HumanMM: Global Human Motion Recovery from Multi-shot Videos

Supplementary Material

Contents

A. Additional Experiment

Further evaluation on *ms*-Motion. To ensure a balanced evaluation and evaluate potential data bias in the *ms*-Motion dataset, where videos in the 2-, 3-, and 4-shot categories are mutually exclusive, we create multiple versions of the same video across different shot categories for further evaluation. Specifically, we randomly select 20 videos and create their 2-, 3-, and 4-shot versions, resulting in a total of 60 videos. We then evaluate SLAHMR [26], WHAM [28], GVHMR [29], and our method on these videos. The results, presented in Tab. 8, follow the same evaluation pattern as those on the full *ms*-Motion dataset in Tab. 1, which indicates the effectiveness of our proposed dataset in evaluating different human motion recovery methods.

Methods	2-shot						3-shot						4-shot					
	PA.↓	WA.↓	RTE↓	ROE↓	Jitter↓	F.S.↓	PA.↓	WA.↓	RTE↓	ROE↓	Jitter↓	F.S.↓	PA.↓	WA.↓	RTE↓	ROE↓	Jitter↓	F.S.↓
SLAHMR [2023]	75.54	326.15	9.35	94.63	63.74	3.15	81.22	523.98	11.59	105.44	74.12	4.55	93.45	811.31	14.85	109.56	80.69	14.12
WHAM [2024]	65.14	339.85	4.65	81.12	**25.29**	2.82	83.10	501.29	7.82	89.88	**25.85**	3.05	100.20	801.63	5.97	94.36	**27.29**	3.98
GVHMR [2024]	66.18	234.96	5.82	92.13	35.77	4.72	72.63	359.99	6.42	97.45	35.63	5.85	90.52	588.40	9.02	96.75	31.66	9.52
Ours	**33.96**	**109.72**	**2.12**	**66.58**	33.49	**1.69**	**40.12**	**131.39**	**3.94**	**68.33**	31.82	**3.01**	**42.64**	**165.23**	**4.32**	**69.15**	32.22	**3.52**

Table 8. **Quantitative results on *ms*-Motion (20)**. We select 20 videos from *ms*-Motion and generate 2-, 3-, and 4-shot versions for each, yielding a total of 60 videos for evaluation. PA. and WA. means PA-MPJPE and WA-MPJPE respectively, while F.S. is Foot Sliding. The results follow a similar trend to those on original *ms*-Motion, showcasing *ms*-Motion's effectiveness in evaluating different HMR methods.

B. Masked DPVO Detailed Algorithm

In this section, we provide a detailed exposition of the algorithm underpinning our proposed Masked DPVO, as introduced in the context of **Comparison on Camera Trajectory Estimation** in Sec. 5.4, building on the foundation of DPVO [76]. In accordance with the principles of DPVO, a scene is conceptualized as a composition of camera poses $\mathbf{G} \in SE(3)^N$ and a collection of square image patches $\mathbf{P}$ derived from video frames. The inverse depth is denoted as $\mathbf{d}$, while pixel coordinates are represented by (x, y). Each patch $\mathbf{P}_k$ is modeled as a $4p^2$ homogeneous array, where p denotes the patch width, as illustrated below,

$$\mathbf{P}_k = \begin{pmatrix} x \\ y \\ 1 \\ \mathbf{d} \end{pmatrix}, \quad \mathbf{x}, \mathbf{y}, \mathbf{d} \in \mathbb{R}^{1 \times p^2}. \tag{6}$$

Let i, j represent the indices corresponding to frame i and frame j, respectively. The projection matrix $\mathbf{P_{kj}}$, which maps the patch $\mathbf{P}_k$, extracted from frame i, to frame j, can then be expressed as follows,

$$\mathbf{P}_{kj} \sim \mathbf{K}\mathbf{G}_j\mathbf{G}_i^{-1}\mathbf{K}^{-1}\mathbf{P}_k. \tag{7}$$

where, $\mathbf{K}$ is a calibration matrix, defined as,

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{8}$$

In the original DPVO framework, patches are selected randomly, as this approach has been shown to achieve comparable improvements in performance. However, a critical issue arises when a selected patch corresponds to a region on a moving object. In such cases, the estimated camera pose $\mathbf{G}$ becomes inaccurate, negatively impacting the reconstruction of human motion in world coordinates. Given that our primary focus involves human-centric video data, where moving humans often occupy a significant portion of the image, excluding regions corresponding to moving humans presents a straightforward yet effective strategy for improving camera pose estimation accuracy.

