gDLS*: Generalized Pose-and-Scale Estimation Given Scale and Gravity Priors
Supplemental Material

Victor Fragoso
Microsoft
victor.fragoso@microsoft.com

Joseph DeGol
Microsoft
joseph.degol@microsoft.com

Gang Hua
Wormpex AI
ganghua@gmail.com

I. Introduction

In this document, we present the full mathematical derivation of gDLS* shown in the main submission. This derivation provides closed form equations that allow a user to implement the proposed method.

II. Incorporating Priors via Regularizers

Section 3 of the main submission describes the proposed formulation to include scale and rotation priors using regularizers. This formulation is:

$$J' = J(R, \mathbf{t}, s, \boldsymbol{\alpha}) + \underbrace{\lambda_s \left(s_0 - s\right)^2}_{J_s} + \underbrace{\lambda_g \|\mathbf{g}_{\mathcal{Q}} \times R\mathbf{g}_{\mathcal{W}}\|^2}_{J_g}, \tag{i}$$

where

$$J(R, \mathbf{t}, s, \boldsymbol{\alpha}) = \sum_{i=1}^{n} \|\alpha_i \mathbf{r}_i - \left(R\mathbf{p}_i + \mathbf{t} - s\mathbf{c}_i\right)\|^2 \tag{ii}$$

is the gDLS [2] pose-and-scale re-projection error cost function from the n 2D-3D correspondences, J_s is the scale prior regularizer, and J_g is the gravity direction constraint imposing a rotation prior.

II.i. Cost Function Depending Only on Rotation

As mentioned in Section 3, the first step to obtain a cost function that we can minimize in a single shot is to rewrite Eq. (i) as a function of the rotation matrix R. To do so, we define

$$\mathbf{x} = \begin{bmatrix}\alpha_1 & \dots & \alpha_n & s & \mathbf{t}^\intercal\end{bmatrix}^\intercal, \tag{iii}$$

a vector holding the depths for each i-th point α_i, the scale s, and translation vector $\mathbf{t}$. We know that the optimal depths, translation, and scale $\mathbf{x}^\star$ vanish the gradient $\nabla_\mathbf{x} J'$ of the cost function J', *i.e.*,

$$\nabla_\mathbf{x} J'\big|_{\mathbf{x}=\mathbf{x}^\star} = \left[\nabla_\mathbf{x} J + \nabla_\mathbf{x} J_s\right]_{\mathbf{x}=\mathbf{x}^\star} = 0. \tag{iv}$$

To satisfy this constraint, we calculate each of the gradients $\nabla_\mathbf{x} J$ and $\nabla_\mathbf{x} J_s$:

$$\begin{aligned}\nabla_\mathbf{x} J &= 2A^\intercal A\mathbf{x} - 2AWb \\ \nabla_\mathbf{x} J_s &= 2P\mathbf{x} - 2P\mathbf{x}_0,\end{aligned} \tag{v}$$

where

$$\begin{aligned} A &= \begin{bmatrix}\mathbf{r}_1 & & & \mathbf{c}_1 & -I \\ & \ddots & & \vdots & \vdots \\ & & \mathbf{r}_n & \mathbf{c}_n & -I\end{bmatrix}, \mathbf{b} = \begin{bmatrix}\mathbf{p}_1 \\ \vdots \\ \mathbf{p}_n\end{bmatrix} \\ P &= \begin{bmatrix}0_{n\times n} & & \\ & \lambda_s & \\ & & 0_{3\times 3}\end{bmatrix}, W = \begin{bmatrix}R & & \\ & \ddots & \\ & & R\end{bmatrix},\end{aligned} \tag{vi}$$

and $\mathbf{x}_0 = \begin{bmatrix}0_n^\intercal & s_o & 0_3^\intercal\end{bmatrix}^\intercal$. Note that

$$J_s = \lambda_s \left(s_0 - s\right)^2 = \left(\mathbf{x} - \mathbf{x}_0\right)^\intercal P \left(\mathbf{x} - \mathbf{x}_0\right). \tag{vii}$$

