Supplementary material
E.T. the Exceptional Trajectories: Text-to-camera-trajectory generation with character awareness

Robin Courant[1], Nicolas Dufour[1,2], Xi Wang[1], Marc Christie[3], and Vicky Kalogeiton[1]

[1] LIX, Ecole Polytechnique, IP Paris
[2] LIGM, Ecole des Ponts, CNRS, UGE
[3] Inria, IRISA, CNRS, Univ. Rennes

A Ethical discussion

We discuss the ethical impact of our method across several aspects:

- *Creative Integrity:* It is a fine line between using AI tool to enhance the human creativity and allowing it to deprive human creative process. Under misusage, the proposed method could diminish the artistic expression instead of support it.
- *Intellectual Property*: The use of AI-generated content raises questions about ownership and copyright. The Intellectual Property ownership of the generated content can be debatable.
- *Job Displacement or Creation*: The automation of certain aspects of filmmaking could lead to concerns about job displacement within the industry, or under proper usage, may also help to create new types of jobs in the domain.

B Exceptional Trajectories dataset (E.T.)

B.1 Additional statistics

We build our E.T. dataset the Condensed Movies Dataset [1] (CMD), encompassing over $30,000$ scenes from $3,000$ diverse movies, totaling more than $1,000$ hours of video. We segment each movie scene into continuous shots by leveraging changes in color and intensity between frames [3].

We show additional statistics of E.T. in Figure 2. We observe that for both camera and character, the majority of trajectories are smaller than 20 meters, i.e. corresponding to a velocity of 20 meters/(300 frames/25 fps) $= 1.67 m.s^{-1}$.

Additionally, in Figure 1, we show extensive examples of E.T. samples.

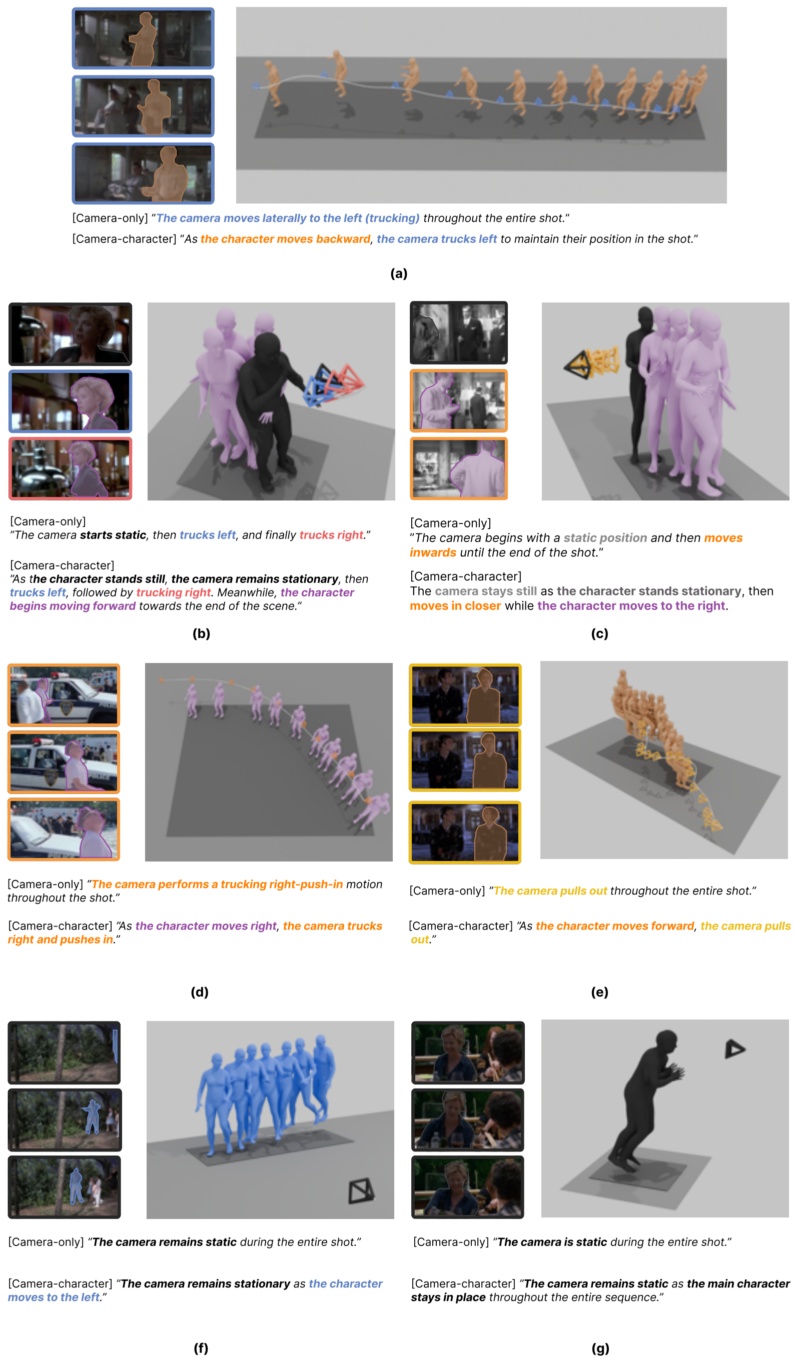
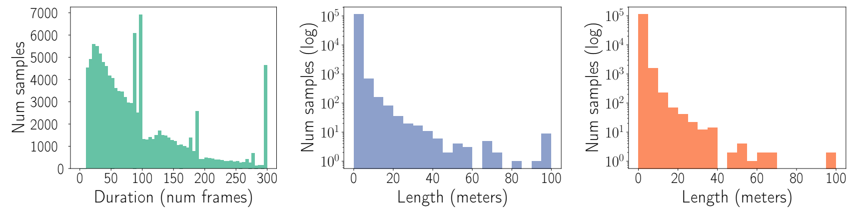
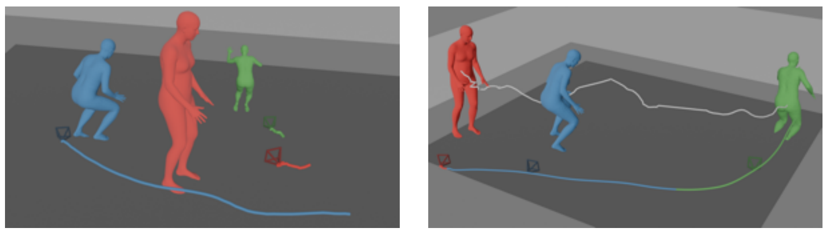


