AI-Based Automatic Cinematic Framing

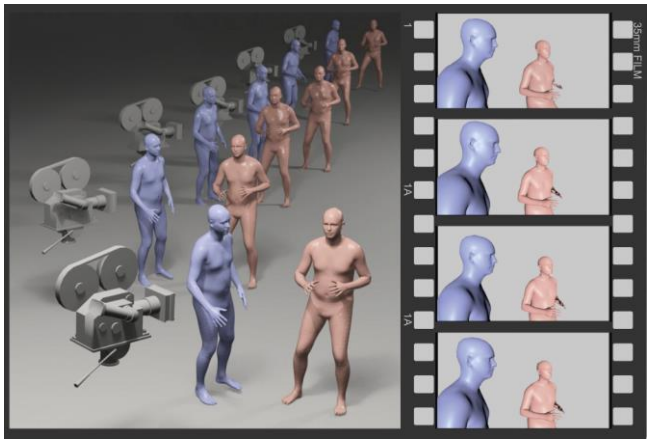
Shang Ni (s5701147)

GitHub URL: https://github.com/poooiil/MasterClass

Dataset in Google Drive: https://drive.google.com/file/d/17NuDXd0YB9mMlhn5DYysVEvJfshUI_dm/view?usp=sharing

1. Abstract

Cinematic camera control serves as a fundamental element of visual storytelling across film, animation, and interactive media, yet it remains a labor-intensive process typically entrusted to specialized artists. Contemporary deep learning approaches automate camera placement and movement from existing footage but rely heavily on extensive, annotated video corpora and exhibit limited generalization to unfamiliar character interactions. A novel framework addresses these limitations by predicting Toric camera parameters directly from two-person 3D motion sequences, thereby eliminating dependence on preexisting visual datasets. This framework employs a dual-stream Transformer to encode each character's motion, applies bidirectional cross-attention to capture inter-character dynamics, and integrates explicit spatial vectors for robust geometric grounding. A streamlined fusion network subsequently regresses per-frame Toric parameters, producing smooth, compositionally balanced camera trajectories.



2. Introduction

Animation is widely recognized as a compelling storytelling medium, uniquely capable of delivering narratives through visual framing. Expressive character portrayals, richly detailed environments, and nuanced lighting collectively form the foundation of animated storytelling. However, the role of cinematography, particularly shot composition and camera positioning, in influencing audience emotion and narrative coherence is frequently underappreciated in the animation industry. Effective shot design goes beyond selecting camera angles or movement trajectories it requires intentional planning and precise positioning to clearly communicate character interactions, relationships, and psychological states. Therefore, the camera serves as an essential tool enabling directors to articulate artistic intentions, significantly enhancing narrative clarity and emotional resonance.

Despite its critical role, identifying optimal cinematographic compositions in animation is traditionally a complex and labor-intensive task, demanding significant expertise from directors and specialized artists. Typically, production teams rely on manual refinement and iterative experimentation to finalize camera movements and settings. This conventional approach is inefficient, often escalating production costs and creating technical obstacles. Small and medium-sized animation studios are particularly affected, as they often lack access to skilled personnel and sufficient technical resources. Consequently, these limitations hinder creative exploration, negatively impacting the overall quality and expressive potential of animated narratives.

Recent advancements in deep learning offer promising avenues to automate the processes of camera blocking and shot composition. Current deep learning methods successfully estimate camera motion and framing from existing visual datasets, closely replicating professional cinematographic techniques. However, a critical drawback of these techniques is their dependency on extensive, predefined visual datasets. This dependence makes their performance highly sensitive to the diversity, quality, and representative nature of the available training data, limiting their adaptability and effectiveness when dealing with novel character interactions or unfamiliar animation contexts.

To address these limitations, this project introduces a novel method that eliminates reliance on predefined visual datasets by leveraging 3D motion data to estimate camera placements and cinematographic composition. My approach specifically processes interactions between pairs of animated characters, utilizing deep learning integrated with Toric features to capture spatial orientations and relative positions in 3D space.