Thus, we can rewrite Eq. (iv) into

$$\nabla_\mathbf{x} J' = A^\intercal A\mathbf{x} - AWb + P\mathbf{x} - P\mathbf{x}_0 = 0, \tag{viii}$$

by combining Equations (iv) and (v). Rearranging terms of Eq. (viii) yields Eq. (5) in the main submission, which is

$$\mathbf{x} = \left(A^\intercal A + P\right)^{-1} A^\intercal Wb + \left(A^\intercal A + P\right)^{-1} P\mathbf{x}_0. \tag{ix}$$

In order to obtain the simplified version of Eq. (ix) shown in Eq. (5) of the main submission and inspired by [2], we rewrite

$$\begin{aligned} A^\intercal A + P &= \begin{bmatrix}I_{n\times n} & B'_{n\times 4} \\ B'^\intercal_{n\times 4} & D'_{4\times 4}\end{bmatrix} \\ &= \underbrace{\begin{bmatrix}I_{n\times n} & B_{n\times 4} \\ B^\intercal_{n\times 4} & D_{4\times 4}\end{bmatrix}}_{A^\intercal A} + \underbrace{\begin{bmatrix}0_{n\times n} & & \\ & \lambda_s & \\ & & 0_{3\times 3}\end{bmatrix}}_{P},\end{aligned} \tag{x}$$

where

$$B = \begin{bmatrix} \mathbf{r}_1^\intercal \mathbf{c}_1 & -\mathbf{r}_1^\intercal \\ \vdots & \vdots \\ \mathbf{r}_n^\intercal \mathbf{c}_n & -\mathbf{r}_n^\intercal \end{bmatrix}$$
$$D = \begin{bmatrix} \sum_{i=1}^n \mathbf{c}_i^\intercal \mathbf{c}_i & \sum_{i=1}^n -\mathbf{c}_i^\intercal \\ \sum_{i=1}^n -\mathbf{c}_i & nI \end{bmatrix}, \quad \text{(xi)}$$

and I is the identity matrix. It is important to mention that $A^\intercal A$ is exactly the same as that of gDLS [2], and thus

$$A^\intercal A + P = \begin{bmatrix} I_{n\times n} & B_{n\times 4} \\ B_{n\times 4}^\intercal & D'_{4\times 4} \end{bmatrix}$$
$$D'_{4\times 4} = \begin{bmatrix} \lambda_s + \sum_{i=1}^n \mathbf{c}_i^\intercal \mathbf{c}_i & \sum_{i=1}^n -\mathbf{c}_i^\intercal \\ \sum_{i=1}^n -\mathbf{c}_i & nI \end{bmatrix}. \quad \text{(xii)}$$

Eq. (ix) requires the inverse of $(A^\intercal A + P)^{-1}$. To compute a closed form relationship, we use the following block matrix expression

$$(A^\intercal A + P)^{-1} = \begin{bmatrix} E_{n\times n} & F_{n\times 4} \\ G_{4\times n} & H_{4\times 4} \end{bmatrix}. \quad \text{(xiii)}$$

Through block matrix inversion, we obtain the following closed-form block matrices:

$$\begin{aligned} E &= I + BHB^\intercal \\ F &= -BH \\ G &= -HB^\intercal \\ H &= (D' - Y)^{-1} \\ Y &= \begin{bmatrix} \sum_{i=1}^n \mathbf{c}_i^\intercal \mathbf{r}_i \mathbf{r}_i^\intercal \mathbf{c}_i & \sum_{i=1}^n -\mathbf{c}_i^\intercal \mathbf{r}_i \mathbf{r}_i^\intercal \\ \sum_{i=1}^n -\mathbf{r}_i \mathbf{r}_i^\intercal \mathbf{c}_i & \sum_{i=1}^n \mathbf{r}_i \mathbf{r}_i^\intercal \end{bmatrix} \end{aligned} \quad \text{(xiv)}$$

