

# Non-rigid Structure-from-Motion: Temporally-smooth Procrustean Alignment and Spatially-variant Deformation Modeling

## Supplementary Material

### Abstract

In this supplementary material, we provide additional materials in the following aspects: 1) solution of the Temporally-smooth Procrustean Alignment (TPA) model and verification of its validity; 2) detailed solution of the ADMM model; 3) additional explanations for spatially-variant deformation modeling; and 4) additional details of the experimental setup, supplementation of missing data experiments, and more visualization results.

### 1. Temporally-smooth Procrustean Alignment

In this section, we first provide details in solving our proposed Temporally-smooth Procrustean Alignment (TPA) module and then report additional experimental results.

#### 1.1. Algorithm for TPA

Denote the 3D shape sequence to be aligned as  $\{\mathbf{S}_i\}_{i=1}^F$ , we set the optimization target of TPA as:

$$\mathcal{L}_{tpa}^Q = \frac{1}{2} \sum_{i=1}^{F-1} \|\mathbf{Q}_i \mathbf{S}_i - \mathbf{Q}_{i+1} \mathbf{S}_{i+1}\|_F^2, \quad (1)$$

where we initialize  $\mathbf{Q}_i = \mathbf{I}_3$ , and  $\mathbf{I}_3 \in \mathbb{R}^{3 \times 3}$  denotes the identity matrix. We solve this optimization problem in a frame-by-frame manner. When  $1 < i < F$ , we transfer the optimization target as:

$$\begin{aligned} \mathcal{L}_{tpa}^{Q_i} &= \frac{1}{2} \sum_{k=i-1}^i \sum_{j=1}^P \|\mathbf{Q}_k \mathbf{s}_{k,j} - \mathbf{Q}_{k+1} \mathbf{s}_{k+1,j}\|_2^2 \\ &= \frac{1}{2} \sum_{j=1}^P (\|\mathbf{Q}_{i-1} \mathbf{s}_{i-1,j} - \mathbf{Q}_i \mathbf{s}_{i,j}\|_2^2 \\ &\quad + \|\mathbf{Q}_i \mathbf{s}_{i,j} - \mathbf{Q}_{i+1} \mathbf{s}_{i+1,j}\|_2^2), \end{aligned} \quad (2)$$

where  $\mathbf{s}_{i,j}$  is the  $j$ -th column of 3D shape  $\mathbf{S}_i$ . Taking the Lie algebra  $\phi_i$  corresponding to the rotation matrix  $\mathbf{Q}_i$  as the optimization variable, the gradient of model (1) can be calculated as:

$$\mathbf{g}_i = \frac{\partial \mathcal{L}_{tpa}^{Q_i}}{\partial \phi_i} = \frac{\partial \frac{1}{2} \sum_{j=1}^P (\|\mathbf{r}_{i,j}^{(1)}\|_2^2 + \|\mathbf{r}_{i,j}^{(0)}\|_2^2)}{\partial \phi_i} \quad (3)$$

$$\begin{aligned} &= \sum_{j=1}^P \left[ \left( \frac{\partial \mathbf{r}_{i,j}^{(1)}}{\partial \phi_i} \right)^T \mathbf{r}_{i,j}^{(1)} + \left( \frac{\partial \mathbf{r}_{i,j}^{(0)}}{\partial \phi_i} \right)^T \mathbf{r}_{i,j}^{(0)} \right] \\ &= \sum_{j=1}^P [(\mathbf{Q}_i \mathbf{s}_{i,j})^\wedge]^T (\mathbf{r}_{i,j}^{(1)} - \mathbf{r}_{i,j}^{(0)}), \end{aligned}$$

where  $\mathbf{r}_{i,j}^{(m)} = \mathbf{Q}_{i-m} \mathbf{s}_{i-m,j} - \mathbf{Q}_{i-m+1} \mathbf{s}_{i-m+1,j}$  is the residual vector, and  $(\mathbf{Q}_i \mathbf{s}_{i,j})^\wedge$  is an approximation to the derivative of the residual  $\mathbf{r}_{i,j}^{(m)}$  with respect to the Lie algebra  $\phi_i$ .

$(\mathbf{a})^\wedge$  is the skew-symmetric matrix of vector  $\mathbf{a}$ , *i.e.*:

$$\left( \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \right)^\wedge = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \quad (4)$$

Similarly, when  $i = 1$  or  $F$ , the gradient  $\mathbf{g}_i$  can be calculated as:

$$\mathbf{g}_i = \begin{cases} -\sum_{j=1}^P [(\mathbf{Q}_i \mathbf{s}_{i,j})^\wedge]^T \mathbf{r}_{i,j}^{(0)}, & i = 1, \\ \sum_{j=1}^P [(\mathbf{Q}_i \mathbf{s}_{i,j})^\wedge]^T \mathbf{r}_{i,j}^{(1)}, & i = F. \end{cases} \quad (5)$$

The gradient  $\mathbf{g}_i$  can be expressed as the product of the Jacobian matrix and the residual, *i.e.*,  $\mathbf{g}_i = \mathbf{J}_i^T \mathbf{r}_i$ .  $\mathbf{J}_i \in \mathbb{R}^{3P \times 3}$  is the Jacobian matrix as follows:

$$\mathbf{J}_i = [(\mathbf{Q}_i \mathbf{s}_{i,1})^\wedge \cdots (\mathbf{Q}_i \mathbf{s}_{i,P})^\wedge]^T, \quad (6)$$

where  $\mathbf{r}_i$  is the column vector stacked from residuals  $\{r_{i,j}^{(1)} - r_{i,j}^{(0)}\}_{j=1}^P$  or  $\{r_{i,j}^{(m)}\}_{j=1}^P$ . After obtaining the Jacobian matrix, we can compute the approximation of the second-order Hessian matrix as  $\mathcal{H}_i = \mathbf{J}_i^T \mathbf{J}_i$  and use numerical optimization methods such as the Levenberg-Marquardt (LM) algorithm [12] to solve for the rotation matrix  $\mathbf{Q} = \text{diag}(\mathbf{Q}_i)$ .

### 1.2. Experiments for TPA

In this subsection, we verify the effectiveness of the TPA module. Let's revisit the setting of the experiments. We randomly sample Lie algebra  $p_n \in \mathbb{R}^{3 \times F}$  from Gaussian distribution  $\mathcal{N}(0, 0.1)$ , and then map it to a block diagonal matrix  $\mathbf{R}_d \in \mathbb{R}^{3F \times 3F}$  consisting of 3D rotations using the Rodrigues' rotation formula. We define the shape sequence disrupted by random rotations as  $\mathbf{S}_{noise} = \mathbf{R}_d \mathbf{S}_{gt}$ , where  $\mathbf{S}_{gt} \in \mathbb{R}^{3F \times P}$  is the GT sequence in the world coordinate. We realign the sequence using the TPA and GPA [10] modules and compare their results. We performed these experiments on the Pickup, Yoga, Stretch, and Drink sequences in MoCap benchmark and the results are shown in Fig. 1.

