GAZED– Gaze-guided Cinematic Editing of Wide-Angle Monocular Video Recordings

K L Bhanu Moorthy
CVIT, IIIT Hyderabad
Hyderabad, India
k.l.bhanu@research.iiit.ac.in

Moneish Kumar *
Samsung R&D Institute
Bangalore, India
moneish04@gmail.com

Ramanathan Subramanian
IIT Ropar, India
s.raamanathan@iitrpr.ac.in

Vineet Gandhi
CVIT, IIIT Hyderabad
Hyderabad, India
vgandhi@iiit.ac.in

Figure 1. We present GAZED: gaze-guided and cinematic editing of monocular and static stage performance recordings. (Top row) GAZED takes as input frames from the original video (*master shots*), and generates multiple *rushes* by simulating several virtual Pan/Tilt/Zoom cameras. Generated rushes are shown below each frame. Eye gaze data are then utilized to *select* the *best* rush (shown in red) for each master shot. The *edited video* (a) vividly presents scene emotions and actions, and (b) adheres to cinematic principles to present aesthetic content. Edited video frames are shown in the bottom row. Best viewed in color.

ABSTRACT

We present **GAZED**– eye GAZe-guided EDiting for videos captured by a solitary, static, wide-angle and high-resolution camera. Eye-gaze has been effectively employed in computational applications as a cue to capture *interesting* scene content; we employ gaze as a proxy to *select shots* for inclusion in the *edited video*. Given the original video, scene content and user eye-gaze tracks are combined to generate an edited video comprising ***cinematically valid*** *actor shots* and *shot transitions* to generate an aesthetic and vivid representation of the original narrative. We model cinematic video editing as an *energy minimization problem* over *shot selection*, whose constraints capture cinematographic editing conventions. Gazed scene locations primarily determine the shots constituting the edited video. Effectiveness of GAZED against multiple competing methods is demonstrated via a psychophysical study involving 12 users and twelve performance videos.

Author Keywords

Eye gaze, Cinematic video editing, Stage performance, Static wide-angle recording, Gaze potential, Shot selection, Dynamic programming

CCS Concepts

•**Information systems** → **Multimedia content creation;**
•**Mathematics of computing** → *Combinatorial optimization;*
•**Computing methodologies** → *Computational photography;*
•**Human-centered computing** → User studies;

INTRODUCTION

Professional video recordings of stage performances are typically created by employing skilled camera operators, who record the performance from multiple viewpoints. These multi-camera feeds, termed *rushes*, are then *edited* together to portray an eloquent story intended to maximize viewer engagement. Generating professional edits of stage performances is both difficult and challenging. Firstly, maneuvering cameras during a live performance is difficult even for experts as there is no option of retake upon error, and camera viewpoints are limited as the use of large supporting equipment

*Work done while at IIIT Hyderabad

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI'20, April 25–30, 2020, Honolulu, HI, USA

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ISBN 978-1-4503-6708-0/20/04…$15.00

DOI: https://doi.org/10.1145/3313831.XXXXXXX

(trolley, crane *etc*.) is infeasible. Secondly, manual video editing is an extremely slow and tedious process and leverages the experience of skilled editors. Overall, the need for (i) a professional camera crew, (ii) multiple cameras and supporting equipment, and (iii) expert editors escalates the process complexity and costs.

Consequently, most production houses employ a large field-of-view static camera, placed far enough to capture the entire stage. This approach is widespread as it is simple to implement, and also captures the entire scene. Such static visualizations are apt for archival purposes; however, they are often unsuccessful at captivating attention when presented to the target audience. While conveying the overall context, the distant camera feed fails to bring out vivid scene details like close-up faces, character emotions and actions, and ensuing interactions which are critical for cinematic storytelling (Fig. 1, top row). Renowned film editor Walter Murch states in his book [23] that a primary objective of film editing is to register the expression in the actor's eyes. If this is not possible, one should attempt to capture the best possible close-up of the actor, even if the original wide-shot may convey the scene adequately.

GAZED denotes an end-to-end pipeline to generate an *aesthetically edited video* from a single static, wide-angle stage recording. This is inspired by prior work [14], which describes how a plural camera crew can be replaced by a *single* high-resolution static camera, and *multiple virtual camera shots* or *rushes* generated by simulating several virtual pan/tilt/zoom cameras to focus on actors and actions within the original recording. In this work, we demonstrate that the multiple rushes can be automatically edited by leveraging user *eye gaze* information, by modeling (virtual) *shot selection* as a discrete optimization problem. Eye-gaze represents an inherent guiding factor for video editing, as eyes are sensitive to *interesting* scene events [26, 28] that need to be vividly presented in the edited video.

Bottom insets in the top row of Fig. 1 show the multiple rushes (which are either manually shot or virtually generated) to choose from at any given time; the objective critical for video editing and the key contribution of our work is to *decide which shot (or rush) needs to be selected to describe each frame of the edited video*. The shot selection problem is modeled as an optimization, which incorporates gaze information along with other cost terms that model *cinematic editing principles*. Gazed scene locations are utilized to define *gaze potentials*, which measure the importance of the different shots to choose from. Gaze potentials are then combined with other terms that model cinematic principles like *avoiding jump cuts* (which produce jarring shot transitions), *rhythm* (pace of shot transitioning), *avoiding transient shots etc*. The optimization is solved using ***dynamic programming*** (https://en.wikipedia.org/wiki/Dynamic_programming).

To validate GAZED, we compare multiple edited versions of 12 performance recordings via a psychophysical study involving 12 users. Our editing strategy outperforms multiple competing baselines such as *random* editing, *wide-shot* framing, *speaker detection*-based editing and *greedy gaze*-based editing. Our contributions include:

(1) **Gaze potential for shot selection:** We model user eye-gaze information via gaze potentials that quantify the *importance* of the different shots (rushes) generated from the original recording. Our algorithm examines locations as well as the extent of gaze clusters in a bottom-up fashion to compute unary and higher-order gaze potentials. Human gaze is known to be more sensitive to high-level scene semantics such as emotions [26, 28], as compared to bottom-up computational saliency methods.

(2) **Novel video editing methodology:** We perform shot selection by minimizing an objective function modeled via gaze potentials plus constraints conveying cinematic principles. GAZED edits a 1-minute video with four performers on stage in 5 seconds on a PC with 7th generation Intel 2.7 GHz i5 processor and 8GB RAM. In contrast, manual editing is both time and effort-intensive.

