Computer Vision for Cinematographic Motion Control

Carnegie Mellon University Robotics Institute

Shaurye Aggarwal, Kaustav Mukherjee, Dr. Ji Zhang

1. Motivation

In movies, music videos, and more, **Robotic Camera Rigs** like the **Bolt™ Jr.+** Is used to take more interesting shots.

However, programming trajectories is **time-consuming** and **difficult**.

Therefore, our goal is to create:
- an **intelligent**
- and **intuitive**

cinematographic system for robotic camera trajectory planning.

2. Problem Formulation

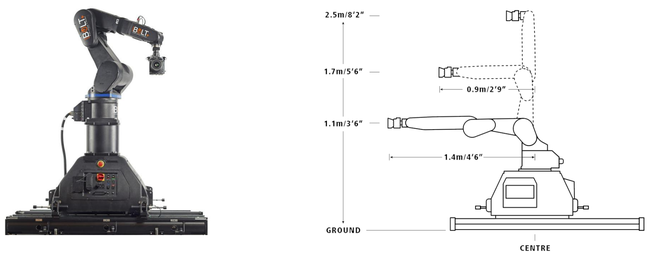


Fig 1: Bolt™ Jr.+ is a 7DoF high-speed camera robot.

Hardware & Software

- **FLAIR Classic** is a proprietary software used to control the **Bolt™ Jr.+**.
- Figure 2 below shows the **control flow** we are trying to **replace**.

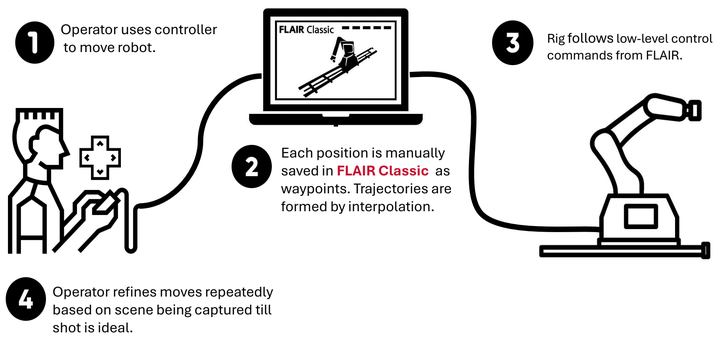


Fig 2: Typical Rig Operation Workflow While Shooting

Our Solutions

- **Control via Demonstration:** enables mimicking phone-based trajectory inputs.
- **Control via Natural Language:** allows for trajectory generation from text or speech.

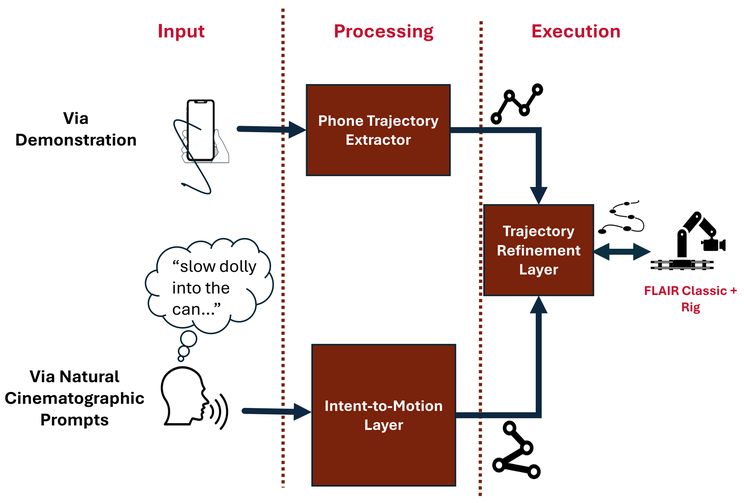


Fig 3: System Overview For Camera Motion Control: Users create camera trajectories using either **phone** or **voice commands**. Trajectories are refined before execution through FLAIR Classic.

3. Control via Demonstration

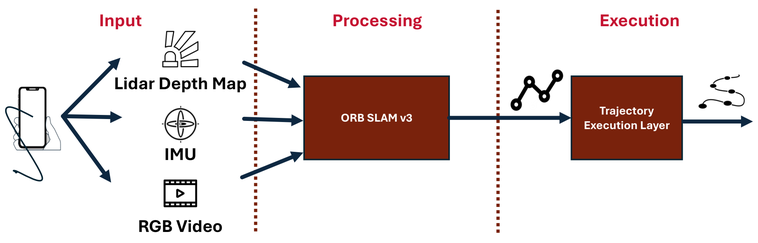


Fig 4: Phone Demonstration to Motion Architecture Diagram

Processing Steps

- **ORB SLAM v3** [1] used to extract **trajectory** from input modalities.
- Imported camera trajectory is **non-smooth** with **inconsistent velocity**, necessitating a **Trajectory Refinement** step.