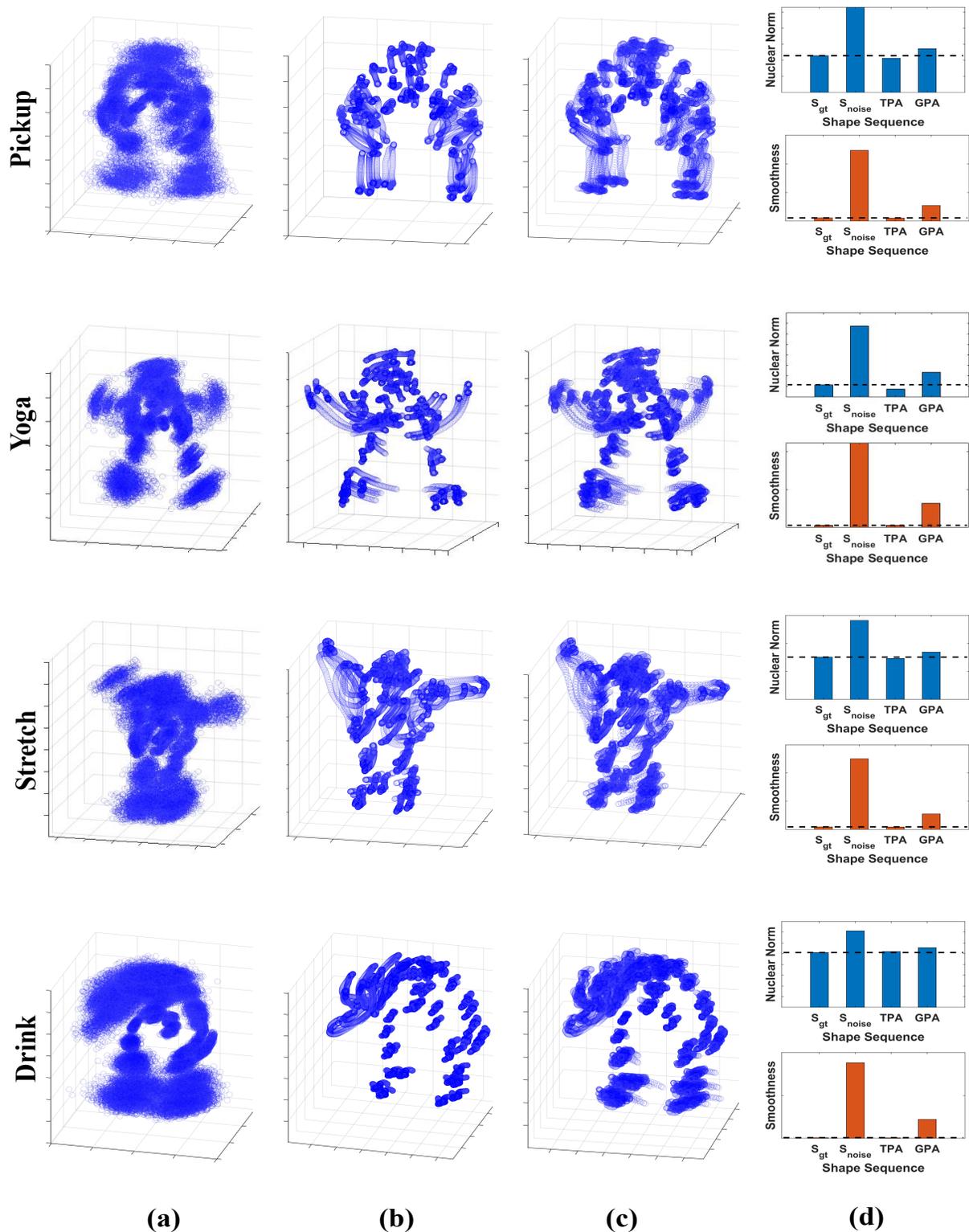


Figure 1. Experiments for testing the alignment capability of TPA module. (a) Visualization of the shape sequences  $S_{noise}$ . (b) Visualization of the TPA-aligned shape sequences. (c) Visualization of the GPA-aligned shape sequences. (d) The low-rank and smoothing properties of GT non-rigid sequences  $S_{gt}$ , randomly rotated disrupted sequences  $S_{noise}$ , and sequences aligned by TPA or GPA.

We centralized all the 3D shapes in the sequence and plotted them in the same coordinate. Column (a) in Fig. 1 is the disrupted sequence  $\mathbf{S}_{noise}$ . Columns (b) and (c) are sequences after TPA or GPA alignment, respectively. Column (d) shows the low-rank and smoothing properties of real sequences  $\mathbf{S}_{gt}$ , noisy sequences  $\mathbf{S}_{noise}$ , TPA-aligned sequences, and GPA-aligned sequences. Fig. 1 demonstrates that the TPA module can efficiently align the 3D shape sequences. Compared with GPA, the TPA-aligned sequences are more similar to the real 3D shape sequences in terms of their low-rank and smoothing properties.

Non-rigid deformation mixed with rigid motion cannot be accurately estimated using the low-rank constraint. As manifested in Fig. 1 (d), shape sequences after alignment by the TPA module have lower nuclear norm and smoothness. From one hand, this demonstrates that the TPA module can effectively separate rigid motion from non-rigid deformation. From the other hand, the properties of the TPA-aligned sequence make it can be better recovered by the low-rank and smoothing constraints. In summary, we utilize the low-rank constraint to recover the TPA-aligned shape sequence, rather than directly applying the low-rank constraint under the coordinate defined by the estimated camera motion [3, 8, 9].

## 2. ADMM Model Solution

### 2.1. Model without Occlusion

In this section, we provide the completed solution of our proposed method. Let's first recall the Lagrangian model to be solved:

$$\begin{aligned} \min_{\Omega} \mathcal{L} = & \frac{\mu_1}{2} \|\mathbf{W} - \Pi \mathbf{S}\|_F^2 + \mu_2 \|\check{\mathbf{S}}^\# \|_* + \frac{\mu_3}{2} \sum_{i=1}^{F-1} \|\mathbf{Q}_i \tilde{\mathbf{S}}_i - \mathbf{Q}_{i+1} \tilde{\mathbf{S}}_{i+1}\|_F^2 \\ & + \frac{\beta}{2} \|\check{\mathbf{S}}^\# - g(\hat{\mathbf{S}}\mathbf{\Lambda})\|_F^2 + \langle \mathbf{Y}_1, \check{\mathbf{S}}^\# - g(\hat{\mathbf{S}}\mathbf{\Lambda}) \rangle \\ & + \frac{\beta}{2} \|\hat{\mathbf{S}} - \mathbf{Q}\tilde{\mathbf{S}}\|_F^2 + \langle \mathbf{Y}_2, \hat{\mathbf{S}} - \mathbf{Q}\tilde{\mathbf{S}} \rangle \\ & + \frac{\beta}{2} \|\tilde{\mathbf{S}} - \mathbf{R}_p \mathbf{S}\|_F^2 + \langle \mathbf{Y}_3, \tilde{\mathbf{S}} - \mathbf{R}_p \mathbf{S} \rangle, \end{aligned} \quad (7)$$

where  $\Omega = \{\mathbf{S}, \tilde{\mathbf{S}}, \hat{\mathbf{S}}, \check{\mathbf{S}}^\#, \mathbf{Q}\}$  denotes the variables to be updated,  $\{\mathbf{Y}_n\}_{n=1}^3$  are the Lagrange multipliers. We then give the update formula for each optimization variable in  $\Omega$ .

**Solution for  $\check{\mathbf{S}}^\#$ .** Selecting all the optimization terms in model (7) that are related to  $\check{\mathbf{S}}^\#$ , the optimization model for  $\check{\mathbf{S}}^\#$  is obtained as:

$$\begin{aligned} \check{\mathbf{S}}^\# = & \arg \min_{\check{\mathbf{S}}^\#} \mu_2 \|\check{\mathbf{S}}^\# \|_* + \frac{\beta}{2} \|\check{\mathbf{S}}^\# - g(\hat{\mathbf{S}}\mathbf{\Lambda})\|_F^2 \\ & + \langle \mathbf{Y}_1, \check{\mathbf{S}}^\# - g(\hat{\mathbf{S}}\mathbf{\Lambda}) \rangle \\ = & \arg \min_{\check{\mathbf{S}}^\#} \frac{\beta}{2} \left\| \check{\mathbf{S}}^\# - \left( g(\hat{\mathbf{S}}\mathbf{\Lambda}) - \frac{1}{\beta} \mathbf{Y}_1 \right) \right\|_F^2 + \mu_2 \|\check{\mathbf{S}}^\# \|_* . \end{aligned} \quad (8)$$

We use the method in [8] to obtain the closed-form solution. We first define the soft-thresholding function  $S_\tau(\sigma) =$

$sign(\sigma) \max(|\sigma| - \tau, 0)$ . Then the closed-form solution  $\check{\mathbf{S}}^\#$  can be given by:

$$\begin{aligned} \check{\mathbf{S}}^\# = & \mathbf{U} \mathcal{S}_{\frac{\Theta \mu_2}{\beta}}(\mathbf{\Sigma}) \mathbf{V}^T, \\ \mathbf{U}, \mathbf{\Sigma}, \mathbf{V} = & \mathbf{SVD} \left( g(\hat{\mathbf{S}}\mathbf{\Lambda}) - \frac{1}{\beta} \mathbf{Y}_1 \right), \end{aligned} \quad (9)$$

where  $\Theta$  is the weight set for different singular values. The larger the singular value tends to be the more significant, and it should correspond to a smaller weight. The weight  $\Theta$  in [8] is set as:

$$\Theta_j = \frac{\xi}{\sigma_j(\mathbf{X}) + \gamma}, \quad (10)$$

where  $\sigma_j(\mathbf{X})$  is the singular value of  $\mathbf{X}$ ,  $\xi$  is a positive number and  $\gamma = 1e^{-6}$ . Here we use the relative magnitude of the singular values to adjust the weight setting, *i.e.*:

$$\tilde{\Theta}_j = \begin{cases} \frac{\xi \cdot \Theta_j}{\sum_{i=1}^{K_s} \Theta_i}, & 1 \leq j \leq K_s, \\ 0, & K_s < j. \end{cases} \quad (11)$$

We set the shape basis dimension  $K_s$  according to the assumption of linear basis combination and truncate the singular values. We can better measure the importance of different singular values by adjusting the weights through normalization.