To address this, we incorporate SAM [86] into the DPVO pipeline. Using the bounding box of a detected human as a prompt, SAM generates a human mask. During the patch extraction process across frames, if a randomly selected patch falls within the human mask, it is excluded and a new patch is selected instead. The resulting patch after applying the human mask is denoted as $\hat{\mathbf{P}}_k$. The corresponding masked projection matrix is subsequently represented as $\mathbf{P}_{kj}$,

$$\hat{\mathbf{P}}_{kj} \sim \mathbf{K}\mathbf{G}_j\mathbf{G}_i^{-1}\mathbf{K}^{-1}\hat{\mathbf{P}}_k. \tag{9}$$

Following the principles of DPVO, a bipartite patch graph is constructed to capture the relationships between patches and video frames. In this graph, edges connect patches and frames that are within a specified distance, defined by the temporal relationship between frame i and frame j). The graph is inherently dynamic, as older frames are discarded as newer frames are introduced. For each edge (k, j) in the patch graph, the visual alignment, determined by the current estimates of depth and camera poses, is evaluated. This is achieved by computing correlation features $\mathbf{C} \in \mathbb{R}^{p \times p \times 7 \times 7}$ (visual similarities), which represent visual similarities. These correlation features are computed as follows,

$$\mathbf{C}_{uv\alpha\beta} = \mathbf{g}_{uv} \cdot \mathbf{f}(\hat{\mathbf{P}}_{kj}(u, v) + \Delta_{\alpha\beta}) \tag{10}$$

where u, v are the index of each pixel in patch k, $\mathbf{f}(\cdot)$ is the bilinear sampling and Δ is 7×7 grid indexed by α and β. Based on extracted features and correlations, DPVO uses MLP layers to predict trajectory update δ_{kj} and confidence weight Σ_{kj}.

Then, a differentiable bundle adjustment (BA) layers is proposed to solve both the depth and camera poses. BA operates on the patch graph and outputs updates to depth and camera pose. The optimization objective of BA is shown as,

$$\sum_{(k,j)\in\mathcal{E}} \left\| \mathbf{K}\mathbf{G}_j\mathbf{G}_i^{-1}\mathbf{K}^{-1}\hat{\mathbf{P}}_k - \left[\hat{\mathbf{P}}'_{kj} + \delta_{kj}\right] \right\|^2_{\Sigma_{kj}} \tag{11}$$

where $\|\cdot\|_\Sigma$ is the Mahalanobis distance and $\hat{\mathbf{P}}'_{kj}$ is the center of $\hat{\mathbf{P}}_{kj}$. DPVO then applies two Gauss-Newton iterations to the linearized objective, optimizing the camera poses and inverse depth component at the same time. The main intuition for this optimization is to refine the camera poses and depth such that the induced trajectory updates agree with the predicted trajectory updates.

C. Masked LEAP-VO Detailed Algorithm

In this section, we illustrate the detailed algorithm of our proposed camera estimation method Masked LEAP-VO. We start with the preliminaries about tracking any point (TAP), which has been used in LEAP-VO [77]. Given an input video sequence $\mathbf{I} = \{I_t\}_{t=1}^{T}$ of length T, where I_t denotes the t-th frame, the goal of TAP is to track a set of query points across these frames. For a specific query point q with the 2D pixel coordinate $x_t^q \in \mathbb{R}^2$ in frame I_t, TAP predicts both the visibility $\mathbf{V}^q = [v_1^q, \cdots, v_T^q]$ and trajectory $\mathbf{X}^q = [x_1^q, \cdots, x_T^q]$ of this point through the whole video. Thus, we have,

$$(\mathbf{X}, \mathbf{V}) = \text{TAP}(\mathbf{I}, x_t^q, I_t) \tag{12}$$

LEAP-VO uses CoTracker [85] to as their TAP module. During inference, CoTracker will produce point feature f_t for each frame I_t and then concatenate these point features into a tensor $\mathbf{F}^q = [f_1, \cdots, f_T]$. With predicted $\mathbf{X}$ and $\mathbf{F}$, LEAP-VO tries to predict whether each point query is on the dynamic object (dynamic label m_d) using anchor-based trajectory comparison to alleviate the negative effect they bring to the camera pose estimation.

Subsequently, LEAP-VO uses temporal probability modeling to get the confidence of estimated trajectories of each point query. Let $X = [\mathbf{a}, \mathbf{b}]$, where $\mathbf{a}, \mathbf{b}$ represent the x and y coordinates of all points in X, respectively. The probability density function for a coordinate is modeled using a multivariate Cauchy distribution,

$$p(\mathbf{a}|I_t,x_i) = \frac{\gamma(\frac{1+T}{2})}{\gamma(\frac{1}{2})\pi^{\frac{T}{2}}|\sum_a|^{\frac{1}{2}}[1+(\mathbf{a}-\mu_a)^T\Sigma_a^{-1}(\mathbf{a}-\mu_a)]^{\frac{1+S}{2}}}, \tag{13}$$

and similarly for $p(\mathbf{b}|I_t, x_i)$. Here, γ denotes the Gamma function. LEAPVO then filters out trajectories shorter than a predefined threshold and inputs the remaining trajectories into a bundle adjustment (BA) stage with sliding window optimization.

However, as we mentioned in Sec. 3.3, the performance of this method is still satisfactory as the exclusion of the moving object is not complete. Since we are dealing with human-centric videos, we can simply apply SAM to get the mask of human and exclude these points. Thus, we set visibility of trajectories that contains points inside human mask to zero. The adjusted BA function is shown as Sec. 3.3 in Sec. 3.3.