(3) **An end-to-end cinematic editing pipeline:** to generate professional videos from a static camera recording. GAZED enables novice users to create professionally edited videos using their eye gaze data and an inexpensive desktop eye-tracker. The *Tobii Eye-X* eye-tracker costing $< €200$ was used to compile eye gaze data in this work. We employ the algorithm of Gandhi *et al.* [14] to generate multiple rushes, followed by dynamic programming to obtain the edited video.

(4) **A thorough user study:** to validate the superiority of GAZED against multiple editing baselines. Results confirm that GAZED outputs are preferred by users with respect to several attributes characterizing editing quality.

RELATED WORK

Editing in virtual 3D environments

The video editing problem has been extensively studied in virtual 3D environments, where cinematographic conventions are computationally modeled to best convey the animated content. Earlier works [8, 16] focused on an idiom based approach. Film idioms [2] are a collection of stereotypical formulas for capturing specific scenes as sequences of shots. Difficulty to formulate every specific scenario significantly limits idiom-based approaches. Moreover, in constrained settings like live theatre, it may not always be feasible to acquire all the ingredient shots of the formula.

Another series of papers pose video editing as a discrete optimization problem, solved using dynamic programming [12, 21, 13, 22]. The key idea is to define the importance of each shot based on the narrative, and solve for the best set of shots that maximize viewer engagement. The work of Elson *et al.* [12] couples the twin problems of camera placement and camera selection; however, details are insufficiently described to be reproduced. Meratbi *et al.* [22] employs a Hidden Markov Model for the editing process, where shot transition probabilities are learned from existing films. They only limit to dialogue scenes, which require manual annotation from actual movies. The work of Galvane *et al.* [13] decouples the camera positioning and selection problems, and is arguably the most comprehensive effort in video editing, precisely addressing several important issues like exact placement of cuts, rhythm and continuity editing rules. While our work is inspired by

these efforts, stage performances are significantly constrained as neither is there freedom to freely place cameras, nor does one have access to the precise scene geometry, character localization, events and action information which is available in 3D environments on which the above works are applicable.

Editing 2D videos

Automated video editing has also been studied in other specialized scenarios. The Virtual Videography system [17] simulates shots captured by virtual pan-tilt-zoom cameras from lecture videos, and the best shot is selected via branch-and-bound optimization. The MSLRCS [31] and AutoAuditorium [3] systems use a small set of fixed cameras including input of presentation slides. Editing is done via a rule-based approach limited to a constrained environment of usually a single presenter in front of a chalkboard/slide screen.

Ranjan *et al.* [27] propose a system for editing group meetings based on speaker detection cues, posture changes, and head orientation. They model editing in the form of simple rules like *cut to the close up of the speaker* when a speaker change is detected and *cut to an overview shot* when multiple people are speaking. In our experiments, we compare GAZED with a similar baseline which focuses on the speaker(s) over time. Doubek *et al.* [11] study the problem of camera selection in surveillance and telepresence settings.

The camera selection problem has been extensively studied in sports events [30, 6, 5, 7]. Earlier works [30, 6] employ Hidden Markov Models to select from within a panoramic view, or from multiple views. Other works [5, 7] take a data-driven approach and train regressors to learn the importance of each camera view at a given time. A recent interesting work by Leake *et al.* [20] proposes an idiom based approach for editing dialogue-driven scenes. Inputs to their system include multiple camera views with the film script, and the output is the most informative set of shots for each line of the dialogue. In contrast to the approaches which work on multi-camera feeds plus scene-related meta-data, GAZED only requires a static wide-angle camera recording of a stage performance, and eye gaze data from one or more viewers (with a high-end eye-tracker, eye gaze recording of the editor/director reviewing the event would be sufficient). Our method does not rely on script alignment, multiple carefully placed cameras, speaker information *etc*. Furthermore, our approach is applicable to a wide variety of scenarios, and we show results on *theatre*, *dance* and *music* performances as part of the supplementary video.

Gaze driven editing

Prior works have also employed eye-gaze for video editing. The gaze is estimated either via head pose or through specialized eye-tracking devices. Takemai *et al.* [29] propose a video editing system based on the participant's gaze direction in indoor conversations. They show results on debate scenarios comprising 3–4 participants, where participant gaze is manually labelled for the entire video. They apply a simple editing rule, *i.e.*, to cut to the close-up of the person that most peers are gazing at. They demonstrate that gaze is able to better convey the flow of conversation compared to a speaker detection

Figure 2. Video retargeting vs. editing: *Video retargeting* (top row) involves estimating the location of a cropping window (defined by the x-coordinate denoted by a green dot) that preserves focal scene content for rendering on a device with different aspect ratio. *Video editing* (bottom row) involves compositing several manually captured or virtually generated shots (two of the simulated shots are shown) to capture focal characters in vivid detail, and selecting at each instant the shot that most engagingly conveys the storyline. Selected shots are rendered at the same aspect ratio as the original video.

based approach. Daigo *et al.* [10] use audience gaze direction to estimate the areas of interest in a basketball match. Park *et al.* [24] and Arev *et al.* [1] identify salient scene regions from the convergence of the field-of-view of multiple social cameras viewing the scene.

Two recent works [18, 25] employ eye gaze data for the problem of *video retargeting*, which is meant to *adapt* an *edited video* designed for a specific display device to a different one (*e.g.*, theatre screen to a mobile device). This is often achieved by virtually moving a cropping window within the original video, which preserves the salient scene content (Fig. 2). These methods primarily solve for an x value at each time point, and allow nominal zoom based on gaze variance. Since these methods do not solve for y, they cannot be used to obtain well-composed framings (*e.g.*, medium shot, close-up) of the scene objects and actions. Substantial zoom in [18, 25] ends up generating odd compositions (frame covering head but not face; actors cropped in an unaesthetic way, *etc*). In contrast, the proposed work emulates the multi-camera video production-plus-editing pipeline, and focal scene actors and events are captured via efficient shot selection. GAZED performs shot selection among several *rushes* (which are either part of the original capture or virtually generated), where each rush is carefully composed based on cinematic rules and is parameterized by the x, y values denoting the cropping window location, along with the zoom level. In summary, the GAZED system is designed to accomplish the (edited) video creation process (not just adaptation), and can analogously work with a multi-camera setting capturing a wide scene.

