Real-Time Video Processing for Dynamic Content Creation

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering by Research

by

Sudheer Achary
20161076
sudheer.achary@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
November 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Real-Time Video Processing for Dynamic Content Creation" by Sudheer Achary, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Vineet Gandhi,
Center for Visual Information Technology,
Kohli Center on Intelligent Systems,
IIIT Hyderabad.

To my family

Acknowledgments

First of all, I would like to express my sincere gratitude to Prof. Vineet Gandhi for introducing me to research. His rigorous process of reviewing research papers before submitting them to conferences made me comprehend the way for exhibiting thoughts. His constant guidance helped me to develop both professionally and personally.

I would also like to thank my colleagues K L Bhanu Moorthy, Ashar Javed, Rohit Girmaji and Adhiraj Anil Deshmukh for their constant support and for providing a positive learning environment.

I express my deepest gratitude toward my parents for their unconditional love and support throughout my journey.

Abstract

Autonomous camera systems are vital in capturing dynamic events and creating engaging videos. However, existing filtering techniques used to stabilize and smoothen camera trajectories often fail to replicate the natural behavior of human camera operators. To address these challenges, our work proposes novel approaches for real-time camera trajectory optimization and gaze-guided video editing. We introduce two online filtering methods: *CineConvex* and *CineCNN*. CineConvex utilizes a sliding window-based convex optimization formulation, while CineCNN employs a convolutional neural network as an encoder-decoder model. Both methods are motivated by cinematographic principles, producing smooth and natural camera trajectories. Evaluation of basketball and stage performance datasets demonstrates superior performance over previous methods and baselines, both quantitatively and qualitatively. With a minor latency of half a second, CineConvex operates at approximately 250 frames per second (fps), while CineCNN achieves an impressive speed of 1000 fps, making them highly suitable for real-time applications.

In the realm of video editing, we present Real Time GAZED, a real-time adaptation of the GAZED framework. It enables users to create professionally edited videos in real-time. Comparative evaluations against baseline methods, including the non-real-time GAZED, demonstrate that Real Time GAZED achieves similar editing results, ensuring high-quality video output. Furthermore, a user study confirms the aesthetic quality of the video edits produced by Real Time GAZED.

With the advancements in real-time camera trajectory optimization and video editing presented, the demand for immediate and dynamic content creation in industries such as live broadcasting, sports coverage, news reporting, and social media content creation can be met more efficiently. The elimination of time-consuming post-production processes and the ability to deliver high-quality videos in today's fast-paced digital landscape are the key advantages offered by these real-time approaches.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The availability of compact and affordable cameras with advanced features, capable of capturing videos at high resolutions like 4K, has made high-quality video recording accessible to a broader audience. While this ease of recording has simplified the process, the subsequent stage of video production - editing - remains a labor-intensive task that demands skill and expertise.

Due to such high availability of video-capturing devices, The demand for immediate and dynamic video production has increased across various industries, including live broadcasting, sports coverage, news reporting, and social media content creation. Creating professional recordings of live stage performances involves skilled camera operators who capture the performance from various angles. These different camera feeds are then combined through editing to create a polished and engaging final product. However, generating these professional edits is a challenging task. Firstly, operating cameras during a live performance is rugged even for experts, as there are no second chances to retake footage, and there are limitations on camera angles due to the impracticality of using large equipment like trolleys or cranes. Secondly, manual video editing is a slow and laborious process that requires the expertise of skilled editors. All in all, the process of producing professional recordings of live performances requires a professional camera crew, multiple cameras and equipment, and experienced editors, which increases the complexity and costs of the process.

Video production encompasses three primary stages: pre-production, production, and post-production. During pre-production, all planning aspects of the video production process are addressed before filming commences. This includes tasks such as scriptwriting, scheduling, and logistical arrangements. Production involves capturing the video content, which entails filming the subjects or scenes of the video. Finally, post-production is the phase where the captured video clips are meticulously combined through video editing that conveys a story or communicates a message effectively. However, it is essential to note that post-production editing can be a tedious process, as it requires a skilled individual to carefully review all the recorded footage and make informed decisions on structuring and presenting the video content.

Many video production companies commonly employ fixed wide-angle cameras positioned at a considerable distance to capture the entire stage. While these static recordings are helpful for archiving

purposes and provide an overall understanding of the context, they often fail to captivate the audience's attention. These distant camera feeds cannot showcase crucial elements of cinematic storytelling, such as close-up shots of faces, capturing emotions and character actions, and highlighting interactions between actors. A high-resolution static camera can replace the need for multiple camera crews by simulating virtual panning, tilting, and zooming within the original recordings. This technique, Virtual PTZ (Pan-Tilt-Zoom) multiple-camera simulation, enhances the viewing experience by introducing variety, emphasis, and clarity to the content. Creating the illusion that the video was recorded using multiple cameras capturing different angles and focal lengths gives the impression of a more dynamic and visually captivating presentation.

Creating an engaging video edit from a wide angle video recording typically involves following steps:

- *Generating Shots*: Identify key moments or scenes in the video where multiple camera angles or movements would enhance the storytelling or visual impact. These shots will serve as the reference points for the virtual camera simulation.

- *Camera Path Planning*: Determine the desired camera movements, such as panning, tilting, and zooming, for each reference shot. Plan the trajectory and timing of these movements to create a natural and cinematic effect.

- *Virtual Camera Placement*: Based on the camera movements planned, position virtual cameras within the video scene. These virtual cameras represent the different viewpoints or angles that would be captured by physical cameras if they were actually present.

- *Shot Transition*: Smoothly transition between virtual cameras by applying seamless cuts, fades, or other transition techniques. Use blending and cross-dissolves to ensure a natural and fluid flow between different camera perspectives.

Identifying key moments in video footage using human gaze can be invaluable. Human perception and understanding of storytelling nuances allow for subjective judgment and interpretation of what constitutes significant or impactful moments. By involving human gaze in the process of identifying key moments, video editors can benefit from their expertise and contextual understanding. However, relying on human gaze for identification of key moments may have some limitations. It can be subjective and dependent on individual perspectives, leading to potential bias. Additionally, the process can be time-consuming, especially for large amounts of footage.

For camera path planning and virtual camera placements we use camera systems that autonomously track individuals, objects, or actions of interest often apply to filter operations to refine raw estimations on a frame-by-frame basis. These systems mimic skilled camera operators' techniques, such as cropping a window around a subject to create a virtual camera that follows their movements. Trajectory optimization is also employed in video stabilization methods to generate videos with smooth, stabilized trajectories. In contrast, professional cinematographers employ a diverse range of stabilization tools

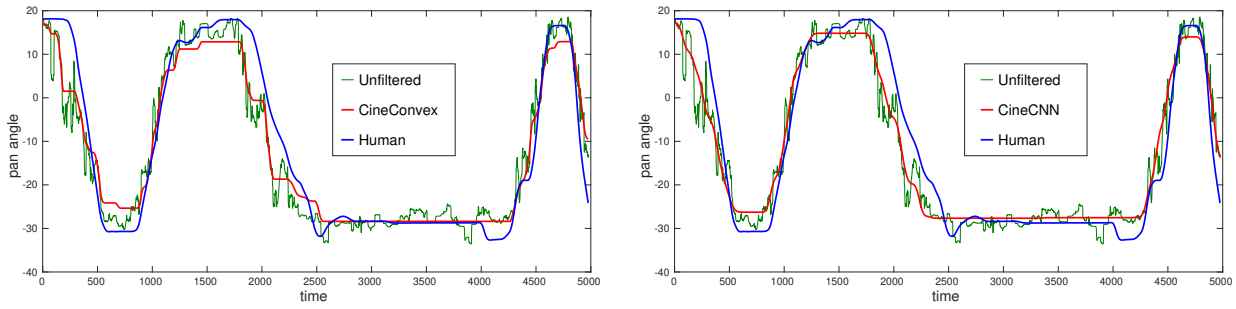

Figure 1.1: Both CineCNN and CineConvex exhibit comparable performance, although CineCNN demonstrates a superior quantitative metric compared to CineConvex.

such as tripods, camera dollies, and steadycams. While optical stabilization systems primarily address high-frequency jitter, they often struggle to eliminate low-frequency distortions that commonly occur during handheld panning shots or videos filmed while walking. These low-frequency distortions can negatively impact the overall stability of the footage.

The goal of trajectory optimization and filtering models is to transform the original trajectory to closely resemble the behavior of experienced camera operators. According to cinematographic literature, a well-executed pan/tilt shot typically consists of three components: an initial static period, a smooth camera movement following the subject's motion, and a final stationary period. Previous research studies [20, 19, 47] have demonstrated that this behavior can be formulated as an optimization problem, resulting in trajectories composed of piece-wise constant, linear, and parabolic segments. This work proposes fast and lightweight real-time camera trajectory optimization models motivated by cinematic principles.

The optimization of camera motion in real-time plays a crucial role in video editing, especially when considering real-time content creation. Real-time video editing has become essential for quickly delivering high-quality videos in today's fast-paced digital environment, eliminating the need for time-consuming post-production processes. We introduce a fast and lightweight real-time version of the gaze-guided video editing framework to meet this demand. Through a comparison with other baselines, including non-real-time methods, we showcase the effectiveness of our real-time editing approach, highlighting its ability to achieve efficient and responsive video editing results.

1.1 Key Contributions

1. ***CineFilter***: We introduce the *CineConvex* formulation, which transforms the offline trajectory optimization process into an online one. Filtering becomes a convex optimization problem with minimal latency by adopting a sliding window approach. We impose constraints on the optimized trajectory based on the previous window's results. Additionally, we present *CineCNN*, a camera trajectory smoothing function learned through a convolutional neural network. By inputting a

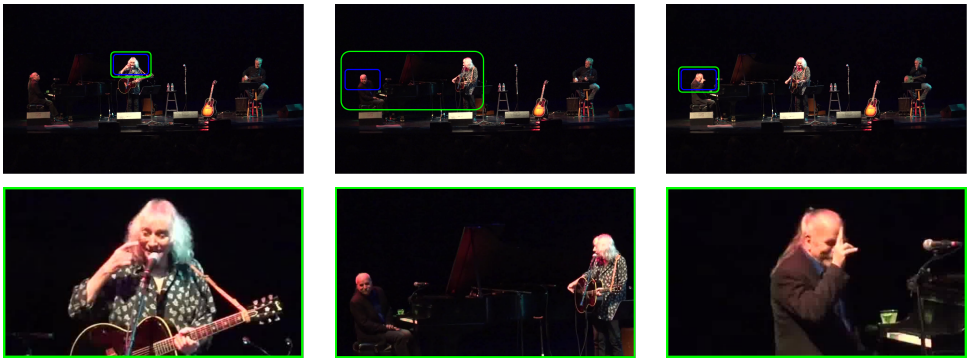

Figure 1.2: The top row displays bounding boxes representing shots selected by two algorithms: Real Time GAZED (*highlighted in green*) and GAZED (*highlighted in blue*). These shots correspond to the frames chosen by each algorithm for video editing. In the bottom row, we can observe the actual cropped shots that result from the selections made by the Real Time GAZED algorithm.

denoised trajectory version, we generate trajectories that closely resemble the behavior of skilled camera operators. This predictive model offers several advantages, including reduced computational load and improved noise robustness. With *CineConvex* and *CineCNN*, we enhance the efficiency and accuracy of camera trajectory optimization, enabling real-time applications with superior performance.

2. ***Real Time GAZED***: We propose *Real Time GAZED*, our innovative real-time adaptation of the gaze-guided video editing framework GAZED. We have transformed the shot selection component into a real-time process by harnessing the core elements of GAZED, including shot generation and gaze potential. This advancement incorporates a small lookahead and shot continuity constraints, ensuring seamless and coherent shot transitions in the edited video. With Real Time GAZED, we enable efficient and dynamic video editing in real-time, revolutionizing the way we create engaging and captivating visual content.

Chapter 2

Background

Cinematography has evolved its conventions and rules for visually conveying information to viewers. This collective set of rules, conventions, and unique vocabulary forms the grammar of cinematography. In this section, we explore the widely accepted terminology and guidelines that dictate the creation and presentation of visual elements. These principles are derived from established cinematography and editing literature, providing a comprehensive overview of the foundations of cinematic language.

2.1 Aspect Ratio

The aspect ratio of an image refers to the proportional relationship between its width and height. It is typically represented in the format W:H, where W denotes the width and H denotes the height. For instance, an aspect ratio of 16:9 indicates that for every 16 units of width, there are 9 units of height. The most prevalent aspect ratios in cinema are 1.85:1 and 2.39:1. Conversely, for television and online videos, commonly used aspect ratios are 4:3 and 16:9.

2.2 Types of Shots

Framing the main subject within a shot offers various possibilities, from capturing their entire body to focusing solely on their eyes. Shot types can be categorized into four main sizes: Long, Full, Medium, and Close-up.

- *Long Shot*: This shot captures the subject or the scene from a distance, providing an overview of the surroundings. It is commonly used to establish a scene or provide context. Hence it is sometimes referred to as an establishing shot.
- *Full Shot*: In a full shot, the entire character is framed from head to toe, occupying a significant portion of the frame. This shot emphasizes action and movement rather than focusing on the character's emotional state.

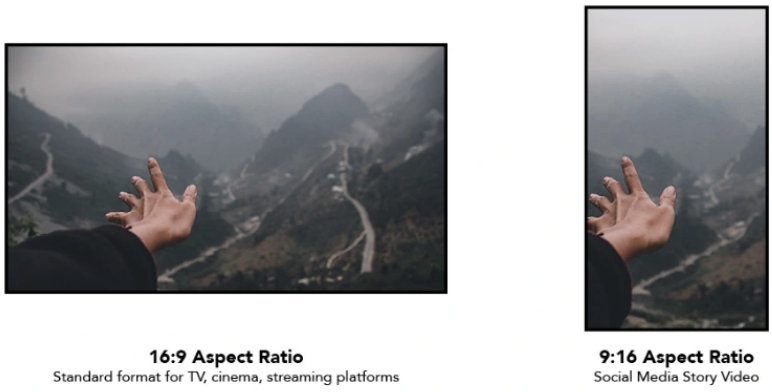

Figure 2.1: The figures depict examples of images with different aspect ratios. One image demonstrates a 16:9 aspect ratio, while the other showcases a 9:16 aspect ratio.

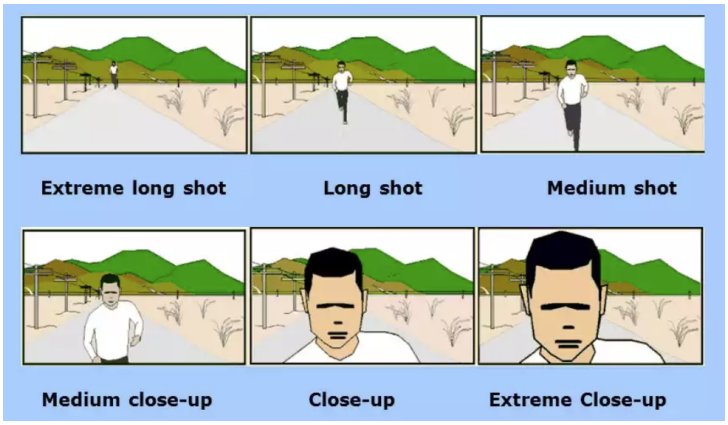

Figure 2.2: The figure displays various shot types, including Long, Medium, and Close-up shots, among others. These shot types represent different framing techniques used in cinematography to capture subjects from different distances and perspectives.

- *Medium Shot*: A medium shot displays a portion of the subject in more detail, typically framing them from the waist up. It is frequently used in films to focus on characters while still providing some context of the environment.

- *Close-Up*: The close-up shot tightly frames part of the subject, such as their head or face. By filling the screen with this level of detail, the shot emphasizes the character's emotions and reactions, becoming the dominant element in the scene.

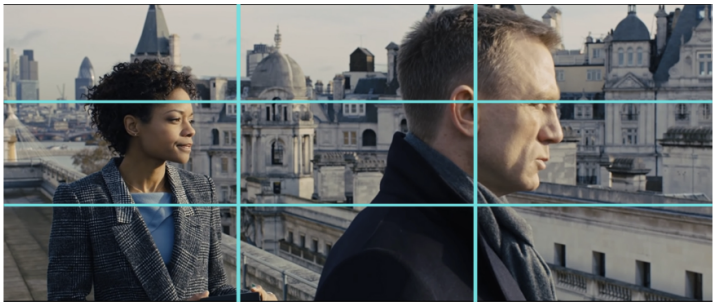

Figure 2.3: The figure illustrates a sample picture that demonstrates the principle of cinematic composition known as the rule of thirds.

2.3 Composition

Composition in cinematography pertains to the framing of the image and the arrangement of elements within it. When visually storytelling through filmmaking, adhering to specific composition guidelines is essential. Commonly used rules in cinematography are the Rule of Thirds etc. These guidelines help create visually pleasing compositions. However, it is important to note that these rules are flexible and can be overlooked when there is a clear motivation and purpose to conceive extraordinary ideas and perspectives.

2.4 Cut

In cinematography, a sequence refers to a cohesive collection of shots forming a distinct narrative unit. A cut is a sudden and abrupt transition from one sequence to another, signaling a shift in the storytelling.

2.5 Gaze Capturing

Eye-tracking technology is commonly employed to capture the human gaze for video editing. This technology records and analyzes where a person directs their gaze while watching a video. It provides valuable insights for understanding viewer attention and implementing gaze-guided editing techniques. This study utilized the Tobii Eye-X eye-tracker, which has a 60Hz sampling rate and offers gaming-level eye-tracking precision. Before recording, the eye tracker was calibrated using the 9-point method. It involves sequentially focusing on nine predefined points on the screen or viewing area to establish a mapping between recorded eye movements and corresponding positions on the display. The video presentation and gaze recording protocol were developed using MATLAB PsychToolbox [28]. The videos were presented to participants in a fixed order for gaze recording purposes.

Figure 2.4: The white dots marked on the video frame in the figure represent the captured human gaze positions.

2.6 Filtering Methods

Autonomous camera systems that track actors in videos are essential for automatic video editing. These systems utilize filtering models to refine raw estimations on a frame-by-frame basis, resulting in trajectories that closely resemble the behavior of experienced camera operators. Our study compares our approach with established filtering methods such as Kalman, Bilateral, and Savitzky-Golay. These methods serve as baselines for comparison, allowing us to assess the effectiveness and performance of our proposed approach.

1. *Savitzky Golay*: The Savitzky-Golay (SG) filter is a data smoothing technique that uses least squares to fit a polynomial of a specified degree within a window of consecutive data points surrounding each point. It calculates the smoothed data point by taking the central value in the window, and this process is applied to each point in the time series.