Fig. 1: Examples E.T. samples. Each subfigure presents frames from the original movie shot (left), and processed camera and character trajectories (right). Additionally, the bottom part showcases the generated camera trajectory caption with or without the character trajectory caption.



(a) Trajectory length (in #frames) **(b)** Camera length (in meters) **(c)** Character length (in meters)

Fig. 2: E.T. statistics.



(a) Before alignment. **(b)** After alignment.

Fig. 3: Raw chunk alignment. We show in (a) the raw independent chunks just after the SLAHMR [7] extraction. In (b) we display the result of the chunk alignment process. Each color (red, blue, green) corresponds to a different chunk.

B.2 Data pre-processing

Chunk alignment. A limitation of SLAHMR [7] is its inability to handle long videos (exceeding 100 frames). Consequently, we divide each shot into chunks of 100 frames and process them independently. However, it produces non-consitant outputs: it exhibits translational bias/offset and different scales, as shown in Figure 3a.

To address this issue, we propose the following alignment method: dividing shots into overlapping chunks, where consecutive chunks share frames, and performing alignment on these overlapping frames. A chunk contains camera trajectories with $SE(3)$ poses represented as $[\mathbf{R}|\mathbf{t}]$ (where $\mathbf{R}$ denotes rotation and $\mathbf{t}$ translation), and 3D human poses described by $\mathbf{V}$ (vertices of a 3D mesh).

Given two consecutive chunks at k and $k+1$, we initially align the cameras. The alignment involves determining a scale parameter s and a $SE(3)$ rigid transformation $[\mathbf{B} \mid \mathbf{b}]$:

$$[\mathbf{R}_k \mid \mathbf{t}_k] = [\mathbf{B}_k \mid \mathbf{b}_k]\,[\mathbf{R}_{k+1} \mid s_k\,\mathbf{t}_{k+1}], \tag{1}$$

$$[\mathbf{R}_k \mid \mathbf{t}_k] = [\mathbf{B}_k\,\mathbf{R}_{k+1} \mid s_k\,\mathbf{B}_k\,\mathbf{t}_{k+1} + \mathbf{b}_k], \tag{2}$$

which simplifies to:

$$(a) \quad \mathbf{R}_k = \mathbf{B}_k\,\mathbf{R}_{k+1}, \tag{3}$$

$$(b) \quad \mathbf{t}_k = s_k\,\mathbf{B}_k\,\mathbf{t}_{k+1} + \mathbf{b}_k. \tag{4}$$