D. Camera Calibration Procedure

In this section, we show the detailed procedure of the *camera calibration* we mentioned in Sec. 3.4.1. We denote the selected feature points of two frames during shot transition in world coordinate as

$$W_1 = [(x_{w1}^{(1)}, y_{w1}^{(1)}, z_{w1}^{(1)}), (x_{w1}^{(2)}, (y_{w1}^{(2)}, z_{w1}^{(2)}), \cdots, (x_{w1}^{(N)}, y_{w1}^{(N)}, z_{w1}^{(N)})]^\top,$$
$$W_2 = [(x_{w2}^{(1)}, y_{w2}^{(1)}, z_{w2}^{(1)}), (x_{w2}^{(2)}, (y_{w2}^{(2)}, z_{w2}^{(2)}), \cdots, (x_{w2}^{(N)}, y_{w2}^{(N)}, z_{w2}^{(N)})]^\top,$$

respectively. Suppose we have the initialized camera translation matrix $\mathbf{T}$ and camera rotation matrix $\mathbf{R}$ during shot transition from Masked LEAP-VO in Sec. 3.3, where

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \tag{14}$$

and since

$$\begin{bmatrix} x_{w1}^{(1)} \\ y_{w1}^{(1)} \\ z_{w1}^{(1)} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_{w1}^{(1)} \\ y_{w1}^{(1)} \\ z_{w1}^{(1)} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \tag{15}$$

Thus, according to Epipolar Constraint,

$$\begin{bmatrix} x_{w2}^{(1)} & y_{w2}^{(1)} & z_{w2}^{(1)} \end{bmatrix} \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} x_{w2}^{(1)} \\ y_{w2}^{(1)} \\ z_{w2}^{(1)} \end{bmatrix} = 0. \tag{16}$$

Thus, by substituting Eq. (15) to Eq. (16),

$$\begin{bmatrix} x_{w2}^{(1)} & y_{w2}^{(1)} & z_{w2}^{(1)} \end{bmatrix} \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_{w1}^{(1)} \\ y_{w1}^{(1)} \\ z_{w1}^{(1)} \end{bmatrix} + \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = 0. \tag{17}$$

Let's denote $[\mathbf{T}]_\times$ as

$$[\mathbf{T}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}. \tag{18}$$

As a result, the essential matrix can be denoted as $\mathbf{E} = [\mathbf{T}]_\times \mathbf{R}$,

$$\mathbf{E} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{19}$$

Since

$$\begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = 0, \tag{20}$$

by constructing essential matrix $\mathbf{E}$ and elimination of the second term, the Eq. (17) can then be rewritten as

$$\begin{bmatrix} x_{w2}^{(1)} & y_{w2}^{(1)} & z_{w2}^{(1)} \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} x_{w1}^{(1)} \\ y_{w1}^{(1)} \\ z_{w1}^{(1)} \end{bmatrix} = 0 \tag{21}$$

As our method is targeted for the in-the-wild multi-shot videos, we typically do not know the intrinsics for each shot. Thus, we assume that through a multi-shot video, the camera intrinsics are the same. The camera intrinsic matrix $\mathbf{K}$ is denoted as,

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \tag{22}$$

Similar as mentioned in Sec. 3.4.1, we denote the coordinates of detected 2D KPTs of two frames in the shot transition as $\mathbf{S}_1 = [(x_1^{(1)}, y_1^{(1)}), (x_1^{(2)}, y_1^{(2)}), \cdots, (x_1^{(N)}, y_1^{(N)})]^\top \in \mathtt{R}^{2\times N}$ and $\mathbf{S}_2 = [(x_2^{(1)}, y_2^{(1)}), (x_2^{(2)}, y_2^{(2)}), \cdots, (x_2^{(N)}, y_2^{(N)})]^\top \in \mathtt{R}^{2\times N}$. Based on intrinsic matrix $\mathbf{K}$,

$$\begin{bmatrix} x_{w2}^{(1)} & y_{w2}^{(1)} & z_{w2}^{(1)} \end{bmatrix} = \begin{bmatrix} x_2^{(1)} & y_2^{(1)} & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{f_x} & 0 & 0 \\ -\frac{o_x}{f_x} & \frac{1}{f_y} & 0 \\ 0 & -\frac{o_y}{f_y} & 1 \end{bmatrix}, \begin{bmatrix} x_{w1}^{(1)} \\ y_{w1}^{(1)} \\ z_{w2}^{(1)} \end{bmatrix} = \begin{bmatrix} \frac{1}{f_x} & 0 & 0 \\ -\frac{o_x}{f_x} & \frac{1}{f_y} & 0 \\ 0 & -\frac{o_y}{f_y} & 1 \end{bmatrix} \begin{bmatrix} x_1^{(1)} \\ y_1^{(1)} \\ 1 \end{bmatrix} \tag{23}$$