**Solution for  $\hat{\mathbf{S}}$ .** The solution model for the optimization variable  $\hat{\mathbf{S}}$  can be expressed as follows:

$$\begin{aligned} \hat{\mathbf{S}} = & \arg \min_{\hat{\mathbf{S}}} \frac{\beta}{2} \|\check{\mathbf{S}}^\# - g(\hat{\mathbf{S}}\mathbf{\Lambda})\|_F^2 + \langle \mathbf{Y}_1, \check{\mathbf{S}}^\# - g(\hat{\mathbf{S}}\mathbf{\Lambda}) \rangle \\ & + \frac{\beta}{2} \|\hat{\mathbf{S}} - \mathbf{Q}\tilde{\mathbf{S}}\|_F^2 + \langle \mathbf{Y}_2, \hat{\mathbf{S}} - \mathbf{Q}\tilde{\mathbf{S}} \rangle, \end{aligned} \quad (12)$$

where  $g(\cdot)$  is an invertible linear operator, so we can rewrite the above equation in a more easily solvable form as:

$$\begin{aligned} \hat{\mathbf{S}} = & \arg \min_{\hat{\mathbf{S}}} \frac{\beta}{2} \|\hat{\mathbf{S}} - \mathbf{Q}\tilde{\mathbf{S}}\|_F^2 + \langle \mathbf{Y}_2, \hat{\mathbf{S}} - \mathbf{Q}\tilde{\mathbf{S}} \rangle \\ & + \frac{\beta}{2} \|g^{-1}(\check{\mathbf{S}}^\#) - \hat{\mathbf{S}}\mathbf{\Lambda}\|_F^2 + \langle g^{-1}(\mathbf{Y}_1), g^{-1}(\check{\mathbf{S}}^\#) - \hat{\mathbf{S}}\mathbf{\Lambda} \rangle. \end{aligned} \quad (13)$$

The closed-form solution of  $\hat{\mathbf{S}}$  can be computed by taking the derivative of the model (13) and equating it to zero:

$$\begin{aligned} \hat{\mathbf{S}}(\mathbf{I}_P + \mathbf{\Lambda}\mathbf{\Lambda}^T) = & \\ \mathbf{Q}\tilde{\mathbf{S}} - \frac{1}{\beta} \mathbf{Y}_2 + g^{-1}(\check{\mathbf{S}}^\#)\mathbf{\Lambda}^T + \frac{1}{\beta} g^{-1}(\mathbf{Y}_1)\mathbf{\Lambda}^T. \end{aligned} \quad (14)$$

**Solution for  $\tilde{\mathbf{S}}$ .** First, we define the first-order smoothing matrix  $\mathbf{H} \in \mathbb{R}^{3F \times 3F}$  as in [3]:

$$\mathbf{H}_{i,j} = \begin{cases} 1, & j = i, i = 1, \dots, 3(F-1), \\ -1, & j = i+3, i = 1, \dots, 3(F-1), \\ 0, & \text{Otherwise.} \end{cases} \quad (15)$$

---

**Algorithm 1** ADMM Optimization Algorithm
 

---

- 1: **Input:** Initialize  $\mathbf{S}, \tilde{\mathbf{S}}, \hat{\mathbf{S}}, \check{\mathbf{S}}^\#, \mathbf{R}_p, \mathbf{Q}, \mathbf{A}, \beta, \beta_{max}, \lambda$  and  $\epsilon = 1e^{-6}$
  - 2: **repeat**
  - 3:   Update  $\check{\mathbf{S}}^\#$  by Eq. (9) yields  $\check{\mathbf{S}}_-^\#$
  - 4:   Update  $\hat{\mathbf{S}}$  by Eq. (14) yields  $\hat{\mathbf{S}}_-$
  - 5:   Update  $\tilde{\mathbf{S}}$  by Eq. (18) yields  $\tilde{\mathbf{S}}_-$
  - 6:   Update  $\mathbf{S}$  by Eq. (20) yields  $\mathbf{S}_-$
  - 7:   **repeat**
  - 8:     Calculate the gradient by Eq. (22) and update  $\mathbf{Q}$  by LM algorithm yields  $\mathbf{Q}_-$
  - 9:     **until Convergence**
  - 10:   Update  $\mathbf{Y}_1$  by  $\mathbf{Y}_1 + \beta (\check{\mathbf{S}}_-^\# - g(\hat{\mathbf{S}}_- \mathbf{A}))$
  - 11:   Update  $\mathbf{Y}_2$  by  $\mathbf{Y}_2 + \beta (\hat{\mathbf{S}}_- - \mathbf{Q} \tilde{\mathbf{S}}_-)$
  - 12:   Update  $\mathbf{Y}_3$  by  $\mathbf{Y}_3 + \beta (\tilde{\mathbf{S}}_- - \mathbf{R}_p \mathbf{S}_-)$
  - 13:   Update  $\beta$  as  $\min(\beta_{max}, \lambda \beta)$
  - 14: **until**  $\|\mathbf{S}_- - \mathbf{S}\|_\infty < \epsilon$
- Output:**  $\mathbf{S}, \mathbf{Q}$
- 

By introducing the matrix  $\mathbf{H}$ , we can denote the TPA module equivalently as:

$$\sum_{i=1}^{F-1} \left\| \mathbf{Q}_i \tilde{\mathbf{S}}_i - \mathbf{Q}_{i+1} \tilde{\mathbf{S}}_{i+1} \right\|_F^2 = \left\| \mathbf{H} \mathbf{Q} \tilde{\mathbf{S}} \right\|_F^2. \quad (16)$$

Then the optimization model for the variable  $\tilde{\mathbf{S}}$  can be expressed as follows:

$$\begin{aligned} \tilde{\mathbf{S}} = \arg \min_{\tilde{\mathbf{S}}} & \frac{\mu_3}{2} \left\| \mathbf{H} \mathbf{Q} \tilde{\mathbf{S}} \right\|_F^2 \\ & + \frac{\beta}{2} \left\| \hat{\mathbf{S}} - \mathbf{Q} \tilde{\mathbf{S}} \right\|_F^2 + \langle \mathbf{Y}_2, \hat{\mathbf{S}} - \mathbf{Q} \tilde{\mathbf{S}} \rangle \\ & + \frac{\beta}{2} \left\| \tilde{\mathbf{S}} - \mathbf{R}_p \mathbf{S} \right\|_F^2 + \langle \mathbf{Y}_3, \tilde{\mathbf{S}} - \mathbf{R}_p \mathbf{S} \rangle. \end{aligned} \quad (17)$$

Calculating the derivative of model (17) and equating it to zero yields the closed-form solution of  $\tilde{\mathbf{S}}$  as follows:

$$\begin{aligned} \left( \frac{\mu_3}{\beta} \mathbf{Q}^T \mathbf{H}^T \mathbf{H} \mathbf{Q} + 2\mathbf{I}_{3F} \right) \tilde{\mathbf{S}} = \\ \mathbf{Q}^T \hat{\mathbf{S}} + \frac{1}{\beta} \mathbf{Q}^T \mathbf{Y}_2 + \mathbf{R}_p \mathbf{S} - \frac{1}{\beta} \mathbf{Y}_3. \end{aligned} \quad (18)$$