GAZED OVERVIEW

GAZED represents an end-to-end system to automate the entire video production process for staged performances. The system takes as input a static, high-definition recording of the scene and eye-tracking data of one or more users who have reviewed the video recording (typically an expert such as the program director/editor). It then outputs an edited video that

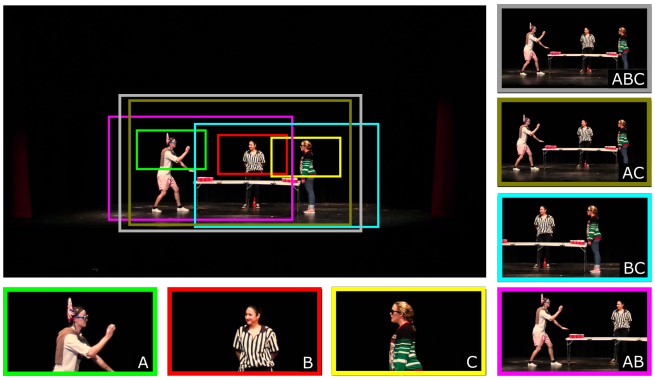

Figure 3. Virtual shot generation: Rushes simulated from original video frame are denoted via colored rectangles. Anti-clockwise from top left: *Master shot* with actors A, B and C. Generated 1-shots for each actor. Three 2-shots (AB, BC, AC) and a 3-shot (ABC comprising all three actors).

adheres to common cinematic principles and is aesthetically pleasing to watch. Like the traditional video production process, we also split the task into two parts (i) The *generation* of cinematically valid shots capturing context as well as actions and facial expressions of performers. (ii) *Selection* of the most appropriate shot at each time for engaging story-telling.

Shot Generation

GAZED takes as input a wide-angle recording captured with a static camera, covering the entire scene. We term each frame in this input wide-angle video as the 'master shot' in the rest of the paper. Our work assumes that tracks (bounding box coordinates) of all the performers/actors are available for all master shots. Our approach is agnostic to the object detection/tracking algorithm; however, in this work, we employ OpenPose [4] as a person detector. Hungarian matching was then employed with the pairwise appearance and proximity costs for generating identity-preserving actor tracks. Tracking errors (if any) were manually corrected. A virtual camera simulation approach [15] was then used to automatically generate multiple zoomed-in shots from the original wide-angle recording. The method simulates multiple virtual pan-tilt-zoom (PTZ) cameras by moving multiple cropping windows (each following a particular actor or a group of actors) inside the master shot. The shot generation problem is cast as a convex optimization, which takes into consideration *composition* (framing/positioning of actors in the frame), *panning* and *cutting* aspects of cinematography.The optimization translates noisy shot estimates into well-composed cinematic shots similar to those generated by professional cameramen.

Given an input video, we first generate the set of all possible shots for every combination of performers in the video. For a video with n performers, there are 2^n-1 combinations possible. Figure 3 illustrates an exemplary video frame, and the set of constituent shots, which include three 1-shots (individual actor shots), three 2-shots (shots containing two actors) and a 3-shot (shot containing all actors). Similarly, for an N actor sequence we generate ${}^nC_1$ 1-shots; ${}^nC_2$ 2-shots; ${}^nC_3$ 3-shots and so on. The master shot along with the generated shots are referred to as *rushes*, on which editing or shot selection has to be performed. Each of the generated shots $S=\{s^i\}_{i=1}^{2^n-1}$ is parameterized by its center $c_t^i=\{x_t^i,y_t^i\}$ and width w_t^i at each video time instant $t=[1...T]$.

We use a Medium Shot (MS) or a Medium Close-Up (MCU) for generating single actor shots (1-shots). A medium shot captures a performer from head to waist, and a medium close-up goes from head to mid-chest. Smaller framings such as the MCU and MS present an actor in vivid detail and enable viewers to better focus on the character's actions and expressions. We use a Full Shot (FS) for multiple actor sequences (2-shots, 3-shots *etc*). A full shot for multiple performers is defined such that it captures each performer from head to toe. Larger framings such as the FS allow for a good deal of context around an actor to be included in the shot; these shots help establish the location where a subject is present and his/her interaction with the surroundings.

Shot Selection

Given the multiple rushes, the next step involves the selection of the shot that most vividly narrates the storyline at each time instant. The GAZED algorithm poses shot selection as a discrete optimization problem, which examines the *importance* of each of the multiple shots generated for every video frame, while adhering to cinematic principles like avoiding cuts between overlapping shots (termed *jump cuts*), avoiding *rapid shot transitions*, maintaining a *cutting rhythm*, *etc*. The importance of each shot at a given time is computed from eye gaze data collected using an eye-tracking device. Cinematic principles are modeled in the form of penalty terms. The final solution is obtained via a search for the optimal path through an editing graph [13]. For a scene with n actors, the editing graph consists of 2^n-1 nodes at each frame (time step) t, where each node represents a rush 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 (rushes) $S_t=\{s_t^i\}_{i=1}^{2^n-1}$ and the raw gaze data g_t^k corresponding to user k at frame t, our algorithm selects a sequence of shots $\varepsilon=\{r_t\}, r_t\in S_t$ for each frame t, minimising the following objective function:

$$E(\varepsilon)=\sum_{t=1}^{T}-ln(G(r_t))+\sum_{t=2}^{T}E_e(r_{t-1},r_t) \qquad (1)$$

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

Gaze potential

A well-edited video should engagingly convey the original narrative in a scene. Hence, it is important that the selection algorithm prefers shots which best showcase the *focal* scene events at any given time. To incorporate this idea into GAZED, we quantitatively measure the importance of each rush at every time instant. Previous works ([20],[13]) estimate the actions/emotions in a given shot by either relying on additional meta-data or bottom-up computational features, which do not account for high-level scene semantics which humans are sensitive to. Jain *et al.* [18] and Rachavarapu *et*

