimization of both unknown camera parameters and scene structure is called **Bundle Adjustment**.

Now, we consider applying bundle adjustment to depth camera. Let $\boldsymbol{X}_{ij} = [X_{ij}\ Y_{ij}\ Z_{ij}]^{\mathbf{T}}$ denotes a depth information of j-th point observed by i-th camera and $\hat{\boldsymbol{X}}_{ij}$ denotes re-projected $\boldsymbol{X}_{ij}$ as

$$
\begin{aligned}
\hat{\boldsymbol{X}}_{ij} &= \begin{bmatrix}\hat{X}_{ij} & \hat{Y}_{ij} & \hat{Z}_{ij}\end{bmatrix}^{\mathbf{T}}\\
&= \begin{bmatrix}\mathrm{comp}(X;\boldsymbol{p}_i,\boldsymbol{X}_j) & \mathrm{comp}(Y;\boldsymbol{p}_i,\boldsymbol{X}_j) & \mathrm{comp}(Z;\boldsymbol{p}_i,\boldsymbol{X}_j)\end{bmatrix}^{\mathbf{T}}\\
&= \begin{bmatrix}(\mathsf{P}_i)_1[\boldsymbol{X}_j^{\mathbf{T}}\ 1]^{\mathbf{T}} & (\mathsf{P}_i)_2[\boldsymbol{X}_j^{\mathbf{T}}\ 1]^{\mathbf{T}} & (\mathsf{P}_i)_3[\boldsymbol{X}_j^{\mathbf{T}}\ 1]^{\mathbf{T}}\end{bmatrix}^{\mathbf{T}}.
\end{aligned}
$$

Then, the cost function Eq. (3) is re-formulated for depth camera as

$$
E\left(\{\boldsymbol{p}_i\},\{\boldsymbol{X}_j\}\right) = \frac{1}{2}\sum_{i,j}\left[(X_{ij}-\hat{X}_{ij})^2+(Y_{ij}-\hat{Y}_{ij})^2+(Z_{ij}-\hat{Z}_{ij})^2\right]. \qquad (5)
$$

Figure 3 illustrates the geometrical relationship.

2 Numerical optimization

Let x represent a set of all unknown variables. The cost function $E(\mathrm{x}) = E\left(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m, \boldsymbol{X}_j, \ldots, \boldsymbol{X}_n\right)$ is non-linear function of x, thus, finding x minimizing Eq. (3) is achieved by iterative numerical optimization. Given an initial guess x, we iteratively update the solution as $\mathrm{x} \rightarrow \mathrm{x} + \delta\mathrm{x}$ until the current solution satisfies some conditions.

The core of bundle adjustment is minimization of re-projection error $E(\mathrm{x})$. Usually, cost function $E(\mathrm{x})$ forms sum of squares, popular solution is least-squares method.

2.1 Newton's method

Newton's method computes second order derivative, or its approximation.

2.1.1 Gauss-Newton method

The solution x minimizing the cost function should satisfy $dE/d\mathrm{x}$.

Newton's method updates current estimate x with an update $\delta\mathrm{x}$ as $\mathrm{x} \rightarrow \mathrm{x} + \delta\mathrm{x}$ and iterates this update until convergence. Applying Tailor expansion around current x, we obtain

$$E(\mathrm{x} + \delta\mathrm{x}) \approx E(\mathrm{x}) + \mathrm{g}^{\mathbf{T}}\delta\mathrm{x} + \frac{1}{2}\delta\mathrm{x}^{\mathbf{T}}\mathtt{H}\delta\mathrm{x}, \tag{6}$$

where g and $\mathtt{H}$ are gradient and Hessian at x as

$$\mathrm{g} = \left.\frac{dE}{d\mathrm{x}}\right|_{\mathrm{x}}, \mathtt{H} = \left.\frac{d^2E}{d\mathrm{x}^2}\right|_{\mathrm{x}} \tag{7}$$

Regarding x as constant, the update $\delta\mathrm{x}$ minimizing the right hand side of Eq. (6) is

$$\mathtt{H}\delta\mathrm{x} = -\mathrm{g}. \tag{8}$$

Therefore, is $\mathtt{H}$ is invertible, the update is $\delta\mathrm{x} = -\mathtt{H}^{-1}\mathrm{g}$. If the current solution x is enough closer to the minimum value, $\mathtt{H}$ is positive-definite that means all eigenvalue is positive. If it's a case, rapid convergence of the above udpate is guaranteed.

However, when x is far from the minimum, there is no guarantee that $\mathtt{H}$ is positive-definite. Furthermore, computing $\mathtt{H}$ is heavy cost. For this issue, Gauss-Newton method approximates $texttt H$ as

$$\mathtt{H} \approx \mathtt{A} \equiv \mathtt{J}^{\mathbf{T}}\mathtt{J}, \tag{9}$$

where $\mathtt{J}$ is Jacovian matrix of $E(\mathrm{x})$ as

$$\mathtt{J} = \frac{dE(\mathrm{x})}{d\mathrm{x}}. \tag{10}$$

With Jacobian, Eq. (8) is rewritten as

$$\mathtt{A}\delta\mathrm{x} = \mathrm{a}, \tag{11}$$

where $\mathrm{a} = -\mathrm{g} = -\mathtt{J}^{\mathbf{T}}\mathrm{e}$.

2.1.2 Levenberg-Marquardt

Levenberg-Marquardt algorithm changes Eq. (11) as

$$(\mathtt{A} + \lambda \mathtt{I})\delta \mathrm{x} = \mathrm{a}, \tag{12}$$

where $\lambda \leq 0$ is a dumping factor. When $\lambda = 0$, LM becomes Gauss-Newton while LM becomes steepest descent with big λ.