Thus, we can substitute Eq. (23) to Eq. (21),

$$\begin{bmatrix} x_2^{(1)} & y_2^{(1)} & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{f_x} & 0 & 0 \\ -\frac{o_x}{f_x} & \frac{1}{f_y} & 0 \\ 0 & -\frac{o_y}{f_y} & 1 \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} \frac{1}{f_x} & 0 & 0 \\ -\frac{o_x}{f_x} & \frac{1}{f_y} & 0 \\ 0 & -\frac{o_y}{f_y} & 1 \end{bmatrix} \begin{bmatrix} x_1^{(1)} \\ y_1^{(1)} \\ 1 \end{bmatrix} = 0. \tag{24}$$

We can then denote fundamental matrix $\mathbf{F}$ as,

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{f_x} & 0 & 0 \\ -\frac{o_x}{f_x} & \frac{1}{f_y} & 0 \\ 0 & -\frac{o_y}{f_y} & 1 \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} \frac{1}{f_x} & 0 & 0 \\ -\frac{o_x}{f_x} & \frac{1}{f_y} & 0 \\ 0 & -\frac{o_y}{f_y} & 1 \end{bmatrix}, \tag{25}$$

and we can rewrite Eq. (24) as,

$$\begin{bmatrix} x_2^{(1)} & y_2^{(1)} & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_1^{(1)} \\ y_1^{(1)} \\ 1 \end{bmatrix} = 0. \tag{26}$$

One clarification is that the fundamental matrix $\mathbf{F}$ here is denoted as essential matrix $\mathbf{E}$ in Eq. (3) in Sec. 3.4.1. After we derive Eq. (26), we can expand it as a linear equation, illustrated as,

$$\left(f_{11}x_1^{(1)} + f_{12}y_1^{(1)} + f_{13}\right) x_2^{(1)} + f_{31}x_1^{(1)} + f_{32}y_1^{(1)} + f_{33} + \left(f_{21}x_1^{(1)} + f_{22}y_1^{(1)} + f_{23}\right) y_2^{(1)} = 0. \tag{27}$$

This illustrates the case for one correspondence, since we have correspondences, we can build up a linear equation system to solve $\mathbf{F}$. Thus, we create a matrix $\mathbf{A}$ based on $\mathbf{S_1}$ and $\mathbf{S_2}$, shown as,

$$A = \begin{bmatrix} x_2^{(1)}x_1^{(1)} & x_2^{(1)}y_1^{(1)} & x_2^{(1)} & y_2^{(1)}x_1^{(1)} & y_2^{(1)}y_1^{(1)} & y_2^{(1)} & x_1^{(1)} & y_1^{(1)} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_2^{(i)}x_1^{(i)} & x_2^{(i)}y_1^{(i)} & x_2^{(i)} & y_2^{(i)}x_1^{(i)} & y_2^{(i)}y_1^{(i)} & y_2^{(i)} & x_1^{(i)} & y_1^{(i)} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_2^{(N)}x_1^{(N)} & x_2^{(N)}y_1^{(N)} & x_2^{(N)} & y_2^{(N)}x_1^{(N)} & y_2^{(N)}y_1^{(N)} & y_2^{(N)} & x_1^{(N)} & y_1^{(N)} & 1 \end{bmatrix}. \tag{28}$$

Let f denotes the flattened version $\mathbf{F}$,

$$\mathbf{A}f = \begin{bmatrix} x_2^{(1)}x_1^{(1)} & x_2^{(1)}y_1^{(1)} & x_2^{(1)} & y_2^{(1)}x_1^{(1)} & y_2^{(1)}y_1^{(1)} & y_2^{(1)} & x_1^{(1)} & y_1^{(1)} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_2^{(i)}x_1^{(i)} & x_2^{(i)}y_1^{(i)} & x_2^{(i)} & y_2^{(i)}x_1^{(i)} & y_2^{(i)}y_1^{(i)} & y_2^{(i)} & x_1^{(i)} & y_1^{(i)} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_2^{(N)}x_1^{(N)} & x_2^{(N)}y_1^{(N)} & x_2^{(N)} & y_2^{(N)}x_1^{(N)} & y_2^{(N)}y_1^{(N)} & y_2^{(N)} & x_1^{(N)} & y_1^{(N)} & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0. \tag{29}$$

Then, we can solve f by applying singular value decomposition (SVD) on $\mathbf{A}$. Decompose $\mathbf{A}$ into three matrices $\mathbf{A} = U\Sigma V^\top$. Substituting into Eq. (29),

$$\begin{aligned} U\Sigma V^\top f &= 0 \\ U^\top U\Sigma V^\top f &= U^\top 0 \\ \Sigma V^\top f &= 0. \end{aligned} \tag{30}$$

Since f is in the null space of $\mathbf{A}$, from decomposition, the null space is spanned by the last columns of V corresponding to zero singular values in Σ. Thus, we can extract the last column of V (denoted as v_n) and assign $f = v_n$. After we get f, we can get the fundamental matrix $\mathbf{F}$. We can then derive essential matrix $\mathbf{E}$ from $\mathbf{F}$ shown as,

$$\mathbf{E} = \mathbf{K}^\top \mathbf{F}\mathbf{K} = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ o_x & o_y & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \tag{31}$$

Then, we can derive camera rotation $\mathbf{R}$ and camera translation $\mathbf{T}$ by applying SVD on essential matrix $\mathbf{E}$,

$$\mathbf{E} = U\Sigma V^\top = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \end{bmatrix}^\top \tag{32}$$

$$\mathbf{R} = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \end{bmatrix}^\top, \mathbf{T} = \begin{bmatrix} u_{13} \\ u_{23} \\ u_{33} \end{bmatrix} \tag{33}$$

The output $\mathbf{R}$ here is denoted as the $\mathbf{R}_{\delta_{\text{cam}}}$ in Sec. 3.4.1.