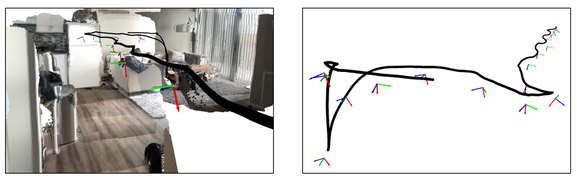


Fig 5: (Left) Video shot on iPhone 15 Pro (Right) Extracted camera pose trajectory

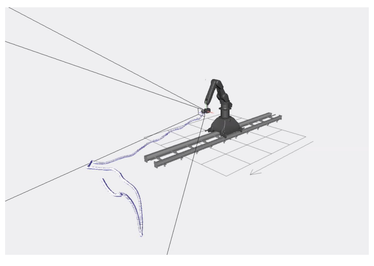


Fig 6: iPhone trajectory is converted to robot joint angles via inverse kinematics.

4. Control via Natural Language

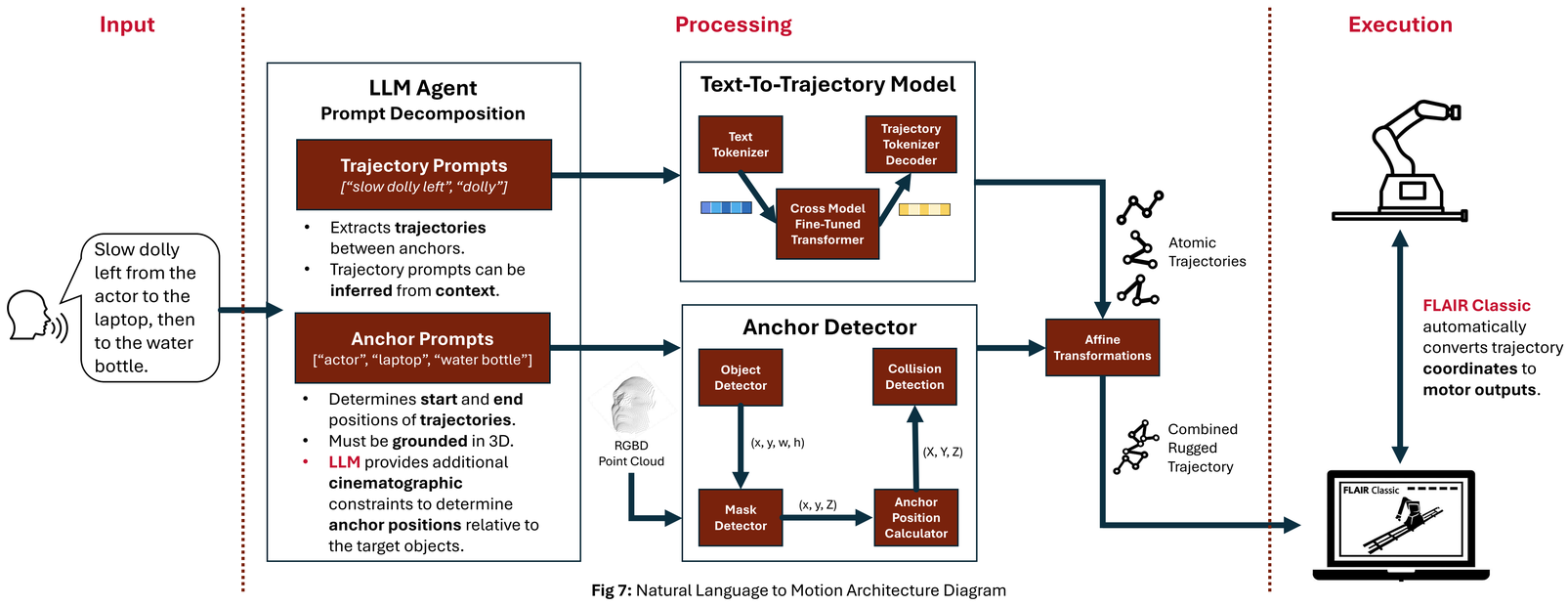


Fig 7: Natural Language to Motion Architecture Diagram

4a. Available Data

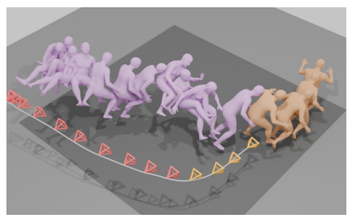


Fig 8: The Exceptional Trajectories Data

The Exceptional Trajectories[3]

- Contains 115'000 **text-trajectory-anchor** pairs.
- Has corresponding **video** data as well.
- Key in training our **text-to-trajectory model**.

