Camera Pose Estimation in Structure-from-Motion

Yiyu Wang, Yuqi Wang, Yuxuan Long

***Abstract*—We investigate the optimization problems in camera pose estimation under the background of rigid Structure-from-Motion. Rotation averaging and translation averaging are studied, with efficient solvers like interior point method and scaled-ADMM implemented. Relevant experiments are also conducted.**

I. Introduction

Structure-from-motion (SfM) is a highly attractive research topic, from which a 3D model can be recovered with a collection of 2D images. SfM methods can be roughly categorized as incremental or global according to their ways to initialize bundle adjustment. Incremental SfM is popular due to its accurate reconstruction result but very slow and might suffer from large drifting errors. Global methods initialize all camera simultaneously and hence may benefit efficiency and accuracy.

In this project, we mainly study the optimization methods used in the camera pose estimation from the perspective of global SfM. There are two key steps involved in the global SfM. The first step is rotation averaging, where a classical approach relying on the Lie group structure of 3D rotation is to be studied. Interior point method is implemented to solve a minimum $\mathcal{L}_1$ error approximation formulation. The subsequent step is translation averaging (i.e. camera location estimation). We investigate a robust convex formulation for translation averaging, which contains the unsquared $\mathcal{L}_2$ loss. Two solvers like scaled-ADMM and IRLS are implemented respectively for this problem. Relevant experiments have been conducted to test the robustness of the algorithms.

II. Rotation Averaging

Given a collection of 2D images, a viewing graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be established by matching image features in pairs. The relative pose (i.e. translation and rotation) between different cameras can be solved by eight-point or five-point method. In this section, we focus on rotation averaging problem. Let's denote the absolute 3D rotation of the k-th camera as $\mathbf{R}_k$, and the relative rotation between camera i and j as $\mathbf{R}_{ij}$, where

$$\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i^{-1}, \forall e_{ij} \in \mathcal{E}. \tag{1}$$

Due to the existence of the noise and outliers in the relative poses, (1) does not necessarily hold for all image pairs. Hence we would like to minimize the summed distances:

$$\underset{\{\mathbf{R}_i\}_{i=1}^{|\mathcal{V}|}}{\arg\min} \sum_{e_{ij} \in \mathcal{E}} d(\mathbf{R}_{ij}, \mathbf{R}_j \mathbf{R}_i^{-1}). \tag{2}$$

We follow the choice from [1] which uses geodesic distance on Special Orthogonal group $SO(3)$ as the distance metric. The Riemannian distance metric on $SO(3)$ is bi-invariant, shown as following. $\|\cdot\|_F$ denotes the Frobenius norm and first-order BCH approximation is used.

$$\begin{aligned} d(\mathbf{R}_1, \mathbf{R}_2) &= d(\mathbf{I}, \mathbf{R}_2 \mathbf{R}_1^{-1}) = \frac{1}{\sqrt{2}} \left\| \log \left(\mathbf{R}_2 \mathbf{R}_1^{-1} \right) \right\|_F \\ &= \left\| \boldsymbol{\omega}(\mathbf{R}_2 \mathbf{R}_1^{-1}) \right\|_2 = \left\| BCH(\boldsymbol{\omega}_2, -\boldsymbol{\omega}_1) \right\|_2 \\ &\approx \left\| \boldsymbol{\omega}_2 - \boldsymbol{\omega}_1 \right\|_2 \end{aligned} \tag{3}$$

A rotation in Lie group can be expressed as some skew-symmetric matrix in Lie algebra with exponential map. The components in this skew-symmetric matrix, denoted as $[\boldsymbol{\omega}]_\times$ are exactly from $\boldsymbol{\theta} = \theta \boldsymbol{e}$, that is the axis-angle representation of the rotation. The following is the geodesic distance between $\mathbf{R}_{ij}$ and $\mathbf{R}_j \mathbf{R}_i^{-1}$.