E. Implementation Details of *HumanMM*

E.1. Training Details

The *ms*-HMR, the trajectory, and foot sliding refiner are trained on the AMASS [78], 3DPW [79], Human3.6M [61], and BEDLAM [90] datasets, evaluate on EMDB and our *ms*-Motion. During training, we introduce random rotational noise (ranging from 0 to 1 radian) along the y-axis to the root orientation Γ and random noise to the body pose θ at random positions to simulate the inaccuracies of pre-estimated human motions caused by shot transitions in multi-shot videos. This strategy enables the network to robustly recover smooth and consistent human motion from noisy initial parameters. The introduction of these noise perturbations stems from the observation that relying solely on our orientation alignment module may fall short in fully aligning the root pose across different shots in challenging scenarios, particularly when significant angular discrepancies occur during shot transitions. To overcome this limitation, our trainable module is designed not only to align root poses with camera parameters but also to ensure smooth transitions in local poses across shot boundaries. This strategy significantly enhances the robustness of our method in managing abrupt orientation changes caused by multi-shot video transitions.

E.2. *ms*-Motion Benchmark Details

Our *ms*-Motion Benchmark is building on existing multi-view datasets AIST [60], H36M [61]. The AIST dataset provides world translation as ground truth. H36M dataset does not include such labels; therefore, we process human world translation from [29] as the ground truth. For benchmarking, the test results were obtained after training *HumanMM* for 80 epochs on a single NVIDIA-A100 GPU (1.3 days). This computational setup ensures efficient convergence of the model while maintaining a high level of accuracy in human motion recovery.

