



HAL
open science

OpenKinoAI: A Framework for Intelligent Cinematography and Editing of Live Performances

Rémi Ronfard, Rémi Colin de Verdière

► **To cite this version:**

Rémi Ronfard, Rémi Colin de Verdière. OpenKinoAI: A Framework for Intelligent Cinematography and Editing of Live Performances. *Leonardo*, 2022, 55 (4), pp.373-377. 10.1162/leon_a_02226 . hal-03778661

HAL Id: hal-03778661

<https://inria.hal.science/hal-03778661>

Submitted on 30 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

OPENKINOAI: A FRAMEWORK FOR INTELLIGENT CINEMATOGRAPHY AND EDITING OF LIVE PERFORMANCES

Rémi Ronfard and Rémi Colin de Verdière,

Univ. Grenoble Alpes, Inria, LJK, CNRS, Grenoble INP, 38000 Grenoble, France.

Email: remi.ronfard@inria.fr.

Submitted: January 4, 2021.

Revised : June 4, 2021.

Published: August 1, 2022.

Abstract

OpenKinoAI is an open source framework for post-production of Ultra High Definition video which makes it possible to emulate professional multiclip editing techniques for the case of single camera recordings. OpenKinoAI includes tools for uploading raw video footage of live performances on a remote web server, detecting, tracking and recognizing the performers in the original material, reframing the raw video into a large choice of cinematographic rushes, editing the rushes into movies, and annotating rushes and movies for documentation purposes. OpenKinoAI is made available to promote research in multiclip video editing of Ultra High Definition video, and to allow performing artists and companies to use this research for archiving, documenting and sharing their work online in an innovative fashion.

1 INTRODUCTION

Adapting a stage performance to screen is a difficult task which presents many challenges even to expert human cinematographers and film editors [1]. We propose the OpenKinoAI framework as a toolset for making AI-based video production accessible to theater professionals and scholars.

Previous work in AI-based video production has investigated video lectures [2], personal videos [3], talk shows [4] and sports [5]. OpenKinoAI is the first platform directly addressing the challenges of artistic live performances such as theater, opera and dance. OpenKinoAI is an open source framework for ingesting video recordings of live performances, breaking them down into meaningful cinematographic rushes and editing them into movies.

Our framework provides support for the various tasks involved in video production, including the processing and handling of Ultra High Definition video and their metadata between a web server and a web client, the synchronized display of the generated movies, and the necessary graphical user interfaces for handling and manipulating multiple timelines. We make the simplifying assumption that the generated movies maintain the temporal continuity of the recorded performances.

As a result, we only provide support for emulating live multi-camera editing in post-production, rather than general non-linear three-point editing, which is left for future work.

2 ARCHITECTURE

OpenKinoAI is an open software environment which implements rush generation methods from previous work [6] and provides an API for future work in intelligent film editing of those rushes. Our software architecture carefully separates tasks in cinematography, which are implemented on the server side, from tasks in film editing, which are implemented on the client side. We describe the main functionalities currently available in OpenKinoAI in terms of video processing (Section 3), rush generation (Section 4), video editing (Section 5) and annotation (Section 6). We then describe extensive experimental results obtained in the last two years while implementing the system and discuss the limitations and perspectives of the OpenKinoAI software.

3 PREPARATION

We compress the original (4k or 6k) source with HEVC codec, and create three low resolution sources (360p, 540p and 1080p), one audio source and create a dash manifest [7] for web viewing. In order to provide fast and comfortable viewing experience, we choose to split the footage into ten minute segments and process them separately.

3.1 Actor detection

We use the Openpose toolkit to detect a set of 25 keypoints for each actor present on stage [8]. We obtain a 2D skeleton for every actor on stage at every frame. The skeleton is based on the dataset Body 25, a point of the skeleton is made of 2D coordinate plus a confidence value which expresses the probability of the detection.

3.2 Tracklet generation

Openpose detection gives a set of actors skeletons for each frame but it does not allow to track actor position over time. The goal here is to link the actor detection through the time with an IoU comparison between frame t and $t+1$. A tracklet is defined as a sequence of openpose detections. To create the sequence, first we iterate over frame. For each actor detection in the frame, we compute an upper bounding box with the neck, nose, eyes and ears position. Then we compare the intersection over union with the previous frame. We link the indexes of detections which have the best intersection over union ratio.