$$d(\mathbf{R}_{ij}, \mathbf{R}_j \mathbf{R}_i^{-1}) = \left\| \boldsymbol{\omega}(\mathbf{R}_j^{-1} \mathbf{R}_{ij} \mathbf{R}_i) \right\|_2 = \left\| \boldsymbol{\omega}(\Delta R_{ij}) \right\|_2 \tag{4}$$

In order to minimize this distance, we apply an iterative approach and hope to adjust absolute rotation $\mathbf{R}_k, \forall k \in [1, N]$ $(N = |\mathcal{V}|)$ in every iteration until the distance satisfying our tolerance. In other words, we are supposed to solve the following system in every iteration.

$$\mathbf{A} \Delta \boldsymbol{\omega}_{global} = \Delta \boldsymbol{\omega}_{rel} \tag{5}$$

Where $\mathbf{A}$ is a $M \times N$ matrix(M is the number of edges) which contains the graph structure information. $\Delta\omega_{rel} = \|\omega(\Delta R_{ij})\|$ and $\Delta\omega_{global}$ is the adjustment of absolute rotation.

A. Interior Point method

(5) can be robustly solved by minimizing the absolute errors:

$$\min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_1 \tag{6}$$

For the simplicity of notation, we let $\mathbf{x} = \Delta\omega_{global}$ and $\mathbf{y} = \Delta\omega_{rel}$ in the following analysis.

In practice, this $\mathcal{L}1$ error approximation problem can also be considered as the linear program

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^N, \mathbf{u} \in \mathbb{R}^M} \quad & \mathbf{1}^T \mathbf{u} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} - \mathbf{u} - \mathbf{y} \preceq 0 \\ & -\mathbf{A}\mathbf{x} - \mathbf{u} + \mathbf{y} \preceq 0 \end{aligned} \tag{7}$$

for the primal-dual algorithm, we define $\mathbf{f}_{u_1} := \mathbf{A}\mathbf{x} - \mathbf{u} - \mathbf{y}$, $\mathbf{f}_{u_2} := -\mathbf{A}\mathbf{x} - \mathbf{u} + \mathbf{y}$, where $\mathbf{f}_{u_1}$ is the vector $(f_{u_1;1} ... f_{u_1;M})^T$ and likewise for $\mathbf{f}_{u_2}, \lambda_{u_1}, \lambda_{u_2}$. The Lagrangian can be written as

$$\mathcal{L}(\mathbf{x}, \mathbf{u}, \lambda_{u_1}, \lambda_{u_2}) = \mathbf{1}^T u + \lambda_{u_1}^T \mathbf{f}_{u_1} + \lambda_{u_2}^T \mathbf{f}_{u_2}. \tag{8}$$

The core of interior point method is the Newton step on perturbed KKT where the complementary slackness condition $\lambda_i f_i = 0$ is replaced by $\lambda_i^k f_i^k = -1/\tau^k$(at the k-th step), and in this case, τ^k increases with the progress of Newton iterations. The perturbed KKT conditions is obtained as following.

- stationary

$$\frac{\partial \mathcal{L}}{\partial(\mathbf{x},\mathbf{u})} = \begin{pmatrix} \mathbf{A}^T(\lambda_{u_1} - \lambda_{u_2}) \\ \mathbf{1} - \lambda_{u_1} - \lambda_{u_2} \end{pmatrix} \tag{9}$$

- complementary slackness

$$\lambda_{u_j;i} f_{u_j;i} = -1/\tau \quad j = 1,2 \quad i = 1,...,M \tag{10}$$

- primal feasibility

$$f_{u_j;i} \leq 0 \quad j = 1,2 \quad i = 1,...,M \tag{11}$$

- dual feasibility

$$\lambda_{u_j;i} \geq 0 \quad j = 1,2 \quad i = 1,...,M \tag{12}$$

introduce notation

$$\mathbf{r}_{dual} = \begin{pmatrix} \mathbf{A}^T(\lambda_{u_1} - \lambda_{u_2}) \\ \mathbf{1} - \lambda_{u_1} - \lambda_{u_2} \end{pmatrix}, \mathbf{r}_{cent} = \begin{pmatrix} -\wedge_{u_1} \mathbf{f}_{u_1} \\ -\wedge_{u_2} \mathbf{f}_{u_2} \end{pmatrix} - \frac{1}{\tau}\mathbf{1} \tag{13}$$