**Solution for S.** The optimization model for the variable  $\mathbf{S}$  can be expressed as follows:

$$\begin{aligned} \mathbf{S} = \arg \min_{\mathbf{S}} & \frac{\mu_1}{2} \left\| \mathbf{W} - \mathbf{\Pi} \mathbf{S} \right\|_F^2 \\ & + \frac{\beta}{2} \left\| \tilde{\mathbf{S}} - \mathbf{R}_p \mathbf{S} \right\|_F^2 + \langle \mathbf{Y}_3, \tilde{\mathbf{S}} - \mathbf{R}_p \mathbf{S} \rangle. \end{aligned} \quad (19)$$

Calculating the derivative of model (19) and equating it to

zero yields the closed-form solution of  $\mathbf{S}$  as follows:

$$\begin{aligned} \left( \frac{\mu_1}{\beta} \mathbf{\Pi}^T \mathbf{\Pi} + \mathbf{I}_{3F} \right) \mathbf{S} = \\ \frac{\mu_1}{\beta} \mathbf{\Pi}^T \mathbf{W} + \mathbf{R}_p^T \tilde{\mathbf{S}} + \frac{1}{\beta} \mathbf{R}_p^T \mathbf{Y}_3. \end{aligned} \quad (20)$$

**Solution for Q.** We have already discussed how to solve the TPA module in Sec. 1.1. But in model (7), the variables in  $\Omega$  are coupled and need to be optimized alternatively, so the updating formula for  $\mathbf{Q}$  needs adjustments. The optimization terms in model (7) containing the optimization variable  $\mathbf{Q}_i, i = 1, \dots, F$  are:

$$\begin{aligned} \mathbf{Q}_i = \arg \min_{\mathbf{Q}_i} & \frac{\mu_3}{2} \sum_{i=1}^{F-1} \left\| \mathbf{Q}_i \tilde{\mathbf{S}}_i - \mathbf{Q}_{i+1} \tilde{\mathbf{S}}_{i+1} \right\|_F^2 \\ & + \frac{\beta}{2} \left\| \hat{\mathbf{S}} - \mathbf{Q} \tilde{\mathbf{S}} \right\|_F^2 + \langle \mathbf{Y}_2, \hat{\mathbf{S}} - \mathbf{Q} \tilde{\mathbf{S}} \rangle \\ = \arg \min_{\mathbf{Q}_i} & \frac{\mu_3}{2} \sum_{k=i-1}^i \sum_{j=1}^P \left\| \mathbf{Q}_k \tilde{\mathbf{s}}_{k,j} - \mathbf{Q}_{k+1} \tilde{\mathbf{s}}_{k+1,j} \right\|_2^2 \\ & + \frac{\beta}{2} \sum_{j=1}^P \left( \left\| \hat{\mathbf{s}}_{i,j} - \mathbf{Q}_i \tilde{\mathbf{s}}_{i,j} \right\|_2^2 + \langle \mathbf{Y}_{2,j}^i, \hat{\mathbf{s}}_{i,j} - \mathbf{Q}_i \tilde{\mathbf{s}}_{i,j} \rangle \right), \end{aligned} \quad (21)$$

where  $\mathbf{Y}_{2,j}^i$  is the rows  $3i - 2$  through  $3i$  and  $j$ -th column of  $\mathbf{Y}_2$ . We denote  $\hat{\mathbf{s}}_{i,j} - \mathbf{Q}_i \tilde{\mathbf{s}}_{i,j}$  as  $\hat{\mathbf{r}}_{i,j}$  and compute the gradient of the Lie algebra  $\phi_i$  corresponding to  $\mathbf{Q}_i$  by imitating Eq. (3):

$$\begin{aligned} \hat{\mathbf{g}}_i = & \frac{\partial \frac{\mu_3}{2} \sum_{j=1}^P \left( \left\| \mathbf{r}_{i,j}^{(1)} \right\|_2^2 + \left\| \mathbf{r}_{i,j}^{(0)} \right\|_2^2 \right)}{\partial \phi_i} \\ & + \frac{\partial \sum_{j=1}^P \left( \frac{\beta}{2} \left\| \hat{\mathbf{r}}_{i,j} \right\|_2^2 + \langle \mathbf{Y}_{2,j}^i, \hat{\mathbf{r}}_{i,j} \rangle \right)}{\partial \phi_i} \\ = & \mu_3 \sum_{j=1}^P \left[ \left( \frac{\partial \mathbf{r}_{i,j}^{(1)}}{\partial \phi_i} \right)^T \mathbf{r}_{i,j}^{(1)} + \left( \frac{\partial \mathbf{r}_{i,j}^{(0)}}{\partial \phi_i} \right)^T \mathbf{r}_{i,j}^{(0)} \right] \\ & + \sum_{j=1}^P \left[ \beta \left( \frac{\partial \hat{\mathbf{r}}_{i,j}}{\partial \phi_i} \right)^T \hat{\mathbf{r}}_{i,j} + \left( \frac{\partial \hat{\mathbf{r}}_{i,j}}{\partial \phi_i} \right)^T \mathbf{Y}_{2,j}^i \right] \\ = & \sum_{j=1}^P \left[ \left( \mathbf{Q}_i \tilde{\mathbf{s}}_{i,j} \right)^\wedge \right]^T \left[ \mu_3 \left( \mathbf{r}_{i,j}^{(1)} - \mathbf{r}_{i,j}^{(0)} \right) + \beta \hat{\mathbf{r}}_{i,j} + \mathbf{Y}_{2,j}^i \right]. \end{aligned} \quad (22)$$

Therefore, we can still update  $\mathbf{Q}_i$  using the TPA optimization algorithm in Sec. 1.1, and only need to adjust the residual vector  $\mathbf{r}_i$  to satisfy the descent direction.

After discussing the solution formulas for each variable, we give the complete optimization Algorithm 1. Since no closed-form solution for updating  $\mathbf{Q}$  exists, another iterative optimization must be embedded in the ADMM algorithm. In experiments, we found that only 1 to 10 iterations are needed to update  $\mathbf{Q}$  well.