2. *Kalman Filter*: The Kalman Filter is a recursive Bayesian estimation technique that updates the current camera state based on previous predictions and the recent observation..

3. *Bilateral Filter*: The bilateral filter is a powerful smoothing filter that is non-linear, adaptive, and capable of preserving edges while reducing noise. It operates by computing a weighted average of neighboring data points, where two Gaussian functions determine the weights. One Gaussian is based on the spatial distance between points, while the other is based on the difference in magnitude between the given point and its neighbors.

4. *MeshFlow*: MeshFlow [33] is an approach that aims to achieve video stabilization with low latency. It optimizes the motion estimation locally at each frame by considering a few previous frames. The optimization process consists of two terms. The first term calculates the mean squared error (MSE) between the predicted and original values of the 1D signal. The second term computes the MSE between the first-order differentials of the expected signal.

2.7 Convex Optimization Solver

Convex optimization refers to a specific class of optimization problems characterized by the convexity of both the objective function and the constraints. In mathematics, a function is considered convex if any two points on the function are connected by a line segment that lies either on or above the graph of the function. Similarly, a convex set is a set where any line segment connecting two points within the set remains entirely within the set. Convex optimization aims to find the optimal solution that minimizes or maximizes the objective function while adhering to the constraints. The advantage of convex optimization is that it provides a well-defined framework with efficient algorithms to solve many optimization problems. Ensuring convexity guarantees that the solution found is globally optimal rather than getting stuck in local optima.

We utilize the Gurobi solver [44] for real-time camera trajectory stabilization, which involves solving a convex optimization problem. Gurobi is a highly efficient solver for tackling various convex optimization problems. It is capable of handling various types of convex optimization, such as linear programming (LP), quadratic programming (QP), second-order cone programming (SOCP), and mixed-integer linear programming (MILP). Several optimization modeling libraries in Python can use the Gurobi solver. One prominent library is CVXPY, a popular Python tool for convex optimization. With CVXPY, you can express convex optimization problems clearly and concisely using mathematical syntax. It supports multiple solvers, including Gurobi, enabling efficient solutions to convex optimization problems.

Chapter 3

Related Work

According to the acclaimed film editor Walter Murch, one of the key goals in film editing is to capture the emotion and expression portrayed through an actor's eyes. And if that's not feasible, the next best option is to aim for a compelling close-up shot of the actor, even if the initial wide-shot seems sufficient to convey the scene. Murch's insight sheds light on the importance of visual storytelling and the power of capturing the subtle nuances of an actor's performance. By focusing on the eyes, which are often considered windows to the soul, editors have the opportunity to connect the audience with the characters on a deeper, more emotional level. This technique allows the audience to truly feel and experience the story unfolding before them.

Exploring the world of video editing in virtual 3D environments has allowed creators to leverage established cinematic techniques to enhance the impact of their animated content. Initially, researchers focused on adopting an idiom-based approach [12] [21] [24], relying on tried-and-true formulas to capture scenes through specific sequences of shots. While this approach proved effective in many cases, it encountered limitations when faced with unconventional scenarios. Each unique situation required a tailored solution, making it challenging to rely solely on pre-established formulas. Additionally, when working with live theatre productions, where spontaneity reigns supreme, the acquisition of all the shots prescribed by these formulas can prove impractical.

A wave of research papers has approached video editing as an intriguing puzzle to solve. By treating it as a discrete optimization problem [17] [18] [31] [34], Most of them have employed dynamic programming techniques to find the best combination of shots that maximize viewer engagement. Let's take a closer look at some of these studies. Elson et al [17] tackle the intertwined challenges of camera placement and selection. However, their approach lacks sufficient detail to be easily replicated. On the other hand, Meratbi et al [31] utilize a Hidden Markov Model to guide the editing process. They learn shot transition probabilities from existing films, albeit focusing solely on dialogue scenes, which requires manual annotation from real movies.

Among these endeavors, Galvane et al [18] work stands out as a comprehensive effort in video editing. They meticulously address crucial aspects, including the precise placement of cuts, adhering to rhythm, and maintaining continuity editing rules. Their contribution is undoubtedly noteworthy.

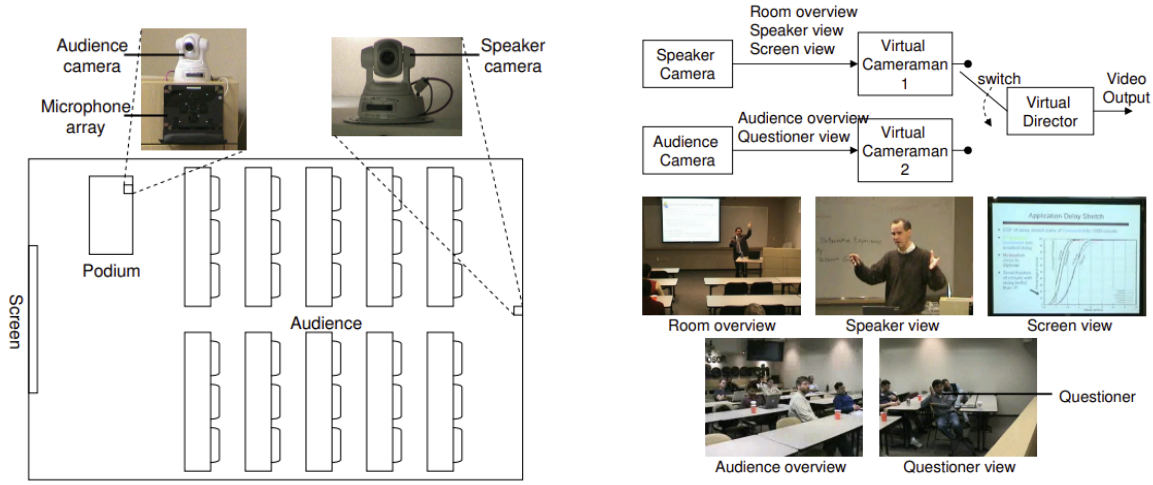

Figure 3.1: The figure illustrates the typical setup of the iCam2 system and the process of video capturing in [52].

While our own research draws inspiration from these innovative works, we face unique challenges when dealing with stage performances. Unlike the freedom to position cameras as desired or access to detailed scene geometry and character information found in 3D environments, our constraints are significantly different. However, we remain driven to find effective solutions for video editing in this specific context.

Besides general automated video editing, there have been intriguing studies exploring automated editing in specific scenarios. One fascinating example is the Virtual Videography system [23], which emulates shots taken by virtual pan-tilt-zoom cameras during lecture videos. By employing a branch-and-bound optimization technique, the system identifies the most captivating shot.

Similarly, the MSLRCS [52] and Auto Auditorium [2] systems employ a handful of fixed cameras, including shots of presentation slides. However, their editing approach is rule-based and primarily suited for constrained environments, typically featuring a lone presenter in front of a chalkboard or slide screen. These innovative systems offer valuable insights into automating the editing process and provide exciting possibilities for enhancing video content in specific contexts. By leveraging cutting-edge techniques and harnessing the power of technology, automated video editing continues to evolve and revolutionize the way we capture and present visual information.

Ranjan et al. [42] have come up with an intriguing system that revolutionizes the editing of group meetings. By incorporating various cues such as speaker detection, posture changes, and head orientation, they have devised a set of simple yet effective rules to guide the editing process. For instance, when a change in speaker is detected, the system automatically cuts to a close-up shot of the speaker to capture their presence and emphasize their role. Similarly, when multiple individuals are engaged in conversation, the system intelligently switches to an overview shot, providing a comprehensive view of the dynamic discussion.

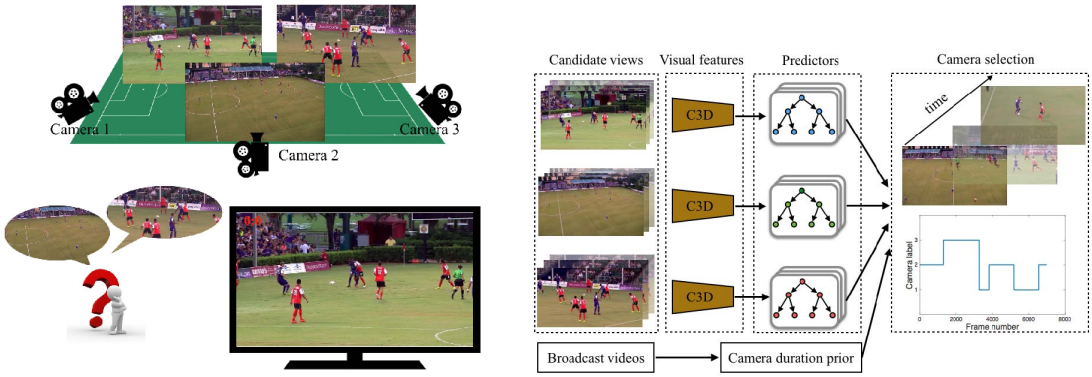

Figure 3.2: The figure demonstrates a method for the automatic selection of cameras in broadcasting soccer games, as described in the work on camera selection for broadcasting [11].

To assess the capabilities of their system, the researchers conducted experiments and compared it to a similar baseline approach that focuses on tracking the speaker(s) throughout the meeting. Additionally, Doubek et al. [16] delve into the realm of surveillance settings, exploring the intricate challenge of camera selection. These studies showcase the innovative strides being made in the field of video editing and camera selection, paving the way for more engaging and visually captivating experiences in various contexts. The field of camera selection in sports events has seen extensive research and development [7] [9] [11] [49]. Previous studies have utilized Hidden Markov Models [9] [49] to choose cameras from panoramic or multiple viewpoints.

Others have taken a data-driven approach, training regressors to understand the significance of each camera view at any given moment. In contrast to these approaches that rely on multi-camera feeds and scene-related metadata, Our method only requires a static wide-angle camera recording of a stage performance and eye gaze data from one or more viewers (even eye gaze recordings of the editor or director reviewing the event using a high-end eye-tracker would suffice). Additionally, our approach is versatile and applicable to various scenarios. So, while others grapple with complex setups and detailed information, our approach offers a simpler yet powerful solution. With just a wide-angle camera and eye gaze data, we can transform any stage performance into a captivating visual experience.

Previous studies have also explored the use of eye gaze in video editing, employing techniques such as head pose estimation or dedicated eye-tracking devices. For instance, Takemai et al. [46] proposed a video editing system that relies on the gaze direction of participants during indoor conversations. Their editing rule was simple but effective: they would cut to a close-up of the person receiving the most gazes from their peers. The results revealed that using gaze information improved the conveyance of conversation dynamics compared to a speaker detection-based approach. Similarly, Daigo et al. [14] utilized the gaze direction of the audience to estimate areas of interest during a basketball match. These studies showcase the potential of eye gaze in enhancing video editing by capturing salient moments and areas of interest. By incorporating gaze information, video editing can become more engaging, capturing the flow of conversations and highlighting key moments in sports or social events.

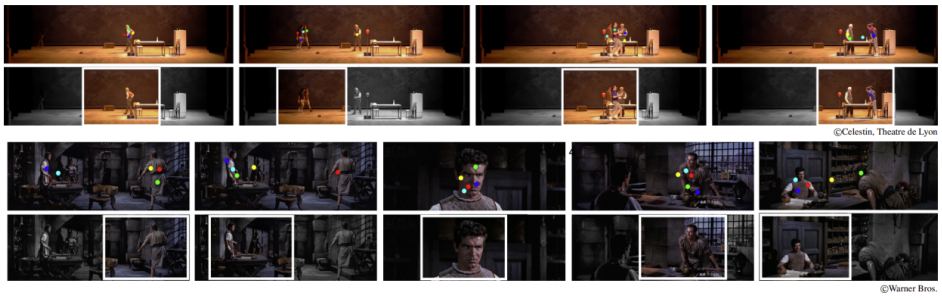

Figure 3.3: The figure displays the algorithm [40], which focuses on retargeting widescreen recordings to smaller aspect ratios. The image showcases the original recording, including overlaid eye gaze data from multiple users (each viewer represented by a distinct color)

Other studies [26] [40] have explored the use of eye gaze data in video retargeting, a process that involves adapting an edited video intended for one display device to another (such as from a theater screen to a mobile device). Typically, this is achieved by digitally moving a cropping window within the original video to preserve important content. However, these existing methods only focus on adjusting the horizontal (x) position and allowing minimal zoom based on gaze variance. As a result, they fail to consider the vertical (y) position, resulting in poorly composed frames and odd zooming effects. This can lead to awkward compositions [26] [40], like cropping actors in an unappealing way or covering their heads but not their faces.

Autonomous camera systems that track actors in a video play a central role in automatic video editing. Research in autonomous camera systems dates back to more than two decades. One of the earliest systems was proposed by Pinhanez and Bobick [39], which aimed at automated camera framing in cooking shows. Their system was based on two types of cameras, a spotting camera that watched the entire area of interest and a robotic tracking camera that followed the verbal instructions from a director to frame the desired targets automatically. The Autoauditorium [3] system extended this idea for lecture videos (a single presenter in front of a screen). The robotic camera in the Autoauditorium system crudely followed the presenter, whose position was estimated each frame using background subtraction on the spotting camera.

A series of works then followed, and we review the computational models for camera movement, proposed in these approaches. Yokoi *et al.* [51] present a method for automated editing of lecture videos. Their work replaces the robotic camera by a virtual camera, i.e., a cropping window moving inside a high-resolution video. They use temporal frame differencing to detect the Region of Interest (RoI) around the presenter and use bilateral filtering to remove the jittery motion introduced by per-frame RoI estimations. A similar digital tracking approach was employed in Microsoft's icam2 system [53]. The contemporary work by [45] also tracks cropped RoI from a panoramic video; however, it employs a Kalman filter for removing the jitter. Such filtering approaches successfully reduce jitter; however, they are not cinematically inspired, and they lead to unmotivated camera movements and fail to keep the camera static. Heuristics have been applied to tackle some of these challenges. For instance, [45] keeps

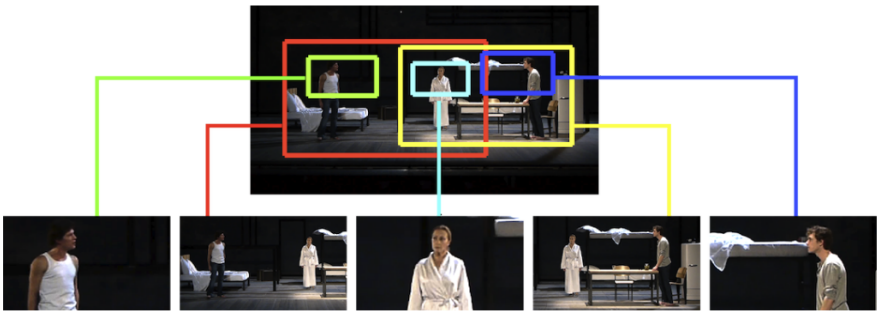

Figure 3.4: The figure depicts method [19] that takes a high-resolution video recorded from a single viewpoint as input. The output of their approach is a collection of synchronized subclips.

RoI unchanged if the Kalman filter predictions of new positions are within a specified distance of the registered position, and the new estimated velocity is below a threshold. However, such heuristics are not applicable in generalized scenarios.

The virtual camerawork has been investigated for the autonomous broadcast of sports. Unlike the classroom environment, where framing a single person is relatively simple, the sports videos have multiple players and fast-moving objects. Diago *et al.* [15] propose an offline system that uses the audience face direction to build an automatic pan control system (pan angle of the broadcasting camera). Ariki *et al.* [1] proposed a heuristic-based editing strategy. However, these approaches focus on the per-frame pan angle estimation and skip details on the camera motion models or smoothing algorithms. Chen *et al.* [8] uses Gaussian Markov Random Fields (MRF) to obtain smooth virtual camera movements. They induce smoothness by inducing inter-frame smoothness priors. Recent work [41] has shown that while inter-frame smoothness priors do induce smoothness, it fails to give aesthetically pleasing camera behavior. They augment it with an additional post-processing optimization to render professional-looking camera trajectories.

The virtual videography [22] system was one of the earliest works to model camera movement inspired by filmmaking literature [4, 48]. They define that a good tracking shot should consist only of smooth motions in a single direction that accelerate and decelerate gradually to avoid jarring the viewer while maintaining the correct apparent motion of the subject. They define a customized parametric function for the motion model and solve for parameters for each moving shot individually. Liu *et al.* [32] demonstrate the applicability of such a motion model in the application of automatic Pan and Scan. Jain *et al.* [25] build upon their work and model the camera trajectories as parametric piece-wise spline curves. However, such parametric motion models have limited applicability due to assumptions on the content (a single pan in each shot). Grundmann *et al.* [20] show that 'professional cameraman like' trajectories consist of piece-wise static, linear, and parabolic segments. They show the applicability of such a motion model for video stabilization. Their approach seems motivated by cinematic ideas, generalizes well, and is deployed on large scale systems like Youtube. A similar formulation has been employed in applications of virtual cinematography in panoramic [19] and 360 videos [47]. However,

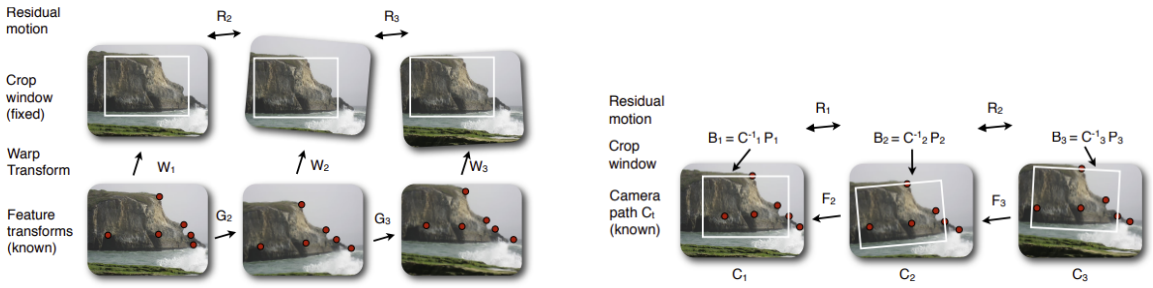

Figure 3.5: The figure illustrates the algorithm [20] for video retargeting, which automatically applies L1-optimal camera paths with controllable constraints. These camera paths generate stabilized videos by eliminating unwanted motions. The computed camera paths consist of constant, linear, and parabolic segments, replicating the camera movements employed by professional cinematographers.

the optimization posed in these works [20, 19, 47] is offline (and non-causal) and in our work, we extend it to online and causal settings.