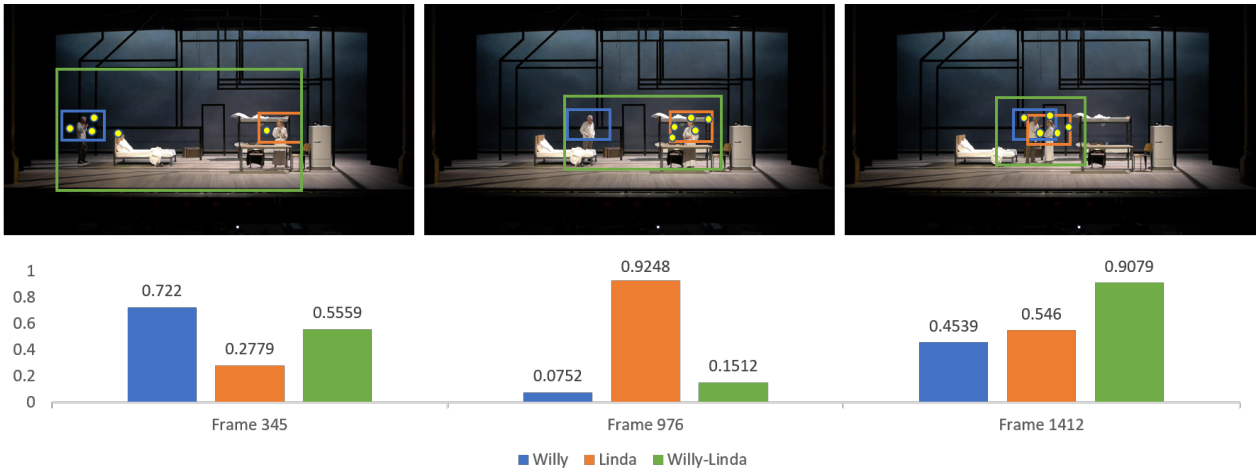

Figure 4. Computation of gaze potentials: The first row illustrates three different frames from the *master shot* and corresponding user gaze points (yellow dots). Generated rushes are denoted via colored rectangles (two *one*-shots and one *two*-shot in each frame). Second row presents gaze potential for each rush in the frame. Note that the gaze potential is higher for 1-shots when gaze points are concentrated (middle frame), and higher for the 2-shot when attention is dispersed (left and right frames).

al. [25] have shown that the gaze data recorded from users enables effective localization of focal scene events. We extend this idea and propose a novel method to calculate the gaze potential of each shot using gaze data from users.

The *point-of-gaze* g_t^k of user k at each frame t is defined by the (x,y) coordinates where the user is looking. To determine the gaze potential for each shot, we adopt a bottom-up approach, in which we first calculate the gaze potential of lower-order 1-shots, which capture individual actors. Gaze potential for shots with multiple actors (higher-order shots) are then computed from the gaze potentials of constituent lower-order shots.

We first compute the distance-to-center for all gaze points in a given frame t, considering 1-shots. The distance measure for a shot s_i at time t with frame center c_t^i is defined as $d_t^i = \sum_k ||c_t^i - g_t^k||_2$. If gaze points are clustered around a particular shot (actor), d_t^i will be small, while d_t^i will be larger if gaze is dispersed away from the shot center. We then use the distance measure d_t^i to compute gaze potentials for 1-shots as $G(s_t^i) = \frac{1}{d_t^i} / \sum_i \frac{1}{d_t^i}$. The denominator is summed only over the 1-shots. The function returns a higher potential (degree of importance) for shots with focused gaze clusters, and lower potential for shots with dispersed gaze points. Fig. 4 presents the computation of gaze potentials for an exemplar shot. We can observe that the 1-shot attracting greater user attention has higher gaze potential.

Now, consider two 1-shots s_t^a and s_t^b at frame t, and their gaze potentials $G(s_t^a)$ and $G(s_t^b)$. Gaze potential $G(s_t^{ab})$ of the *combined 2-shot* for actors a and b is given by:

$$G(s_t^{ab}) = G(s_t^a) + G(s_t^b) - |G(s_t^a) - G(s_t^b)| \qquad (2)$$

If the gaze is equally distributed among the two constituent 1-shots, the resulting 2-shot will have a high gaze potential, implying that the combined shot of the two actors has more value. Conversely, if the gaze is focused on only one of the 1-shots the 2-shot gaze potential will be lower, implying that

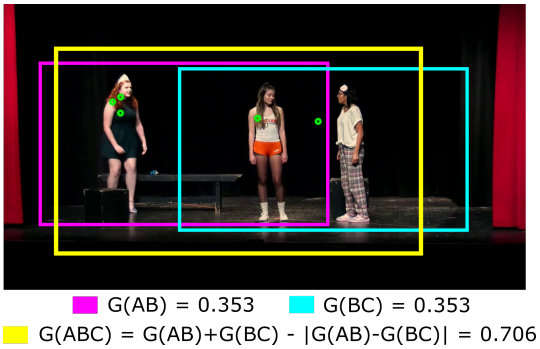

Figure 5. Actors are ordered from left to right as A,B,C. Gaze potential for the higher order ABC shot (or 3-shot) is computed from the constituent 2-shots AB and BC.

the combined shot is less valuable. A similar hierarchy can be followed for computing the gaze potential for higher-order shots. For instance, gaze potentials of two 2-shots $G(s^{ab})$ and $G(s^{bc})$ can be used to compute gaze potential of a 3-shot $G(s^{abc})$, when the actors appear on screen in the order a,b,c on moving left to right. Gaze potential computation is illustrated in Figures 4 and 5. Fig. 4 presents gaze potential computation for a 2-shot, while Fig. 5 shows computation of a 3-shot's gaze potential from two 2-shots.

Editing Cost

While gaze cues enable identification of the most important shots from the viewer's perspective, a shot selection methodology that is purely guided by gaze may not be optimal. This is owing to two reasons: (1) Even high-end eye-tracking hardware is prone to noisy measurements, especially over time; (2) If users constantly keep shifting their focus between two actors, frequently cutting between these two shots may negatively impact viewing experience of the edited video.

Therefore, an important objective of video editing is to ensure smooth shot transitions that sustain a continuous and clear narration. This is often achieved by relying on a well estab-

O(willy,linda,0) = 0 O(willy, linda, 0.22) = 1.1 O(willy,linda,0.41) = 1000

Figure 6. Shot-overlap cost computation: Figure shows three different master shots, and overlap cost is evaluated over switching between the two 1-shots in each case. The overlap cost (O) increases from 0 to 1.1 to 1000 as IoU for the 1-shots varies from 0 to 0.22 to 0.41. In essence, jump cuts are precluded when there is significant overlap between actor shots.

lished set of rules such as avoiding jump cuts, maintaining left to right ordering of on-screen characters *etc*. Another important element of film editing style is the cutting *rhythm*, which denotes the average time duration between cuts. Cutting rhythm can be manipulated to control the style and tone of the scene. We model these important elements of the video editing process as penalty terms in the objective function denoted by Eq. 1. We introduce three types of penalty terms, namely (a) shot transition cost (T), (b) shot overlap cost (O) and (c) cutting rhythm cost (R). The total cost for transitioning from shot r_{t-1} to shot r_t $(r_{t-1}, r_t \in S)$ is the cumulative sum of all these costs, *i.e.*,

$$E_e(r_{t-1}, r_t) = T(r_{t-1}, r_t) + O(r_{t-1}, r_t, \gamma) + R(r_t, r_{t-1}, \tau) \quad (3)$$

Shot transition cost

Frequent shot transitions may not allow the viewer enough time to comprehend the scene, and rapid cuts can disrupt the viewing experience. To avoid frequent transitions from one shot to another, we introduce the transition cost. Given two shots s_t^p and s_{t+1}^q across consecutive time steps, the transition cost is defined as

$$T(s_t^p, s_{t+1}^q) = \begin{cases} 0 & p = q \\ \lambda & p \neq q \end{cases} \quad (4)$$

where λ is the transition cost parameter. The above term penalizes cuts, and motivates shot sustenance over time.