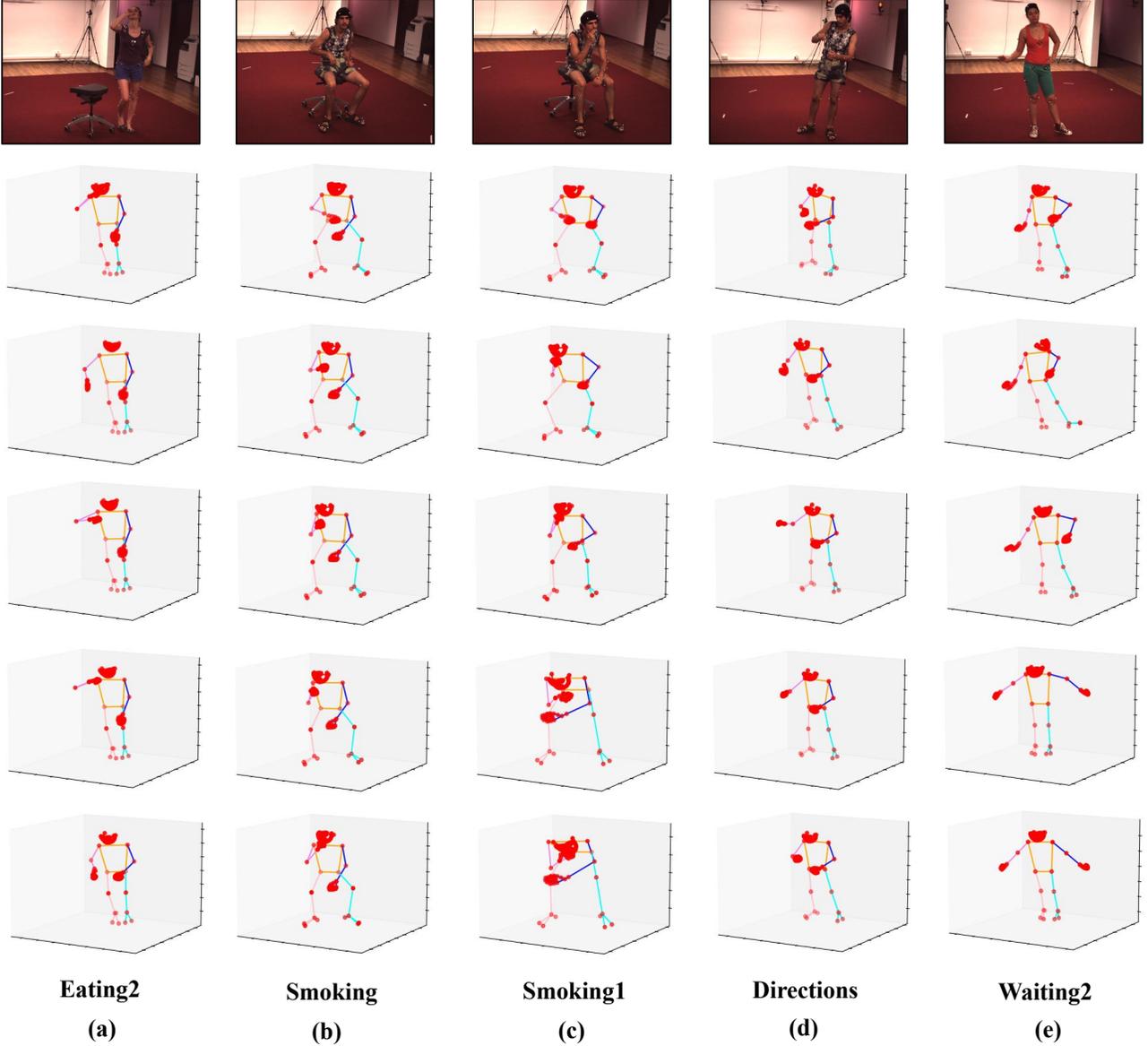


Figure 2. Visualization of the 3D shapes in the H3WB NRSfM dataset. The first row shows the pictures in the original H36M dataset, and the other rows display the GT 3D shapes in each sequence.

## 2.2. Model with Occlusion

In real-world scenes, the captured images are often obscured and it is difficult to observe all the keypoints in each frame. Assume  $\mathbf{o}_i \in \mathbb{R}^{1 \times P}$ ,  $i = 1, \dots, F$  is the mask vectors and  $o_{i,j} = 1$  if  $j$ -th point in the  $i$ -th frame is visible, otherwise 0. Then to solve the occlusion problem, we introduce the mask matrix  $\mathbf{O} \in \mathbb{R}^{2F \times P}$  and correct the data term constraint  $\mathcal{F}(\cdot)$  as follows:

$$\frac{\mu_1}{2} \|\mathbf{O} \odot (\mathbf{W} - \mathbf{IIS})\|_F^2, \quad (23)$$

where  $\mathbf{O} = [\mathbf{1}_2 \otimes \mathbf{o}_1; \dots; \mathbf{1}_2 \otimes \mathbf{o}_F] \in \mathbb{R}^{2F \times P}$ . Since the occlusion is different for each frame, centralization for the observation matrix  $\mathbf{W}$  does not guarantee that the translation between shapes is eliminated. Therefore re-centering of the shapes is required before alignment using the TPA module, which only requires modification of  $\tilde{\mathbf{S}}$ :

$$\tilde{\mathbf{S}}_i = \mathbf{R}_{pi} \mathbf{S}_i \mathbf{T}, \quad i = 1, \dots, F, \quad (24)$$

where the definition of  $\mathbf{T} = \mathbf{I} - \frac{1}{P} \mathbf{1} \mathbf{1}^T$  is translation removal matrix. The completed model after adding occlusion is as follows:

$$\begin{aligned}
& \min_{\mathbf{S}, \mathbf{Q}} \frac{\mu_1}{2} \|\mathbf{O} \odot (\mathbf{W} - \mathbf{\Pi}\mathbf{S})\|_F^2 + \mu_2 \|\check{\mathbf{S}}^\# \|_* + \\
& \frac{\mu_3}{2} \sum_{i=1}^{F-1} \|\mathbf{Q}_i \tilde{\mathbf{S}}_i - \mathbf{Q}_{i+1} \tilde{\mathbf{S}}_{i+1}\|_F^2 \\
& s.t. \begin{cases} \check{\mathbf{S}}^\# = g(\hat{\mathbf{S}}\mathbf{\Lambda}) \\ \tilde{\mathbf{S}}_i = \mathbf{Q}_i \tilde{\mathbf{S}}_i, i = 1, \dots, F \\ \tilde{\mathbf{S}}_i = \mathbf{R}_{pi} \mathbf{S}_i \mathbf{T}, i = 1, \dots, F \end{cases}
\end{aligned} \quad (25)$$

The update formulas for variables  $\Omega$  can be obtained by imitating the above solution procedure. Since the improvement of the model is only related to the optimization variable  $\tilde{\mathbf{S}}, \mathbf{S}$ , we only need to adjust their update formulas.

**Solution for  $\tilde{\mathbf{S}}$  under occlusion.**  $\mathbf{S}$  in model (25) needs to be centralized and then transformed by rotation  $\mathbf{R}_{pi}$  to get  $\tilde{\mathbf{S}}$ . Thus we only need to replace  $\mathbf{S}$  in Eq. (18) with  $\mathbf{S}\mathbf{T}$ , *i.e.*:

$$\begin{aligned}
& \left( \frac{\mu_3}{\beta} \mathbf{Q}^T \mathbf{H}^T \mathbf{H} \mathbf{Q} + 2\mathbf{I}_{3F} \right) \tilde{\mathbf{S}} = \\
& \mathbf{Q}^T \hat{\mathbf{S}} + \frac{1}{\beta} \mathbf{Q}^T \mathbf{Y}_2 + \mathbf{R}_p \mathbf{S} \mathbf{T} - \frac{1}{\beta} \mathbf{Y}_3.
\end{aligned} \quad (26)$$

**Solution for  $\mathbf{S}$  under occlusion.** We solve the 3D shape  $\mathbf{S}_i$  frame-by-frame and the optimization model is as follows:

$$\begin{aligned}
\mathbf{S}_i = \arg \min_{\mathbf{S}_i} & \frac{\mu_1}{2} \|\mathbf{O}_i \odot (\mathbf{W}_i - \mathbf{\Pi}_i \mathbf{S}_i)\|_F^2 \\
& + \frac{\beta}{2} \|\tilde{\mathbf{S}}_i - \mathbf{R}_{pi} \mathbf{S}_i \mathbf{T}\|_F^2 + \langle \mathbf{Y}_3^i, \tilde{\mathbf{S}}_i - \mathbf{R}_{pi} \mathbf{S}_i \mathbf{T} \rangle,
\end{aligned} \quad (27)$$

where  $\mathbf{O}_i = \mathbf{1}_2 \otimes \mathbf{o}_i$ . We define  $\mathbf{M}_i = \text{diag}(\mathbf{o}_i) \in \mathbb{R}^{P \times P}$ , then model (27) can be rewritten as:

$$\begin{aligned}
\mathbf{S}_i = \arg \min_{\mathbf{S}_i} & \frac{\mu_1}{2} \|(\mathbf{W}_i - \mathbf{\Pi}_i \mathbf{S}_i) \mathbf{M}_i\|_F^2 \\
& + \frac{\beta}{2} \|\tilde{\mathbf{S}}_i - \mathbf{R}_{pi} \mathbf{S}_i \mathbf{T}\|_F^2 + \langle \mathbf{Y}_3^i, \tilde{\mathbf{S}}_i - \mathbf{R}_{pi} \mathbf{S}_i \mathbf{T} \rangle.
\end{aligned} \quad (28)$$

Calculating the derivative of model (28) and equating it to zero yields the solution of  $\mathbf{S}_i$  as follows:

$$\begin{aligned}
& \frac{\mu_1}{\beta} \mathbf{\Pi}_i^T \mathbf{\Pi}_i \mathbf{S}_i \mathbf{M}_i^2 + \mathbf{S}_i \mathbf{T}^2 = \\
& \frac{\mu_1}{\beta} \mathbf{\Pi}_i^T \mathbf{W}_i \mathbf{M}_i^2 + \mathbf{R}_{pi}^T \tilde{\mathbf{S}}_i \mathbf{T} + \frac{1}{\beta} \mathbf{R}_{pi}^T \mathbf{Y}_3^i \mathbf{T}.
\end{aligned} \quad (29)$$