Carr *et al.* [6] proposed a hybrid camera to aesthetic video generation in the context of basketball games. They combined robotic cameras to coarsely track the game and augmented it with virtual camera simulations to get smoother camera movements. They show that the loss of image resolution can be minimized by using a hybrid system. They investigate a causal moving average filtering and a non-causal l_1 trend filtering [27] to filter the crude trajectories obtained by following the centroid of player detections. They extend their work by learning per frame pan angle predictors based on the player positions in the game and further smooth it using savitzky Golay filter [43]. In [10], they merge the camera position prediction and smoothing into a unified framework. These works [27, 10] rely on a supervised signal generated by a synchronized human camera operator, which is difficult to obtain and also makes them domain-specific. In contrast, the proposed filter in our work is unsupervised.

Our approach to real time camera trajectory stabilization is also related to the work of Liu *et al.* [33], which proposed a framework for online video stabilization of casually captured videos. Their method performs video stabilization by optimizing vertex level motion trajectories. It runs with a single frame latency and runs optimization locally at every frame using a few previous frames. However, such exact optimization only applies to minimal cost functions like mean squared error + L2 norm smoothness (as in [33]). Such minimal functions fail to give the desired cinematic behavior. Moreover, running the optimization on every frame is computationally expensive when a closed-form solution does not exist. We tackle this problem by controlling strides in a sliding window optimization. The proposed *CineConvex* filter: (a) uses a cinematically motivated cost function, and (b) provides more structure due to historical constraints and peek into the future frames. Moreover, beyond a standalone optimization for each window, the proposed *CineCNN* filter can learn priors/structures from previous data as well as temporal dependencies within a sequence.

In summary, Our approach to real time camera trajectory stabilization, limits to the problem of one-dimensional trajectory filtering/stabilization. The work is agnostic to the application and is directly

applicable in variety of frameworks [51, 45, 8, 22, 6, 10, 33, 19, 53]. We compare our approach against trajectory filtering approaches employed in these frameworks and our approach to real time gaze guided video editing system is designed not only for video adaptation but also for the entire video creation process. It can seamlessly integrate with a multi-camera setup, enabling the capturing of wide scenes while ensuring the focus remains on the most important actors and events and edits in real time making it more usable in settings like live stage performances and broadcasting sports event.

Chapter 4

Real Time Unsupervised Filtering for Autonomous Camera Systems

Autonomous camera systems that track a person, object, or action of interest often employ a filtering operation on top of raw per frame estimations (e.g., a virtual camera following an instructor by cropping a window from a static camera around him every frame). Trajectory optimization is also useful in video stabilization methods, where the video is re-rendered with the stabilized trajectory. However, most of these methods [20, 19, 47] are offline and require the entire trajectory during the stabilization process. Such offline filtering/optimization methods cannot be adopted for real-time applications such as live broadcasts (e.g., sports, lectures, live performances). Chen [10] made an attempt to tackle this problem in real-time by learning a regression model using a specific ground truth signal generated by a human operator for a basketball game dataset. The learned model however, is inherently tied to the behavior specific to the particular basketball setting and its ground truth. Such a supervised approach is not applicable to arbitrary trajectory optimization. Since the proposed framework does not depend on any supervision (from a manual operator [10] or synthetic simulations [50]), it can cater to a large variety of applications requiring camera trajectory stabilization. We evaluate our method on the cases of sports and live theatre.

The pan angle of a human cameraman compared to those generated by filtering per frame pan angle predictions using different algorithms over a basketball video sequence.The plots show the filtered outputs corresponding to several baseline algorithms and our proposed approach. The proposed CineFilters mimic professional cameraman-like behaviour and results in piece-wise static, linear and parabolic segments. Other filters fail to give perfect static segments, have sharp corners and sudden direction changes. Although they smooth the trajectories well, they are not appropriate for applications in autonomous camera systems.

Data is being captured and transmitted more rapidly in modern day through devices that have limited storage and computation power. In applications like speech analysis, stork market analysis, live-streaming, biomedical data (eeg, ecg signal), we require filters that can do noise cancellation, trend observation (in case of financial time series analysis, macroeconomics etc) and information enhancement (in case of edge-preserving smoothened images) in an online fashion. In another setting, like

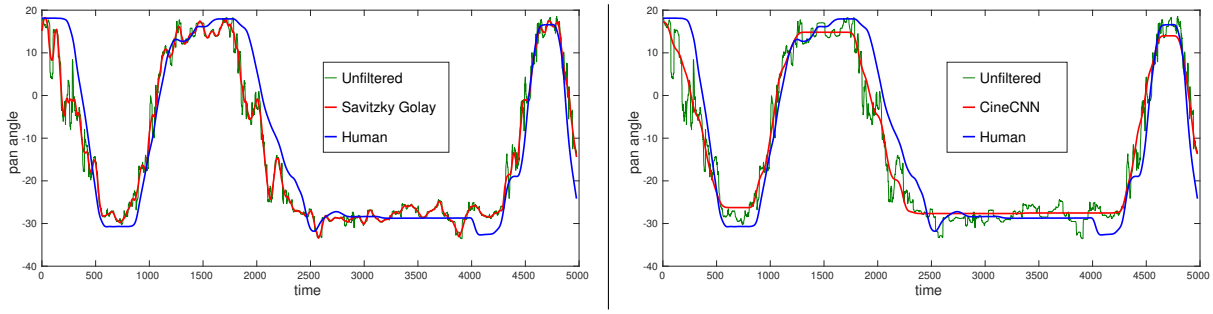

Figure 4.1: SG filter tries to follow the crude pan angle, CineCNN tries to maintain a small difference with pan angle to enforce a cinematographically motivated camera behaviour

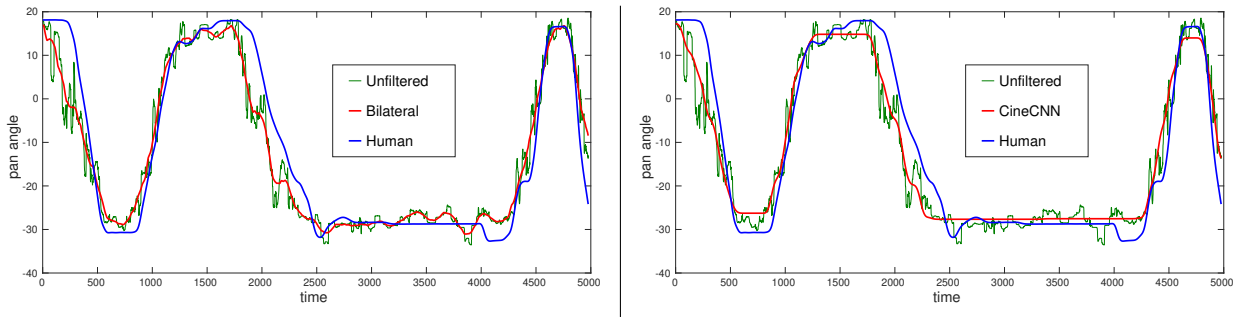

Figure 4.2: Bilateral filter smoothly captures the object of interest but has sudden direction changes

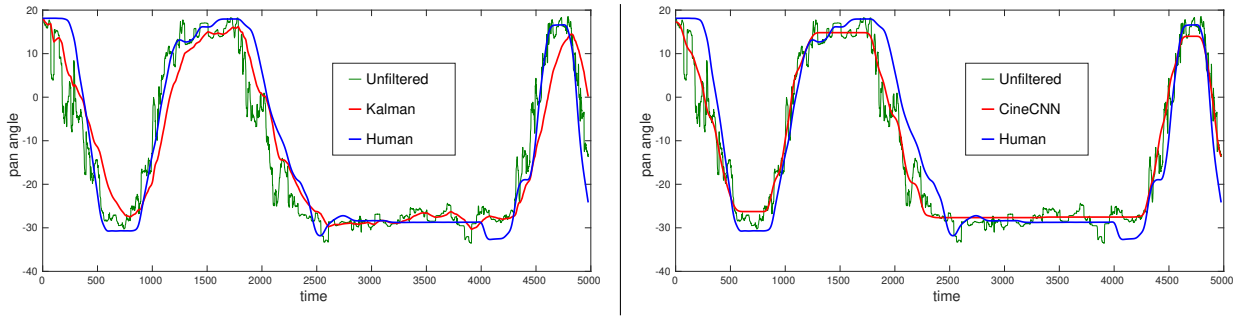

Figure 4.3: Kalman filter also has similar behaviour to that of Bilateral

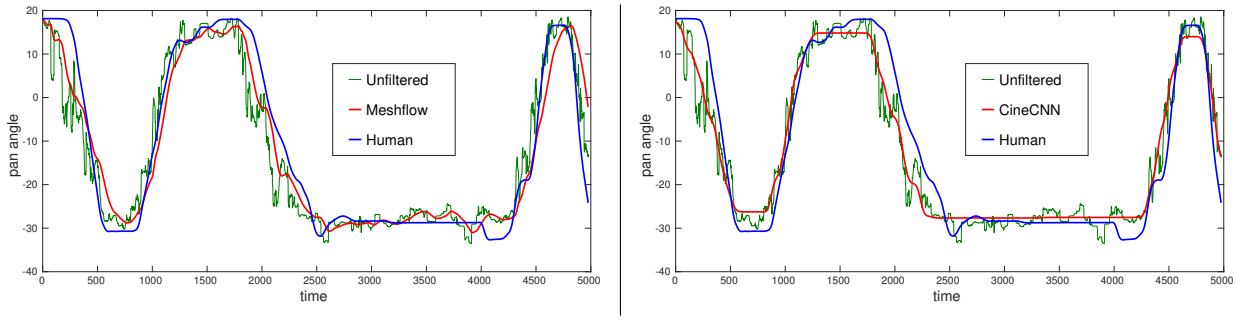

Figure 4.4: Mesh flow lacks on static segments, Where as CineCNN has piece-wise static, linear and parabolic segments

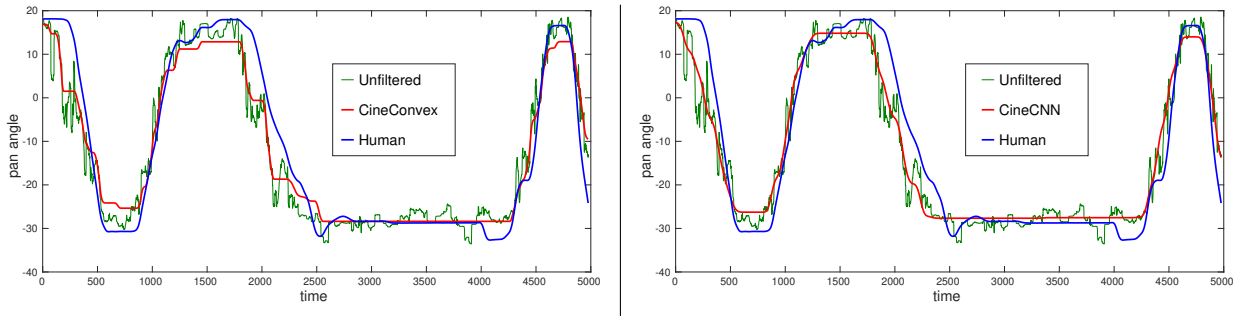

Figure 4.5: CineCNN & CineConvex performs equally better where as CineCNN had a better quantitative metric over CineConvex.

in space missions, where huge amounts of data is collected, we need methods that would throw away irrelevant redundant information while enhancing anomalous information.

In video-stabilization, we need filters that would output signals which are a combination of constant, linear and parabolic segments. It is achieved using an L1-norm based filter. This behaviour is considered optimal because it obeys with cinematic principles. A casually shot video from any hand-held device has high frequency jitter which is mostly removed using stabilizing equipment leaving the low-frequency distortions. A wide range of filters have been proposed for noise cancellation, signal smoothing, flattening etc. Offline filters have been around for a while, where filtering is done as a post-processing step, but there have not been methods that would modify the signal on-the-fly that remove the noise in l1-norm style and store it back on the device

Real-time prediction of smooth camera trajectories is an ill-posed problem since, at any given instance, the predicted trajectory can be smoothed in multiple ways depending on the future input sequences. Thus, using direct optimization on trajectories like previous offline works in a piece-wise manner (sliding window) for the online prediction can make the system very brittle. In this paper, we study the effect of allowing a little peek into the future and applying causal constraints on pre-stabilized trajectories. We show that the proposed sliding window formulation of *CineConvex* filter can closely mimic an offline global optimization with a latency of just half a second. We further investigate parametric models trained with data that can learn the multimodal distribution of a smooth camera trajectory in an online setting. We hypothesize that such predictive models can reduce the dependence on the latency, especially in high noise scenarios. Furthermore, the prediction itself can be done at much faster rates.

To this end, we propose *CineCNN*, a novel CNN based filtering approach that can learn from patterns existing across multiple samples in the dataset. A key issue when training such models is the availability of large labeled datasets. However, our training is unsupervised and uses a set of generic objective functions on top of total variation denoising. Thus, we do not require any ground truth labels for training, and *CineCNN* is independent of the idiosyncrasies of human labels from specific settings. Moreover, our objective functions also encourage cinematographically motivated behaviors like the prediction of true static segments without any residual motions and avoidance of sudden jerks, which is something

missing from the past works. A motivating illustration is shown in Figure 4.5, where we compare our approach with filtering mechanisms prevalent in previous works. In summary, we make the following contributions:

1. We propose *CineConvex* formulation to adapt offline trajectory optimization [20, 19, 47] into an online one. Filtering is posed as a convex optimization in sliding window fashion with minor latency and constraints on the optimized trajectory from the previous window.

2. We propose *CineCNN* for learning a camera trajectory smoothing function. A total variation denoised input is sent to a convolutional neural network to convert into trajectories which mimic cameraman behaviour. The predictive model offers various advantages, especially in terms of computation load and robustness to noise.

3. We propose a novel deep learning based trajectory filtering approach which works in a causal fashion and is suitable for real time applications. Our key novelty is to propose an unsupervised prediction network which has loss functions adaptive to the local structure of the trajectory. We show that our approach is able to mimic professional cameraman behaviour in a casual manner and outperforms the other approaches.

4. Our models can predict smooth trajectories in real-time (1000 FPS for the *CineCNN* and 250 FPS for *CineConvex* on an Intel Xeon CPU). *CineCNN* has a low model complexity (400KB in size), which enables easy deployment in live settings.

5. Both our approaches work in an unsupervised way and do not need any expensive ground truth labels as compared to previous approaches [10]. This makes our model-training independent of the setting in which the label was obtained, thus making it easier to extend to new settings.

6. Our quantitative and qualitative evaluations show that our approach can mimic professional cameraman behavior and outperforms the baselines and the prior art.

In this chapter, we describe our models and their unsupervised optimization procedure. Before going into the problem formulation, we list the desiderata that guide our modeling choices.

1. The smoothing filter should be online and run in real-time.
2. It should not require labeled supervisory data (e.g. ground truth trajectories from human experts).
3. Unmotivated camera movements should be avoided. For instance, when the subject in the frame is stationary, the camera behavior should be static to ensure a pleasant viewing experience.
4. Trajectory smoothing should be robust to outliers in the case of sudden changes in camera trends (should avoid abrupt jerks).
5. It should avoid the accumulation of drift, which is a common problem observed in real-time systems.

4.1 Introduction

We first define our problem as that of predicting a smooth camera trajectory given a stream of incoming noisy trajectory positions. Let $X_t = \{x_0, x_1, ...x_t\}$ be the noisy input sequence that has arrived until frame t and $Y_{t-1} = \{y_0, y_1, ...y_{t-1}\}$ be the smoothed output sequence predicted until the previous frame $t-1$. We wish to learn a nonlinear causal filter specified by a parametric model $y_t = f(X_t, Y_{t-1})$, which predicts the smoothed output for the current frame t. At any timestep, predicting the next smooth frame is ideally a function of both the past and future trajectory directions, which is feasible to model using offline approaches. In the online scenario that we wish to operate in, learning the above mapping is an ill-posed problem as multiple solutions can exist depending on where the future trajectory goes. Therefore it makes sense to accommodate a small number of future frames into the model to constrain the direction of the trajectory.

In this work, we propose two different methods to solve the above problem.

1. The first model is an online filter that performs convex optimization on a combination of objectives. Since the final objective is convex, we can run a per-sample solver to obtain a global minimum at each timestep. However, these methods have a dependency on the solver at deployment time and are computationally intensive if run at each frame in an online fashion. These methods also do not learn a data-based model of trajectory behavior; hence, they might not be fit for cases with high variance in data statistics.

2. The second model we propose is a learning-based solution combining total variation denoising with a 1D convolution neural network (CNN). The CNN gives advantage over convex optimization approach as it can learn trajectory patterns from data. Moreover, at inference, a forward pass through this model is faster and computationally cheaper than a convex solver.

The ideal camera trajectory should be composed of three types of segments, namely static segments, constant velocity segments, and segments with constant acceleration, all transitioning in a smooth manner. As opposed to previous works that use ground truth data from human operators or create large datasets for deep learning based methods, we build our model through an unsupervised multi-objective loss function that enforces such behavior without the need to collect labeled data.

4.2 CineConvex filter

We first describe the convex optimization-based solution for online smoothing. The model works in a sliding window manner. The sliding window configuration is shown in Figure 4.7. It consists of three fragments. The first fragment is called the present window of size p from timestep t to $t + p$ where t is the current time. This fragment gets updated in the final predictions after optimizing the current sliding window. It is also the step-size by which we shift the window after each optimization. The second fragment is called the buffer window, which spans timestep $t - b$ to t. It is the historical trajectory

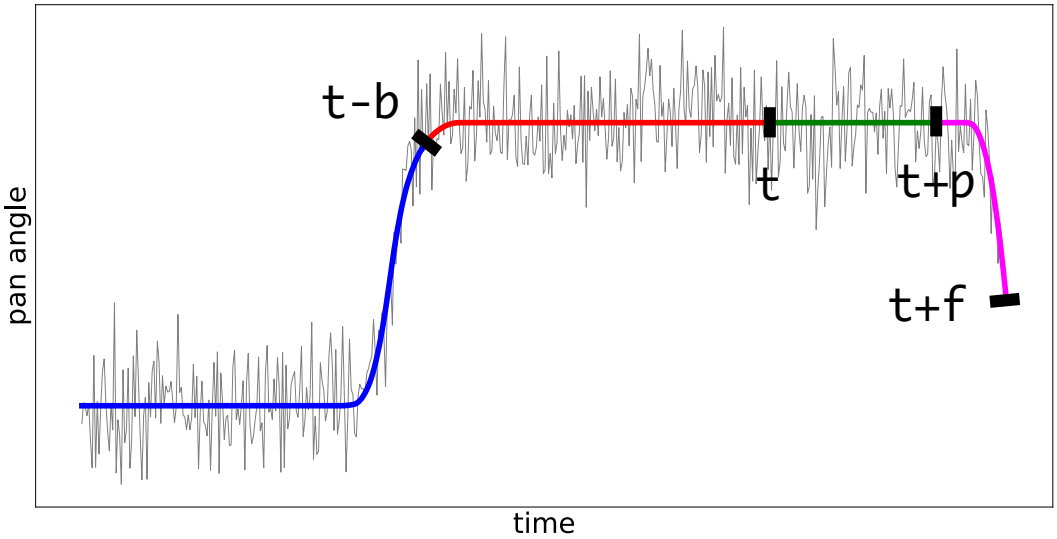

Figure 4.6: The sliding window configuration for our CineFilter models. At timestep t, the model stores b timesteps of the past buffer (in red), has access to f timesteps of the future (in pink) and shifts with a stride of p after each prediction (in green).

information we use in the optimization. The third fragment is called the future window which holds timestep t to $t+f$ with $t+f > t+p > t$ and includes the present window. It provides future context for the optimization. We optimize the trajectories from timestep $t-b$ to $t+f$ and shift all windows by p after each optimization.

