Eurographics / ACM IMX Joint Workshop WICED x Cinemotions (2023)
H.-Y. Wu, S. Jolly, A. Bruckert and L. Lévêque (Editors)

Unsupervised Scouting and Layout for Storyboarding in Movie Pre-production

Will Kerr, Tom S. F. Haines and Wenbin Li

Department of Computer Science,
University of Bath, United Kingdom

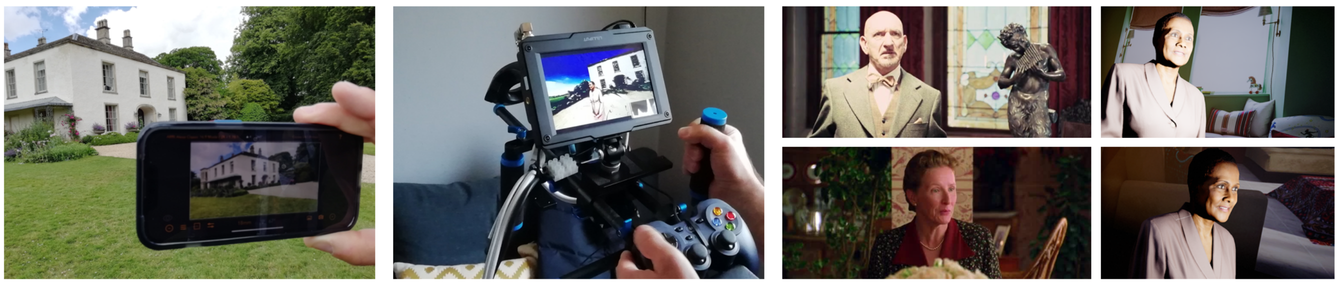


Figure 1: *Hybrid scouting workflow and system output. L to R: 1) traditional scouting on location; 2) virtual scouting using tracked cameras and 3D models; 3) reference film image examples courtesy of [SKSB21]; 4) system output showing automatic staging when tasked with imitating visual features of the previous column*

Abstract

We introduce a system that analyses film content to provide expert camera pose recommendations to cinematographers, so they can rapidly experiment with shots in a virtual space. Shot experimentation to produce storyboards is a crucial preparation step prior to filming, with the time and skill required to create expert storyboards high. Additionally, the convergence between the desired aesthetic and what the physical environment allows is traditionally explored by scouting on location, which can also be costly. We hypothesise that autonomous scouting in the virtual world by moving cameras, actors and lights to achieve specific composition traits is efficient and supports creativity. Our system automates aspects of scouting, storyboarding, and staging without being directed by limiting rule-based cinematography techniques, and instead by data-driven unsupervised clustering.

CCS Concepts
• Computing methodologies $\rightarrow$ Image processing; Computational photography; • Human-centered computing $\rightarrow$ Human computer interaction (HCI);

1. Introduction

Film pre-production develops a director's vision of a script. The process is creative with high requirements for time, skill and resources. A critical deliverable from this stage is a storyboard, which supports planning for efficient filming on shooting days. Creating a storyboard can be summarised into two main tasks:

Conceiving shots, whereby the professional team imagine how each section of script will translate into the 2D frame captured. Experience, imagination, script, and previous film content may all contribute to the shot decision.

Realising a storyboard to document the aforementioned conceived shot. Traditionally crafted by hand on paper, technological advances have provided the ability to transfer or create storyboards using virtual environments. They allow quick, low cost experimentation without the need to physically be on location, and offer greater realism than hand drawn sketches. However, the task of placing virtual cameras, actors and lights is currently predominantly manual.

These two tasks present a combined opportunity for automation. Firstly, for an appropriate shot-type to be recommended, and secondly, for that to be realised in the virtual environment by solving for camera pose. Our hypothesis is that stylised and professional-level camera poses can be recommended to cinematographers in the virtual space, as an aid in the shot-planning stage. We believe that imitating shot composition from previous film content is a valid and robust prior which can be guided from sources appropriate to the story. This will support the creativity of film making without constraining cinematographers, enabling a convenient and guided method for explore environments to produce shots in desired style.

1.1. Contributions

Aligned with the tasks of conceiving shots and realising a storyboard, as described above, our contributions are as follows:

1. A system that analyses existing film content to identify image features characteristic of shot styles using clustering.
2. Unreal Engine tools to enable the optimisation of camera, actor, and light positions according to the above.

Through a user study we show user preference of system parameters, demonstrating a step towards autonomous storyboard creation.

2. Background

Virtual environments offer the flexibility to develop ideas without the need to travel to and manipulate physical environments. They can offer enhanced visualisations for planning real shoots, and actual footage for a full computer generated imagery (CGI) movie or game. Recent surveys [CON08; Ron21] discussed the history and advancements in virtual cinematography. The latter splits film making into decoupage (the choice of camera shots), mise-en-scene (the staging of events in front of the camera), and montage (the editing process), and how researchers have tackled each aspect in the application of games and virtual cinematography. Several complete autonomous cinematography environments have been proposed, integrating virtual environments with cinematographic control, as discussed below. Some have used Finite State Machines, geometric limits, and hard cinematographic rules to produce media based on machine readable scene descriptions as input [HCS96]. Refining the trajectories to mimic real-life camera hardware like rails has been considered [HAB17].