In order to calculate the closed-form solution of  $\mathbf{S}_i$ , we use the property  $\text{vec}(\mathbf{A}\mathbf{X}\mathbf{B}) = (\mathbf{B}^T \otimes \mathbf{A})\text{vec}(\mathbf{X})$  to equivalently represent Eq. (29) as:

$$\begin{aligned}
& \left[ \frac{\mu_1}{\beta} (\mathbf{M}_i^2 \otimes \mathbf{\Pi}_i^T \mathbf{\Pi}_i) + (\mathbf{T}^2 \otimes \mathbf{I}_3) \right] \text{vec}(\mathbf{S}_i) = \\
& \text{vec} \left( \frac{\mu_1}{\beta} \mathbf{\Pi}_i^T \mathbf{W}_i \mathbf{M}_i^2 + \mathbf{R}_{pi}^T \tilde{\mathbf{S}}_i \mathbf{T} + \frac{1}{\beta} \mathbf{R}_{pi}^T \mathbf{Y}_3^i \mathbf{T} \right).
\end{aligned} \quad (30)$$

The update formulas for the other optimization variables in Model (25) remain unchanged.

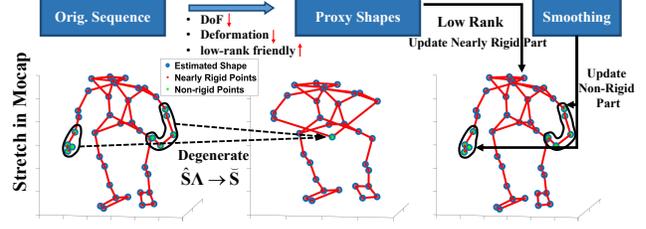


Figure 3. Illustration of the proxy shape construction and shape update principle.

### 3. Supplement to Spatially-variant Modeling

In this section, We provide an additional explanation of the construction of the proxy shape and how it plays a role in optimization, refer to Fig. 3.

**1) Definition of proxy shape.** We divide the non-rigid object into two regions with different deformation degrees through frequency domain analysis of 3D trajectories. Then by spatial weighting, we merge the single super point degenerated from “Non-Rigid” part with “Nearly Rigid” points to form the **proxy shapes**, *i.e.*,  $\check{\mathbf{S}} = \hat{\mathbf{S}}\mathbf{\Lambda}$ . The weight matrix  $\mathbf{\Lambda}$  defined by the feature mapping  $\phi(\cdot)$  (Eq. (12) in them main text) is rank-deficient (when  $\alpha_r < 1$ ), so the resultant proxy sequence  $\check{\mathbf{S}}$  has fewer degrees of freedom than  $\hat{\mathbf{S}}$  and more satisfies the low-rank constraint.

**2) How SWNN works.** We enforce the low-rank constraint on proxy shapes and update new shapes by Eq. (14). The new  $\hat{\mathbf{S}}$  is mainly composed of  $\check{\mathbf{S}}$  after low-rank regularization and  $\tilde{\mathbf{S}}$  after smoothing regularization. Since  $\mathbf{\Lambda}$  is rank-deficient, the low-rank constrained  $\check{\mathbf{S}}$  after inverse transformation only retains the structural information of nearly rigid part. Therefore,  $\hat{\mathbf{S}}$  updates nearly rigid part mainly through low-rank, and non-rigid part is fine-tuned by  $\tilde{\mathbf{S}}$ .

Our method combines low-rank and smoothing constraints through spatial weighting, which improves the accuracy of the reconstruction by avoiding the over-penalization of localized drastic deformations with low-rank constraint. To verify the validity of the combination, we designed the comparison experiment presented in Fig. 4. We compare the reconstruction error of Eq. (7) with methods using only low-rank or smoothing constraints on the NRSfM Challenge dataset. For the low-rank-only method, we compared with R-BMM [8], which is an improvement on the “prior-free” classical method BMM [3]. For the approach using only the smooth prior, we removed the low-rank constraint term from Eq. (7) and compared with it, *i.e.*:

$$\begin{aligned}
\min_{\check{\mathbf{S}}, \mathbf{S}, \mathbf{Q}} \mathcal{L} = & \frac{\mu_1}{2} \|\mathbf{W} - \mathbf{\Pi}\mathbf{S}\|_F^2 + \frac{\mu_2}{2} \sum_{i=1}^{F-1} \|\mathbf{Q}_i \tilde{\mathbf{S}}_i - \mathbf{Q}_{i+1} \tilde{\mathbf{S}}_{i+1}\|_F^2 \\
& + \frac{\beta}{2} \|\tilde{\mathbf{S}} - \mathbf{R}_p \mathbf{S}\|_F^2 + \langle \mathbf{Y}_1, \tilde{\mathbf{S}} - \mathbf{R}_p \mathbf{S} \rangle.
\end{aligned} \quad (31)$$

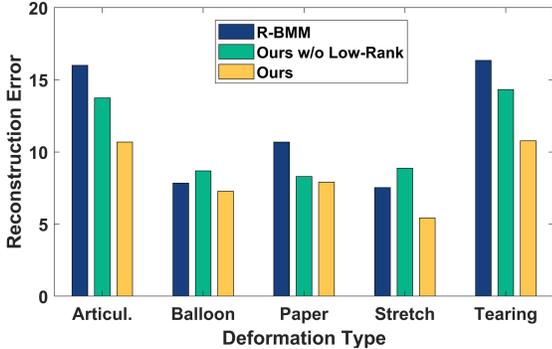


Figure 4. Comparison of the reconstruction errors of our model with methods using only low-rank or smoothing constraints on the NRSfM Challenge dataset.

As shown in Fig. 4, the reconstruction error of our method is lower than that of the low-rank-only/smooth-only method on all types of deformation, and there is a significant improvement on Articul., Balloon and Tearing.

Our approach can effectively couple low-rank and smoothing constraints to improve the stability of the algorithm over different types of deformations. Moreover, statistical prior methods tend to be sensitive to the statistical properties of the data, which are also related to the complexity of the corresponding object deformation. In other words, it is not enough to only mine the smooth prior to complement the low-rank constraints, so finding constraints that are more general and insensitive to the statistical properties of the data is a feasible direction for improvement.

## 4. Additional Experiments

In this section, we first add the details of dataset processing and algorithm implementation and then report additional experiment results.

### 4.1. Implementation Details

For parameters in ADMM Algorithm 1, we refer to [9] to initialize  $\beta = 1e^{-4}$ ,  $\beta_{max} = 1e^{10}$ ,  $\Lambda = 1.1$  and set  $\{\mathbf{Y}_n\}_{n=1}^3$  to zero matrices. The weights  $\mu_1, \mu_2, \mu_3$  of the optimization objective (7) are set to 1, 0.1, 0.1 by default. We can adjust the sizes of  $\mu_2, \mu_3$  according to the shape sequences' low-rank and smoothing properties. In addition, we found if the shape sequence does not satisfy the low-rank property well, increasing the weight  $\mu_1$  of the reprojection term can effectively improve the reconstruction results, *e.g.*,  $\mu_1 = 1e^1$  generally on the NRSfM Challenge dataset and  $\mu_1 = 1e^2$  on kinect and rug in the Semi-dense dataset. For the estimation of proxy shape, we adjust the ratio of nearly rigid points  $\alpha_r$  between 0 to 1 depending on the deformation characteristics of the object.  $\delta_r$  is set to  $\frac{1}{3}$  on Mocap and NRSfM Challenge datasets and 0 on Semi-dense and H3WB datasets.

As described in Sec. 3,  $\alpha_r$  determines the optimization

approach used for different regions of the 3D structure, so  $\alpha_r$  should depend on the specific characteristics of the object's deformation, *e.g.*, spatial coherence, continuity, *etc.* I adopted a testing interval of 0.2 for parameter selection initially, followed by a finer search around promising results using a step size of 0.1. This granularity proved sufficient for achieving good outcomes, eliminating the need for further refinement in the search steps. The stability test for our method on  $\alpha_r$  is illustrated in Fig. 7. In most cases, the algorithm is stable with respect to  $\alpha_r$  once a rough selection interval has been determined, and thus careful screening of  $\alpha_r$  is not necessary. When we have a large amount of data, we can segment the keypoints by learning a certain distribution without setting hyperparameters, like [4].