More specifically, we enforce the predicted trajectory to be close to the original trajectory in some distance metric. Additionally, we also enforce the velocity and the accelerations of the two trajectories to be close to each other by including the distance between the first and second order derivatives of the trajectory into the objective. This leads to predictions which are close to the original signal and smooth due to the higher order derivatives.

The optimization procedure for each sliding window includes: (a) a term to enforce the predicted trajectory to be close to the original trajectory in some distance metric and (b) L1-norm of the first-order, second-order and third-order derivatives over the optimized trajectory to induce piece-wise static, linear and parabolic behavior. L1-norm has the property to avoid residual motions (e.g., when the path is meant to be static, it leads to truly static outputs) and avoids the superposition between the constant, linear, and parabolic segments. The final objective J^m with respect to timestep t, where m indexes the m^{th} optimization procedure, is given by:

$$J^m(t) = \lambda_0 D_0^m(t) + \lambda_1 D_1^m(t) + \lambda_2 D_2^m(t) + \lambda_3 D_3^m(t) \tag{4.1}$$

where,

$$D_0^m(t) = \sum_{i=t-b}^{t+f} (x^m(i) - y^m(i))^2 \tag{4.2}$$

$$D_1^m(t) = \sum_{i=t-b}^{t+f} \mid y^m(i+1) - y^m(i) \mid \tag{4.3}$$

$$D_2^m(t) = \sum_{i=t-b}^{t+f} \mid y^m(i+2) - 2y^m(i+1) + y^m(i) \mid \tag{4.4}$$

$$D_3^m(t) = \sum_{i=t-b}^{t+f} \mid y^m(i+3) - 3y^m(i+2) + 3y^m(i+1) - y^m(i) \mid \tag{4.5}$$

The λs are hyperparameters found using cross-validation and p, b and f are values we fix heuristically.

Note that this *CineConvex* model has a latency of f timesteps, stores an extra b timesteps from the past, and is run after every p timesteps. Reducing f can reduce the latency, but will lead to less smooth results since multiple future trajectory directions are possible. Decreasing p will lead to frequent trajectory updates but will affect the speed of the filtering operation due to the larger number of optimizations required. Finally, increasing b will provide further historical context; however, that will increase the time required for each individual optimization. Hence the window-related parameters offer a way to balance the speed and accuracy trade-off.

Another problem encountered during this optimization is to maintaining continuity over optimizations on consecutive sliding windows. To mitigate this, we place a hard equality constraint on the past trajectories being currently predicted ($y^m(t-b)$ to $y^m(t)$) and the past trajectories already predicted during the previous optimization ($y^{m-1}(t-b)$ to $y^{m-1}(t)$). Finally, the objective is changed to the following:

$$\begin{aligned} J^m(t) &= \lambda_0 D_0^m(t) + \lambda_1 D_1^m(t) + \lambda_2 D_2^m(t) + \lambda_3 D_3^m(t) \\ &\text{s.t., } y^m(k) = y^{m-1}(k)\ \forall k \in \{t-b, ..., t\} \end{aligned} \tag{4.6}$$

We use the state of the art Gurobi solver [44] for minimizing the objective function (the fastest solver in the MIPLIB 2017 Benchmark [35]).

4.3 CineCNN filter

The *CineConvex* model described above allows us to obtain a global minima for a given fragment, but there are two potential issues with it. Firstly, the optimization is performed on a per-sliding window basis and needs to run frequently during the online operation of the filter. In addition to computational burden and it has a vital dependency on the speed of the solver. Secondly, since there is no data-based learning involved and the optimization described only works at window-level trajectories, the model cannot build a global model for the variation in data statistics that can be encountered by the model. Moreover, since the objective is always an approximation to the behavior we want in the real world, having some inductive biases directly from the data might help in learning better models. This motivates our use of a data-driven model to solve the smoothing problem.

We can pose the problem of estimating the current smooth trajectory position as a sequence modeling problem. However, popular sequence prediction models like HMMs, RNNs, and LSTMs operate

sequentially and have a high memory footprint. Moreover, we found in our experiments that 1D encoder-decoder CNN based architecture better learns the local structure over the more complex recurrent counterparts and also provides improved performance. To promote perfectly static trajectories, wherever possible, we first filter the input signal using a 1D Total Variation (TV) denoising algorithm [13] and then pass it to the CNN. The TV output leads into staircase artefacts whenever the camera movement occurs. The CNN smooths out the staircase artefacts and the results into trajectories that mimic professional camera behavior (having smooth transitions between perfectly static segments). We use an extremely fast direct (non iterative) TV denoising algorithm [13]. In the absence of smooth trajectory labels, we use a similar unsupervised loss function for training the CNN as the convex optimization, without the first-order term. The loss function penalizes the squared distance between the original trajectory and the predictions along with the second, and third-order derivative terms.

Another important observation was that neural network predictions struggle to give perfectly static trajectories. Consequently, we use direct and fast 1D Total Variation (TV) denoising algorithm [13] along with 5-layered encoder-decoder CNN model. The framework takes in the noisy trajectory as input then filters and predicts the smooth output trajectory. Also direct 1D TV denoising algorithm is appropriate for real-time processing of an incoming stream of data, as it locates the jumps one after the other by forward scans. We also include an additional loss term, which penalizes the predictions linearly if they deviate beyond a certain threshold from an original trajectory. This loss is implemented using a shifted ReLU function.

The final loss function for the model is as follows:

$$L = \lambda_0 D_0 + \lambda_2 D_2 + \lambda_3 D_3 \tag{4.7}$$

$$L_{SafetyNet} = \sum_{i=1}^{n} ReLU(\mid y(i) - x(i) \mid -\delta) \tag{4.8}$$

The parameter δ defines the permissible limit for deviation from the original trajectory before this term starts contributing to the final loss function. The first three terms D_1, D_2, D_3 are the same as before, but from timestep 1 to n.

While training, the input sequence from all trajectories are divided into overlapping subsets of $n = 512$ frames. The inference happens on a sliding window of 32 frames, which is possible since the network is fully convolutional. During inference, when only the first frame has arrived, we left-pad the input with repeated values of the first trajectory position and use this as the input. Since there is no optimization at test time, but only a filter-forward pass through the model, we can make a prediction at each time step i.e., with $p = 1$ in figure 4.7. Also, note that there is no explicit constraint that enforces trajectory continuity across predictions like the one needed in the convex optimization formulation. The model has a structural inductive bias (1D convolution) that merges information from local trajectory positions, thus aiding continuity. The data itself, which is fed as a sequence, also provides an additional implicit constraint on the smoothness of the predictions.

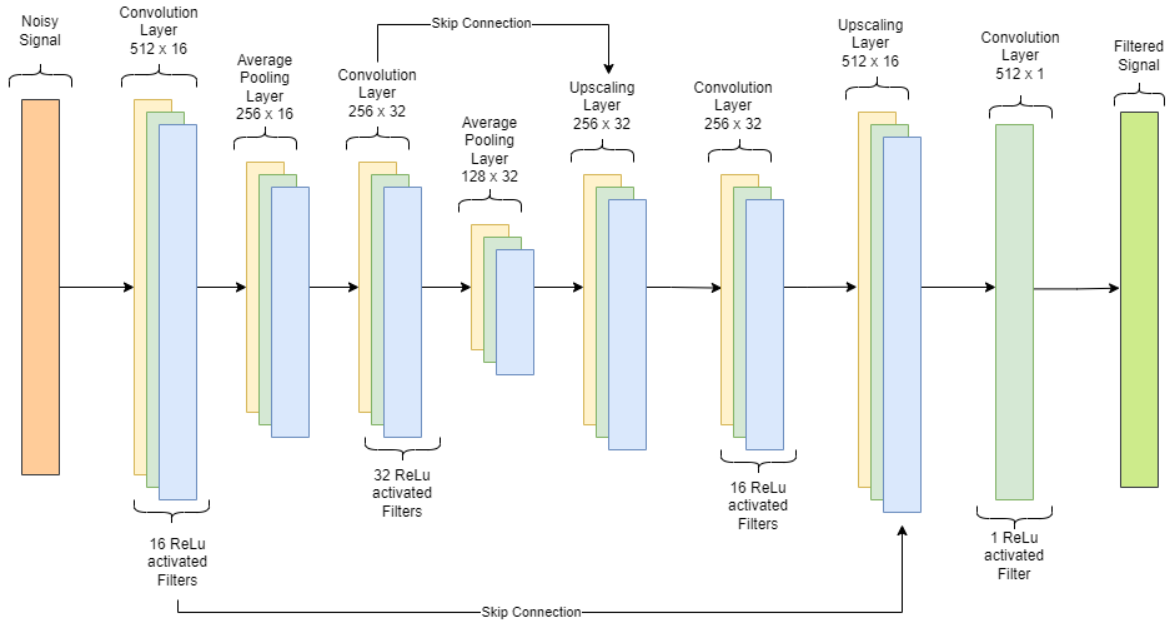

Figure 4.7: CineCNN has CNN based 1D encoder-decoder network with skip connections similar to U-Net architecture

As the new frames keep coming in during inference, we keep appending them to the right of this initial input trajectory and taking the most recent 32 frames as the input for each timestep. The prediction formulation at inference time is the same as that of the convex optimization. Since there is no optimization at test time, but only a filter-forward pass through the model, the time is taken for the predictions is not a bottleneck to the latency, and we can run the system after every timestep.

In this section we explain our loss term which is a combination of penalty terms that arise from different aspects of the camera behaviour. These terms are designed specifically to obey cinematographic principles as mentioned in [48]. In an offline setting, this is formulated as a global optimization function where, given a camera trajectory for the entire duration, an l1 norm based optimization is designed. This comprises a constant term, to mimic a static camera, a linear term to mimic a constant velocity camera. And a third parabolic term to transition smoothly from static to constant velocity camera and from constant velocity to a static camera [29] While the offline global optimization function gives the desired camera behaviour that mimics a professional cameraman, we cannot use a closed form optimization function to achieve the same behaviour in a real-time scenario simply because we do not have access to the future frames. Alternatively, the optimization function can be used as a loss function in training a network.

4.4 Implementation details

For *CineCNN*, we use a 5-layered 1D encoder-decoder based CNN model that has skip connections similar to U-Net architecture with a kernel size of 3 and with 16, 32, 32, 16 and 1 filters respectively. Each CNN layer has a relu based activation on the outputs. The videos from TLP dataset [37] and Basketball dataset [10] (described in Section 4.6) are used to train the model. We sub-sample from the full trajectory of the video at random frames and create a set of input trajectories each of a fixed length of 512 and use these as single instances of training data for the model.

We create $98k$ such instances through sub-sampling. The model is trained for 20 epochs on basketball and TLP datasets, each with a batch size of 16. The adam optimizer is used during training. We follow a learning rate schedule that decays the learning rate by a factor of 0.1 if the validation loss plateaus for more than 4 epochs. The weights associated with each loss term are obtained empirically and $(\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ for stage performance dataset are $(1.0, 1000, 50, 2000)$ and for basketball dataset are $(1.0, 2000, 100, 3000)$. The values for p, f, b are $8, 16, 64$, respectively, for the *CineConvex* model. For the *CineCNN*, we set $p = 1$, $f = 16$ and $b = 16$.

4.5 Experiments

Availability of ground truth is challenging for most applications. The approach being unsupervised, we show that, availability of ground truth does not have any role to play in improving the efficiency of our system. We only use ground truth in the evaluation metric to assess our approach.

In this section, we show the results achieved by our approach on radically different video sequences. These include both indoor and outdoor videos, videos with static and dynamic environment. Unlike other application specific approaches that we have discussed earlier, we show that our filter works well for any given application. The generalizability of our approach makes it easy to incorporate into autonomous cameras systems. We compare our results with five baselines algorithms commonly used for online filtering. We show that our filter gives cinematography driven results. In fact, ours is the first online filter that is carefully designed to satisfy all the constraints that arise in mimicking a human cameraman like camera motion.

We evaluate our approach for the automated broadcast of basketball matches [10] and staged performances [19]. We compute the results of *CineConvex* and *CineCNN* with four algorithms commonly used for online filtering in previous works. We also perform ablation studies to demonstrate the effect of the parameters p, f, b on model performance. We now describe the datasets, baselines, and evaluation strategy in detail.

4.6 Datasets

1. **Basketball dataset:** We use the Basketball dataset proposed by Chen et al. [10]. This dataset consists of a video recording of a high school basketball match taken from two different cameras. One wide-angle camera is installed near the ceiling and looks at the entire basketball area. The feed from the wide-angle camera is used to detect players and compute features summarizing the current state of the scene. The second broadcast camera is placed at the ground level and is manually operated by a human expert. The evaluation task is to predict the pan angle for a robotic camera, given the current state of the match observed by the wide camera. The pan angle of the human-operated camera is considered as the ground truth (calculated by computing its homography with respect to the wide-angle camera). The dataset consists of 50 segments of 40 seconds each (overall 32 minutes of in-play data), out of which 48 segments are used for training, and 2 are used for validation and testing. Similar to [10], we train a Random Forest regressor to obtain per frame pan angle predictions. For learning the Random Forest regressor, the game features computed using the wide-angle camera are used as input and the human operator pan angle as ground truth labels. The per-frame predictions give an extremely noisy output, which are then subjected to a filtering operation. We perform comparisons on the output of CineFilter models and the baselines with the human operator trajectory as the ground truth.

2. **Stage Performance dataset:** We build a Stage Performance dataset that comprises of two wide-angle recordings of staged performances each of 12 and 10 minutes, respectively (a dance and a theatre performance). The videos are selected from the Track Long and Prosper (TLP) Dataset [37]. The original recordings were done using a static wide-angle camera covering the entire action in the scene. The noisy object trajectory sequences for these recordings are obtained using the MD-Net tracker [38]. A complete 12 minute sequence is used for training, and the 10 minute sequence is used for testing. The filters are evaluated on the task of virtual camerawork, following an actor on the stage based on the output of the tracker. Since there is no ground truth available for these videos, we use the offline optimizer from [19] as the ground truth trajectory.

   We evaluate on 3 minute sequences in both recordings which is kept aside for testing and the rest is used for training. We then simulate the virtual cameras (sub-shots) filtered using the baselines approaches and the proposed Cinefilter. We use the recently proposed single object tracking dataset called Track Long and Prosper (TLP) [37] to get the noisy tracked object trajectories as input to train our model. The TLP dataset contains a wide variety of trajectories like roaming animals and flying jets etc. whereas the Stage Performance dataset is recorded in a controlled setting. We perform a quantitative and qualitative evaluation on the Stage Performance dataset in the form of a user study.

3. **Theatre dataset:** We curate a new Stage Performance dataset that comprises of 6 wide angle recordings of staged performances. The original recordings were done using a static wide angle

camera covering the entire action in the scene. We evaluate these sequences on the task of virtual camera simulation [19]. A total of 13 virtual camera sub-shots are generated by moving a cropping window inside the original recordings, by first crudely following the noisy actor tracks. We then simulate the virtual cameras (sub-shots) filtered using the baselines approaches and the proposed Cinefilter. To show generalisation across different settings, none of the videos from Stage Performance dataset is used for training the Cinefilter model. Instead, we use the recently proposed single object tracking dataset called Track Long and Prosper (TLP) [37] to get the noisy tracked object trajectories as input to train our model. The TLP dataset contains a wide variety of trajectories like roaming animals and flying jets etc. whereas the Stage Performance dataset is recorded in a controlled setting. We perform a qualitative evaluation on the Stage Performance dataset in the form of a user study.

We first employ an offline approach as proposed in [19] to simulate the sub-shots, and this is considered as the ground truth trajectories. We perform quantitative evaluation of the generated sub-shots from the proposed algorithm and the baselines, considering offline optimization as ground truth. We further perform a user study to compare the output from different approaches.

4.7 Baselines

Most of the baseline online filters that we have compared our model with show a basic smoothing of the original signal but it is far from the behaviour we desire, refer to figure 4.5. These filters cannot be used in synthesize videos online that are aesthetically pleasing. They would need some kind of hard-coding instructions based on the underlying application

1. **Savitzky Golay:** SG filter performs data smoothing using least squares to fit a polynomial of a chosen degree within a window of consecutive data points around every point. It takes the central value in the window as the new smoothed data point and this step is performed at each point on the time series. SG filter is non-causal and in our case we choose a window size of 51, giving a latency of 25 frames. The degree of the polynomial is set to 3.

2. **Kalman Filter:** Kalman Filter is a recursive Bayesian estimation method that can be used for updating the current smooth camera state at every time step using previous predictions and the current observation. We set the parameters of the filter similar to [10].

3. **Bilateral Filter:** Bilateral filter is a non linear, adaptive, edge preserving and noise reducing smoothing filter. It uses a weighted-average approach where the weights at any given data point is computed using two Gaussians, one Gaussian on the difference of spatial distance and the other on the difference between the magnitudes of the given point with every other point in the surrounding window. It is also non-causal and we use a window size of 64, giving the latency of 32 frames.

4. **MeshFlow:** MeshFlow [33] runs with a single frame latency and runs optimization locally at every frame using a few previous frames. The optimization functions consists of two different terms. The first term is the MSE over the predicted and original values of the 1D signal. The second term is a MSE over the first order differential of the predicted signal. Window size of 20 values is considered for the experiments. It solves the problem of video stabilization by dividing the image into uniform patches and representing each patch with a vertex, known as mesh vertex. Motion at every mesh vertex is computed by matching images features. It solves a convex optimization problem to smoothen out the temporal changes at each mesh vertex to obtain an optimal single motion between every pair of consecutive frames. We use our CineFilter model instead of the convex optimization to obtain a smooth camera motion between every consecutive pair of images.