called the dual, central residuals at point $(\mathbf{x}, \mathbf{u}, \lambda_{u_1}, \lambda_{u_2})$. $\wedge$ is a diagonal matrix with $\wedge_{ii} = \lambda_i$. Primal residual doesn't show up because there is no equality constraints in the formulation. Both $\mathbf{r}_{dual}$ and $\mathbf{r}_{cent}$ are supposed to be $\mathbf{0}$, and therefore we can view this as a nonlineasr system of equations and solve this by Newton's method. For the primal variables, the Newton step is the solution to

$$\begin{aligned} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix} &:= \begin{pmatrix} \mathbf{A}^T \sum_{11} \mathbf{A} & \mathbf{A}^T \sum_{12} \\ \sum_{12} \mathbf{A} & \sum_{11} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{u} \end{pmatrix} \\ &= \begin{pmatrix} -(1/\tau) \cdot \mathbf{A}^T(-\mathbf{f}_{u_1}^{-1} + \mathbf{f}_{u_2}^{-1}) \\ -\mathbf{1} - (1/\tau) \cdot (\mathbf{f}_{u_1}^{-1} + \mathbf{f}_{u_2}^{-1}) \end{pmatrix} \end{aligned} \tag{14}$$

where

$$\textstyle\sum_{11} = -\wedge_{u_1} \mathbf{f}_{u_1}^{-1} - \wedge_{u_2} \mathbf{f}_{u_2}^{-1} \tag{15}$$

$$\textstyle\sum_{12} = \wedge_{u_1} \mathbf{f}_{u_1}^{-1} - \wedge_{u_2} \mathbf{f}_{u_2}^{-1} \tag{16}$$

$\mathbf{F}$ is diagonal with $F_{ii} = f_i$. $\Delta\mathbf{u}$ can be solved by

$$\Delta\mathbf{u} = \textstyle\sum_{11}^{-1}(\mathbf{w}_2 - \sum_{22} \mathbf{A}\Delta\mathbf{x}) \tag{17}$$

and $\Delta\mathbf{x}$ can be obtained from solving

$$\mathbf{A}^T \textstyle\sum_x \mathbf{A}\Delta\mathbf{x} = \mathbf{w}_1 - \mathbf{A}^T \sum_{22} \sum_{11}^{-1} \mathbf{w}_2 \tag{18}$$

where $\sum_x = \sum_{11} - \sum_{12}^2 \sum_{11}^{-1}$ and the updates for dual variables are given by

$$\Delta\lambda_{u_1} = \wedge_{u_1}^{-1}(-\Delta\mathbf{x} + \Delta\mathbf{u}) - \lambda_{u_1} - (1/\tau)\mathbf{f}_{u_1}^{-1} \tag{19}$$

$$\Delta\lambda_{u_2} = \wedge_{u_2}^{-1}(-\Delta\mathbf{x} + \Delta\mathbf{u}) - \lambda_{u_2} - (1/\tau)\mathbf{f}_{u_2}^{-1} \tag{20}$$