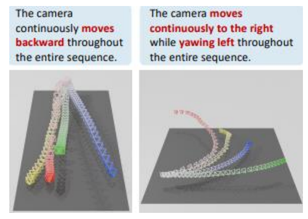


Fig 9: SLAHMR and VDA Outputs

GenDoP [5]

- Contains 29'000 **text-trajectory-video-depth** pairs.
- 27 translation and 7 rotation motions.
- Text prompt describe **camera movements**, interactions with scene, and **directorial intent**.

4b. Anchor Detection

1. Inputs

- **Intel Realsense D435** used to obtain **RGB** and **depth** image.
- User **prompt** collected:
  "Dolly left from the actor to the laptop, then to the water bottle."

Fig 10: RGB Scene Input

2. Object Detection

- **YOLO-World**[2] used for **open-vocabulary** object detection.
- **SAM2** [4] used for segmentation masks.
- **LLM** used to match between detected objects and desired anchors.

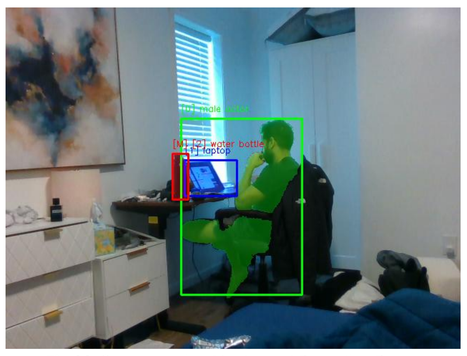


Fig 11: Segmentation Masks and Bounding Boxes

3. Cinematographic Constraints

- **LLM** prompted for **cinematographic** constraints:

Direction
Which side of the object should the camera look at?

Frame Coverage
How much of the frame should the object take up?

Offset
Should the object be centered or off to the side?

- These determine the **camera's position** relative to the **object**.

4. Anchor Selection

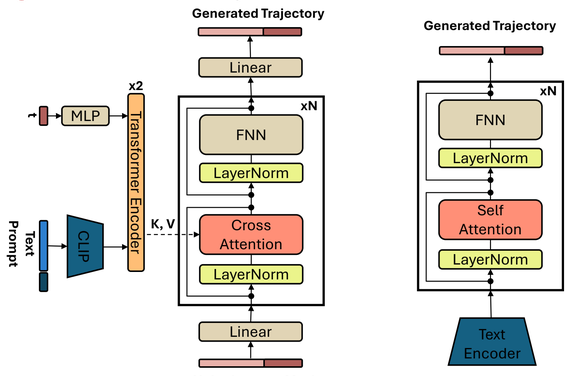
- **Anchors** calculated with **depth, segmentation masks**, and **cinematographic constraints**.
- Finalized after **collision detection** using the pointcloud.

Fig 12: Anchor Positions (R, G, B)

4c. Text-To-Trajectory Generation

Model Architectures

a. **Diffusion** based transformer via **Cross-Attention** Modules [3]
b. **Auto-Regressive** decoder-only transformer via **Self-Attention** Modules [5]



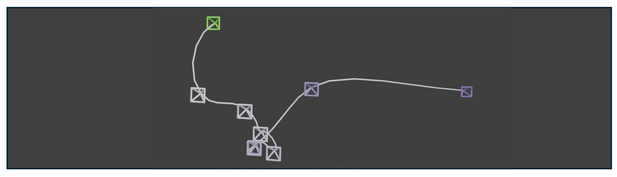
(a) Diffusion Based Transformer

(b) Auto-Regressive, Decoder-Only Transformer

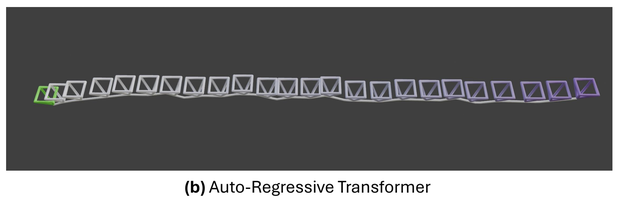
Fig 13: Text-to-Trajectory Transformer Architecture Diagrams

Results

- **Autoregressive model (b)** better captures **intent**, generating more reasonable trajectories than the **diffusion transformer (a)**.
- Few **cinematographic terms** are missing from datasets, leading to partially **incorrect trajectories**.