4.8 Evaluation Metric

We perform quantitative experiments on the Basketball dataset and qualitative experiments on the Stage Performance dataset. For quantitative experiments on the Basketball dataset, we use the data from the human operator as ground truth and compute two different metrics, one measuring the closeness to the original signal and other measuring the movement profile. The first metric (precision metric) is the mean squared error between the filtered trajectory predictions and the ground truth trajectories (pan angle corresponding to the expert human operator). Assuming that the human operator selects the pan angle that best showcases the activity happening in the game, this term penalizes trajectories that deviate from the angles at which important activities are happening. The second metric (smoothness metric) is the absolute difference in the slopes between the predicted and the ground truth trajectory. It measures the ability to mimic human cameraman-like behavior. It penalizes any movement of the predicted trajectory when the human operator is static and also penalizes the predicted trajectory if it moves in a different direction or at a different speed than the ground truth. We define the two terms as Precision Loss and Smoothness Loss:

1. The first measures the effectiveness to convey the important actions in the game and the second term measures the smoothness of the camera trajectories.Assuming that a skilled human operator selects the pan angle which best showcase the activity happening in the game, we approximate the first term as the mean squared error of the predicted trajectory with the pan angle of the human operator.

2. As mentioned earlier, a pleasant viewing experience is conveyed when there is a steady camera motion. Hence, along with the magnitude of camera position, we also introduce the velocity of the camera in our evaluation metric. This not only measures where the camera is looking which is given by the Mean Squared Error (MSE) term, but also where the camera should be moving in the next time step. This is to say, the absolute slope difference term evaluates if our camera

system moves in the same direction as the ground truth camera with a similar speed. Our final metric for evaluation is sum of precision and smoothness loss terms. precision term captures how well our model is close to input signal, where as smoothness term measures how well it preserves smoothness.

$$precision = \sum_{t} (y(t) - \hat{y(t)})^2 \tag{4.9}$$

$$smoothness = \sum_{t} \mid \frac{dy(t)}{dt} - \frac{d\hat{y(t)}}{dt} \mid \tag{4.10}$$

$y(t)$ is the prediction, $\hat{y(t)}$ is the ground truth from the human operator, lower the precision & smoothness loss, the better the predictions.

Although quantitative metrics can give a reasonable estimate about the effectiveness of the predictions, the final gold standard is the human perception of the rendered videos using the filtered trajectories. For instance, an aesthetically pleasing viewing experience is more important even if it comes at the cost of increased precision loss. Hence, in addition to quantitative metrics, we also evaluate our model qualitatively in the form of a comprehensive user study. The study is done on the output shots obtained from the task of virtual camera simulation on 14 small sequences from different videos of stage performances.

4.9 Results

4.9.1 Quantitative Evaluation

We compare *CineConvex* and *CineCNN* against the baselines on the precision loss (Equation 4.9) and smoothness loss (Equation 4.10) metrics. The results are summarized in Figure 4.11.

For the Basketball dataset, which has the noisier motion of the two datasets, we see that the proposed models are competent on the precision metric to other approaches; however, they bring significant improvements over the smoothness metric. *CineCNN* gives more than three times improvement over the other baselines. We also observe that *CineCNN* gives better performance in high noise situations over the *CineConvex* filter.

For the Stage Performance dataset, which has relatively noise-free trajectories, *CineConvex* and *CineCNN* give similar performances. The proposed methods notably outperform the baselines over both precision and smoothness metric. The use of a ground truth generated using similar loss terms (offline) might give an added advantage to the proposed models. However, offline optimization [20, 19] is shown to extremely effective and considered to closely mimic human cameraman behaviour. The efficacy of the proposed models (over baselines) is further affirmed by the user study presented in the following section.

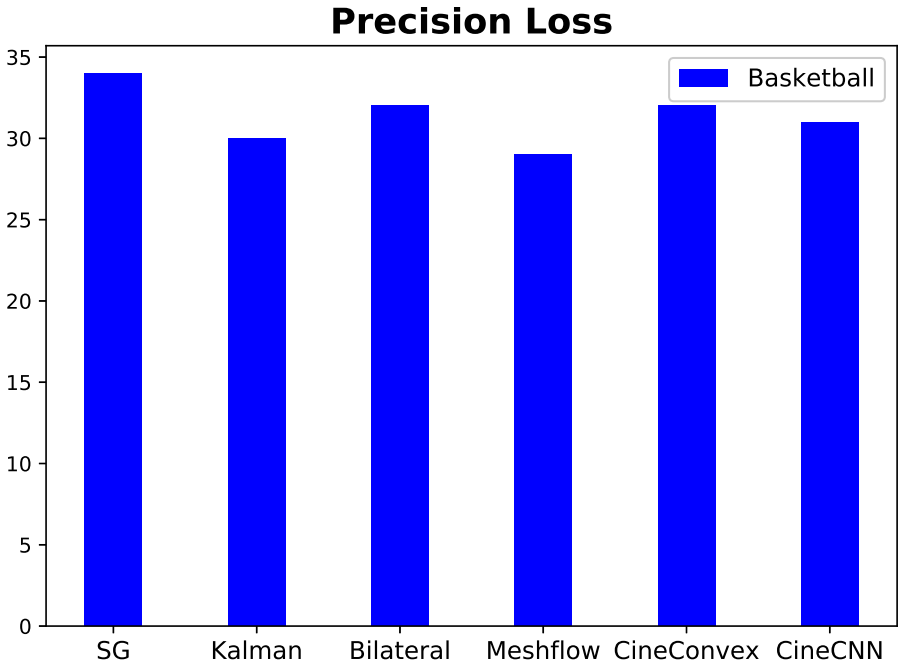

Figure 4.8: Precision loss for our approach and the baselines on the Basketball dataset.

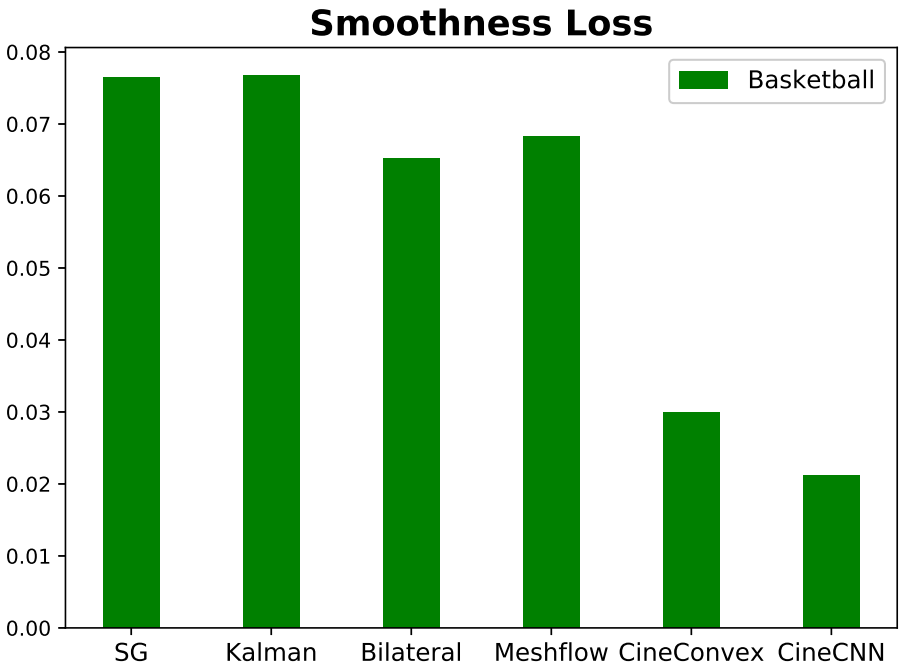

Figure 4.9: Smoothness loss for our approach and the baselines on the Basketball dataset.

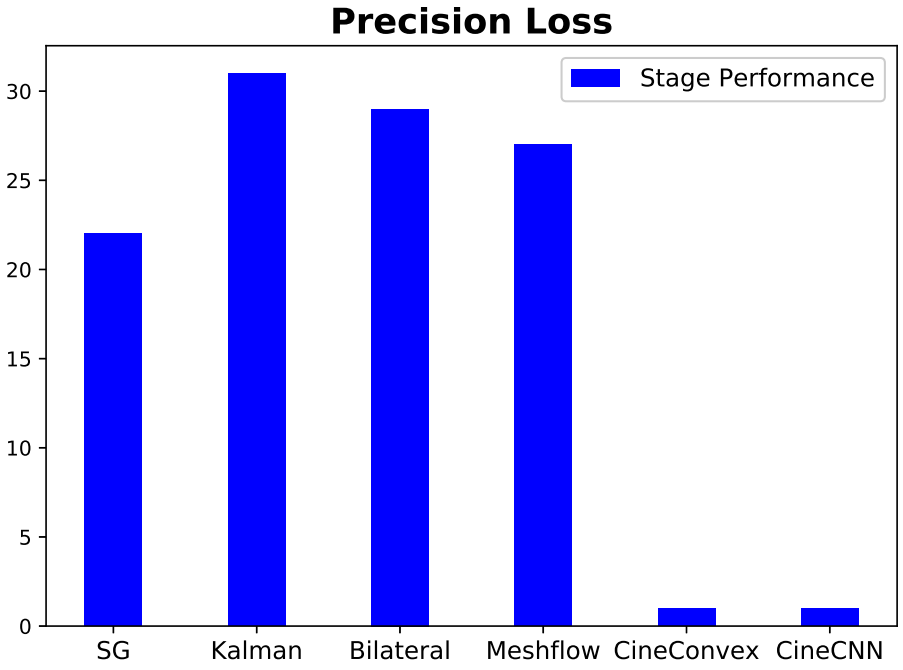

Figure 4.10: Precision for our approach and the baselines on the Stage Performance dataset.

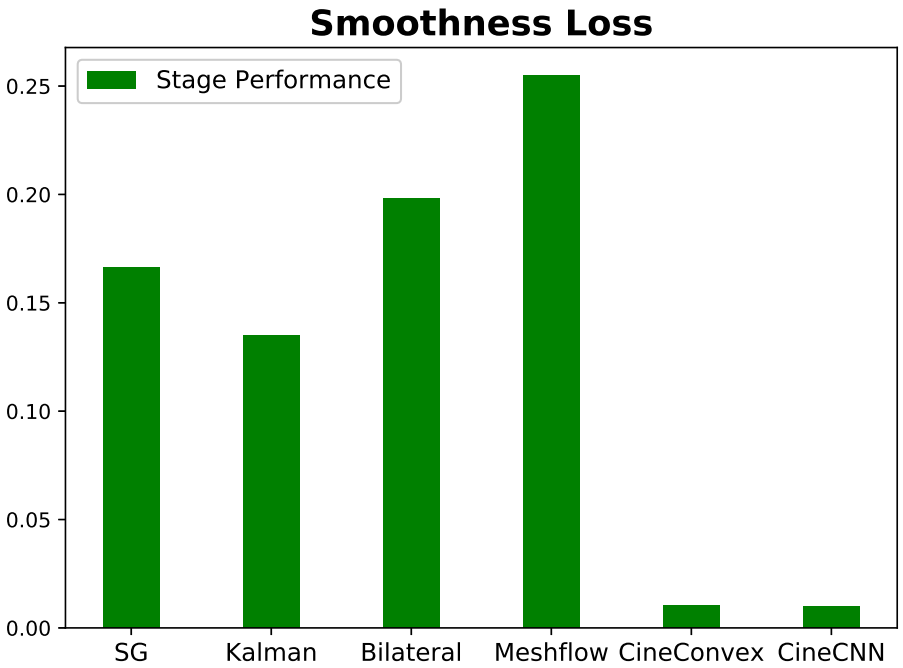

Figure 4.11: Smoothness loss for our approach and the baselines on the Stage Performance dataset.

CNN	Win	Loss	No Preference
vs Kalman	13	0	1
vs SG	14	0	1
vs Meshflow	11	1	2
vs Bilateral	11	1	2
vs Gurobi	3	0	11

Table 4.1: User study results of baselines approaches vs *CineCNN* filter.

However to note is the difference in the time efficiency of these two filters and the latency introduced due to the hyperparameters which we analyze in the ablations in Section 4.9.3.

4.9.2 User Study and Visual Inspection

Although the precision & smoothness loss is a reasonable way to assess our model, it may not be able to measure the adherence to cinematic principles accurately and, more importantly, the final aesthetics of the rendered video. For instance, a smoother trajectory may be preferred by the user even if it is slightly drifted and increases the precision loss. Similarly, unmotivated movement can appear distracting to the user, even if they are extremely minute and may not significantly contribute to the loss. To this end, we complement our evaluation using qualitative methods to account for the perceptual metrics i.e., how the proposed filtering method performs against the baselines in terms of aesthetics of the rendered video; we perform a study with 14 users.

We select 14 small video clips from a diverse set of wide-angle stage recordings, which are different from the two sequences in Stage Performance dataset (used in training). The average duration of the clips is 25 seconds. We evaluate each of the proposed and baseline filters for the virtual camera simulation task [19]. The filters are applied over the per frame shot estimations obtained from noisy actor tracks.

In each trial, the participants are shown two videos in a side by side manner, one rendered using *CineCNN* and the other using *CineConvex* or one of the baselines. They are instructed to choose the video that is more aesthetically appealing and better mimics human camerawork. They are also given an option to choose neither (if they do not have a clear preference, and both videos are reasonably similar in terms of aesthetics). Each user watches nine pairs of videos; therefore, each pair of videos is watched exactly once. The left and right ordering for videos is randomly switched. The results are illustrated in Table 4.1.

The proposed methods are significantly preferred compared to the other baselines. Bilateral is the most competent approach among the baselines, and it works well in sequences where actors are continually moving. Kalman filter has minimal drift and is preferred in a few cases, as it maintains the shot compositions well. We present some of these rendered comparison videos in the supplementary material. *CineCNN* is consistently chosen over *CineConvex*, which correlates with the precision &

Gurobi	Win	Loss	No Preference
vs Kalman	12	1	1
vs SG	14	0	1
vs Meshflow	11	1	2
vs Bilateral	10	1	3
vs CNN	0	3	11

Table 4.2: User study results of baselines approaches vs our *CineConvex* filter.

p \ f	4	8	16	32
4	(51.7, 0.05)	(37.2, 0.05)	(31.7, 0.04)	(32.0, 0.03)
8	-	(35.9, 0.04)	(31.1, 0.04)	(31.6, 0.03)
16	-	-	(31.7, 0.04)	(30.1, 0.04)
32	-	-	-	(31.1, 0.04)

Table 4.3: Ablation study of *CineConvex*, comparing model performance (Precision loss, Smoothness loss), across various present window size (p) and future window size (f) combinations.

smoothness loss shown in Figure 4.11. On the other hand, since the Basketball dataset does not have publicly available video sequences, we point the readers to the predicted trajectory comparison for all the baselines and between *CineConvex* and *CineCNN* models in Figure 4.5. The proposed *CineCNN* filter outperforms other methods on the Basketball dataset (in terms of smoothness, lack of sharp jerks, and lack of residual motion), as also indicated by the precision & smoothness Score in Figure 4.11.

4.9.3 Ablative Experiments

In this section, we discuss results for ablation experiments across different values for the window hyperparameters p and f and show performance and speed varies across the two proposed model formulations. Table 4.4 shows the influence of the present window size (p) and future window size (f) on the (precision loss, smoothness loss) and speed of *CineConvex* filter. The tables show how increasing the present window width can improve the speed of the filtering operation, but also needs an increase in the future frames, thus increasing latency. Also, there exists a middle ground for the p and f values, which balances the performance with speed. We can contrast this with *CineCNN*, which has a constant speed of around 1000 FPS with $p = 1$. For *CineCNN*, the values of $f = (4, 8, 16, 32)$ gets respective precision & smoothness loss of $((31.1, 0.05), (31.9, 0.03), (31.9, 0.02), (31.6, 0.02))$. Like the *CineConvex*, the *CineCNN* has only very slight improvement for $f = 32$ over $f = 16$, so we use $f = 16$ in our experiments. The ablation experiments again show the various speed-accuracy-latency trade-offs associated with both models across the choice of hyperparameters.

p \ f	4	8	16	32
4	147	135	125	98
8	-	250	227	172
16	-	-	417	333
32	-	-	-	714

Table 4.4: Ablation study of *CineConvex*, comparing model speed (frames per second), across various present window size (p) and future window size (f) combinations.

Baselines	Precision	Smoothness
Kalman	28.1	0.03
SG	35.3	0.04
Meshflow	28.2	0.03
Bilateral	35.2	0.04
CineConvex	31.6	0.02

Table 4.5: Total Variation pre-processed evaluation of *CineConvex* vs baseline approaches on the Stage Performance dataset.

4.9.4 Pre-processed Evaluation

The proposed *CineCNN* model performs a TV denoising prior to sending the data through the CNN. For a fair comparison, we pre-process the tracks with TV denoising on other baselines as well and compare the outputs. The results are presented in Table 4.6. We observe that TV denoising brings minor improvements in precision and smoothness metrics for the baselines, however, significantly behind the performance of CineCNN. The baselines filters also fail to tackle the staircase artefacts and lead to jerks.

Baselines	Precision	Smoothness
Kalman	29.1	0.144
SG	19.1	0.155
Meshflow	27.0	0.158
Bilateral	37.0	0.138
CineCNN	0.94	0.009

Table 4.6: Total Variation pre-processed evaluation of *CineCNN* vs baseline approaches on the Stage Performance dataset.

Baselines	Residual Error
Kalman	0.0609
SG	0.1509
Meshflow	0.0798
Bilateral	0.0592
CineConvex	0.0005
CineCNN	0.0001

Table 4.7: Residual motion loss of baseline approaches vs *CineFilter* models on Stage Performance Dataset.

4.10 Residual Motion

According to professional cinematographic practices [48], a steady camera behaviour is necessary for pleasant viewing experience. A camera movement without enough motivation may appear irritating to the viewer, hence the camera should remain static in case of small and unmotivated movements. The ideal camera trajectory should be composed of three types of segments, namely static segments, constant velocity segments, and segments with constant acceleration, all transitioning in a smooth manner. Small residual motions (even minuscule) are displeasing. Since baselines filters are not cinematically motivated, they exhibit residual motion.

A residual motion can be clearly stated as an unmotivated camera movement. We quantatively show this by computing the smoothness metrics only on parts where the ground truth is perfectly static for at least 4 seconds (i.e 128 frames) and compute smoothness loss of the output produced by *CineFilter* models and other baselines on these corresponding segments. The results presented in Table 4.7 clearly demonstrate the efficacy of *CineConvex* and *CineCNN* in terms of providing perfectly static camera trajectories. (Also, To find perfectly static segments we compute first-order derivative on offline optimization (ground truth) and choose all the segments of length 128 frames or longer with first-order derivative less than $1e-8$.)