Overlap cost

While cutting from one shot to another, the overlap between the two framings should be sufficiently low, otherwise it results in a cut that gives the effect of jumping in time, and is hence termed a *jump cut*. To prevent jump cuts, we introduce a overlap cost.

$$O(s_t^p, s_{t+1}^q, \gamma) = \begin{cases} 0 & \gamma \leq \alpha \\ \frac{\mu\gamma}{\alpha} & \alpha \leq \gamma \leq \beta \\ \nu & \gamma \geq \beta \end{cases} \quad (5)$$

Here, γ is the overlap ratio defined as the intersection over union (IoU) of two distinct shots s_t^p, s_{t+1}^q. The overlap cost is a piece-wise function where no cost is incurred when IoU (γ) is below threshold α, linear cost is incurred with IoU between thresholds α and β, and high penalty ν incurs when the IoU is greater than β. Figure 6 illustrates how overlap cost varies in different scenarios.

Rhythm cost

Frequency of cuts plays a key role in video editing. Shot length greatly effects how a scene is perceived by the audience. Slower rhythm or longer shots invoke a sense of stillness, which is often adopted in romantic scenes where emotions have to be portrayed. Faster rhythm or shorter shots are used when high energy or chaos has to be shown; this is commonly employed in action sequences. To control the cut rhythm, we introduce a cost based on shot duration. We define rhythm cost as:

$$R(s_t^p, s_{t-1}^q, \tau) = \begin{cases} \gamma_1 \left(1 - \frac{1}{1+\exp(l-\tau)}\right) & p \neq q \\ \gamma_2 \left(1 - \frac{1}{1+\exp(\tau-m)}\right) & p = q \end{cases} \quad (6)$$

where transition has been made from shot s_{t-1}^p to s_t^q, upon s_{t-1}^p prevailing for τ seconds. l and m are rhythm parameters and γ_1 and γ_2 are scaling constants. The rhythm cost is a case-wise penalty. When transition is made from a shot to itself $(p = q)$, we have a monotonically increasing penalty which becomes significant at $\tau = m$ seconds and accumulates over time. This motivates the introduction of a new cut. Once a cut occurs $(p \neq q)$, a monotonically decreasing function is triggered and adds a high penalty if another cut is introduced before $\tau = l$ seconds. The two terms together control the cutting rhythm.

Optimizing Edits

We solve Equation 1 using dynamic programming. The algorithm outputs a sequence of shots r_t for each frame t selected from the set of shots generated over time $\{S_t\}$. We build a cost matrix $C(r_t, t)$ (where $r_t \in s_t^i$, $t = [1..T]$) where each cell is computed with recurrence relation resulting from Eq. 1 :

$$C(r_t, t) = \begin{cases} -ln(G(r_t)) & t = 1 \\ \min_k [C(r_k, t-1) - ln(G(r_t)) + E_e(r_k, r_t)] & otherwise \end{cases} \quad (7)$$

The matrix is built along the time dimension. For each cell in the matrix, we compute and store the minimum cost to reach it. Once the matrix is built, we then perform backtracking to retrieve the sequence of optimal shots. We present the original wide angle recording during the first four seconds of the edited video as an *establishing shot* and only optimize over the remaining video frames.

Personalizing Edits

Parameters of GAZED are either inspired from literature or empirically set. For example, transitions with less than 20% ($\alpha = 0.2$) overlap are fluid and more than 40% ($\beta = 0.4$) overlap appear abrupt. The rhythm parameter m is set to 7 seconds based on the average shot length used in movies over two decades [9]. $\gamma 1 = 100$ and $\nu = 1000$ are kept relatively high, to avoid extremely fast cuts and to avoid jump cuts respectively. Most parameters are generic and fixed; however, some parameters like m and l can be varied for personalization (faster pace or allowing shorter shots). Since our algorithm is computationally efficient, it enables interactive content exploration. In the supplementary material, we illustrate example results of the same video edited at different pace/rhythm.

EXPERIMENTS

To examine if GAZED video editing, which incorporates both gaze information and cinematic principles results in a vivid, engaging and aesthetic rendering of a stage performance recorded with a static and distant camera, we performed a user study detailed below.

Materials and Methods

We selected a total of 12 stage performance videos for evaluation, five of them recorded at 4K resolution (3840×2160 pixels) and another seven in Full HD (1920×1080 pixels). These videos are wide-angle recordings, where a static camera captures the entire scene-of-interest, and the videos are devoid of any pan/cut/zoom operations. The 12 videos depict a rich variety of events such as *music concerts*, *dance performances* and *dialogue scenes* from theatrical acts. These videos were carefully selected so as to have diversity in the pace of the scenes (*e.g.*, slow and fast dance movements, long monologues and rapid conversations), periodically necessitating pans and cuts to captivate viewer attention and present them with a detailed view of the center of action. Cumulatively, the 12 videos lasted 12 minutes and 4 seconds, with individual videos ranging from 45–80 seconds.

The GAZED system envisages an editor/director reviewing a stage recording, and implicitly selecting the shots of interest over time via eye-gaze in lieu of manual selection. Expensive eye-trackers that accurately record eye-gaze at high speed ($\geq$ 1000 samples/second) exist today. However, we recorded gaze using a highly affordable ($< €200$) eye-tracker in this work to demonstrate that GAZED can generate professional edits even with cheap eye tracking hardware.

To compensate for tracking inaccuracies and sampling limitations of low-end eye trackers, five student users with normal or corrected vision were recruited for recording gaze. All five users were naive to the purpose of the study, and had not earlier watched any of 12 videos used in our experiments. Participants viewed the 12 videos re-sized to 1920×1080 pixels size on a 15.6 inch PC monitor. Viewers sat about 60 cm away from the screen, and ergonomic settings were adjusted prior to the experiment.

The *Tobii EyeX* eye-tracker having a 60 Hz sampling rate and gaming-level eye tracking precision was used to record gaze data. The tracker was calibrated via the 9-point method before recording began. MATLAB PsychToolbox [19] was used to develop the video presentation and gaze recording protocol. The 12 videos were presented to users in a fixed order for gaze recording.