Notably, the rotation estimated by SLAHMR remains consistent across chunks, implying $\mathbf{B}_k = \mathbf{I}$, and simplifying Equations 3 and 4 :

$$(a) \quad \mathbf{R}_k = \mathbf{R}_{k+1}, \tag{5}$$

$$(b) \quad \mathbf{t}_k = s_k\, \mathbf{t}_{k+1} + \mathbf{b}_k. \tag{6}$$

Subsequently, alignment entails determining the scaling factor s and translational bias $\mathbf{b}$. These parameters can be accurately estimated using the least-square method [2], as represented by:

$$\begin{bmatrix}\mathbf{t}_k\ \mathbf{I}\end{bmatrix}\begin{bmatrix}s_k\\ \mathbf{b_k}\end{bmatrix} = \mathbf{t}_{k+1}, \tag{7}$$

which can be further expressed as:

$$\begin{bmatrix}t_k^x & 1 & 0 & 0\\ t_k^y & 0 & 1 & 0\\ t_k^z & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}s_k\\ b_k^x\\ b_k^y\\ b_k^z\end{bmatrix} = \begin{bmatrix}t_{k+1}^x\\ t_{k+1}^y\\ t_{k+1}^z\end{bmatrix}. \tag{8}$$

We also seek the alignment transform Δ_b, such that:

$$[\mathbf{R}_{k+1} \mid s_k\, \mathbf{t}_{k+1} + \mathbf{b}_k]\, \mathbf{\Delta}_b = [\mathbf{R}_{k+1} \mid \mathbf{t}_{k+1}], \tag{9}$$

resulting in:

$$\mathbf{\Delta}_b = [\mathbf{R}_{k+1} \mid s_k\, \mathbf{t}_{k+1} + \mathbf{b}_k]^{-1}\, [\mathbf{R}_{k+1} \mid \mathbf{t}_{k+1}]. \tag{10}$$

Considering the inverse of a 4x4 transformation matrix representing a rigid transformation:

$$\begin{bmatrix}\mathbf{R}^T & -\mathbf{R}^T\mathbf{t}\\ \mathbf{0} & 1\end{bmatrix}, \tag{11}$$

we obtain from Eq. 10:

$$\mathbf{\Delta}_b = \begin{bmatrix}\mathbf{R}_{k+1}^T & -\mathbf{R}_{k+1}^T(s\mathbf{t}_{k+1} + \mathbf{b}_k)\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_{k+1} & \mathbf{t}_{k+1}\\ \mathbf{0} & 1\end{bmatrix}, \tag{12}$$

$$\mathbf{\Delta}_b = \begin{bmatrix}\mathbf{I} & \mathbf{R}_{k+1}^T(\mathbf{t}_{k+1} - (s\mathbf{t}_{k+1} + \mathbf{b}_k))\\ \mathbf{0} & 1\end{bmatrix}. \tag{13}$$

Ultimately, to align the 3D human poses based on their vertices V:

$$\begin{bmatrix}\mathbf{V}_k^T\\ 1\end{bmatrix} = \mathbf{\Delta}_b\begin{bmatrix}\mathbf{V}_{k+1}^T\\ 1\end{bmatrix} = \begin{bmatrix}\mathbf{V}_{k+1}^T + \mathbf{R}_{k+1}^T(\mathbf{t}_{k+1} - (s_k\mathbf{t}_{k+1} + \mathbf{b}_k))\\ 1\end{bmatrix}, \tag{14}$$

$$\mathbf{V}_k = \mathbf{V}_{k+1} + (\mathbf{t}_{k+1} - (s_k\mathbf{t}_{k+1} + \mathbf{b}_k))^T\mathbf{R}_{k+1}. \tag{15}$$

The alignment process outcome is illustrated in Figure 3b.

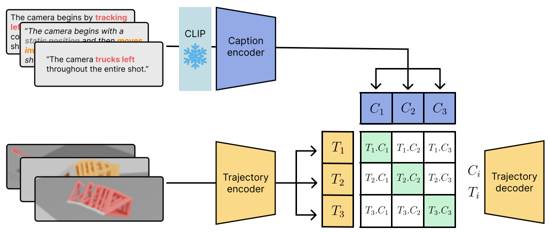
Data cleaning. The extracted trajectories have limitations from the data extraction method [7], including discontinuities, ruptures and jerky motions. To address this, we first clean the data by removing outliers (i.e., discontinuous segments), with a velocity threshold. Specifically, we eliminate trajectory points holding velocities greater than the 95th percentile of the overall trajectory velocity multiplied by a scaling factor. Subsequently, the trajectory is partitioned into sub-trajectories without outliers. Finally, we use Kalman filter on each chunk to reduce residual jerkiness and enhance overall smoothness.