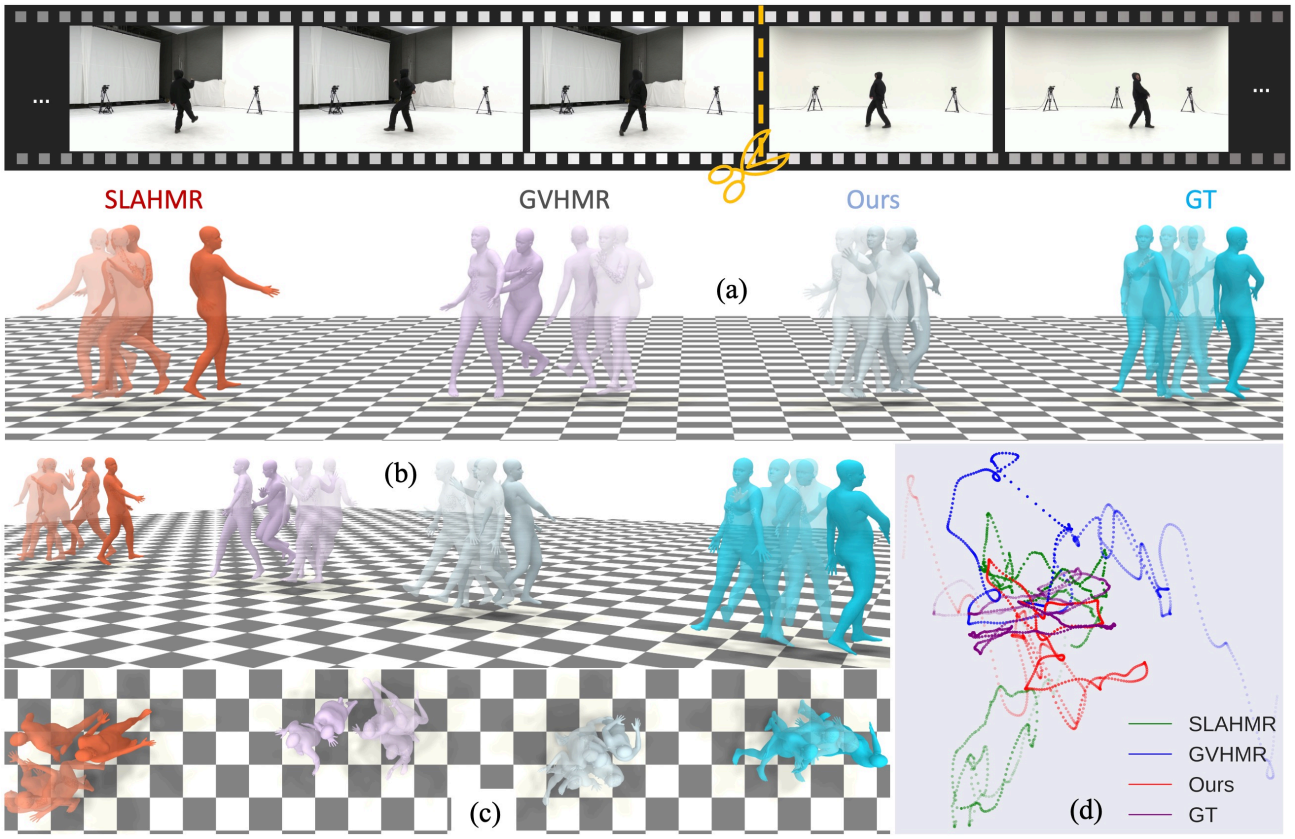
E.3. Baseline Setting

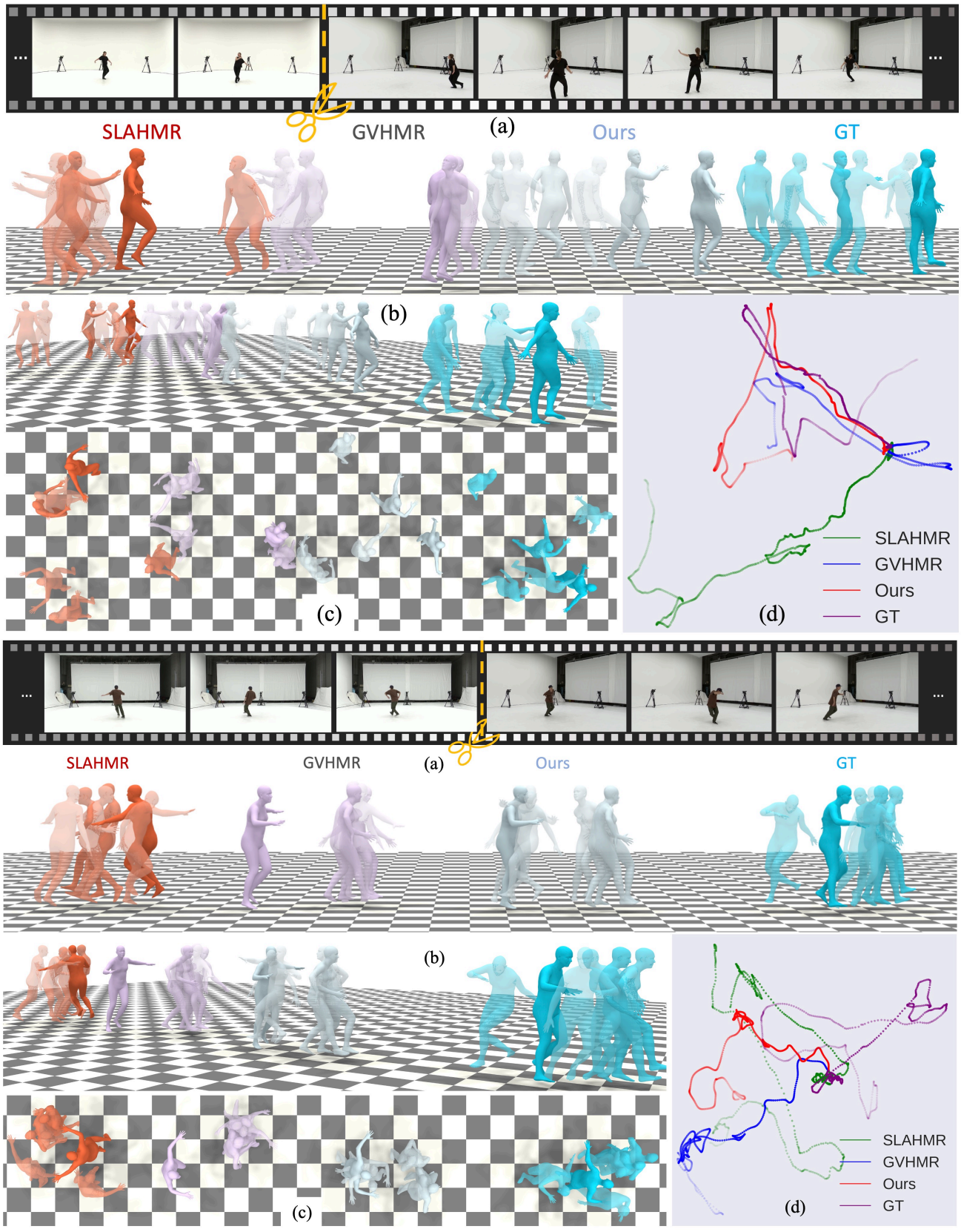
The baseline mentioned in Tab. 5 in Sec. 5.4 is implemented as follows. For the input multi-shot videos, we process them by using our proposed *shot transition detector* and *human and camera parameters initialization* as shown in Fig. 3 in Sec. 3. Next, we directly concatenate the human translation, root orientation, and body pose based on the relative offsets between frames. This method could achieve global motion recovery in a few scenarios with simple motions. However, our observations reveal that this approach suffers from noticeable issues, such as foot sliding, motion truncation, and motion collapse. As there were no existing methods for global human motion recovery, this approach can serve as our baseline for conducting ablation studies to evaluate the effectiveness of our proposed training and optimization modules.