Video Editing Baselines

As outlined previously, *video retargeting* and *video editing* are two different problems. As illustrated in Fig. 2, *video retargeting* only involves estimation of a center of the cropping window location (single x-coordinate) within the original video at each time. In contrast *video editing* requires generation of multiple virtual shots with desired compositions (parameterized by x, y and zoom value at each frame) followed by shot selection. Therefore, a comparison between the GAZED editing algorithm and state-of-the-art gaze-based video retargeting methods [18, 25] would neither be meaningful nor fair. We instead compare GAZED against four competent video editing baselines, namely, *random*, *wide*, *greedy gaze* and *speaker-based*, which are described below. Edits obtained via these strategies for exemplar recordings are presented in the supplementary video. For fair comparison, the *minimum shot duration* parameter l is set to 1.5 seconds, and the initial scene is established via presentation of the *master shots* (original video frames) for the first four seconds for all methods (except wide baseline).

Random

The Random (Ran) baseline is the weakest or the most context-insensitive video editing method. In this approach, one shot among the multiple rushes is arbitrarily selected for rendering every l seconds independent of the scene information.

Wide

The Wide baseline is motivated by the idea of video retargeting, and is in principle equivalent to the letterboxing method described in [18]. In the Wide shot selection strategy, the widest possible shot covering all stage performers is always preferred. This wide shot represents a zoomed-in version of the master shot, and denotes the smallest bounding box that covers all performers. As an illustrative example, the 'ABC' shot in Figure 3 denotes the Wide baseline.

Greedy gaze

The greedy gaze (GG) editing algorithm greedily selects the shot with *maximum* gaze potential to render at every time instant. In terms of shot selection, the main difference between the GG baseline and GAZED is as follows: while GAZED performs a *global* optimization by minimizing Equation 1 to derive the optimal sequence of shots $\{r_t\}$ to present over time, the GG baseline directly selects the presentation shot r_t based on the *local gaze potential optimum* at time t. In addition, since this editing strategy is solely guided by gaze information without adhering to cinematic editing guidelines, frequent switching of shots may occur which would negatively impact comprehension of the scene content and consequently, the viewing experience. To preclude the occurrence of transient shots, we impose a minimum shot duration of 1.5 seconds as specified by the the l parameter.

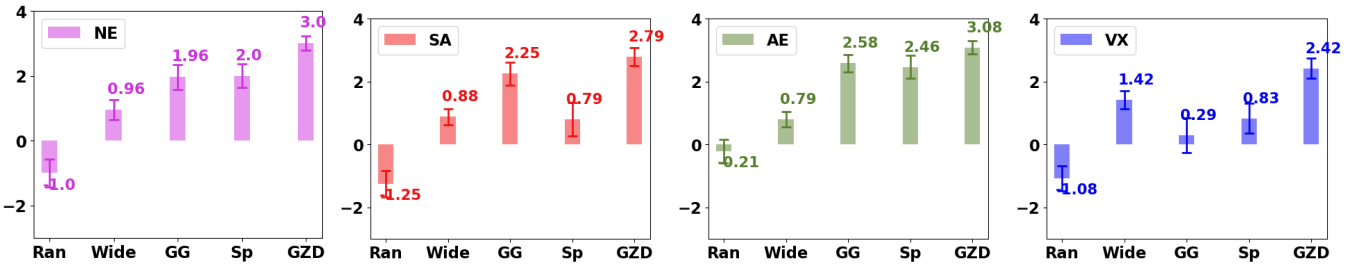

Figure 7. Bar plots denoting mean user ratings for the different evaluation attributes and editing methodologies. Error bars denote standard error of mean. Best viewed in color and under zoom.

Speaker-based

Speaker cues have proved useful for editing dialog-driven scenes [27, 20]. Ranjal *et al.* [27] and Leake *et al.* [20] favor selection of shots where the speaker is clearly visible. Our speaker-based (Sp) editing baseline similarly selects the 1-shot that best captures the speaker from among the rushes. For the purpose of this work, speaker information in each video was annotated manually. When more than one person speaks simultaneously, their combined shot is selected. The algorithm persists with its current selection till a change of speaker occurs. A minimum shot duration (specified by l) is enforced to avoid rapid shot transitions. If silence of more than 10 seconds is observed, the *wide* shot is selected for the next time instant.

User Study

To evaluate the video editing qualities of GAZED (GZD) against the above baselines, we conducted a psychophysical study with 12 users (different from those used for compiling gaze data) and the 12 video recordings described earlier. Edited versions of the 12 videos were generated via the four baseline plus GAZED strategies (for fair comparison, all parameters of GAZED were kept same across all videos). Upon viewing the original video, each user viewed the five edited video versions in random order on the same PC screen used for gaze recording. We designed the study such that each user viewed the original and edited versions of two stage recordings, so that six users cumulatively viewed all the 12 recordings. This resulted in a 12 (video types) $\times$ 2 (user ratings/video) $\times$ 5 (editing strategies) factor design.

Users were naive to the strategy employed for generating each edited version that they watched. On viewing each edited version of the original, users had to *compare* the edited version against the original, and provide a Likert rating on a [-5,5] scale for each of the attributes described below. These attributes were adopted from the retargeting study described in [25]– we note here that while video editing and retargeting are technically different, they are nevertheless designed to direct the viewer's attention onto *focal scene events* given rendering constraints. Therefore, psychophysical questions posed for evaluating video retargeting methods are also relevant for video editing. 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?*

Users were educated regarding these attributes, and about cinematic video editing conventions prior to the study. Users had to rate for questions (1)–(4), *relative* to a reference score of '0' for the original video. A *positive* score would therefore imply that the edited version was *better* than the original for the attribute in question, while a *negative* score conveyed that the edited version was *worse* than the original with respect to the criterion. User responses were collated, and mean scores were computed for each criterion and editing strategy over all videos (see Figure 7). Statistics and inferences from the user study are presented below.

Results and Discussion

A two-way balanced analysis of variance (ANOVA) on the compiled NE, SA, AE and VX user scores revealed the main effect of *editing strategy* on user opinions ($p < 0.000001$ for all four attributes), but no effect of *video type*. An interaction effect was also noted for AE scores ($p < 0.005$). We hypothesized that incorporating gaze cues plus cinematic rules for shot selection would generate an engaging, vivid and aesthetic edited output; Figure 7 validates our hypothesis as the GAZED editing strategy elicited the highest user scores for all the four considered attributes.