3. Background Research

Automatic cinematography involves AI-driven techniques that determine camera positions, orientations, and movements to optimally capture scenes without human intervention. Existing methodologies predominantly fall into two categories: video-based learning and

rule-based procedural methods. Video-based approaches utilize deep neural networks trained on extensive annotated datasets to replicate established cinematographic styles. Although effective, these methods suffer from limited generalization and dependency on expensive labeled datasets.

Conversely, rule-based methods encode cinematographic principles as explicit procedural rules, lacking flexibility and the ability to handle complex dynamic interactions. Recent advancements in transformer-based architectures have demonstrated remarkable capabilities in sequence modeling tasks, including natural language processing and motion prediction, indicating their potential suitability for cinematography tasks.

Thus, leveraging transformer architectures directly on 3D motion data offers a promising alternative that combines the learning capabilities of neural networks with the flexibility required in dynamic cinematography.

4. Methodology

4.1 Data Collection

Video clips were pulled from movies, TV shows, and stage performances to cover a range of camera styles. SHOTDECK annotations helped pinpoint each shot in the original footage. Around those timestamps, continuous six-second segments were manually checked and trimmed, keeping only clips that show two people interacting and feature clear, uninterrupted camera movement.

4.2 IoU-based Tracking

For each video clip, bounding boxes of the two main characters are detected and tracked frame by frame. By calculating the Intersection-over-Union (IoU) metric, the bounding boxes are accurately propagated across consecutive frames. This method ensures consistent identification of characters and reliable tracking stability, enabling precise extraction of character images from each frame throughout the entire video sequence.

4.4 3D Pose Extraction

The cropped character images are input into the MeTRAbs model, which first estimates 2D keypoints and subsequently reconstructs relative 3D poses based on the SMPL-22 joint model. By leveraging perspective geometry optimization, the model additionally recovers the absolute global translation vector for the root joint, which specifies each character's position in a global coordinate system. Consequently, each character's 3D motion data comprises both relative joint coordinates (22 joints relative to the root joint) and the absolute root displacement vector. As the MeTRAbs algorithm places the camera coordinate system at the origin by default, subsequent spatial transformations are necessary for accurate camera pose alignment. To maintain consistency in sequence length across the dataset, motion sequences are uniformly adjusted through cropping or padding to exactly 120 frames, corresponding approximately to standardized interaction segments of 4–5 seconds.

4.5 Data Processing and Normalization

After extracting the raw 3D joint data, Gaussian filtering is applied to smooth each joint trajectory along the temporal dimension, significantly reducing jitter and enhancing motion coherence. To ensure consistent representation between the two characters, their skeletal data are aligned systematically. This alignment involves translating each character's skeleton so that the root joint coincides with the coordinate origin, and subsequently rotating both skeletons into a standardized orientation. The standardized orientation is defined by aligning the vector connecting the left and right shoulders parallel to the global x-axis. Following alignment, each joint pose is represented using a 6D rotation matrix, a formulation known for its numerical stability compared to traditional Euler angles or quaternion representations. Consequently, each frame for each character consists of 22 joints, each encoded as a 6-dimensional rotation vector.

Additionally, a virtual 23rd joint is added to each character to record their overall shift in space. This joint uses a six-element vector: the first three values are the real movement offsets and the last three are just zeros. After this step, the motion data for both characters has the shape (120 frames, 23 joints, 6 values per joint).

4.6 Feature Flattening and Final Input Representation

The motion features for each character, originally represented as tensors of dimensions (120 frames, 23 joints, 6 dimensions per joint), are reshaped into two-dimensional tensors of shape (120, 138) by flattening the joint and dimension axes. This 138-dimensional vector serves as the primary motion input for the learning model, enabling it to learn a direct mapping from dual-character motion sequences to corresponding camera parameters.

To enhance the model's understanding of the spatial relationship between the characters, an additional feature is computed: the relative orientation between the two root joints, encoded as a 9-dimensional rotation offset vector. This vector captures the directional alignment of the characters relative to each other, providing valuable context for interpreting their interactions. These relative orientation vectors, denoted as D1 and D2, can either be utilized independently or concatenated with the primary motion features to offer auxiliary supervisory signals during training.