After discussing the update direction, here comes the other important element in the algorithm, which is the step size. A multi-stage backtracking line search method is applied to find a good step size that should maintain $\mathbf{f}_{u_1} < 0$,$\mathbf{f}_{u_1} < 0$, $\lambda_{u_1} > 0$, and $\lambda_{u_2} > 0$. The first step is to get a largest step size $s_{max} \leq 1$ that makes $\lambda_{u_1} + s\Delta\lambda_{u_1} \geq 0$ and $\lambda_{u_2} + s\Delta\lambda_{u_2} \geq 0$

$$s_{max} = \min\{1, \min\{-\lambda_{u_j;i}/\Delta\lambda_{u_j;i} : \Delta\lambda_{u_j;i} < 0\}\}$$

then, with parameters $\alpha, \beta \in (0,1)$, we set $s = 0.99 s_{max}$ and

- update $s = \beta s$, until $f_{u_j;i} < 0 \quad j = 1,2 \quad i = 1,...M$
- update $s = \beta s$, until $\left\|r(\mathbf{x}^+, \mathbf{u}^+, \lambda_{u_1}^+, \lambda_{u_2}^+)\right\|_2 \leq (1 - \alpha s)\left\|r(\mathbf{x}, \mathbf{u}, \lambda_{u_1}, \lambda_{u_2})\right\|_2, \quad r = (\|r_{dual}\|_2^2 + \|r_{cent}\|_2^2)^{1/2}$

At last but not least, the algorithm stops when the surrogate duality gap(the dual iterates are not necessarily feasible for the original dual problem, hence "surrogate") is under threshold and r is small enough. The surrogate duality gap is defined below.

$$\eta = -\mathbf{f}_{u_1}^T \lambda_{u_1} - \mathbf{f}_{u_2}^T \lambda_{u_2} \tag{21}$$

III. Translation Averaging

Once absolute rotations are obtained from rotation averaging, the pairwise direction between connected cameras can be immediately determined. Then, the camera location recovery is an inverse problem that asks for estimating the absolute coordinates given only the relative direction vectors.

For the known viewing graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we are only given the pairwise direction observations $\{\mathbf{v}_{ij} \in \mathbb{R}^3 | e_{ij} \in \mathcal{E}, \|\mathbf{v}_{ij}\|_2 = 1\}$, where $\mathbf{v}_{ij}$ is the observed unit direction vector pointing from the camera j to camera i. Hence, the location recovery problem is to recover all camera positions $\{\mathbf{t}_i \in \mathbb{R}^3\}_{i=1}^{|\mathcal{V}|}$ up to a global scale and translation. For convenience, let $N = |\mathcal{V}|$ and $M = |\mathcal{E}|$.

We follow the robust convex formulation from ShapeFit [2], where the objective is the $\mathcal{L}_1$ norm of a set of unsquared distances:

$$\begin{aligned} \min_{\{\mathbf{t}_i \in \mathbb{R}^3\}_{i=1}^N} \quad & \sum_{e_{ij} \in \mathcal{E}} \left\|\mathbf{P}_{ij}^c(\mathbf{t}_i - \mathbf{t}_j)\right\|_2 \\ \text{s.t.} \quad & \sum_{e_{ij} \in \mathcal{E}} \mathbf{v}_{ij}^T(\mathbf{t}_i - \mathbf{t}_j) = 1, \sum_{i=1}^N \mathbf{t}_i = \mathbf{0}, \end{aligned} \tag{22}$$