### 4.2. H3WB Dataset Processing

The H3WB dataset [14] is an entire human body 3D dataset extended from the H36M dataset [7]. However, this dataset provides 2D annotations and their corresponding 3D structures frame by frame rather than in a sequence. We screened with the criterion of being as contiguous as possible and obtained five sequences from **S1 Eating2**, **S6 Smoking**, **S6 Smoking1**, **S6 Directions**, and **S7 Waiting2**. Since the motion amplitude of these sequences is vast, we remove the frames with significant mutations and use spline functions to interpolate the remaining parts to obtain five more realistic human action sequences: Eating2 (185 frames), Smoking (180), Smoking1 (265), Directions (245), and Waiting2 (335). To obtain the coordinates of the 2D keypoints under the orthogonal projection model, we set up the camera motion in reference to [1].

We show a partial 3D shape sequences of the H3WB NRSfM dataset in Fig. 2. We used the experimental setup in Sec. 1.2 to test the low-rank and smoothing properties of the H3WB dataset, and the results are displayed in Fig. 8. The sequences in the H3WB dataset generally have higher nuclear norm and first-order smoothing errors compared to the pickup sequence in Mocap. Recovering the 3D shapes in the H3WB dataset using low-rank and smoothing constraints is more complicated. However, it is worth noting that the TPA-aligned sequences have better low-rank and smoothing properties than the GT sequences, which somewhat guarantees the validity of the low-rank regularization.

### 4.3. Additional Experiments on Missing Data

In reality, the movement of an object often leads to occlusion of different regions, resulting in missing 2D observations obtained from camera shots. Therefore, the stability of the algorithm on the missing dataset is significant. We show some tests on missing data in the main text, and we will add more quantitative results in this section.

We use the model in Sec. 2.2 and follow the settings in [10] to randomly add masks (occlusion rate 30%) to the

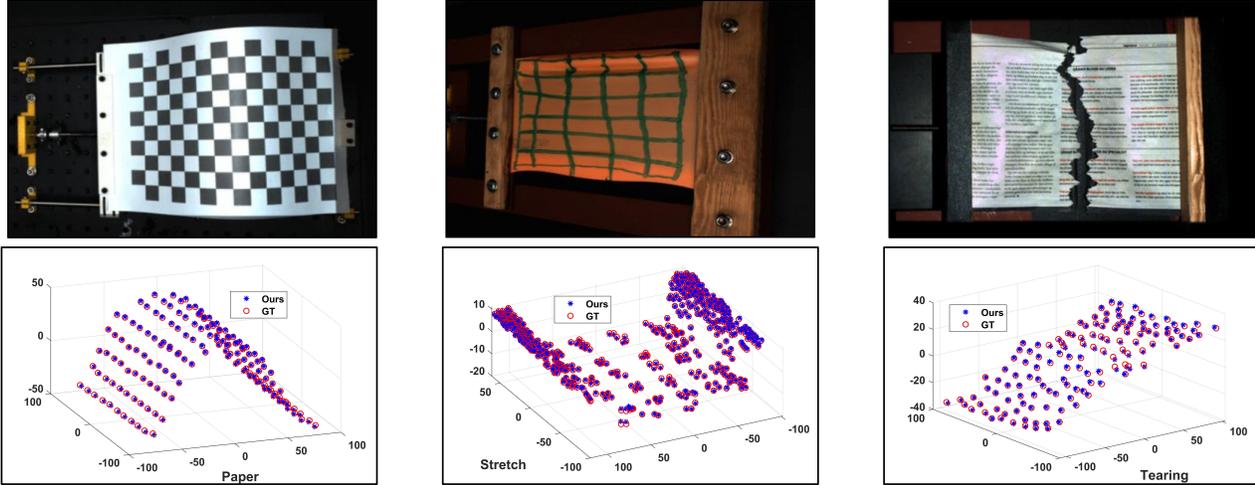


Figure 5. Qualitative Results on the NRSfM Challenge Benchmark. The first row shows the images in the dataset, and the second row shows the reconstruction results of our method compared to the GT.

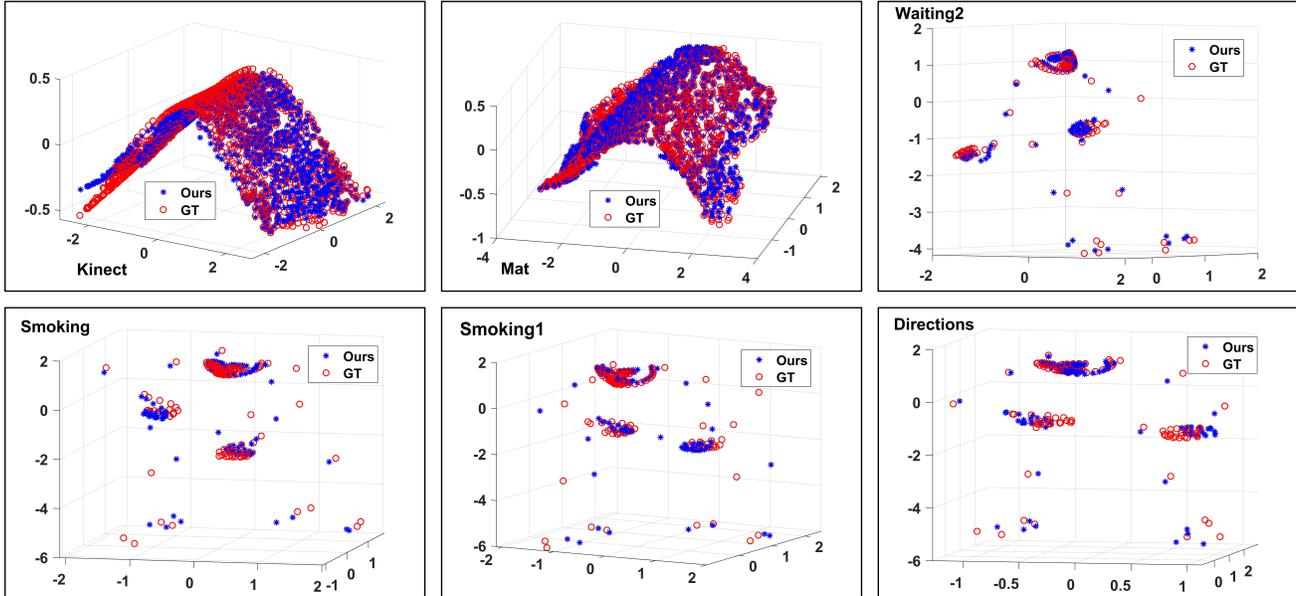


Figure 6. Qualitative Results on the Semi-dense and H3WB dataset. The reconstruction results of our method on the Semi-dense dataset are closer to the GT shapes, while the H3WB dataset is very challenging and cannot yet be accurately reconstructed.

2D observation matrix  $W$ . Before reconstruction, we first solve the low-rank approximation of the observation matrix to complement it [2], which is important for the initialization of the camera motion. Tab. 1 shows the test results on the remaining sequences of the Mocap dataset and the H3WB dataset. Our method achieves the best performance on sequences other than Shark and Walking. The experimental results show that the reconstruction errors on data with and without occlusion display consistency. And the comparison between them indicates that our method still

possesses stability under random occlusion settings (around 30% occlusion).

Simulating occlusion by randomly adding masks is friendly to the recovery of observation matrix, whereas the occlusion scenario in reality tends to be more complex. We tested on the real occlusion data provided by the NRSfM Challenge Dataset, and the results are shown in Fig. 9. The figure shows the reconstruction error comparison with the CSF2 [5] on two sequences Balloon and Stretch. Our approach is superior in terms of mean performance, regardless

Table 1. Comparison of reconstruction errors on missing data in Mocap and H3WB datasets. ‘-’ indicates that test results are unavailable. Ours(w/) and Ours(w/o) represent the test results with and without occlusion respectively.