The target output for the model is represented by a 6-dimensional Toric camera parameter vector for each frame, encapsulating focal length, elevation angle, azimuth angle, and camera distance, among other parameters. The Toric formulation proposed by Lino and Christie(https://dl.acm.org/doi/10.1145/2766965) is employed, as it directly couples camera movements to the on-screen layout of the two characters while remaining independent of aspect ratio and focal length. Specifically, Toric coordinates integrate screen-space positions (2D) of the characters and their world coordinates (3D), along with the world coordinate of the camera itself. Consequently, the resulting Toric representation forms a sequence of vectors of shape (120, 6), providing a coherent and compositionally meaningful training target. One significant advantage of the Toric representation lies in its ability to describe visually intuitive camera-character compositions without altering their respective world coordinates, thereby offering a decoupled yet visually logical approach to cinematographic framing.

Tensor Representation Across Processing Stages:

Stage	Shape	Description
Original 3D Joint Positions	(120, 22, 3)	Raw 3D coordinates (X, Y, Z) of SMPL-22 joints for each frame
6D Joint Rotations + Root Translation	(120, 22, 6) + (120, 1, 6)	6D rotation vectors for 22 joints, plus the root node translation vector (also

		in 6D format, with the last three dimensions set to zero)
Adding Global Offset as a Virtual Joint	(120, 23, 6)	A “virtual joint” is appended to represent the global displacement vector D, extending the skeleton to 23 joints
Flattened Motion Features	(120, 138)	23×6 = 138
Toric Camera Parameters	(120, 6)	Target output

5. Transformer model

The Transformer model is trained to map two synchronized motion sequences and their initial spatial offsets to a dynamic camera trajectory. The inputs consist of two motion tensors of shape (120, 138) and two 9-dimensional offset vectors, while the target output is a sequence of Toric camera parameters of shape (120, 6). Each character’s motion stream is first embedded and augmented with learnable positional encodings, then processed by a multi-layer Transformer encoder to capture temporal dependencies. A bidirectional cross-attention module enables each character’s representation to attend to the other’s evolving context, allowing the network to model interaction cues such as reaction timing, coordinated gestures, and opposing behaviors. Concurrently, two shared-weight vector processors transform the initial offset vectors into relation-aware embeddings, which are then tiled across all time steps. The four resulting feature streams are concatenated and passed through a fully connected fusion network with ReLU activations and layer normalization. The final linear projection produces the six Toric parameters for each frame. Mean squared error serves as the training loss, and upon convergence the trained weights are saved to model_no_att.pth for subsequent inference.

6. Z-score

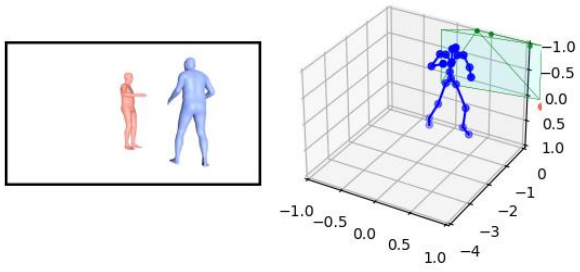
Normalization: Before training, each feature type was standardized using statistics computed on the training set. Three pairs of means and standard deviations were obtained: one for the motion features (Mean_motion, Std_motion), one for the global offset vectors D (Mean_D, Std_D), and one for the Toric camera parameters (Mean_C, Std_C). During preprocessing, both the motion and D features were transformed via z-score normalization by subtracting

their respective means and dividing by their standard deviations. This procedure ensures that each dimension of these inputs has zero mean and unit variance, which promotes numerical stability and accelerates convergence during model training.

Denormalization: After inference, all three feature types are restored to their original scales by applying the inverse z-score transform. Motion and D outputs are multiplied by Std_motion and Std_D, respectively, then have Mean_motion and Mean_D added. Predicted Toric parameters are likewise recovered by multiplying by Std_C and adding Mean_C. This ensures that the final outputs correspond to real-world motion coordinates, global offsets, and camera parameters.