where $\mathbf{P}_{ij}^c \in \mathbb{R}^{3\times3}$ is the complementary projector of $\mathbf{P}_{ij} = \mathbf{v}_{ij}\mathbf{v}_{ij}^T$, i.e. $\mathbf{P}_{ij}^c = \mathbf{I} - \mathbf{v}_{ij}\mathbf{v}_{ij}^T$. The null space of $\mathbf{P}_{ij}^c$ is only spanned by $\mathbf{v}_{ij}$, so that minimizing $\left\|\mathbf{P}_{ij}^c(\mathbf{t}_i - \mathbf{t}_j)\right\|_2$ can force $\mathbf{t}_i - \mathbf{t}_j$ to be parallel to the direction $\mathbf{v}_{ij}$. The linear constraints in (22) fix the scale and zero the mean of the camera locations. Let $\mathbf{t} \in \mathbb{R}^{3N}$ which stacks all $\mathbf{t}_i$, we can create linear operator $\mathbf{Q}_{ij} \in \mathbb{R}^{3\times3N}$ such that $\mathbf{t}_i - \mathbf{t}_j = \mathbf{Q}_{ij}\mathbf{t}$. Then, the first constraint in (22) can be expressed as $\mathbf{b}^T\mathbf{t} = 1$, where $\mathbf{b} = \sum_{e_{ij} \in \mathcal{E}} \mathbf{Q}_{ij}^T \mathbf{v}_{ij}$. Also, let $\mathbf{A} \in \mathbb{R}^{3\times3N}$ be a linear operator such that $\mathbf{A}\mathbf{t} = \sum_i \mathbf{t}_i$, then the second constraint in (22) is simply equivalent to $\mathbf{A}\mathbf{t} = \mathbf{0}$. We hence define the feasible set as $\mathcal{F} := \{\mathbf{t} \in \mathbb{R}^{3N} | \mathbf{b}^T\mathbf{t} = 1, \mathbf{A}\mathbf{t} = \mathbf{0}\}$.

A. ADMM

(22) can be reformulated as

$$\begin{aligned} \min_{\mathbf{t} \in \mathcal{F}, \{\mathbf{y}_{ij} \in \mathbb{R}^3\}_{e_{ij} \in \mathcal{E}}} \quad & \sum_{e_{ij} \in \mathcal{E}} \left\|\mathbf{P}_{ij}^c \mathbf{y}_{ij}\right\|_2 \\ \text{s.t.} \quad & \mathbf{y}_{ij} = \mathbf{Q}_{ij}\mathbf{t} \ \forall e_{ij} \in \mathcal{E}. \end{aligned} \tag{23}$$

This can be solved by scaled-ADMM [2], with the augmented Lagrangian

$$L_\tau(\mathbf{t},\mathbf{y},\boldsymbol{\lambda}) = \sum_{e_{ij}\in\mathcal{E}} \left\|\mathbf{P}^c_{ij}\mathbf{y}_{ij}\right\|_2 + \frac{\tau}{2}\left\|\mathbf{Q}_{ij}\mathbf{t}-\mathbf{y}_{ij}+\boldsymbol{\lambda}_{ij}\right\|_2^2, \tag{24}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{3M}$ stacks all the dual variables $\boldsymbol{\lambda}_{ij} \in \mathbb{R}^3$ and $\mathbf{y} \in \mathbb{R}^{3M}$ stacks all $\mathbf{y}_{ij}$.

ADMM updates are then given as

$$\mathbf{t}^{(k+1)} = \underset{\mathbf{t}\in\mathcal{F}}{\arg\min}\, L_\tau(\mathbf{t},\mathbf{y}^{(k)},\boldsymbol{\lambda}^{(k)}) \tag{25}$$

$$\mathbf{y}^{(k+1)} = \underset{\mathbf{y}\in\mathbb{R}^{3M}}{\arg\min}\, L_\tau(\mathbf{t}^{(k+1)},\mathbf{y},\boldsymbol{\lambda}^{(k)}) \tag{26}$$

$$\boldsymbol{\lambda}_{ij}^{(k+1)} = \boldsymbol{\lambda}_{ij}^{(k)} + \mathbf{Q}_{ij}\mathbf{t}^{(k+1)} - \mathbf{y}_{ij}^{(k+1)},\ \forall e_{ij}\in\mathcal{E}. \tag{27}$$

By stacking all $\mathbf{Q}_{ij}$ into $\mathbf{Q} \in \mathbb{R}^{3M\times 3N}$, we can notice that (25) is equivalent to

$$\mathbf{t}^{(k+1)} = \underset{\mathbf{t}\in\mathcal{F}}{\arg\min}\left\|\mathbf{Q}\mathbf{t}-\mathbf{y}^{(k)}+\boldsymbol{\lambda}^{(k)}\right\|_2^2. \tag{28}$$