Method	Shark	Face	Walking	Eating2	Smoking1	Directions	Smoking	Waiting2
CSF2[6]	0.0653	0.0412	0.1033	0.2556	0.3659	0.3025	0.4058	0.1862
PND[10]	0.0166	0.0177	<b>0.0469</b>	0.2167	0.2954	0.3054	0.4863	0.2256
PMP[11]	<b>0.0116</b>	0.0174	0.0507	-	-	-	-	-
Ours(w/)	0.0268	<b>0.0154</b>	0.0796	<b>0.1657</b>	<b>0.1971</b>	<b>0.2858</b>	<b>0.1804</b>	<b>0.1113</b>
Ours(w/o)	0.0258	0.0144	0.0710	0.1767	0.2040	0.2688	0.1693	0.1097

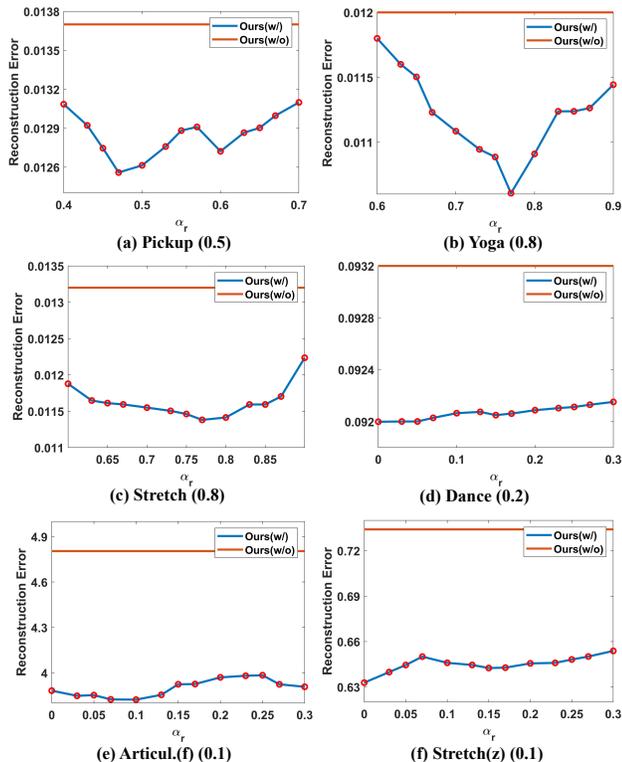


Figure 7. Stability test for hyperparameter  $\alpha_\tau$  on different sequences. Straight lines indicate the results without the SWNN module and broken lines indicate the reconstruction errors for different  $\alpha_\tau$  settings. To the right of the sequence names are the  $\alpha_\tau$ -values corresponding to the results reported in the main text.

of whether the data is occluded or not. In addition, the occlusion rate is the main factor affecting the performance of the algorithm. The average occlusion rate under all camera motion types for Balloon is 38%, while for Stretch it is 13%. As a result the method’s accuracy degradation is more obvious on Balloon. Apart from the occlusion rate, the accuracy of our method also relies on the results of matrix completion. Matrix completion based on the low-rank assumption tends to fail for some special occlusion scenarios, *e.g.*, the object is completely invisible at some moments (such as Articulated/tricky in NRSfM Challenge Dataset, the object is completely occluded in the first 35 frames) and excessive occlusion (such as Tearing/tricky with the occlusion rate of 56%). Searching for more robust matrix-completion algo-

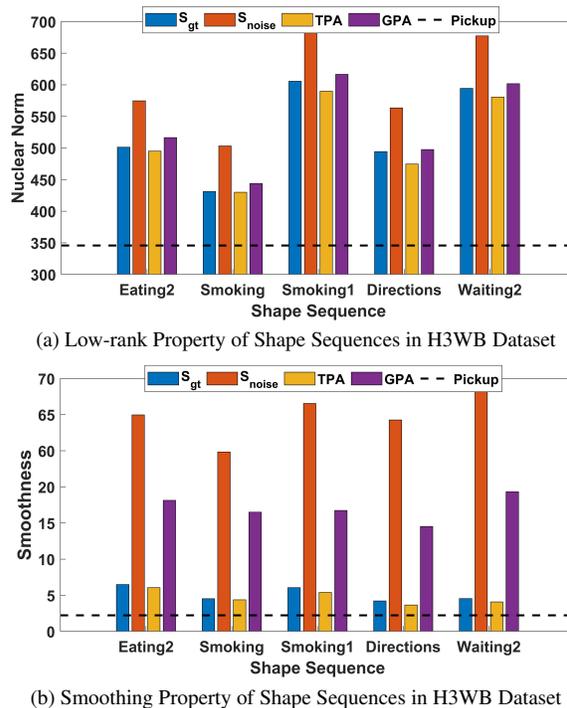


Figure 8. Analyzing the low-rank and smoothing properties of shape sequences in the H3WB dataset. The black dotted line represents the values of the metrics for the pickup sequence in the Mocap dataset. The comparison reveals that the sequences in the H3WB dataset have a greater magnitude of motion and are more difficult to recover using low-rank and smoothing constraints.

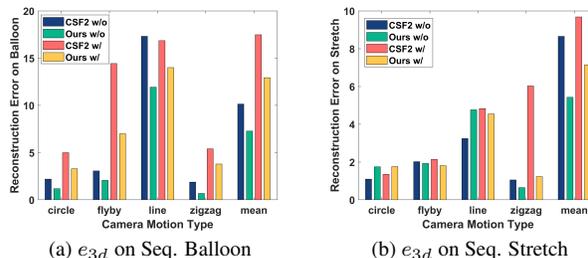


Figure 9. Reconstruction error on missing data in NRSfM Challenge Dataset. For a better comparison, the figure shows the test results under multiple camera motion types and compares the reconstruction errors without missing data.

gorithms or updating the observation matrix in iterations [13] are potential solutions.

#### 4.4. Additional Qualitative Results

In this section, we provide more visualizations of the 3D reconstruction results of our method. Fig. 5 shows additional qualitative results on the NRSfM Challenge dataset, and Fig. 6 illustrates 3D reconstruction results compared with GT on the Semi-dense and H3WB datasets.

#### References

- [1] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. In *Adv. Neural Inform. Process. Syst.*, pages 41–48, 2008. 7
- [2] Ricardo Cabral, Fernando De la Torre, João P Costeira, and Alexandre Bernardino. Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. In *Int. Conf. Comput. Vis.*, pages 2488–2495, 2013. 8
- [3] Yuchao Dai, Hongdong Li, and Mingyi He. A simple prior-free method for non-rigid structure-from-motion factorization. *Int. J. Comput. Vis.*, 107(2):101–122, 2014. 3, 6
- [4] Alessio Del Bue, Xavier Llad, and Lourdes Agapito. Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1191–1198, 2006. 7
- [5] Paulo FU Gotardo and Aleix M Martinez. Computing smooth time trajectories for camera and deformable shape in structure from motion with occlusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(10):2051–2065, 2011. 8
- [6] Paulo FU Gotardo and Aleix M Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3065–3072, 2011. 9
- [7] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1325–1339, 2013. 7
- [8] Suryansh Kumar. Non-rigid structure from motion: Prior-free factorization method revisited. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 51–60, 2020. 3, 6
- [9] Suryansh Kumar and Luc Van Gool. Organic priors in non-rigid structure from motion. In *Eur. Conf. Comput. Vis.*, pages 71–88, 2022. 3, 7
- [10] Minsik Lee, Jungchan Cho, Chong-Ho Choi, and Songhwai Oh. Procrustean normal distribution for non-rigid structure from motion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1280–1287, 2013. 1, 7, 9
- [11] Minsik Lee, Chong-Ho Choi, and Songhwai Oh. A procrustean markov process for non-rigid structure recovery. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014. 9
- [12] Jorge Nocedal and Stephen J Wright. *Numerical Optimization*. Springer, New York, NY, USA, 2e edition, 2006. 1
- [13] Marco Paladini, Alessio Del Bue, Marko Stosic, Marija Dodig, Joao Xavier, and Lourdes Agapito. Factorization for non-rigid and articulated structure using metric projections. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2898–2905, 2009. 9
- [14] Yue Zhu, Nermin Samet, and David Picard. H3wb: Human3.6m 3d wholebody dataset and benchmark. In *Int. Conf. Comput. Vis.*, pages 20166–20177, 2023. 7