3.3 Actor identification

Once we have generated a small enough number of tracklets, we need to attach them with the corresponding actors names. In previous work, we learned appearance models for each actor [9]. In practice, however, actors change costumes quite frequently, either off stage between scenes, or even on stage in the course of a single scene. As a result, the appearance models need to be updated quite frequently. Lighting may also change the appearance of actors quite dramatically. As a result, we instead allow users to assign tracklets to actors manually by dragging and dropping them in the actor timelines in the web interface, as shown in Figure 2.

4 FRAMING

OpenKinoAI implements a generic, optimization-based reframing method first described by Gandhi et al. [10], for automatically reframing high resolution video into a variety of shot types and shot sizes. The new implementation is efficient and robust to large video file sizes, and has been tested on extended live performances lasting up to two hours in resolution up to 6K. The framing view of the web interface is shown in Figure 3.

4.1 Computing desired camera frames

Given a desired shot size and subject matter, we compute a desired camera frame at each time step in the video, taking into account stage constraints (S), gaze constraints (G) and composition constraints (Keepout and Pullin). Gandhi et al. take special care of undesired actors who accidentally appear within the camera frame by choosing between two possible strategies. Whenever possible, undesired actors are kept out of the camera frame with a repulsive force applied to the camera frame during optimization. When undesired actors come too close to the desired actors, this strategy may fail. Instead, undesired actors can be pulled in to the camera frame in those cases. In OpenKinoAI, we take a slightly different approach where we momentarily change the subject matter and size of the desired shot as needed to pull in or keep out the undesired actors.

4.2 Best effort optimization

Given the desired framings at each time step, we then stabilize the camera by solving a large convex optimization problem, as described by Gandhi et al. [11]. For efficiency, this computation is performed in slices of ten seconds and the continuity between slices is enforced with hard constraints. The computation time is fast and independent of the video resolution.

5 VIDEO EDITING

OpenKinoAI provides tools for online video editing using all available rushes into movie scenes. The client application is shown in Figure 4. Users can drag and drop rushes to a timeline and refine the editing by moving transition between shots interactively. This mode of editing mimics live editing techniques used in real-time multi-camera production [12].

Both the rushes and the edited movie scenes inherit the metadata computed during preparation (i.e. names and screen locations of performing actors, gaze directions and moving directions).

6 ANNOTATION

In addition to rush generation and multicamera editing, OpenKinoAI provides a web interface for manually transcribing and annotating live performances. The annotation view of the web interface is shown in Figure 5. It is engineered to provide visual support for various annotation tasks, by showing the appropriate framings.

Typically, speech transcription can be facilitated by showing close up shots of the speaking actors, allowing to quickly identify speakers and transcribing their speech even in cluttered conversation, with loud music and sound effects on stage. Stage directions annotation can be facilitated by showing full shots of the moving actors together with an ensemble shot of the entire stage. The annotation of other scenographic devices (lighting, video projection, stage decoration) can be facilitated by showing a wide shot of the entire stage.

OpenKinoAI does not currently provide support for automatic speech transcription, automatic recognition of stage directions or automatic annotations of scenography. Our experience is that those are difficult problems in most practical situations. Instead, we provide annotation tools for creating reliable ground truth annotation, suitable for broadcasting and documentation of theatre art. An open direction for future work will be to collect such ground truth data and use them as training data for specialized neural networks attempting to recognize actors and transcribe their speech and stage actions automatically.

7 IMPACT

OpenKinoAI is an open source software architecture which makes it possible for theatre companies to gain better artistic and technical control over the cinematography and the editing of single-camera video recordings. With Ultra High Definition cameras now available at reasonable prices, we hope OpenKinoAI will be useful for creating movies of live performances and publishing them online. The OpenKinoAI toolset emulates professional video production tools which were previously not available to a general audience. This is important for theatre companies who can use our system to build memories of their creative work and make them available to the general public at a limited cost.