Applying the KKT condition to (28) raises the KKT equation:

$$\begin{aligned}\mathbf{Q}^T\mathbf{Q}\mathbf{t}+\tilde{\mathbf{A}}^T\mathbf{q} &= \mathbf{Q}^T\mathbf{r}^{(k+1)},\\ \tilde{\mathbf{A}}\mathbf{t} &= [\mathbf{0}_{1\times3},1]^T,\end{aligned} \tag{29}$$

where $\tilde{\mathbf{A}} = [\mathbf{A}^T,\mathbf{b}]^T$, $\mathbf{r}^{(k+1)} = \mathbf{y}^{(k)} - \boldsymbol{\lambda}^{(k)}$ and $\mathbf{q}\in\mathbb{R}^4$ is the dual variable of (28). Let $\mathbf{B} = (\mathbf{Q}^T\mathbf{Q})^{-1}\tilde{\mathbf{A}}^T$, solving (29) explicitly gives the full update in $\mathbf{t}$:

$$\mathbf{t}^{(k+1)} = \mathbf{Q}^\dagger\mathbf{r}^{(k+1)} - \mathbf{B}(\tilde{\mathbf{A}}\mathbf{B})^{-1}(\tilde{\mathbf{A}}\mathbf{Q}^\dagger\mathbf{r}^{(k+1)} - [\mathbf{0}_{1\times3},1]^T), \tag{30}$$

To update $\mathbf{y}$, (26) can be indeed decomposed into subproblem

$$\mathbf{y}_{ij}^{(k+1)} = \underset{\mathbf{y}_{ij}\in\mathbb{R}^3}{\arg\min}\left\|\mathbf{P}^c_{ij}\mathbf{y}_{ij}\right\|_2 + \frac{\tau}{2}\left\|\mathbf{Q}_{ij}\mathbf{t}^{(k+1)}+\boldsymbol{\lambda}_{ij}^{(k)}-\mathbf{y}_{ij}\right\|_2^2. \tag{31}$$

Let $\mathbf{z}_{ij} = \mathbf{Q}_{ij}\mathbf{t}^{(k+1)} + \boldsymbol{\lambda}_{ij}^{(k)}$, (31) can be solved approximately by $\mathcal{L}_1$ norm proximal operator [2]:

$$\mathbf{y}_{ij}^{(k+1)} = \mathbf{P}_{ij}\mathbf{z}_{ij} + \operatorname{sign}(\mathbf{P}^c_{ij}\mathbf{z}_{ij})\circ\max(0,|\mathbf{P}^c_{ij}\mathbf{z}_{ij}| - \frac{1}{\tau}). \tag{32}$$

B. IRLS

As inspired by LUD solver [3], (22) can be solved by Iteratively Reweighted Least Squares (IRLS), with the weights $\{w_{ij}^{(k)}\}_{e_{ij}\in\mathcal{E}}$ updated in every iteration:

$$\mathbf{t}^{(k+1)} = \underset{\mathbf{t}\in\mathcal{F}}{\arg\min}\sum_{e_{ij}\in\mathcal{E}} w_{ij}^{(k)}\left\|\mathbf{P}^c_{ij}\mathbf{Q}_{ij}\mathbf{t}\right\|_2^2, \tag{33}$$

$$w_{ij}^{(k+1)} = \left(\left\|\mathbf{P}^c_{ij}\mathbf{Q}_{ij}\mathbf{t}^{(k+1)}\right\|_2^2+\delta\right)^{-\frac{1}{2}}, \forall e_{ij}\in\mathcal{E}, \tag{34}$$

where δ is a very small positive constant for avoiding the zero division. The weights $\{w_{ij}\}_{e_{ij}\in\mathcal{E}}$ are initially set to one.