4.11 Summary

We present two innovative unsupervised methods for real-time filtering of noisy trajectories. The first method, called *CineConvex*, tackles the filtering problem by formulating it as a convex optimization task within individual sliding windows. We solve this optimization problem using iterative convex solvers. The second method, known as *CineCNN*, approaches filtering as a prediction task using a convolutional neural network. These unsupervised methods offer several advantages, including their versatility across various applications, high frame rates, and low memory usage. To evaluate the performance of our proposed methods, we compare them against commonly used online filters in two different scenarios: basketball games (which involve fast-paced movements and multiple players) and theatre plays (which

focus on individual actors on a stage). Through extensive quantitative and qualitative experiments, we demonstrate that our methods outperform existing filtering techniques, making them a more favorable choice for trajectory filtering in online automated camera systems. These findings highlight the potential of our methods to significantly enhance the quality and accuracy of real-time trajectory filtering.

Chapter 5

Real Time Gaze-guided Cinematic Editing

The demand for immediate and dynamic content creation arises from real-time video editing. This capability is crucial in various industries, such as live broadcasting, sports coverage, news reporting, and social media content creation. It eliminates the need for time-consuming post-production processes and enables faster delivery of high-quality videos in today's fast-paced digital landscape. We present Real Time GAZED, a real-time version of the GAZED - gaze guided video editing framework [36]. We also show our results with Real Time GAZED and compare them with other baselines, including GAZED, to show with real-time editing, our results are close to non real-time method. To further strengthen our claim, we conducted a user study to conclude the video edits were aesthetically pleasing.

5.1 Introduction

Real Time GAZED refers to an updated version of the gaze-guided video editing process that operates in real-time. This pipeline allows for the creation of an edited video using only a single static wide-angle camera feed that captures the entire stage. The concept of Real Time GAZED is influenced by previous work such as GAZED and other related studies that explore the idea of replacing a multiple camera crew setup with a single high-resolution static camera. By simulating multiple virtual pan/tilt/zoom cameras, the system focuses on actors and actions within the original recording, enabling the generation of multiple virtual camera shots. The human gaze plays a significant role in guiding video editing decisions, as our eyes naturally focus on important aspects of a scene that should be highlighted in the edited video. Building upon the previous research, our approach utilizes user eye gaze data and automatically generate multiple video clips by treating shot selection as a real-time discrete optimization problem, we make use of a minimal lookahead into the future to determine the most relevant moments to capture.

The primary goal of video editing is to determine the most appropriate shots to include in each frame of the edited video. To accomplish this, the shot selection process is formulated as an optimization task, where various factors, including gaze information and cinematic editing principles, are taken into account. The use of gaze information helps in identifying important areas within the scene, which are then

assigned gaze potentials that quantify the significance of the available shots. These gaze potentials are combined with other factors that adhere to cinematic principles, such as avoiding abrupt cuts, maintaining rhythm, avoiding transient shots, and ensuring real-time continuity by considering the previously selected shots during optimization. Dynamic programming is employed to solve the optimization problem efficiently. To evaluate the real-time performance of the GAZED system, a user study involving 8 participants is conducted. Multiple edited versions of stage performance recordings are edited using Real Time GAZED and compared against several baseline methods, including wide-shot framing, speaker detection-based editing, greedy gaze-based editing and GAZED

The results of the study demonstrate that Real Time GAZED outperforms the baseline methods in terms of editing quality and performs equally good compared to GAZED. Thus concluding even with real time version the incorporation of gaze information, along with other cinematic principles, leads to more effective and engaging video edits.

Our contributions can be summarized as follows:

1. We have created an end-to-end cinematic editing pipeline that operates in real time. This pipeline allows for the generation of professional-quality videos from a static camera recording. This approach involves selecting shots based on an objective function that incorporates gaze potentials and adheres to cinematic principles and shot continuity constraints. This system empowers even novice users to create polished and well-edited videos by utilizing their eye gaze data and an affordable desktop eye tracker.

2. We have conducted a comprehensive user study to validate the effectiveness of the method compared to various editing baselines. The results clearly demonstrate that users prefer the outputs generated in terms of several attributes that characterize the quality of the editing.

5.2 Method

In Real Time GAZED method we used GAZED video editing pipeline as a backbone along with its shot generation, gaze potential components. We can safely use these standalone components even in real time settings as these rely on contextual information at any given time instant and rather not use information across time. We build upon existing shot selection component with an additional terms to support shot continuity cost constraint. Below we briefly explain the functionality of these components.

5.2.1 Shot Generation

We use the shot generation component as-is from the GAZED video editing pipeline. Also, the aspect of the shot generation component is its ability to operate in real time without relying on temporal video information. Instead, it focuses solely on the actors present in each frame at a specific moment. This means that the system can swiftly generate shots without considering the context across multiple

Figure 5.1: The bounding boxes in the middle row showcases the shots that have been selected using Real Time GAZED (*highlighted in green*) and GAZED (*highlighted in blue*). These shots represent the frames chosen by the respective algorithms for video editing. Moving to the top and bottom rows, we can observe the actual cropped shots resulting from the selections made by Real Time GAZED and GAZED, respectively. These cropped shots provide a closer look at the specific segments of the video that have been chosen for further processing. The visual comparison between the two algorithms gives us valuable insights into their performance and the differences in their selected shots.

frames. It's like capturing the essence of the scene in an instant, allowing for dynamic and on-the-fly shot selection. The result is a fast and efficient process. This component takes a wide-angle recording captured by a static camera, which provides a comprehensive view of the entire scene. Each frame in this input video is considered a master shot. To automatically generate more engaging shots, we employ a virtual camera simulation technique [19] based on a method called multi-virtual pan-tilt-zoom (PTZ) cameras. Using the information from bounding boxes of performers/actors by leveraging [5] in each master shot, we move multiple cropping windows within the frame to simulate various virtual PTZ cameras. These cameras focus on specific actors or groups of actors, creating zoomed-in shots that add depth and intimacy to the original wide-angle recording. This shot generation process involves convex optimization, considering composition, panning, and cutting techniques used in professional cinematography. It transforms rough shot estimates into well-composed cinematic shots reminiscent of those captured by skilled cameramen. When processing an input video, we generate a comprehensive set of possible shots for every combination of performers in the scene. For a video with n performers, we create $n * (n + 1)/2$ combinations of shots. We generate n number of 1-shots for sequences with N actors, followed by $N - 1$ number of 2-shots, $N - 2$ number of 3-shot type, and so on, capturing different arrangements of performers.

To ensure the generated shots are visually appealing, we utilize a Medium Shot (MS) for single actor shots (1-shots). A medium shot frames the performer from head to waist, while a medium closeup fo-

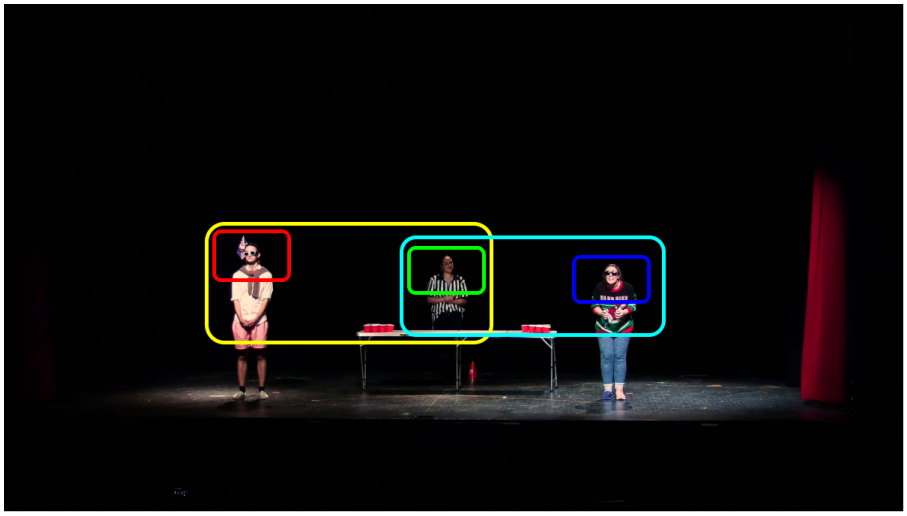

Figure 5.2: The figure illustrates the various bounding boxes generated within a frame. These bounding boxes serve as visual indicators of the different perspectives and compositions that can be captured in a single frame. These generated shots are used in Real Time GAZED algorithm.

cuses from head to mid-chest, offering an intimate perspective. For sequences involving multiple actors, we employ a Full Shot (FS) that captures each performer from head to toe, providing a comprehensive view of the group's dynamics. By implementing these techniques, we enhance the visual impact of the edited videos and create a more immersive viewing experience for the audience. We denote set of shots (S) generated from a frame (master-shot)

$$S = \{s_i\}_{i=1}^{n*(n+1)/2} \tag{5.1}$$

5.2.2 Shot Selection

In our video editing pipeline, the next crucial step after generating shots is selecting the most compelling shot that effectively tells the story at each moment. However, we couldn't directly utilize the shot selection component from the GAZED pipeline as it heavily relies on the complete temporal information of the video. To overcome this limitation, we made modifications to the shot selection component by considering only a small time frame, typically ranging from half to one second, instead of the entire video. Additionally, we introduced an extra penalty term called shot continuity to ensure a smooth transition between shots.

The process of shot selection is treated as a discrete optimization problem, where we assess the importance of each of the multiple shots generated for every video frame. During this assessment, we adhere to fundamental cinematic principles such as avoiding abrupt cuts between overlapping shots (known as jump cuts), preventing rapid shot transitions, and maintaining a cohesive cutting rhythm. To determine the importance of each shot at a given moment, we rely on eye gaze data collected using an eye-tracking device. Moreover, we incorporate cinematic principles into the optimization process

through penalty terms that guide the shot selection. By making these modifications and incorporating eye gaze data and cinematic principles, we enhance the overall editing process, ensuring that the selected shots effectively convey the story and captivate the audience.

For a scene with n actors, the editing graph consists of $n * (n + 1)/2$ nodes at each frame t, where each node represents a shot and edges across time steps represent a transition from one shot to another (denoting a cut) or to itself (no cut). Formally, given a sequence of frames $t = [1..T]$ the set of generated shots $S^t = \{s_i^t\}_{i=1}^{n*(n+1)/2}$ and the raw gaze data g_k^t corresponding to user k at time t, our algorithm selects a sequence of shots $\epsilon = \{s^t\}$ where $s^t \in S^t$, by minimising the following objective function:

$$E(\epsilon) = \sum_{t=1}^{T} -ln(G(s^t)) + \sum_{t=2}^{T} E_e(s^{t-1}, s^t) \tag{5.2}$$

where $E_e(s^{t-1}, s^t)$ denotes cost for switching from one shot to another and $G(s^t)$ is a unary cost that represents the gaze potential (modeling importance) for each shot.

5.2.2.1 Gaze Potential

In our process of selecting the best shots for editing, each generated shot is assigned a score that helps the optimization algorithm find the most optimal path through the editing graph. One crucial component in this ranking is the Gaze Potential, which effectively captures the most important scene events at any given moment. When editing a video, it is essential to ensure that the final result captivatingly conveys the original narrative of each scene. Unlike previous methods [30], [18] that rely on additional metadata or computational features to estimate actions or emotions in a shot, which often overlook high-level scene semantics that humans are sensitive to, we utilize gaze data recorded from users. This approach has proven to be effective in accurately localizing focal scene events.

We choose to use the Gaze Potential component from GAZED as it has several advantages. It is fast, does not depend on temporal context within the video, and can be computed in real-time. For 1-shots, we calculate the gaze potential using sum of the distance between each user k gaze point g_k^t and the center of a shot s_i^t is c_i^t at frame t, where $d_i^t = \sum_k (c_i^t - g_k^t)^2$ and accumulate d_i^t as an inverted sum so that, The function returns a higher potential for shots with focused gaze clusters, and lower potential for shots with dispersed gaze points.

$$G(s_i^t) = \frac{\frac{1}{d_i^t}}{\sum_i \frac{1}{d_i^t}} \tag{5.3}$$

For higher-order shots like 2-shots, 3-shots, and so on, we adopt a bottom-up approach where we compute the gaze potential of higher-order shots based on lower-order shots. Let's consider the example of two 1-shots. If the gaze is evenly distributed between the two constituent 1-shots, the resulting 2-shot will have a high gaze potential. This implies that the combined shot of the two actors holds more value. Conversely, if the gaze is focused on only one of the 1-shots, the gaze potential for the 2-shot will be

lower, indicating that the combined shot is less valuable. This hierarchical approach can also be applied to compute the gaze potential for higher-order shots.

$$G(s_{ab}^t) = G(s_a^t) + G(s_b^t) - \|G(s_a^t) - G(s_b^t)\| \tag{5.4}$$

A similar hierarchy can be followed for computing the gaze potential for higher-order shots. For instance, gaze potentials of two 2-shots $G(s_{ab}^t)$ and $G(s_{bc}^t)$ can be used to compute gaze potential of a 3-shot $G(s_{abc}^t)$, when the actors appear on screen in the order a, b, c on moving left to right. By incorporating gaze potential in our shot selection process, we ensure that the most important elements and interactions within a scene are highlighted, resulting in a more engaging and valuable final video.

5.2.2.2 Editing cost

We have enhanced the shot selection component to adapt to real-time settings, unlike the implementation in GAZED, which focused solely on offline processing. This process involves computing a cost matrix using gaze potential and incorporating penalty terms inspired by cinematic principles to avoid jump cuts, abrupt transitions, and other undesirable effects. We maintain the same penalty terms used in GAZED to construct the cost matrix. Moreover, the computation of the cost matrix is fast and can be performed in real time.

To achieve shot selection in real time, we leverage the ongoing construction of the cost matrix. At any given moment t, while the cost matrix is being built for a future time $t + f$ (with f serving as a look-ahead duration in the cost matrix), we process the cost matrix information between time t and $t + f$ to make shot selection decisions at time t. Now, let's delve into the details of the cinematically motivated penalty terms and the process of utilizing the cost matrix information from t to $t + f$ for shot selection.

We introduce three types of penalty terms - shot transition cost (T), shot overlap cost (O), and cutting rhythm cost (R). These penalty terms contribute to the total cost associated with transitioning from one shot, denoted as s^{t-1}, to another shot, denoted as s^t. Both s^{t-1} and s^t belong to the set of available shots, denoted as S. The cumulative sum of these penalty costs determines the overall cost of transitioning between shots. By incorporating these penalty terms and dynamically processing the cost matrix information, we ensure that shot selections are made in real time while maintaining cinematic principles. This approach creates an engaging video editing experience for the viewer.

$$E_e(s^{t-1}, s^t) = T(s^{t-1}, s^t) + O(s^{t-1}, s^t, \gamma) + R(s^{t-1}, s^t, \tau) \tag{5.5}$$

- *Shot transition cost* - To ensure that the viewer has sufficient time to understand and appreciate the scene, it is important to minimize frequent shot transitions and avoid abrupt cuts that can disrupt the viewing experience. In order to address this issue, we introduce the concept of transition cost. This cost quantifies the undesirability of transitioning from one shot, denoted as s_i^t at time t, to

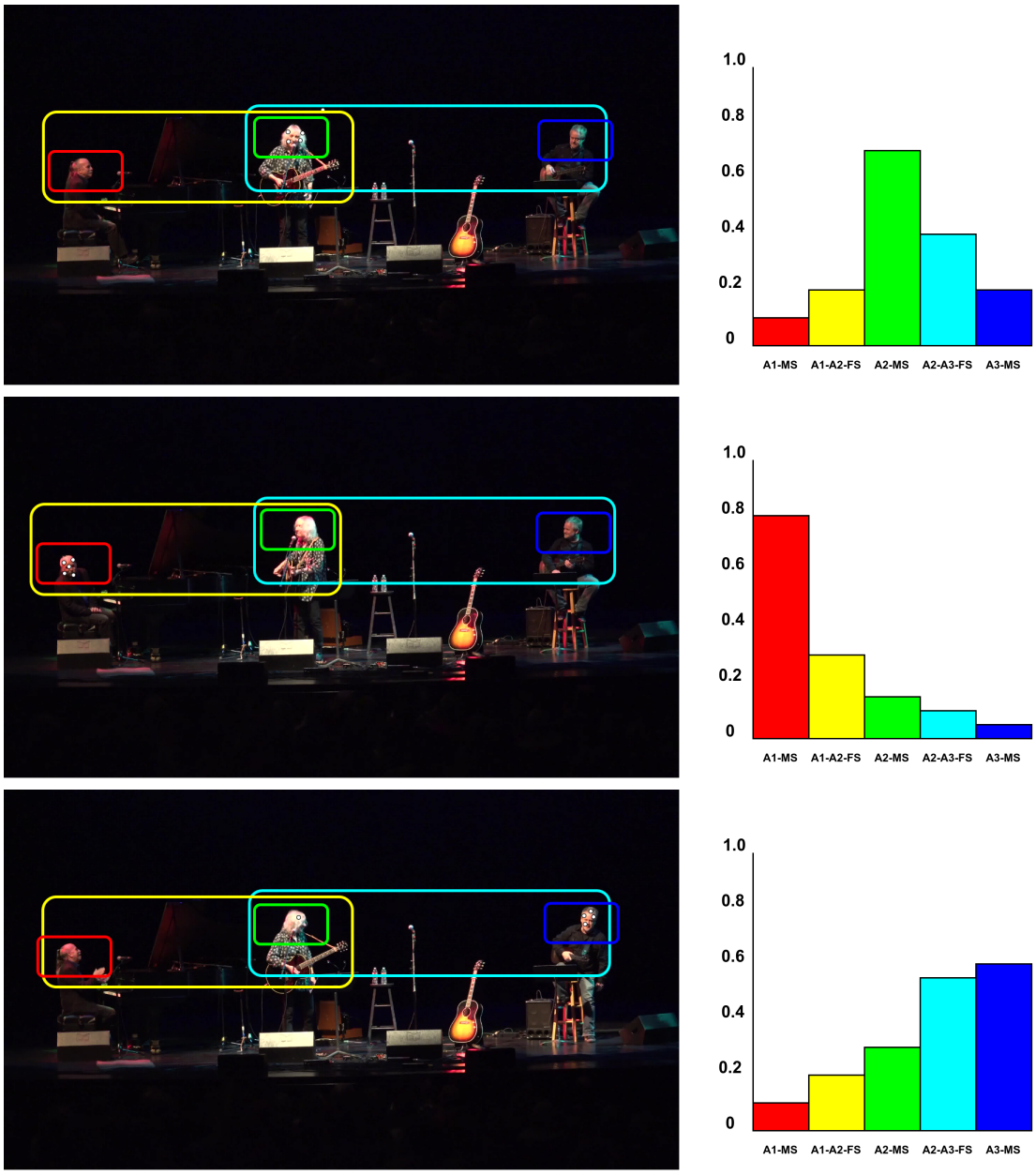

Figure 5.3: The figure showcases the behavior of the gaze potential function in response to human gaze. It provides a visual representation of this interaction by highlighting white dots that indicate the precise locations where the human gaze is directed within the frame. To better understand the influence of gaze on the scene, accompanying histograms are displayed beside each frame. These histograms present the gaze potential cost associated with each bounding box in the scene. To make it even more intuitive, the color-coded bars in the histograms correspond to the respective bounding boxes, allowing for a quick and easy comparison. For instance, the green bar in the histogram represents the gaze potential cost of the bounding box highlighted in green. This insightful visual representation offers a comprehensive understanding of the relationship between human gaze and gaze potential.

another shot, denoted as s_j^{t+1} at time $t+1$.

$$T(s_i^t, s_j^{t+1}) = \begin{cases} 0 & i = j \\ \lambda & i \neq j \end{cases} \tag{5.6}$$

where λ is a transition cost parameter.