Like in gDLS [2], we use matrices U, S, and V to simplify Eq. (ix), *i.e.*,

$$(A^\intercal A + P)^{-1} A^\intercal = \begin{bmatrix} U \\ S \\ V \end{bmatrix}, \quad \text{(xv)}$$

where

$$U = \begin{bmatrix} \mathbf{r}_1^\intercal & & \\ & \ddots & \\ & & \mathbf{r}_n^\intercal \end{bmatrix} + B \begin{bmatrix} S \\ V \end{bmatrix}$$
$$\begin{bmatrix} S \\ V \end{bmatrix} = -HB^\intercal \begin{bmatrix} \mathbf{r}_1^\intercal & & \\ & \ddots & \\ & & \mathbf{r}_n^\intercal \end{bmatrix} + H \begin{bmatrix} \mathbf{c}_1^\intercal & \dots & \mathbf{c}_n^\intercal \\ -I & \dots & -I \end{bmatrix}$$
$$= H \begin{bmatrix} \mathbf{c}_1^\intercal - \mathbf{c}_1 \mathbf{r}_1 \mathbf{r}_1^\intercal & \dots & \mathbf{c}_n^\intercal - \mathbf{c}_n \mathbf{r}_n \mathbf{r}_n^\intercal \\ \mathbf{c}_1 \mathbf{c}_1^\intercal - I & \dots & \mathbf{c}_n \mathbf{c}_n^\intercal - I \end{bmatrix}. \quad \text{(xvi)}$$

We can simplify Eq. (ix) further. To do this, we focus on simplifying the term encoding the scale prior, yielding

$$(A^\intercal A + P)^{-1} P \mathbf{x}_0 = \begin{bmatrix} E & F \\ G & H \end{bmatrix} \underbrace{\begin{bmatrix} 0 & & \\ & \lambda_s & \\ & & 0 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} 0 \\ s_0 \\ 0 \end{bmatrix}}_{\mathbf{x}_0},$$
$$= \lambda_s s_0 \underbrace{\begin{bmatrix} F_1 \\ B_1 \end{bmatrix}}_{\mathbf{l}} \quad \text{(xvii)}$$

where F_1 and B_1 are the first column of the matrix F and B, respectively. Combining Equations (xvii) and (xv) allows us to rewrite Eq. (ix) as follows:

$$\mathbf{x} = \begin{bmatrix} U \\ S \\ V \end{bmatrix} W\mathbf{b} + \lambda_s s_0 \mathbf{l}, \quad \text{(xviii)}$$

which is the bottom part of Eq. (5) in the main submission. Eq. (xviii) provides a linear relationship between depths, scale, and translation and the rotation matrix. The explicit relationships are the following

$$\begin{aligned} \alpha_i(R) &= \mathbf{u}_i^\intercal W\mathbf{b} + \lambda_s s_o \mathbf{l}_i \\ s(R) &= SW\mathbf{b} + \lambda_s s_o \mathbf{l}_{n+1} \\ \mathbf{t}(R) &= VW\mathbf{b} + \lambda_s s_o \mathbf{l}_{\mathbf{t}}, \end{aligned} \quad \text{(xix)}$$

where $\mathbf{u}_i^\intercal$ is the i-th row of matrix U, $\mathbf{l}_j$ is the j-th entry of the vector $\mathbf{l}$, and $\mathbf{l}_{\mathbf{t}}$ corresponds to the last three entries of the vector $\mathbf{l}$. Specifically, the entries of vector $\mathbf{l}$ are

$$\mathbf{l} = \begin{bmatrix} \mathbf{l}_1 \\ \vdots \\ \mathbf{l}_n \\ \mathbf{l}_{n+1} \\ \mathbf{l}_{\mathbf{t}} \end{bmatrix} = \begin{bmatrix} F_{1,1} \\ \vdots \\ F_{n,1} \\ H_{1,1} \\ H_{2:4,1} \end{bmatrix}, \quad \text{(xx)}$$