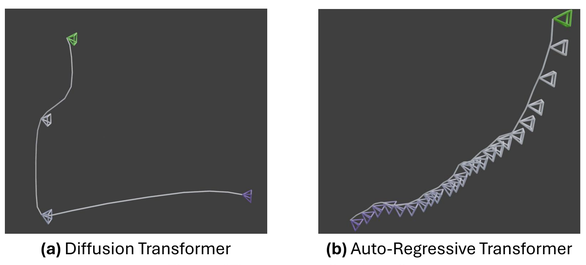


(a) Diffusion Transformer



(b) Auto-Regressive Transformer

Fig 14: "The camera orbits left"



(a) Diffusion Transformer **(b)** Auto-Regressive Transformer

Fig 15: "The camera tilts down"

4d. Text-To-Trajectory Generation

Combining Trajectories and Anchors

- Trajectories transformed between anchors using **affine transformations**.
- Coordinates converted to:
  - **Camera Positions** - where the camera is.
  - **Target Positions** – where the camera should be looking at.
- Considerations made to ensure **robot positions** are **valid**.

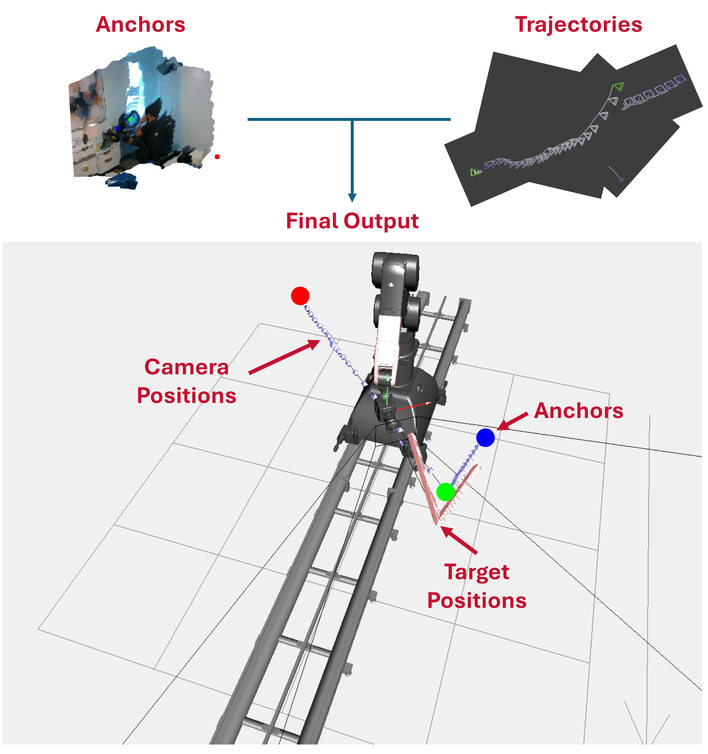


Fig 16: Final Trajectory Imported to FLAIR Classic

References

[1] Campos, C., Elvira, R., Rodriguez, J. J. G., Montiel, J. M. M., & Tardos, J. D. (2021). ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6), 1874–1890. https://doi.org/10.1109/tro.2021.3075644

[2] Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., & Shan, Y. (2024). YOLO-World: Real-time open-vocabulary object detection. arXiv preprint arXiv:2401.17270. https://arxiv.org/abs/2401.17270

[3] Courant, R., Dufour, N., Wang, X., Christie, M., & Kalogeiton, V. (2024). E.T. the exceptional trajectories: Text-to-camera-trajectory generation with character awareness. arXiv preprint arXiv:2407.01516. https://arxiv.org/abs/2407.01516

[4] Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). SAM 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714. https://arxiv.org/abs/2408.00714

[5] Zhang, M., Wu, T., Tan, J., Liu, Z., Wetzstein, G., & Lin, D. (2025). GenDoP: Auto-regressive camera trajectory generation as a director of photography. arXiv preprint arXiv:2504.07083. https://arxiv.org/abs/2504.07083