F. Visualization of Comparison between Existing Methods

F.1. Visualization of Comparison between Existing Methods

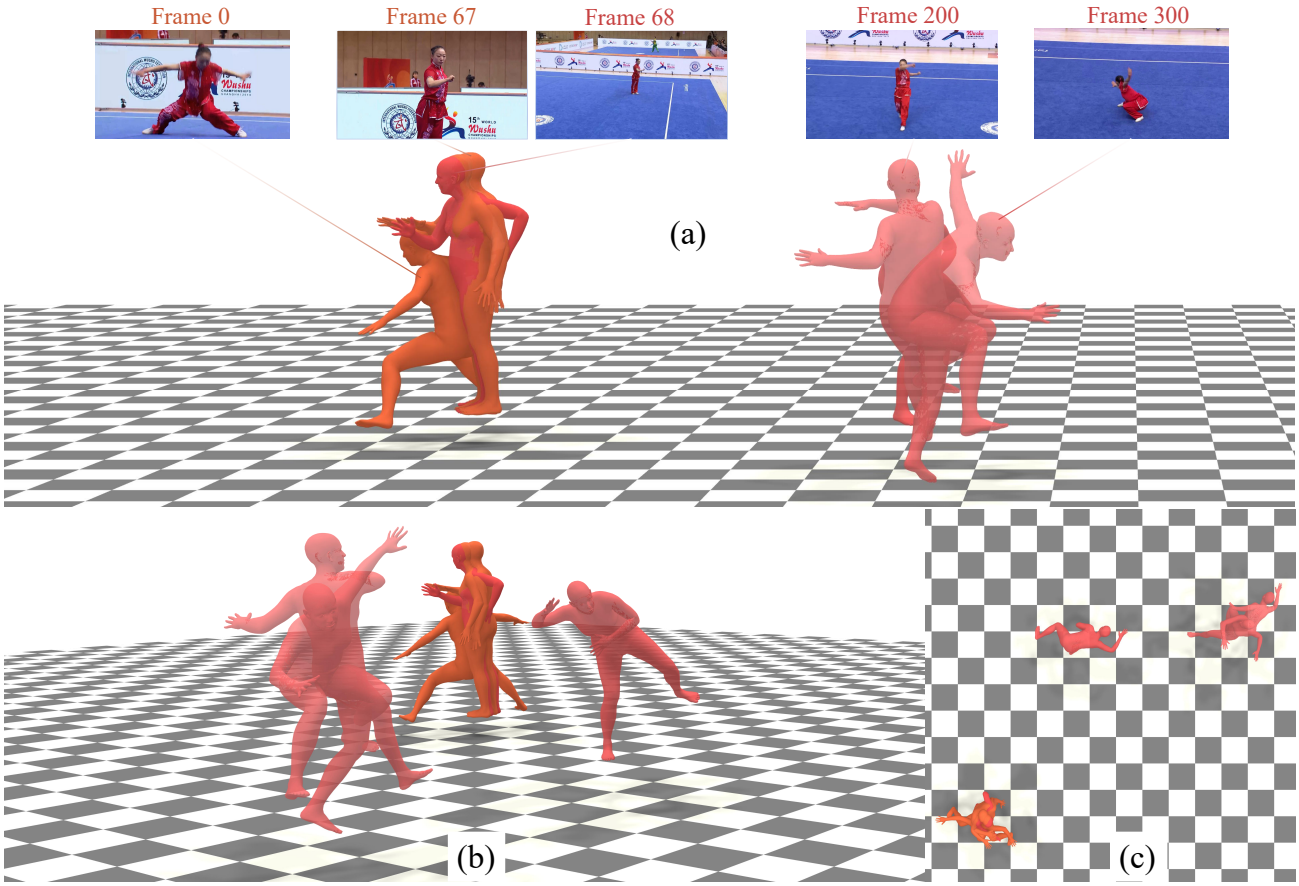
In this section, we present visual comparisons between our proposed method and existing approaches, including SLAHMR [26], GVHMR [29], as well as the ground truth. These comparisons aim to highlight the advancements achieved by our method in accurately reconstructing human motion. To ensure a comprehensive evaluation, we provide visualizations from multiple viewpoints: a side view (a), an alternative side view (b), a top-down view (c), and reconstructed motion trajectories (d). The side view (a) allows for a detailed examination of the overall pose accuracy and alignment over time, emphasizing the consistency and anatomical plausibility of the reconstructed movements. The alternative side view (b) provides additional insights into the depth and spatial relationships of the poses, capturing nuances that might be less apparent from a single perspective. The top-down view (c) reveals the positional alignment and the spatial coherence of the trajectories, showcasing the robustness of our approach in reconstructing dynamic and complex motions. Finally, the motion trajectories (d) offer a quantitative and qualitative representation of the movement paths, highlighting the differences between methods and their ability to accurately track and reproduce the ground truth trajectories. These visual comparisons underscore our method ability in delivering precise, consistent, and realistic human motion recovery from multi-shot videos.