where $H_{2:4,1}$ represent the last three entries of the first column of H. We can use these explicit relationships (*i.e.*, Eq. (i) and Eq. (xx)) to rewrite the main cost function as one depending only on rotation parameters. To do so as clearly as possible, we define

$$\begin{aligned}\mathbf{e}_i &= \alpha_i(R)\mathbf{r}_i - (R\mathbf{p}_i + \mathbf{t}(R) - s(R)\mathbf{c}_i) \\ &= \left(\mathbf{u}_i^\intercal W\mathbf{b} + \lambda s_0 F_{i,1}\right)\mathbf{r}_i - R\mathbf{p}_i \\ &\quad - \left(VW\mathbf{b} + \lambda s_0 H_{2:4,1}\right) \\ &\quad + \left(SW\mathbf{b} + \lambda s_0 H_{1,1}\right)\mathbf{c}_i \\ &= \underbrace{\mathbf{u}_i^\intercal W\mathbf{b}\mathbf{r}_i - R\mathbf{p}_i - VW\mathbf{b} + SW\mathbf{b}\mathbf{c}_i}_{\boldsymbol{\eta}_i}. \\ &\quad + \underbrace{\lambda s_0\left(F_{i,1}\mathbf{r}_i - H_{2:4,1} + H_{1,1}\mathbf{q}_i\right)}_{\mathbf{k}_i} \\ &= \boldsymbol{\eta}_i + \mathbf{k}_i\end{aligned} \tag{xxi}$$

As noted in gDLS [2] paper, $\boldsymbol{\eta}_i$ can be factored out as follows:

$$\begin{aligned}\boldsymbol{\eta}_i &= \left(\mathbf{r}_i\mathbf{r}_i^\intercal - I\right)\left(R\mathbf{p}_i - SW\mathbf{b}\mathbf{c}_i + VW\mathbf{b}\right) \\ &= \underbrace{\left(\mathbf{r}_i\mathbf{r}_i^\intercal - I\right)\left[L(\mathbf{p}_i) \quad -\mathbf{c}_i SL(\mathbf{b}) \quad VL(\mathbf{b})\right]}_{M_i}\text{vec}(R),\end{aligned} \tag{xxii}$$

where $\text{vec}(R)$ vectorizes a rotation matrix R, and $L(\mathbf{z})$ is a function that computes a matrix such that $R\mathbf{z} = L(\mathbf{z})\text{vec}(R)$. Since we use the rotation representation of Upnp [1], *i.e.*,

$$\text{vec}(R) = \left[q_1^2\ q_2^2\ q_3^2\ q_4^2\ q_1q_2\ q_1q_3\ q_1q_4\ q_2q_3\ q_2q_4\ q_3q_4\right]^\intercal, \tag{xxiii}$$

then the function $L(\cdot)$ is

$$L(\mathbf{z})^\intercal = \begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{z}_3 \\ \mathbf{z}_1 & -\mathbf{z}_2 & -\mathbf{z}_3 \\ -\mathbf{z}_1 & \mathbf{z}_2 & -\mathbf{z}_3 \\ -\mathbf{z}_1 & -\mathbf{z}_2 & \mathbf{z}_3 \\ 0 & -2\mathbf{z}_3 & 2\mathbf{z}_2 \\ 2\mathbf{z}_3 & 0 & -2\mathbf{z}_1 \\ -2\mathbf{z}_2 & 2\mathbf{z}_1 & 0 \\ 2\mathbf{z}_2 & 2\mathbf{z}_1 & 0 \\ 2\mathbf{z}_3 & 0 & 2\mathbf{z}_1 \\ 0 & 2\mathbf{z}_3 & 2\mathbf{z}_2 \end{bmatrix}. \tag{xxiv}$$