The quadratic objective in (33) can be indeed simplified as $\mathbf{t}^T\mathbf{L}^{(k)}\mathbf{t}$, with $\mathbf{L}^{(k)} = \sum_{e_{ij}\in\mathcal{E}} w_{ij}^{(k)}\mathbf{Q}_{ij}^T\mathbf{P}^c_{ij}\mathbf{Q}_{ij}$. It is then obvious that (33) is also a constrained quadratic program, similar to (28). So, (33) can be solved explicitly as:

$$\mathbf{t}^{(k+1)} = (\mathbf{L}^{(k)})^{-1}\tilde{\mathbf{A}}^T(\tilde{\mathbf{A}}(\mathbf{L}^{(k)})^{-1}\tilde{\mathbf{A}}^T)^{-1}[\mathbf{0}_{1\times3},1]^T. \tag{35}$$

IV. Experimental Results

A. Rotation Averaging

In practice, all elements in Δw_{global} and Δw_{rel} are described as axis-angle representation when feeding into the linear system. That is to say, we have to solve three such systems in one iteration because there are three components determining the axis while $\mathbf{A}$ matrix remains unchanged.

In order to test our interior point method and simplify the problem, we generated random data in experiments. We built a system $\mathbf{A}\mathbf{x} = \mathbf{y}$ with the size of $\mathbf{A} = 100\times25$, $\mathbf{x} = 25\times1$ and $\mathbf{y} = 25\times1$, then we contaminated 30% $\mathbf{y}$ value. As you can check in the code, we can still perfectly recover $\mathbf{x}$, which means our algorithm works very well. Note, our experiment is for illustrative purpose only. In real-life, $\Delta\omega_{global}$ and $\Delta\omega_{rel}$ would be considered and solving it with iteration method is desired. Once $\Delta\omega_{rel}$ is small enough, then rotation averaging is done. However, the result can still be improved by applying an IRLS algorithm [4].

B. Translation Averaging

When the $\mathcal{L}_2$ objective in (22) is squared, the problem (22) turns to be a constrained Least-Squares (LS), which can be directly solved by the first iteration of IRLS. For the location recovery problem (22), we hence have three algorithms: LS, ADMM (i.e. (25), (26) and (27)), IRLS (i.e. (33) and (34)).

a) Experiment Setup: For IRLS and ADMM, the number of iterations is set to 100. For ADMM, we set $\tau = 0.1$, and $\mathbf{y},\lambda$ are initialized to zeros. For IRLS, we set $\delta = 10^{-5}$. The camera locations are randomly created with 100 cameras. There are 980 edges in the viewing graph.

b) Robustness Test: To test robustness of the algorithms, zero-mean Gaussian noise was added on the pairwise direction observations $\{\mathbf{v}_{ij}\}_{e_{ij}\in\mathcal{E}}$. Mean Absolute Error (MAE) of the recovered locations was computed. Note that recovered locations were finally scaled and translated to the original coordinate system, as shown in Figure 2 from the Appendix. By varying the noise variance, MAEs for all algorithms are shown in Figure 3 from the Appendix. We could find that IRLS and ADMM produce much lower error than LS, indicating the robustness of unsquared $\mathcal{L}_2$ formulation. ADMM slightly outperforms IRLS. However, ADMM explodes once the noise variance becomes large (e.g. 0.04). Interestingly, IRLS could remain stable at such case. But only IRLS has to solve a new linear system in every iteration.

V. Summary

In this project, we explored the optimization methods included in global Structure-from-Motion pipeline and we showed that interior point method is very powerful when dealing with the non-trivial rotation averaging problem. As found in translation averaging, the unsquared $\mathcal{L}_2$ loss always leads to a more robust estimation then the squared $\mathcal{L}_2$ loss. IRLS has stable performance but suffers from higher order time complexity as compared with ADMM. The future work would be to make the algorithms more robust under various circumstances meanwhile keeping at high efficiency.

References

[1] A. Chatterjee and V. M. Govindu, "Efficient and robust large-scale rotation averaging," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 521–528.