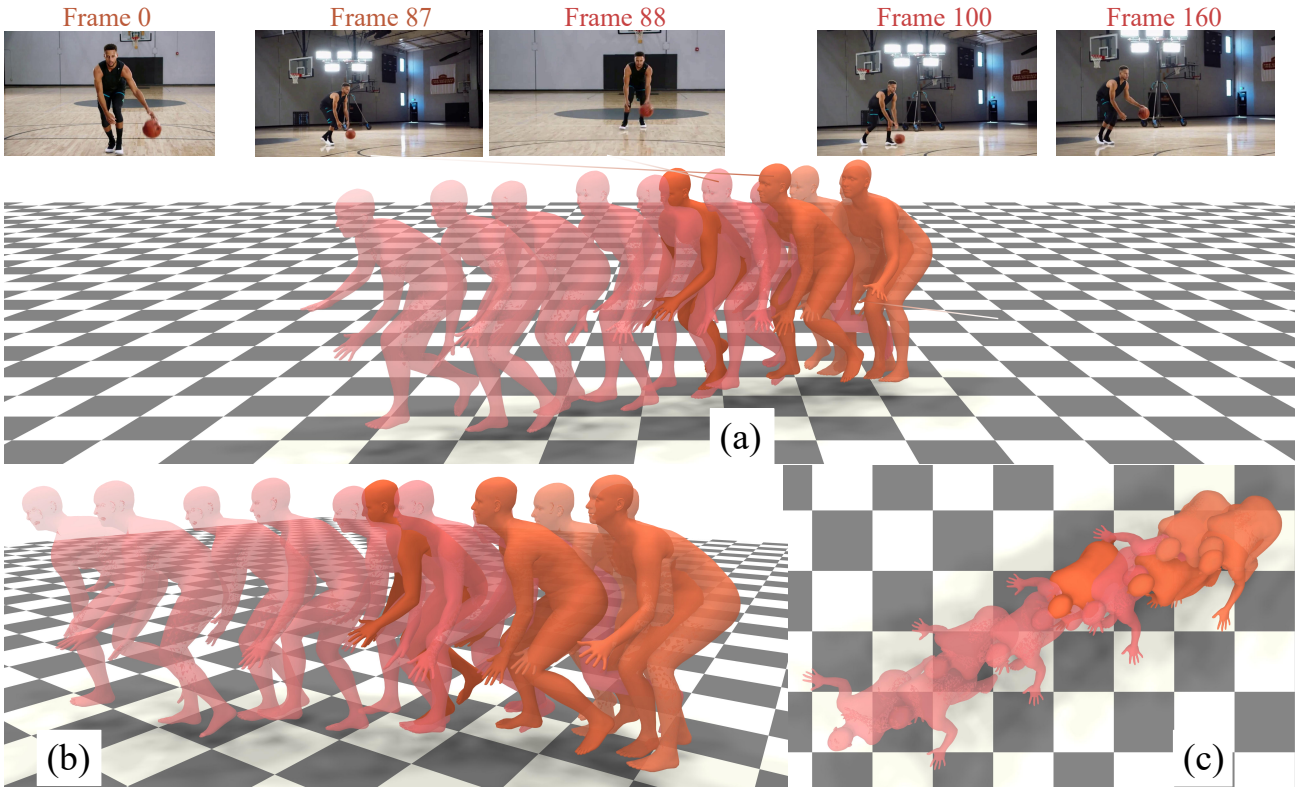



SLAHMR
GVHMR
(a)
Ours
GT
(b)
(c)
(d)
SLAHMR
GVHMR
Ours
GT
SLAHMR
GVHMR
(a)
Ours
GT
(b)
(c)
(d)
SLAHMR
GVHMR
Ours
GT

F.2. Visualization of in-the-wild multi-shot video

To further validate the performance of our proposed method, we apply *HumanMM* on a set of in-the-wild videos and provide the visualizations as shown in the figure. The frames at the top illustrate key moments in the video sequence, with the corresponding reconstructions visualized in 3D space below. The 3D reconstructions effectively capture the dynamic human poses and trajectories over time. In subfigure (a), the side view highlights the accurate reconstruction of intricate human motion, illustrating the consistency and smoothness of the trajectories even for fast and complex movements. Subfigure (b) presents an alternative side view, offering another perspective that underscores the spatial coherence of the reconstructions. Subfigure (c) provides a top-down view, offering an insightful perspective into the overall trajectory and positional accuracy of the reconstructed poses as they evolve over time. This view particularly emphasizes the spatial distribution and alignment of the poses in the reconstructed scene. These visualizations collectively demonstrate the robustness of *HumanMM* in handling challenging, dynamic motions in diverse environments, further showcasing its applicability to real-world scenarios.




Frame 0
Frame 87
Frame 88
Frame 100
Frame 160
(a)
(b)
(c)