B.3 Dataset creation pipeline

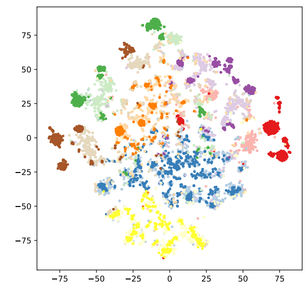
Motion tagging. We tune the parameters of our motion tagging method using the dataset introduced in [4]. This small dataset of 75 short clips includes annotated sequences of pure camera motion. For the character trajectory tagging, we extended this dataset by annotating human trajectories. We select parameters (i.e. mainly threshold values) that corresponds to the best classification metrics described in Section 5 of the main manuscript.

Caption generation. We show the prompt used for caption generation (see Section 3.2 of the main manuscript):

```
You act as a camera operator writing a technical script for camera
motion descriptions.

Given a rough outline of the camera motion and main character motion,
write the camera motion description according to the main character
motion.

The sentence should be short, and factual. Do not mention frame
indices.

# Examples
Outline: Total frames 209.
    [Camera motion] Between frames 0 and 154: boom top, Between
    frames 155 and 209: static.
    [Main character motion] Between frames 0 and 146: move up,
    Between frames 147 and 209: static.
Description: While the character climbs up, the camera follows them
with a boom top, and as soon as the character stops, it remains
static.
# End of examples

Outline: Total frames {CURRENT_NUM_FRAME}.
    [Camera motion] {CURRENT_CAMERA_DESCRIPTION}.
    [Main character motion] {CURRENT_CAMERA_DESCRIPTION}.
Description:
```



(a) Overview of CLaTr framework. CLaTr projects both text and camera trajectories into a common latent space using encoders. Self-similarity is then computed, and a shared-weight decoder decodes both text and camera trajectory features back into a camera trajectory.



(b) t-SNE visualization of CLaTr embedding of text (vivid colors) and trajectory (pastel colors). Each color corresponds to a K-Mean cluster of the text embedding.

C Contrastive Language-Trajectory embedding (CLaTr)

Text-trajectory retrieval						Trajectory-text retrieval					
R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓	R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10↑	MedR ↓
19.73	31.67	40.8	52.08	64.69	5.0	11.15	17.25	20.91	26.5	34.66	28.0

Table 1: CLaTr evaluation. We report the retrieval scores of CLaTr on the E.T. dataset.

We show in Figure 4a the overview of the CLaTr framework as described in Section 4.2 of the main manuscript.

Implementation details. We train CLaTr with a batch size of 32 using the AdamW optimizer with a learning rate of $1e-5$. The set the weight of the reconstruction loss at 1.0, of the latent loss at $1.0e-5$, of the KL loss at $1.0e-5$, and of the contrastive loss at 0.1. The model has 6 layers with a hidden dim of 256 and 4 attention heads. We use dropout of 0.1. Similar to Director, we set the default temporal input size to 300 and use masking to handle inputs with fewer than 300 frames. We represent the camera trajectory with the 6D continuous representation for rotation [8] combined with the 3D translation component.

CLaTr Evaluation. Table 1 presents standard retrieval performance measures from [5,6]. Recall at rank k (R@k) indicates the percentage of times the correct caption is within the top k results (higher is better). Median rank (MedR) is also reported, where lower values are better.

As shown in Table 1, text-to-trajectory metrics outperform trajectory-to-text metrics. This may be because text descriptions are more ambiguous and varied in describing trajectories, making it easier to match a text description to a unique trajectory than to match a trajectory to a specific description among many possibilities.

CLaTr embedding. We show in Figure 4b a t-SNE visualization of CLaTr text (vivid colors) and trajectory (pastel colors) embeddings. We applied K-Means clustering to the text embeddings and visualized the corresponding clusters on the trajectory embeddings to assess the consistency of the joint embedding. Notably, we find that text clusters are preserved in the trajectory space, with vivid and pastel clusters overlapping, indicating a robust alignment between text and trajectory representations.

References

1. Bain, M., Nagrani, A., Brown, A., Zisserman, A.: Condensed movies: Story based retrieval with contextual embeddings. In: ACCV (2020)
2. Björck, Å.: Least squares methods. Handbook of numerical analysis (1990)
3. Castellano, B.: Pyscenedetect. https://github.com/Breakthrough/PySceneDetect (2014)
4. Courant, R., Lino, C., Christie, M., Kalogeiton, V.: High-level features for movie style understanding. In: ICCV-W (2021)
5. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3d human motions from text. In: CVPR (2022)
6. Petrovich, M., Black, M.J., Varol, G.: TMR: Text-to-motion retrieval using contrastive 3d human motion synthesis. In: ICCV (2023)
7. Ye, V., Pavlakos, G., Malik, J., Kanazawa, A.: Decoupling human and camera motion from videos in the wild. In: CVPR (2023)
8. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: CVPR (2019)