OpenKinoAI was used extensively to document rehearsals and performances of a 2018 stage adaptation of Mary Shelley's *Frankenstein* by Jean-François Peyret [13]. A single 4K camera was used with a crew of just one person. A large variety of shots (Figure 6) were created in post-production for documenting the complex, improvisation-based creative process of Peyret and his actors [14] and for broadcasting live performances in various versions and formats, including multi-screen installations. Those options, not available to traditional live multi-camera broadcasts, offer promising new directions for documenting live performances.

OpenKinoAI is made available both as an online service allowing end-users to freely upload, prepare, reframe, edit and annotate their own footage [15]; and as a toolkit allowing software developers to configure their own server for non-commercial applications [16]. A typical server is a Linux machine with a fast GPU and large scale storage device. A typical client is a laptop computer running a web browser in any operating system.

We hope the software can be useful to other researchers in intelligent cinematography and editing as a generic environment where they can implement their own tools for generating cinematographic rushes and editing them into movies.

8 CONCLUSION

OpenKinoAI provides all the necessary tools for generating and annotating cinematographic rushes from raw video footage in Ultra High Definition. OpenKinoAI enforces a strict separation between heavy duty video processing on the server side, and interactive control and immediate feedback on the client side. Directions for future work include support for split-screen composition [17], automatic rough-cut editing [18] and non-linear three-point editing. OpenKinoAI is distributed in the hope that it can serve as a common test bed and application

framework for future research in intelligent cinematography and editing and facilitate their deployment in real life applications.

ACKNOWLEDGMENTS

OpenKinoAI was supported by Inria and the French National Research Agency in the framework of the "Investissements d'avenir" program (ANR-15-IDEX-02) through the Cross Disciplinary Program Performance Lab.

We thank Jean-François Peyret, Jeanne Balibar, Jacques Bonnaffé, Victor Lenoble and Joel Maillard for allowing us to reproduce images of their rehearsal work. We thank Julie Valéro for continuous support and advice and Frédéric Devernay for invaluable contributions to earlier versions of the project.

References and Notes

- [1] R. Ronfard et al. 2015. Capturing and Indexing Rehearsals: The Design and Usage of a Digital Archive of Performing Arts. In Digital Heritage. IEEE, Grenade, Spain, 533–540.
- [2] Cha Zhang, Jim Crawford, Yong Rui, and Li-wei He. 2005. An Automated End-to-End Lecture Capturing and Broadcasting System. In ACM International Conference on Multimedia. 808–809.
- [3] Michael L. Gleicher and Feng Liu. 2008. Re-Cinematography: Improving the Camerawork of Casual Video. ACM Trans. Multimedia Comput. Commun. Appl. 5, 1 (Oct. 2008), 28.
- [4] C. Wright et al. 2020. AI in production: video analysis and machine learning for expanded live events coverage. SMPTE Motion Imaging Journal 129.2 (2020), 36–45.
- [5] Meng L Chen J and Little J. 2018. Camera Selection for Broadcasting Soccer Games. In Winter Conference on Applications of Computer Vision (WACV), 427–435.
- [6] Vineet Gandhi, Rémi Ronfard, and Michael Gleicher. 2014. Multi-Clip Video Editing from a Single Viewpoint. In European Conference on Visual Media Production.
- [7] Iraj Sodagar. 2011. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. IEEE MultiMedia 18, 4 (Oct. 2011), 62–67.
- [8] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. 2017. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In Computer Vision and Pattern Recognition (CVPR), 1302–1310.
- [9] V. Gandhi, R. Ronfard. 2013. Detecting and Naming Actors in Movies Using Generative Appearance Models. In Computer Vision and Pattern Recognition (CVPR), 3706–3713.
- [10] Gandhi et al. [6]
- [11] Gandhi et al. [6]
- [12] Mitch Jacobson. 2010. Mastering MultiCamera Techniques: From Preproduction to Editing and Deliverables Focal Press, Oxford.
- [13] R. Ronfard, J. Valero. 2020. OpenKinoAI and the audiovisual notebook : a plural solution for rehearsal studies. European Journal of Theatre and Performance, 334-375.
- [14] <https://www.youtube.com/channel/UCuDwJmgBxFW40UrQ2uixE2g>
- [15] <http://OpenKinoAI.inria.fr>
- [16] <http://gitlab.inria.fr/ronfard/openkinoai>
- [17] Moneish Kumar, Vineet Gandhi, Rémi Ronfard, and Michael Gleicher. 2017. Zooming On All Actors: Automatic Focus+Context Split Screen Video Generation. Computer Graphics Forum 36, 2 (2017), 455–465.
- [18] Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational Video Editing for Dialogue-driven Scenes. ACM Trans. Graph. 36, 4.