Examining individual attributes, *post-hoc* independent t-tests on NE scores revealed a significant difference between GAZED vs. Speech ($p < 0.05$), GAZED vs. GG ($p < 0.05$), GAZED vs. Wide ($p < 0.000005$) and GAZED vs. Random ($p < 0.000001$). Differences were also noted between Random and the remaining baselines ($p < 0.000005$ for Speech, $p < 0.00001$ for GG and $p < 0.001$ for Wide), and between Speech vs. Wide ($p < 0.05$). These results cumulatively convey that carefully compositing shots that provide a close-up view of the focal actor(s) and action(s) is critical for effective narration. The GG, Sp and GAZED strategies that are designed to focus on actors and actions deemed important via speech or gaze-based cues achieve better scores than the Ran and Wide baselines which can only achieve incorrect/inefficient framing of the scene characters. The Random baseline which selects shots independent of scene content performs worst, while a

vivid presentation of gaze-encoded salient scene events via GAZED is perceived as the best narrative.

With respect to conveying scene actions, GAZED performs significantly better than Speech-based ($p < 0.005$), Wide ($p < 0.00001$) and Random ($p < 0.000001$). Also greedy gaze achieves significantly higher scores than both Speech ($p < 0.05$) and Wide ($p < 0.005$). The Wide and Sp baselines perform similarly for this criterion, while Ran performs the worst as expected. These results point to the fact that speaker cues may not be as effective for conveying focal events in stage performances; this is quite plausible as a particular actor may assume the role of a narrator, while another performs the actions of interest. Alternatively, one performer may verbally introduce co-performers to the audience, in which case the Sp baseline would still focus on the introducer instead of the introducee. In such cases therefore, eye gaze is more accurate at capturing events and actors of interest than speech. The Wide baseline, which captures the entire scene context at all times and thereby can only present low-resolution views of performers to viewers, performs similar to Sp. The GG strategy which vividly captures events of maximal interest at each time instant is nevertheless effective at conveying scene actions and performs second best.

Considering how the different editing strategies convey actor emotions, we observe that the GG, Sp and GAZED methods perform fairly similarly. These three approaches perform significantly better than the Wide and Ran baselines ($p < 0.0005$ for all pairwise comparisons). Wide still performs better than Random editing ($p < 0.05$). These observations can be explained as follows– wherever appropriate, the GG, Sp and GAZED editing methods capture the speaker or focal actor in the scene as a close-up shot, which enables an effective rendering of facial emotions to the viewer. The Wide baseline captures all scene actors at each video frame, and can therefore only present facial movements at (relatively) low-resolution to viewers.

Cinematic editing principles constitute a key component of the GAZED shot selection process– incorporating them in the objective function defined by Equation 1 is critical for producing a seamless, smooth, visually engaging and aesthetic edited output. Specifically, including the shot transition and overlap costs would minimize the number of *cuts* in the edited video, and frequent cuts can distort viewer comprehension of the scene context (*e.g.*, knowledge of scene actor locations), and thereby negatively impact viewing experience. Note from the editing baseline descriptions that none of them incorporate these editing principles for shot selection, except for enforcement of a minimum shot duration of 1.5 seconds via the l parameter to preclude transient shots.

We hypothesized that the incorporation of cinematic editing rules in the shot selection framework, coupled with the capturing of focal scene events would maximize viewing experience of the edited video. Consistent with our expectation, GAZED scores highest among the five methods for viewing experience, and performs significantly better than the Sp ($p < 0.01$), GG ($p < 0.005$), Wide ($p < 0.05$) and Ran ($p < 0.000001$) baselines. Wide performs second best scoring marginally better than GG ($p = 0.0752$) and significantly better than Ran ($p < 0.00001$), but only comparable to Sp. The superiority of Wide over GG and Sp can be attributed to the fact that the entire scene context is always visible to the viewer at all times in this editing strategy; Both Sp and GG are designed to cut routinely to focus on the (sometimes incorrectly) perceived action of interest in the scene, and such frequent cutting can lead to a jarring viewing experience. In contrast, Wide presents a relatively consistent framing of the scene to viewers at all times, resulting in a *smooth* edited video. Finally, both Sp and GG elicit a better viewing experience than Ran ($p < 0.05$ in either case).

SUMMARY AND CONCLUSION

This work presents the GAZED framework for automatically editing stage performance videos captured using a single, static, wide-angle and high-resolution camera employing user gaze cues. As human eyes are known to be sensitive to focal scene events, GAZED translates user gaze data into *gaze potentials*, which quantify the *importance* of the multiple rushes generated from the master shot via the shot generation process; the gaze potential serves a natural measure for guiding *shot selection*, and the shot selection process directly impacts the quality of the edited video. While GAZED shot selections are primarily driven by user attention, cinematic editing principles such as *avoiding jump cuts*, *precluding transient shots* and *controlling shot-change rhythm* are also modeled within an energy minimization function guiding the shot selection process. These cinematic rules facilitate the generation of a smooth, vivid, visually engaging and aesthetic output as confirmed by user opinions compiled from a psychophysical study.

Our user study compares the GAZED framework against four editing baselines, namely, *Random*, *Wide*, *Greedy Gaze* and *Speech-based*. While the Random baseline is employed to demonstrate that editing strategies should be guided by scene content, others are inspired by prior literature. The Wide baseline is motivated by letterboxing for video targeting; the speech baseline is included to mimic [27] and [20] for editing stage recordings. Greedy gaze-based editing is used to showcase how modeling cinematic editing rules can generate a smooth and aesthetic edited video.

The user study reveals salient aspects of the different editing strategies. GAZED performs best, while Random performs worst with respect to four essential attributes of the edited video. Speech-based editing scores highly with respect to conveying actor emotions, but is perceived as ineffective at capturing focal scene actions; this observation reveals that speech cannot effectively reveal the salient actors/actions in stage performances. GG editing conveys both scene actions and action emotions convincingly, validating the utility of gaze cues for encoding focal scene events. However, it does not incorporate cinematic rules, which results in a poor viewing experience. Conversely, the Wide baseline scores low with respect to the NE, SA and AE attributes as it does not focus on individual actors. However, it consistently frames the entire scene resulting relatively smooth edited output, which elicits a moderate viewing experience.

The computational efficiency of the algorithm allows for efficient exploration and creating personalized edits. GAZED edits a minute long video (24 fps) with three performers in 1.5 seconds. The computation only grows linearly with video length and editing a 30 minute video with three performers take s about 49 seconds. With four actors GAZED takes about 5 seconds for editing a minute long video and 162 seconds for a 30 minute video. These computations are done a PC with 7th generation Intel 2.7 GHz i5 processor and 8GB RAM. The exhaustive list of possible shots will grow exponentially with increasing number of actors, which will significantly increase the computation. However, not all these shots may be useful to convey the scene and the user (editor) can pick the relevant set of shots to select from and generate the final editing.