By substituting the relationships shown in Eq. (xix) and the factorizations shown in Eq. (xxii) into Eq. (i), we obtain the following relationships:

$$\begin{aligned}J'_{\text{gDLS}} &= \sum_{i=1}^n \mathbf{e}_i^\intercal\mathbf{e}_i = \sum_{i=1}^n \boldsymbol{\eta}_i^\intercal\boldsymbol{\eta}_i + 2\mathbf{k}_i^\intercal\boldsymbol{\eta}_i + \mathbf{k}_i^\intercal\mathbf{k}_i \\ &= \sum_{i=1}^n \text{vec}(R)^\intercal M_i^\intercal M_i\text{vec}(R) + 2\mathbf{k}_i^\intercal M_i\text{vec}(R) + \mathbf{k}_i^\intercal\mathbf{k}_i \\ &= \text{vec}(R)^\intercal \underbrace{\left(\sum_{i=1}^n M_i^\intercal M_i\right)}_{M_{\text{gDLS}}}\text{vec}(R) + \\ &\quad \underbrace{2\left(\sum_{i=1}^n \mathbf{k}_i^\intercal M_i\right)}_{\mathbf{d}_{\text{gDLS}}^\intercal}\text{vec}(R) + \underbrace{\sum_{i=1}^n \mathbf{k}_i^\intercal\mathbf{k}_i}_{k_{\text{gDLS}}} \\ &= \text{vec}(R)^\intercal M_{\text{gDLS}}\text{vec}(R) + 2\mathbf{d}_{\text{gDLS}}^\intercal\text{vec}(R) + k_{\text{gDLS}}\end{aligned}; \tag{xxv}$$

$$\begin{aligned}J'_s &= \lambda_s\left(s_0 - s(R)\right)^2 \\ &= \lambda_s\left(SL(\mathbf{b})\text{vec}(R) + \lambda_s s_0 H_{1,1} - s_0\right)^2 \\ &= \text{vec}(R)^\intercal \underbrace{\left(\lambda_s L(\mathbf{b})^\intercal S^\intercal SL(\mathbf{b})\right)}_{M_s}\text{vec}(R) + \\ &\quad \underbrace{2\,\lambda_s\left(s_0 - \lambda_s s_0 H_{1,1}\right)SL(\mathbf{b})}_{\mathbf{d}_s^\intercal}\text{vec}(R) + \\ &\quad \underbrace{\lambda_s\left(\lambda_s s_0 H_{1,1} - s_0\right)^2}_{k_s} \\ &= \text{vec}(R)^\intercal M_s\text{vec}(R) + 2\mathbf{d}_s^\intercal\text{vec}(R) + k_s\end{aligned} \quad ;\text{and} \tag{xxvi}$$

$$\begin{aligned}J'_g &= \lambda_g\|\mathbf{g}_\mathcal{Q} \times R\mathbf{g}_\mathcal{W}\|^2 \\ &= \text{vec}(R)^\intercal \underbrace{\left(\lambda_g L(\mathbf{g}_\mathcal{W})^\intercal \lfloor\mathbf{g}_\mathcal{Q}\rfloor_\times^\intercal \lfloor\mathbf{g}_\mathcal{Q}\rfloor_\times L(\mathbf{g}_\mathcal{W})\right)}_{M_g}\text{vec}(R) \\ &= \text{vec}(R)^\intercal M_g\text{vec}(R)\end{aligned}. \tag{xxvii}$$

The symbol $\lfloor\cdot\rfloor_\times$ indicates the skew symmetric matrix. By putting together the components of the cost, we end up with the final cost function

$$J' = J'_{\text{gDLS}} + J'_s + J'_g, \tag{xxviii}$$

which is Eq. (10) in the main submission.

References

[1] L. Kneip, H. Li, and Y. Seo. Upnp: An optimal o(n) solution to the absolute pose problem with universal applicability. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014. 3

[2] C. Sweeney, V. Fragoso, T. Höllerer, and M. Turk. gDLS: A scalable solution to the generalized pose and scale problem. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014. 1, 2, 3