FIGURES



Fig. 1. KinoAI workflow. Bottom: The stage is filmed with a fixed camera in ultra-high definition. Top: We generate candidate shots with different compositions on the server side and allow remote users to edit them into movies on the client side. (© Rémi Ronfard/Inria)

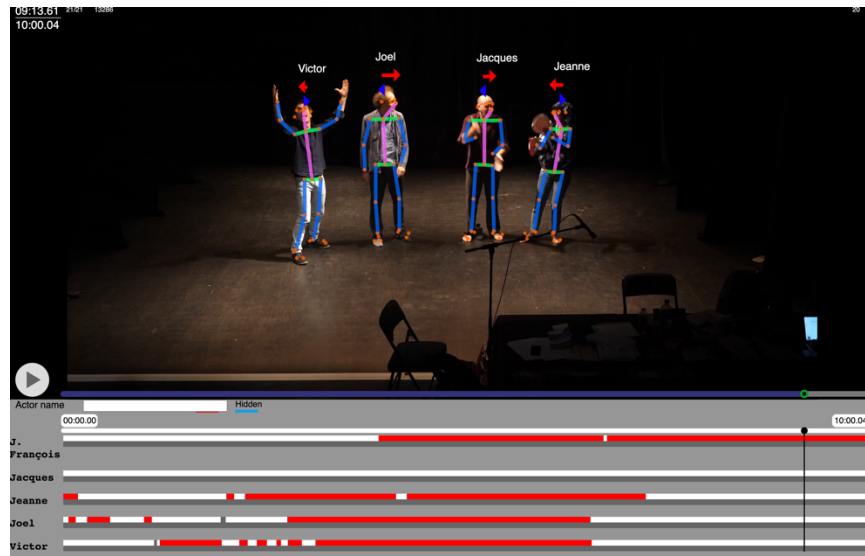


Fig. 2. Preparation view. OpenKinoAI provides tools for labeling tracklets from OpenPose detections and assigning them to timelines. (© Rémi Ronfard/Inria)

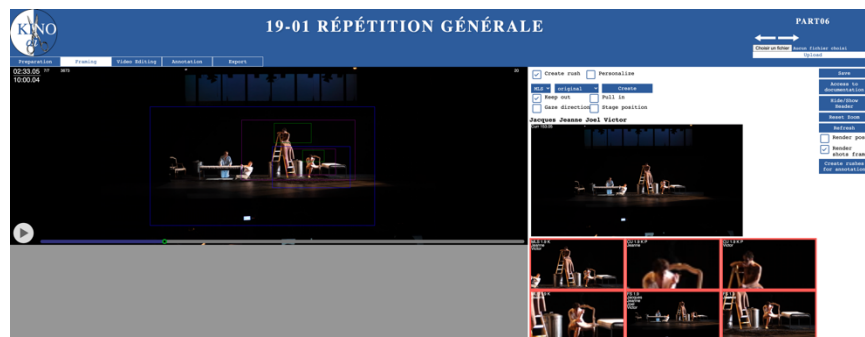


Fig. 3. Framing view. OpenKinoAI provides tools for choosing shot sizes and compositions and previewing the generated rushes. (© Rémi Ronfard/Inria)



Fig. 4. Video editing view. OpenKinoAI provides tools for online video editing using all available rushes into movie scenes. (© Rémi Ronfard/Inria)



Fig. 5. Annotation view. OpenKinoAI provides tools for annotating rushes and movie scenes. (© Rémi Ronfard/Inria)



Fig. 6. Gallery of shots computed by OpenKinoAI during rehearsals of *The making of monsters* by Jean-François Peyret (after Mary Shelley's *Frankenstein*). (© Rémi Ronfard/Inria)