- *Overlap cost* - To avoid the jarring effect of sudden time jumps, known as jump cuts, it is crucial to keep the overlap between two framings at a sufficiently low level. When there is a high overlap between two shots, it can create a visual discontinuity that disrupts the flow of the video. To address this, we introduce an overlap cost as a penalty term. This cost encourages smooth transitions between shots and helps maintain a coherent and seamless viewing experience. By minimizing the overlap cost, we ensure that the resulting edits are visually appealing and free from abrupt interruptions.

$$O(s_i^t, s_j^{t+1}, \gamma) = \begin{cases} 0 & \gamma \leq \alpha \\ \frac{\mu\gamma}{\alpha} & \alpha \leq \gamma \leq \beta \\ \upsilon & \gamma \geq \beta \end{cases} \tag{5.7}$$

- *Rhythm cost* - The frequency of cuts in video editing is a crucial element that significantly influences the audience's perception of a scene. The length of each shot has a profound impact on how the scene is experienced. Longer shots with a slower rhythm evoke a sense of calm and stillness, often used in romantic scenes to convey emotions. On the other hand, shorter shots with a faster rhythm are employed to create a dynamic and energetic atmosphere, commonly seen in action sequences. To control and manipulate the cut rhythm effectively, we introduce a cost factor based on the duration of each shot. This allows us to define a rhythm cost, which determines the overall impact of the shot's length on the editing process.

$$R(s_i^t, s_j^{t+1}, \tau) = \begin{cases} \gamma_1(1 - \frac{1}{1+\exp{(l-\tau)}}) & i \neq j \\ \gamma_2(1 - \frac{1}{1+\exp{(\tau-m)}}) & i = j \end{cases} \tag{5.8}$$

We incorporate the defined penalty terms into the computation of the cost matrix (C). The cost matrix is constructed along the time dimension, where each cell represents the minimum cost required to reach that specific point in time. In building the cost matrix, we utilize recurrence relation 5.9 that takes into account the information from the previous shot, the current shot's gaze potential, and other penalty terms. By evaluating this recurrence relation for each cell in the cost matrix, we can determine the optimal path and associated costs to navigate through the shots over time. This enables us to determine the most cost-effective and visually compelling editing path for the video.

$$C(s_j^t, t) = \begin{cases} -ln(G(s_j^t)) & t = 1 \\ min_i(C(s_i^{t-1}, t-1) - ln(G(s_j^t)) + E_e(s_i^{t-1}, s_j^t) & otherwise \end{cases} \tag{5.9}$$

We use the notation s_i^t to represent a specific shot s_i from the set of generated shots S at a given time t. Additionally, we introduce the method $Backtrack_i(c, t)$, which allows us to backtrack from a state c at time t to its preceding state b. This backtracking process enables us to identify the state b that led to the current state c during the forward pass in the cost matrix. By utilizing this method, we can effectively move backward in time to a state that is located i time steps earlier.

The $Future$ penalty term F encompasses the cost associated with choosing a specific shot s_k^{t+f} at time $t + f$ when considering the entire path leading up to that point.

$$F_k = C(s_k^{t+f}, t + f) \tag{5.10}$$

The $Continuity$ term in our methodology aims to address the smooth transition between shots by penalizing the cost associated with switching from a previously selected shot, denoted as p, to a new shot s. To calculate this term, we perform a backtrack operation starting from a shot (state) s_k^{t+f} in the future at time $t + f$. This backtrack operation allows us to trace back $f - 1$ time steps and determine the cost of transitioning from the previously selected shot p to the current shot s. Here, s is obtained by applying the backtrack function

$$s = Backtrack_{f-1}(s_k, t + f) \tag{5.11}$$

By considering the continuity term in our cost function 5.12, we ensure that the editing process maintains a seamless flow and avoids jarring transitions between shots. This term allows us to evaluate the transition cost between shots, taking into account the previously selected shot and the desired shot at a future time.

$$Continuity_k = C(p, t) - ln(G(Backtrack_{f-1}(s_k, t+f))) + E_e(p, Backtrack_{f-1}(s_k, t+f)) \tag{5.12}$$

Rather than relying on a comprehensive cost matrix for selecting optimal shots, we have devised a real-time shot selection process that incorporates a small look-ahead duration f to make informed decisions. At each frame, we follow a set of steps to determine the most suitable shot. There are two key considerations we take into account: a minimum shot duration constraint (l) and the use of a shot timer θ, which is crucial for the rhythm penalty term.

Our real-time shot selection process ensures that each shot satisfies the minimum duration requirement l to maintain coherence and avoid abrupt transitions. Additionally, we employ a shot timer θ to keep track of the duration of the selected shots. This timer is essential for accurately assessing the rhythm penalty, enabling us to maintain a consistent pacing and flow throughout the video. By implementing this approach, we achieve a dynamic shot selection process that takes into account both temporal constraints and the desired rhythm of the video.

1. If the shot timer $\theta \leq l$ we adhere to a strict constraint. In this case, we select the previous shot as the current shot and increment the shot timer $\theta = \theta + 1$ to keep track of elapsed time for the shot. In simpler terms, if the time allotted for a shot is within the predefined limit, we maintain continuity by keeping the same shot as the previous one.

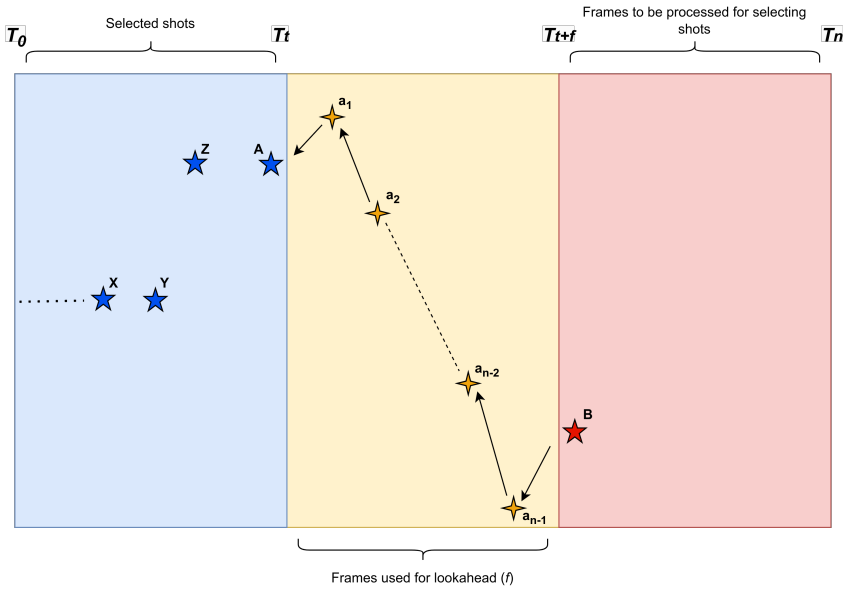

Figure 5.4: The figure provides a visual representation of how the cost matrix operates within the shot selection component of the Real Time GAZED pipeline. It offers insights into the sequential process of shot selection over time. The blue region highlights the frames that have already been processed by the Real Time GAZED algorithm for shot selection. It showcases the algorithm's ability to analyze and make decisions based on the visual content within these frames. As indicated by the labels $X, Y, Z, \& A$ several shots have been selected within the timeframe from the start T_0 up to the current time T_t. The yellow region denotes the frames used for lookahead, providing a glimpse into the frames that are considered for future shot selection. The labels $a_1, a_2, ..., a_{n-1}$ represent the intermediary shots that are assessed by backtracking before the final shot B is chosen at a time T_{t+f}. This lookahead approach allows for more informed and strategic shot selection. Finally, the red region represents the frames that are yet to be processed by the Real Time GAZED algorithm. These frames are awaiting analysis and decision-making to determine the subsequent shots.

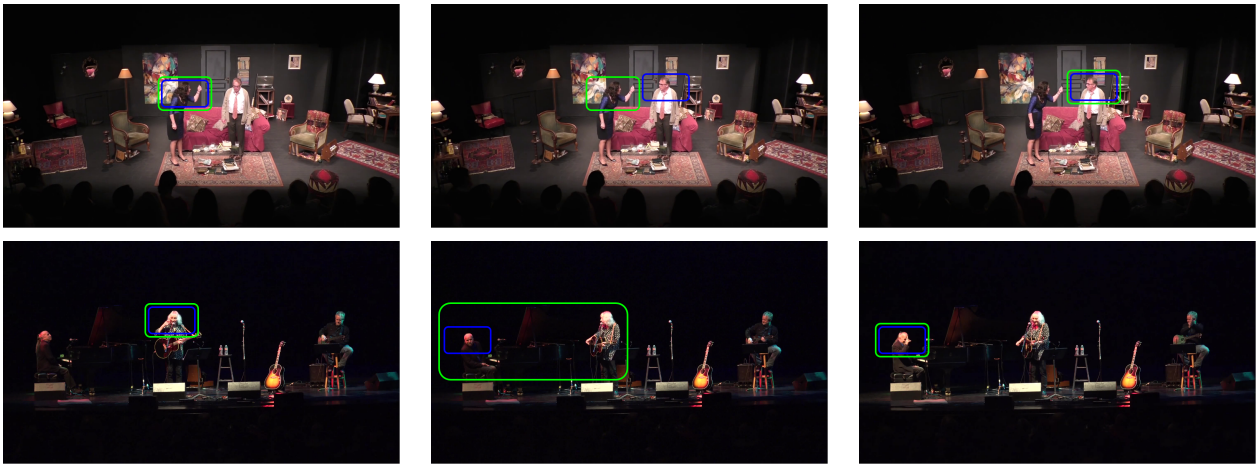

Figure 5.5: The figure showcases a visual comparison of shot selections made by Real Time GAZED (highlighted in green) and GAZED (highlighted in blue) for two different videos. Each video is represented in a separate row. What stands out is that the shot selections made by Real Time GAZED exhibit a catching up behavior with GAZED, given that it operates in real time. While there may be intermediary differences in shot selection, as depicted in the middle column, Real Time GAZED dynamically adjusts its selection to minimize the overall cost and align with the shot chosen by GAZED. This observation highlights the effectiveness of Real Time GAZED in maintaining comparable shot selections to its non-real-time counterpart while operating in real time.

2. Else if the shot timer $\theta \geq l$, we proceed with the shot selection process by minimizing objective function 5.13. This objective function helps us identify the most suitable shot to select. Additionally, we reset the shot timer θ if the selected shot s differs from the previously selected shot, ensuring a fresh start for the shot timer θ. However, if the selected shot s is the same as the previous shot, we increment the shot timer θ to continue the sequence which gets penalized by rhythm cost R.

$$min_k(F_k + \alpha * Continuity_k) \tag{5.13}$$

The objective function 5.13 is formulated as a combination of penalty terms that consider both future shots and shot continuity. However, it requires a tuning parameter, denoted as α, to achieve the desired optimization. The term F_k represents the cumulative cost of previous shots, which is calculated using a recurrence relation. Without careful consideration, the objective function may prioritize minimizing F_k alone. When α is set to a higher value, there is a risk of the objective function overly emphasizing the continuity penalty term ($Continuity_k$). To strike a balance between the cumulative cost of previous shots (F_k) and shot continuity ($Continuity_k$), it is advisable to choose α in proportion to the look-ahead duration (f). By adjusting α proportionally to f, we ensure that the optimization process takes into account the importance of both the cumulative cost of previous shots and maintaining continuity in a more balanced manner.

5.3 Comparison Baselines

To evaluate the effectiveness of Real Time GAZED, we compare it against four robust video editing baselines: Wide, Greedy Gaze, Speaker-based, and GAZED itself. In order to ensure fair comparison, we set the minimum shot duration parameter (l) to 1.5 seconds.

5.3.1 Wide

The Wide baseline approach is inspired by the concept of video retargeting. It selects the widest shot possible, encompassing all performers on the stage. This wide shot is a zoomed-in version of the master shot, capturing the smallest bounding box that includes all the performers.

5.3.2 Greedy Gaze

The Greedy Gaze editing algorithm greedily selects the shot with the highest gaze potential at each time instant. It directly chooses the presentation shot based on the local gaze potential optimum at time t. However, since this approach solely relies on gaze information without considering cinematic editing principles, it may result in frequent shot switches that could hinder the understanding of the scene and degrade the overall viewing experience. To mitigate this issue, we enforce a minimum shot duration of 1.5 seconds (specified by parameter l).

5.3.3 Speaker-based

Speaker cues have shown promise in editing dialog-driven scenes [42] [30]. Our Speaker-based baseline (Sp) selects the 1-shot that best captures the speaker among the rushes. For this study, speaker information in each video was manually annotated. When multiple individuals speak simultaneously, their combined shot is selected. The algorithm maintains the current selection until a change in the speaker occurs. To avoid rapid shot transitions, a minimum shot duration (specified by l) is enforced. If a silence of more than 10 seconds is detected, the wide shot is chosen for the next time instant.

5.3.4 GAZED

We also compare against the original GAZED framework itself, which serves as a baseline for our real-time version. This allows us to assess the improvements and performance of Real Time GAZED in comparison to the offline GAZED approach.

5.4 User Study

To assess the video editing capabilities of GAZED (GZD) compared to the aforementioned baselines, we conducted a psychophysical study involving 8 participants (distinct from those used for collecting

gaze data) and 4 video recordings. Different editing strategies, including Wide, Greedy Gaze, Speaker-based, GAZED, and Real Time GAZED, were applied to generate edited versions of these videos. To ensure fairness, all parameters of GAZED remained consistent across all videos. During the study, participants first watched the original video, followed by the randomly presented edited versions. We designed the study in a way that each participant viewed the original and edited versions of two stage recordings, resulting in a total of 4 (video types) x 2 (user ratings/video) x 5 (editing strategies) combinations.

Participants were unaware of the specific editing strategy employed for each version they watched. After viewing each edited version, they were asked to compare it to the original and rate it on a scale ranging from -5 to 5 for various attributes. The attributes of interest included:

1. Narrational Effectiveness (**NE**): How effectively did the edited video convey the original narrative ?

2. Scene Actions (**SA**): How well did the edited video capture actor movements and actions ?

3. Actor Emotions (**AE**): How well did the edited video capture actor emotions ?

4. Viewing Experience (**VX**): How would you rate the edited video for aesthetic quality ?

By collecting ratings for these attributes, we aimed to evaluate and compare the video editing performances of GAZED and the baseline strategies in terms of visual appeal, narrative coherence, emotional impact, shot transitions, and overall editing quality. Prior to the study, participants were provided with information about the specific attributes and cinematic video editing conventions. They were then asked to rate each attribute using a scale relative to a reference score of '0' assigned to the original video. A positive score indicated that the edited version performed better than the original in terms of the specific attribute, while a negative score indicated that the edited version performed worse. The ratings provided by the participants were collected and the mean scores for each attribute and editing strategy were calculated across all videos.

5.4.1 Narrational Effectiveness (NE)

These findings collectively emphasize the importance of skillfully composing shots that capture close-up views of the key actors and actions for effective storytelling. The Greedy Gaze (GG), Speaker-based (Sp), GAZED and Real Time GAZED strategies, which prioritize actors and actions based on speech or gaze cues, outperform the Wide baselines in terms of their ability to capture the essence of the scene. Unlike the Wide approach, which often results in inefficient framing of the scene characters.

5.4.2 Scene Actions (SA)

The Wide and Speaker-based (Sp) baselines demonstrate similar performance in this aspect. These findings suggest that relying solely on speaker cues may not be as effective in capturing focal events

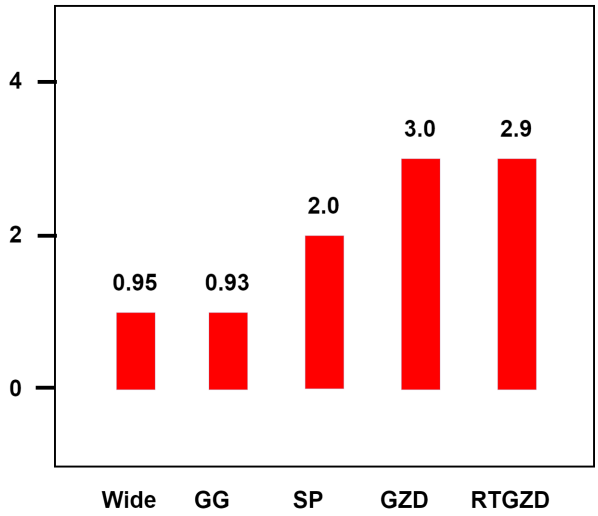

Figure 5.6: Each bar in the histogram denotes the minimum and maximum user rating of narrational effectivenes (NE) for each baseline Wide, Greedy gaze (GG), Speaked based (Sp), GAZED (GZD) and Real Time GAZED (RTGZD)

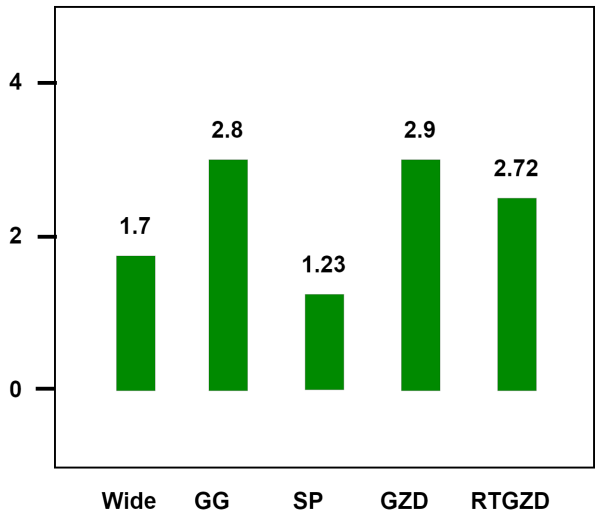

Figure 5.7: Each bar in the histogram denotes the minimum and maximum user rating of scene actions (SA) for each baseline Wide, Greedy gaze (GG), Speaked based (Sp), GAZED (GZD) and Real Time GAZED (RTGZD)

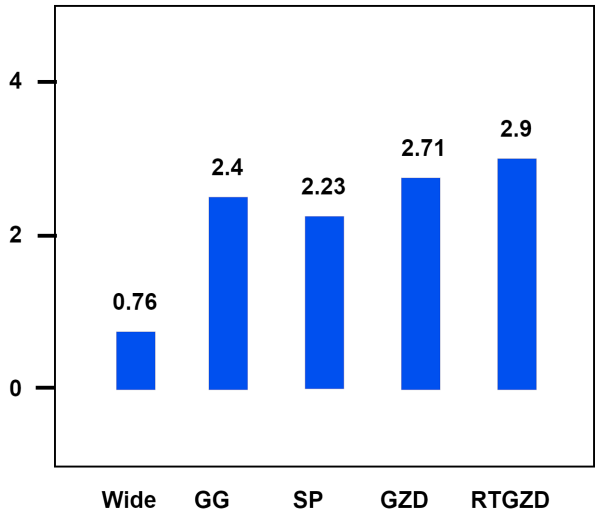

Figure 5.8: Each bar in the histogram denotes the minimum and maximum user rating of action emotions (AE) for each baseline Wide, Greedy gaze (GG), Speaked based (Sp), GAZED (GZD) and Real Time GAZED (RTGZD)

during stage performances. For instance, if one performer introduces other co-performers to the audience verbally, the Sp baseline may still prioritize the introducer instead of the introducee. In such cases, eye gaze proves to be more accurate in capturing the events and actors of interest compared to speech. The Wide baseline, which captures the entire scene context at all times, can only provide low-resolution views of the performers to viewers, resulting in similar performance to Sp. On the other hand, the Greedy Gaze (GG) strategy, which dynamically captures events of maximum interest at each time instant, proves to be effective in conveying scene actions and performs well.