[2] T. Goldstein, P. Hand, C. Lee, V. Voroninski, and S. Soatto, "Shapefit and shapekick for robust, scalable structure from motion," in *European Conference on Computer Vision*. Springer, 2016, pp. 289–304.

[3] O. Ozyesil and A. Singer, "Robust camera location estimation by convex programming," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2674–2683.

[4] A. Chatterjee and V. M. Govindu, "Robust relative rotation averaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 958–972, 2018.

Appendix

A. Figures from Experiments

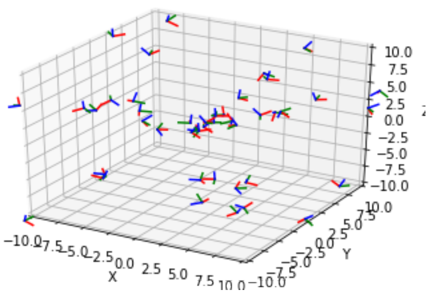

Figure 1: This is a simple demo for viewing graph

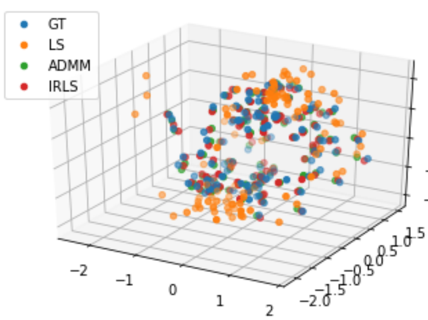

Figure 2: Recovered camera locations (noise variance set to 0.03). GT indicates ground-truth camera locations

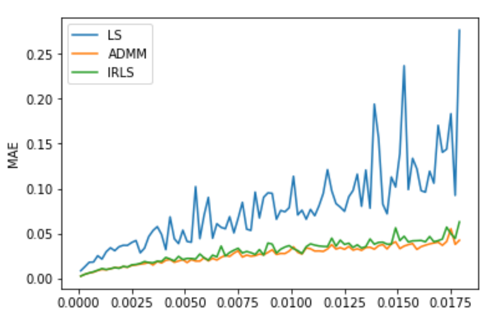

Figure 3: MAE of recovered locations versus noise variance

B. Working on Real Data

The figures 4, 5 and 6 are obtained from Theiasfm library with real data. Theiasfm implemented global structure-from-motion pipeline with the same rotation averaging and translation averaging algorithm as we discussed above. Due to the lack of computational power of our laptop, we only tested on small dataset, hence the sparse point cloud.

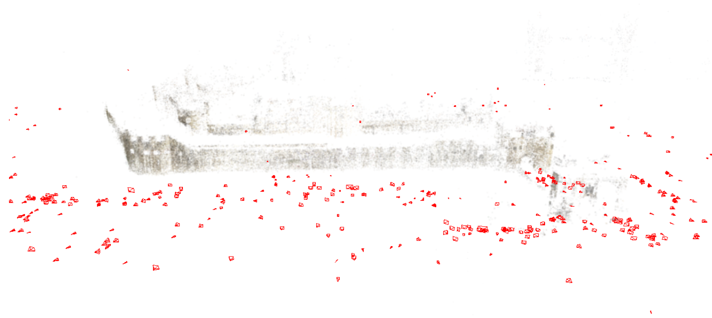

Figure 4: Global Structure-from-Motion for London Tower

Figure 5: Global Structure-from-Motion for Madrid Metropolis

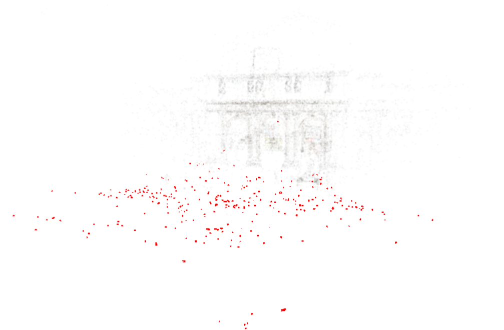

Figure 6: Global Structure-from-Motion for NYC Library