7. Visualization

The final visualization stage begins by loading the predicted joint trajectories and Toric camera parameters from disk. For each frame, the Toric parameters are converted into a world-space camera position and orientation, while the 6D rotation representations of each joint are transformed back into 3D joint locations. A 3D scene is assembled that places two character meshes according to their reconstructed skeletons and inserts the computed camera with its specified field of view. This scene is rendered offscreen to generate the 2D camera view, which is then displayed alongside a 3D plot showing both the skeletal motion and the evolving camera path. Each resulting composite image is saved, providing a clear side-by-side comparison of predicted framing and underlying character movement for qualitative evaluation.



8. Experiments

8.1 Dataset and Training Setup

The dataset each 120 frames long at 20 fps. 80% of the samples were used for training and the remaining 20% for testing. During training, sequences were randomly batched (batch size

= 32) and optimized for 20,000 steps using the Adam optimizer with a learning rate of 1×10^{-5}. All input features (motion tensors and initial offsets) were standardized via z-score normalization before being fed into the network.

8.2 Evaluation Metrics

Three complementary metrics are employed for evaluation. FrameFID assesses the framing quality of individual frames by comparing the ratio of visible body parts and their screen-space projections. SeqFID measures the distributional distance between generated and ground-truth camera trajectories, using features extracted by a self-supervised VAE-Transformer encoder. Pose Distance Error (PDE) quantifies geometric fidelity by averaging the per-frame positional discrepancy between predicted and ground-truth Toric parameters.

8.3 Ablation Study

The individual contributions of the cross-attention module and the explicit spatial vector input were assessed via an ablation study comprising two variants: one lacking the cross-attention mechanism and one omitting the spatial vectors. The table presents the results. In the absence of the spatial offset input (w/o Offset), all three evaluation metrics deteriorate markedly: the PDE increases substantially, and both FrameFID and SeqFID suffer pronounced declines, highlighting the indispensable role of geometric cues in preserving framing accuracy and temporal coherence. When the cross-attention component is removed (w/o Att), a moderate degradation occurs across all metrics, underscoring its importance for capturing inter-character interactions. By contrast, the complete model—integrating both spatial offsets and cross-attention, yields the lowest PDE, FrameFID, and SeqFID, demonstrating that these elements act synergistically to generate precise, smooth, and cinematographically coherent camera trajectories.

Variant	PDE	FrameFID	SeqFID
w/o Offset	0.644	1.128	2.377
w/o Att	0.473	0.443	0.850
Ours	0.451	0.368	0.628

9. Critical Evaluation

Several limitations of the current framework warrant careful consideration. At the most immediate level, the visual output remains relatively coarse: skeletal reconstructions can

exhibit joint misalignments, occlusion handling is imperfect, and camera transitions sometimes lack cinematic fluidity.

Beyond visual fidelity, the dataset's scale and computational demands pose practical constraints. The collection sequences do not fully capture the diversity of human interaction styles, and the multi-layer Transformer architecture incurs significant inference cost, hindering real-time or large-batch processing.

From a stylistic standpoint, the model learns an average cinematographic language without the ability to adapt to different emotional tones or shot types. There is no mechanism for specifying close-ups, wide shots, or genre-specific conventions.

Finally, the approach is inherently limited to dyadic interactions and does not generalize to scenes involving three or more participants, nor does it account for complex environmental dynamics such as background clutter or dynamic occlusions. Addressing these issues would require enhancements in output resolution, expansion and diversification of training data, integration of style-conditioning controls, and extension of the architecture to support multi-actor scenarios.

10. Conclusion

This work has demonstrated the feasibility of predicting cinematic camera trajectories directly from dual-character 3D motion data, eliminating reliance on large annotated video corpora. A dual-stream Transformer architecture effectively captures both individual movement dynamics and inter-character interactions, while explicit spatial encoding grounds these signals in global space. Despite remaining challenges in visual fidelity, real-time performance, and stylistic control, this approach lays a solid foundation for future extensions to multi-actor scenarios and user-driven shot specifications. Continuous refinement of data diversity, model efficiency, and integration of stylistic conditioning will further bridge the gap between automated cinematography and professional filmmaking practice.