5.4.3 Actor Emotions (AE)

We find that the GG, Sp, GAZED, and Real Time GAZED techniques yield comparable performance. In fact, these approaches outperform the Wide baseline by a significant margin. This can be attributed to the fact that the GG, Sp, GAZED, and Real Time GAZED methods effectively capture the speaker or main actor in the scene through close-up shots, allowing for the clear conveyance of facial expressions and emotions to viewers. On the other hand, the Wide baseline captures all actors in the scene in each video frame, resulting in relatively low-resolution presentation of facial movements to the audience.

5.4.4 Viewing Experience (VX)

We hypothesized that by incorporating cinematic editing principles and capturing focal scene events, our shot selection framework would enhance the viewing experience of the edited video. As expected, both GAZED and Real Time GAZED performed exceptionally well, receiving the highest scores for viewing experience among the five methods tested. The superiority of the Wide baseline over Greedy

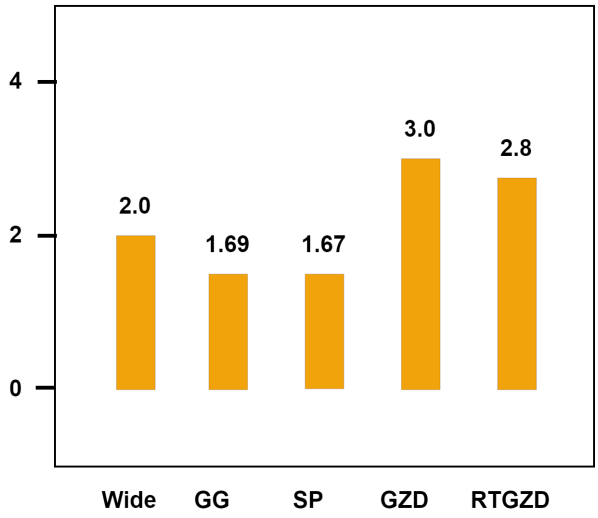

Figure 5.9: Each bar in the histogram denotes the minimum and maximum user rating of viewing experince (VX) for each baseline Wide, Greedy gaze (GG), Speaked based (Sp), GAZED (GZD) and Real Time GAZED (RTGZD)

Gaze (GG) and Speaker-based (Sp) can be attributed to the fact that the Wide strategy ensures the entire scene context is always visible to the viewer. On the other hand, GG and Sp frequently cut between shots, focusing on perceived actions of interest, which can disrupt the viewing experience. In contrast, Wide maintains a more consistent framing of the scene throughout, resulting in a smoother and more enjoyable edited video.

5.5 Summary

This study introduces Real Time GAZED, a modified version of the GAZED framework designed for real-time editing of stage performance videos. Unlike the original GAZED, Real Time GAZED utilizes user gaze cues to guide the shot selection process in real time. It incorporates cinematic editing principles such as avoiding abrupt transitions, eliminating transient shots, and controlling the rhythm of shot changes. By optimizing an energy minimization function with a small lookahead, Real Time GAZED ensures smooth and visually engaging edited videos.

To evaluate the performance of Real Time GAZED, we conducted a user study comparing it to four editing baselines: Wide, Greedy Gaze, and Speech-based. The Wide baseline draws inspiration from letterboxing techniques used in video targeting, while the Speech-based baseline mimics approaches employed in previous studies [42] and [30] on editing stage recordings. The Greedy Gaze baseline demonstrates the impact of incorporating cinematic editing rules on the smoothness and aesthetics of the edited video. Real Time GAZED shares the same limitations as the original GAZED framework, but its real-time processing capability transforms editing into a dynamic task rather than a post-processing

one. The user opinions collected from a psychophysical study confirm the effectiveness of Real Time GAZED in producing visually pleasing and engaging edited videos.

Chapter 6

Conclusion and Future work

In conclusion, this thesis presents two unsupervised methods, CineConvex and CineCNN, for online trajectory filtering in automated camera systems. The CineConvex formulation solves filtering as a convex optimization problem within sliding windows, while the CineCNN formulation employs a convolutional neural network for trajectory prediction. These methods offer high frame rates, low memory usage, and can be applied across various applications. Through extensive experiments in basketball games and theatre plays, the proposed methods outperform existing online filters, making them a superior choice for trajectory filtering in real-time camera systems.

Building upon these trajectory filtering techniques, the thesis introduces Real Time GAZED, a modified version of the GAZED framework tailored for real-time editing of stage performance videos. Real Time GAZED incorporates user gaze cues to guide the shot selection process in real time, adhering to cinematic editing principles such as smooth transitions, avoiding abrupt changes, and controlling shot rhythm. By optimizing an energy minimization function with a small lookahead, Real Time GAZED ensures visually appealing and engaging edited videos. To evaluate the performance of Real Time GAZED, a user study was conducted comparing it to four editing baselines: Wide, Greedy Gaze, Speaker-based, and GAZED itself. The results showed that Real Time GAZED outperforms the baselines, producing visually pleasing and engaging edited videos. Although Real Time GAZED shares some limitations with the original GAZED framework, its real-time processing capability transforms editing into a dynamic task, enabling immediate and dynamic content creation.

Overall, this thesis contributes to the advancement of online trajectory filtering and real-time video editing. The proposed methods provide efficient and effective solutions for trajectory filtering in automated camera systems, while Real Time GAZED offers a practical and engaging approach to real-time video editing, catering to the demands of today's fast-paced digital landscape in real time video processing for dynamic content creation

Chapter 7

Related Publications

1. **CineFilter: Unsupervised Filtering for Real Time Autonomous Camera Systems** *WICED 2020 Workshop on Intelligent Cinematography and Editing*

    - ***Sudheer Achary***, K. L. Bhanu Moorthy, Ashar Javed, Nikitha Shravan, Vineet Gandhi, and Anoop M. Namboodiri

2. **Assessing active speaker detection algorithms through the lens of automated editing** *WICED x Cinemotions IMX2023 Joint Workshop on "Intelligent Cinematography and Editing" and "Emotions in Movies"*

    - Rohit Girmaji, ***Sudheer Achary***, Adhiraj Deshmukh and Vineet Gandhi

3. **Real Time GAZED** *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV 2023) [Under review]*

    - ***Sudheer Achary***, Rohit Girmaji, Adhiraj Deshmukh and Vineet Gandhi

Bibliography

[1] Y. Ariki, S. Kubota, and M. Kumano. Automatic production system of soccer sports video by digital camera work based on situation recognition. In *Eighth IEEE International Symposium on Multimedia (ISM'06)*, pages 851–860. IEEE, 2006.

[2] M. Bianchi. Automatic video production of lectures using an intelligent and aware environment. In D. S. Doermann and R. Duraiswami, editors, *Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia, MUM 2004, College Park, Maryland, USA, October 27-29, 2004*, volume 83 of *ACM International Conference Proceeding Series*, pages 117–123. ACM, 2004.

[3] M. Bianchi. Automatic video production of lectures using an intelligent and aware environment. In *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, pages 117–123. ACM, 2004.

[4] J. Cantine et al. *Shot by shot: A practical guide to filmmaking*. ERIC, 1995.

[5] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(1):172–186, 2021.

[6] P. Carr, M. Mistry, and I. Matthews. Hybrid robotic/virtual pan-tilt-zom cameras for autonomous event recording. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 193–202. ACM, 2013.

[7] C. Chen, O. Wang, S. Heinzle, P. Carr, A. Smolic, and M. H. Gross. Computational sports broadcasting: Automated director assistance for live sports. In *Proceedings of the 2013 IEEE International Conference on Multimedia and Expo, ICME 2013, San Jose, CA, USA, July 15-19, 2013*, pages 1–6. IEEE Computer Society, 2013.

[8] F. Chen and C. De Vleeschouwer. Personalized production of basketball videos from multi-sensored data under limited display resolution. *Computer Vision and Image Understanding*, 114(6):667–680, 2010.

[9] F. Chen and C. D. Vleeschouwer. Personalized production of basketball videos from multi-sensored data under limited display resolution. *Comput. Vis. Image Underst.*, 114(6):667–680, 2010.

[10] J. Chen, H. M. Le, P. Carr, Y. Yue, and J. J. Little. Learning online smooth predictors for realtime camera planning using recurrent decision trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4688–4696, 2016.

[11] J. Chen, L. Meng, and J. J. Little. Camera selection for broadcasting soccer games. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*, pages 427–435. IEEE Computer Society, 2018.

[12] D. B. Christianson, S. E. Anderson, L. He, D. Salesin, D. S. Weld, and M. F. Cohen. Declarative camera control for automatic cinematography. In W. J. Clancey and D. S. Weld, editors, *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, USA, August 4-8, 1996, Volume 1*, pages 148–155. AAAI Press / The MIT Press, 1996.

[13] L. Condat. A direct algorithm for 1-d total variation denoising. *IEEE Signal Processing Letters*, 20(11):1054–1057, 2013.

[14] S. Daigo and S. Ozawa. Automatic pan control system for broadcasting ball games based on audience's face direction. In H. Schulzrinne, N. Dimitrova, M. A. Sasse, S. B. Moon, and R. Lienhart, editors, *Proceedings of the 12th ACM International Conference on Multimedia, New York, NY, USA, October 10-16, 2004*, pages 444–447. ACM, 2004.

[15] S. Daigo and S. Ozawa. Automatic pan control system for broadcasting ball games based on audience's face direction. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 444–447. ACM, 2004.

[16] P. Doubek, I. Geys, T. Svoboda, and L. Van Gool. Cinematographic rules applied to a camera network. 04 2009.

[17] D. K. Elson and M. O. Riedl. A lightweight intelligent virtual cinematography system for machinima production. In J. Schaeffer and M. Mateas, editors, *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference, June 6-8, 2007, Stanford, California, USA*, pages 8–13. The AAAI Press, 2007.

[18] Q. Galvane, R. Ronfard, C. Lino, and M. Christie. Continuity editing for 3d animation. In B. Bonet and S. Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 753–762. AAAI Press, 2015.

[19] V. Gandhi, R. Ronfard, and M. Gleicher. Multi-clip video editing from a single viewpoint. In *Proceedings of the 11th European Conference on Visual Media Production*, page 9. ACM, 2014.

[20] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR 2011*, pages 225–232. IEEE, 2011.

[21] L. He, M. F. Cohen, and D. Salesin. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In J. Fujii, editor, *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996, New Orleans, LA, USA, August 4-9, 1996*, pages 217–224. ACM, 1996.

[22] R. Heck, M. Wallick, and M. Gleicher. Virtual videography. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 3(1):4, 2007.

[23] R. Heck, M. N. Wallick, and M. Gleicher. Virtual videography. *ACM Trans. Multim. Comput. Commun. Appl.*, 3(1):4, 2007.

[24] A. Inoue, H. Shigeno, K. Okada, and Y. Matsushita. Introducing grammar of the film language into automatic shooting for face-to-face meetings. In *2004 Symposium on Applications and the Internet (SAINT 2004), 26-30 January 2004, Tokyo, Japan*, pages 277–282. IEEE Computer Society, 2004.

[25] E. Jain, Y. Sheikh, A. Shamir, and J. Hodgins. Gaze-driven video re-editing. *ACM Transactions on Graphics (TOG)*, 34(2):21, 2015.

[26] E. Jain, Y. Sheikh, A. Shamir, and J. K. Hodgins. Gaze-driven video re-editing. *ACM Trans. Graph.*, 34(2):21:1–21:12, 2015.

[27] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 trend filtering. *SIAM review*, 51(2):339–360, 2009.

[28] M. Kleiner, D. Brainard, D. Pelli, A. Ingling, R. Murray, and C. Broussard. What's new in psychtoolbox-3. *Perception*, 36:1–16, 01 2007.

[29] M. Kumar, V. Gandhi, R. Ronfard, and M. Gleicher. Zooming on all actors: Automatic focus+ context split screen video generation. In *Computer Graphics Forum*, volume 36, pages 455–465. Wiley Online Library, 2017.

[30] M. Leake, A. Davis, A. Truong, and M. Agrawala. Computational video editing for dialogue-driven scenes. *ACM Trans. Graph.*, 36(4):130:1–130:14, 2017.

[31] C. Lino, M. Chollet, M. Christie, and R. Ronfard. Computational model of film editing for interactive storytelling. In M. Si, D. Thue, E. André, J. C. Lester, T. J. Tanenbaum, and V. Zammitto, editors, *Interactive Storytelling - Fourth International Conference on Interactive Digital Storytelling, ICIDS 2011, Vancouver, Canada, November 28 - 1 December, 2011. Proceedings*, volume 7069 of *Lecture Notes in Computer Science*, pages 305–308. Springer, 2011.

[32] F. Liu and M. Gleicher. Video retargeting: automating pan and scan. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 241–250. ACM, 2006.

[33] S. Liu, P. Tan, L. Yuan, J. Sun, and B. Zeng. Meshflow: Minimum latency online video stabilization. In *European Conference on Computer Vision*, pages 800–815. Springer, 2016.

[34] B. Merabti, M. Christie, and K. Bouatouch. A virtual director using hidden markov models. *Comput. Graph. Forum*, 35(8):51–67, 2016.

[35] MIPLIB 2017, 2018. http://miplib.zib.de.

[36] K. L. B. Moorthy, M. Kumar, R. Subramanian, and V. Gandhi. GAZED- gaze-guided cinematic editing of wide-angle monocular video recordings. In R. Bernhaupt, F. F. Mueller, D. Verweij, J. Andres, J. McGrenere, A. Cockburn, I. Avellino, A. Goguey, P. Bjøn, S. Zhao, B. P. Samson, and R. Kocielnik, editors, *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, pages 1–11. ACM, 2020.

[37] A. Moudgil and V. Gandhi. Long-term visual object tracking benchmark. *arXiv preprint arXiv:1712.01358*, 2017.

[38] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4293–4302, 2016.

[39] C. S. Pinhanez. *Intelligent studios: Using computer vision to control TV cameras*. 1995.

[40] K. K. Rachavarapu, M. Kumar, V. Gandhi, and R. Subramanian. Watch to edit: Video retargeting using gaze. *Comput. Graph. Forum*, 37(2):205–215, 2018.

[41] K. K. Rachavarapu, M. Kumar, V. Gandhi, and R. Subramanian. Watch to edit: Video retargeting using gaze. In *Computer Graphics Forum*, volume 37, pages 205–215. Wiley Online Library, 2018.

[42] A. Ranjan, J. P. Birnholtz, and R. Balakrishnan. Improving meeting capture by applying television production principles with audio and motion detection. In M. Czerwinski, A. M. Lund, and D. S. Tan, editors, *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008, 2008, Florence, Italy, April 5-10, 2008*, pages 227–236. ACM, 2008.

[43] A. Savitzky and M. J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[44] G. Solver. Gurobi optimization, 2019.

[45] X. Sun, J. Foote, D. Kimber, and B. Manjunath. Region of interest extraction and virtual camera control based on panoramic video capturing. *IEEE Transactions on Multimedia*, 7(5):981–990, 2005.

[46] Y. Takemae, K. Otsuka, and N. Mukawa. Impact of video editing based on participants' gaze in multiparty conversation. In E. Dykstra-Erickson and M. Tscheligi, editors, *Extended abstracts of the 2004 Conference on Human Factors in Computing Systems, CHI 2004, Vienna, Austria, April 24 - 29, 2004*, pages 1333–1336. ACM, 2004.

[47] C. Tang, O. Wang, F. Liu, and P. Tan. Joint stabilization and direction of 360\deg videos. *ACM Transactions on Graphics (TOG)*, 2018.

[48] R. Thompson and C. Bowen. *Grammar of the Shot*. Taylor & Francis, 2013.

[49] J. Wang, C. Xu, E. Chng, H. Lu, and Q. Tian. Automatic composition of broadcast sports video. *Multim. Syst.*, 14(4):179–193, 2008.

[50] M. Wang, G.-Y. Yang, J.-K. Lin, A. Shamir, S.-H. Zhang, S.-P. Lu, and S.-M. Hu. Deep online video stabilization. *arXiv preprint arXiv:1802.08091*, 2018.

[51] T. Yokoi and H. Fujiyoshi. Virtual camerawork for generating lecture video from high resolution images. In *2005 IEEE International Conference on Multimedia and Expo*, pages 4–pp. IEEE, 2005.

[52] C. Zhang, Y. Rui, J. Crawford, and L. He. An automated end-to-end lecture capture and broadcasting system. 4(1):6:1–6:23, 2008.

[53] C. Zhang, Y. Rui, J. Crawford, and L.-W. He. An automated end-to-end lecture capture and broadcasting system. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 4(1):6, 2008.