Limitations of the current GAZED implementation include (1) the inability to *pan* between shots as the shot selection process only induces *cuts* in the edited video; (2) being able to achieve only a single value of *zoom* corresponding to a medium/medium close-up compositing for the 1-shots, and (3) noise induced by the eye-tracking hardware which can be largely alleviated through the user of high-end trackers, and recording gaze in bursts to enable drift correction. Achieving the capability to pan between shots, and gradually zoom on faces would be the focus of future work.

Nevertheless, the utility of GAZED can be appreciated with an understanding of the complexity of the video editing process, which is extremely tedious and effort intensive. In this regard, speech-based editing [20] which (a) is specific to dialog scenes, (b) requires the film script as input and (c) takes about 110-217 minutes for scene pre-processing, still represents a utility tool. Similarly, the GAZED system can be used by editors to inexpensively and quickly obtain the first-cut edit of performance videos and also explore different editing styles (*e.g.*, by varying the rhythm parameter), so that they can invest more time and effort in devising refinements and creative edits where necessary.

ACKNOWLEDGEMENT

This work was supported in part by Early Career Research Award, ECR/2017/001242, from Science and Engineering Research Board (SERB), Department of Science & Technology, Government of India. Special thanks to Remi Ronfard, Claudia Stavisky, Auxane Dutronc and the cast and crew of 'Death of a salesman'.

REFERENCES

[1] Ido Arev, Hyun Soo Park, Yaser Sheikh, Jessica Hodgins, and Ariel Shamir. 2014. Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 81.

[2] Daniel Arijon. 1976. Grammar of the film language. (1976).

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

[4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*.

[5] Christine Chen, Oliver Wang, Simon Heinzle, Peter Carr, Aljoscha Smolic, and Markus H. Gross. 2013. 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*.

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

[7] Jianhui Chen, Lili Meng, and James J Little. 2018. Camera Selection for Broadcasting Soccer Games. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 427–435.

[8] David B Christianson, Sean E Anderson, Li-wei He, David H Salesin, Daniel S Weld, and Michael F Cohen. 1996. Declarative camera control for automatic cinematography. In *AAAI/IAAI, Vol. 1*. 148–155.

[9] James E Cutting and Ayse Candan. 2015. Shot durations, shot classes, and the increased pace of popular movies. (2015).

[10] Shinji Daigo and Shinji Ozawa. 2004. 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*. ACM, 444–447.

[11] Petr Doubek, Indra Geys, Tomáš Svoboda, and Luc Van Gool. 2004. Cinematographic rules applied to a camera network. In *Omnivis2004: The fifth Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*. Prague, Czech Republic: Czech Technical University, 17–29.

[12] David K Elson and Mark Riedl. 2007. A lightweight intelligent virtual cinematography system for machinima production. (2007).

[13] Quentin Galvane, Rémi Ronfard, Christophe Lino, and Marc Christie. 2015. Continuity editing for 3D animation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[14] Vineet Gandhi, Remi Ronfard, and Michael Gleicher. 2014a. Multi-clip video editing from a single viewpoint. In *Proceedings of the 11th European Conference on Visual Media Production*. ACM, 9.

[15] Vineet Gandhi, Remi Ronfard, and Michael Gleicher. 2014b. Multi-clip video editing from a single viewpoint. *Proceedings of the 11th European Conference on Visual Media Production - CVMP 14* (2014). DOI: http://dx.doi.org/10.1145/2668904.2668936

[16] Li-wei He, Michael F Cohen, and David H Salesin. 1996. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 217–224.

[17] Rachel Heck, Michael Wallick, and Michael Gleicher. 2007. Virtual videography. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 3, 1 (2007), 4.

[18] Eakta Jain, Yaser Sheikh, Ariel Shamir, and Jessica Hodgins. 2015. Gaze-driven video re-editing. *ACM Transactions on Graphics (TOG)* 34, 2 (2015), 21.

[19] M. Kleiner, D. Brainard, and D. Pelli. 2007. In *Perception ECVP Abstract Supplement*, Vol. 36.

[20] Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational video editing for dialogue-driven scenes. *ACM Trans. Graph.* 36, 4 (2017), 130–1.

[21] Christophe Lino, Mathieu Chollet, Marc Christie, and Rémi Ronfard. 2011. Computational model of film editing for interactive storytelling. In *International Conference on Interactive Digital Storytelling*. Springer, 305–308.

[22] Billal Merabti, Marc Christie, and Kadi Bouatouch. 2016. A Virtual Director Using Hidden Markov Models. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 51–67.

[23] Walter Murch. 2001. *In the blink of an eye: A perspective on film editing*. Silman-James Press.

[24] Hyun S Park, Eakta Jain, and Yaser Sheikh. 2012. 3d social saliency from head-mounted cameras. In *Advances in Neural Information Processing Systems*. 422–430.

[25] Kranthi Kumar Rachavarapu, Moneish Kumar, Vineet Gandhi, and Ramanathan Subramanian. 2018. Watch to Edit: Video Retargeting using Gaze. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 205–215.

[26] Subramanian Ramanathan, Harish Katti, Raymond Huang, Tat-Seng Chua, and Mohan Kankanhalli. 2009. Automated Localization of Affective Objects and Actions in Images via Caption Text-Cum-Eye Gaze Analysis. In *ACM International Conference on Multimedia*. 729–732.

[27] Abhishek Ranjan, Jeremy Birnholtz, and Ravin Balakrishnan. 2008. Improving meeting capture by applying television production principles with audio and motion detection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 227–236.

[28] Ramanathan Subramanian, Divya Shankar, Nicu Sebe, and David Melcher. 2014. Emotion modulates eye movement patterns and subsequent memory for the gist and details of movie scenes. *Journal of vision* 14, 3 (2014), 1–18.

[29] Yoshinao Takemae, Kazuhiro Otsuka, and Naoki Mukawa. 2004. Impact of video editing based on participants' gaze in multiparty conversation. In *CHI'04 extended abstracts on Human Factors in Computing Systems*. ACM, 1333–1336.

[30] Jinjun Wang, Changsheng Xu, Engsiong Chng, Hanqing Lu, and Qi Tian. 2008. Automatic composition of broadcast sports video. *Multimedia Systems* 14, 4 (2008), 179–193.

[31] Cha Zhang, Yong Rui, Jim Crawford, and Li-Wei He. 2008. An automated end-to-end lecture capture and broadcasting system. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 4, 1 (2008), 6.