Virtual pose solvers with textual direction allow a director to describe a shot using the language of cinematography with which they are familiar e.g. close-up, medium shot, dolly, zoom [MBC14; Gal15; GR17; RGBM22; LCL18]. This requires upstream translation of the script into shot descriptions, either manual or automatic.

Pose solvers by feature imitation depend on the data and visual features they are imitating. They may analyse camera motion from real or virtual shots [SDM*14] to mimic a specific movies style. This idea may be extended to also extracting the poses of actors, such that relative offsets between actor(s) and camera are recreated [JWW*20]. Goals based on camera movement, framing and head orientation have been demonstrated [YSP*21] to translate well into a virtual environment.

Virtual camera fine-motion controllers have been developed to offer fine-grain motion control of the cameras include imitating camera shake [AMJ*20; KRE*14] and subject zoom via image cropping [WAE*20]. Whilst these can improve realism, they rely on an rough established relationship between camera and actor within the scene, or else be applied to existing camera trajectories.

Many **optimization methods** have been applied to the task of exploring possible camera poses to minimise some objective. In particular, particle swarm optimization (PSO) [KE95] has been demonstrated to improve six degrees of freedom (DoF) and lens field of view (FoV) solvers when compared against grid search methods [BDER08]. They can incorporate cinematographic principles such as rule of thirds, object position, colour palette [BR14], and provide the ability to solve temporal trajectories for moving cameras [PHPU16]. Neural techniques have been investigated to propose cut scenes in the style of a nominated director [EG22] after being trained on hand-annotated clips from those directors, and through the use of Reinforcement Learning (RL) [YYWR22].

Automatic storyboard creation considers the workflow of previsualization. While preexisting tool may inconveniently realise the desired outcome new tools should benefit from a user-centric workflow design [MVWM19]. For the task of building a sequential storyboard from a script Anyi et al. [AXY*23] has investigated a professional shot discriminator, that ranks shots based on their quality. With ResNet-50 as the basis for analysing individual frames the model is trained to discriminate between shots created by professionals and random shots, on the principle the random shots will be of low quality. Their system uses a camera script (a textual description of camera motion) to propose shots as renders from a game environment. The discriminator then identifies the best proposals.

To expand on the state of the art we combine elements from Barry et al. [BR14] and Anyi et al. [AXY*23] with unsupervised clustering to generate stylistic targets for camera poses. Our novel contribution is to use unsupervised shot classification to drive composition-based virtual camera pose solvers (an autonomous virtual 'scouting' process) in order to provide creative, efficient and convenient storyboard recommendations to new filming tasks.

3. Methodology

Below describes firstly the constraints identified to manage project scope, followed by requirements. This is followed by a step by step description of our approach.

Constraints 1) All examples in related work mentioned in section 2 consider the framing of humans (real, or virtual models) as the main subject. To back this decision, a set of Hollywood films was analysed for presence of face. Figure 2 shows the proportion of frames with various numbers of faces, throughout a distribution of 213 films. The high proportion of zero faces detected is likely due to false negative face detector results due to scale, codec artifact, or motion blur within the frame. We therefore focused on solving for images with only 1 actor in frame. This also supported a step-wise approach, in that starting with the simple configuration of a single actor simplifies the solving logic; future work may enable additional actor placement. As such, face detection becomes an important feature in our work, developed as described in section 3.1. 2) Only consider the analysis of high-quality film - e.g. Hollywood grade films, and not user-generated content from social media. This is to ensure learnt priors are guided by content more likely to be well produced, and provides a more likely recognisable reference to users for scenarios where style from a specific film is desired. 3) Analysed media was not required to be in native resolution, since higher resolutions (full HD or 4K) proved slower to analyse without clear benefit.

Requirements defined to achieve the goals set out in section 1.1,

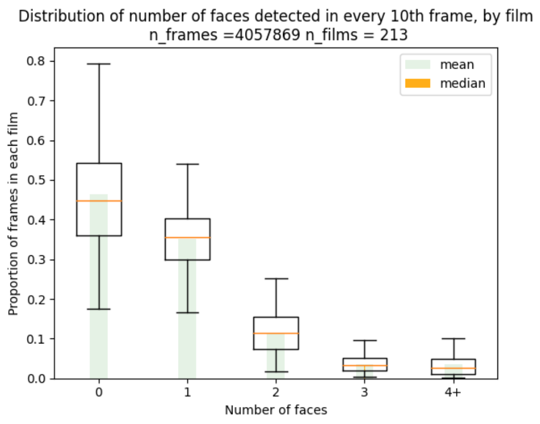


Figure 2: *Distribution of number of faces detected in every 10th frame of 213 films. The high proportion of frames with zero faces is suspected to be face detector failures on small scale, codec compressed, or motion blurred frames.*

were as follows: 1) the ability to extract relevant image features plus a method of summarising these feature using clusters; 2) the ability to externally control the virtual environment camera, actors and lights, plus enable autonomous exploration of those elements when tasked with an image-feature based goal; 3) wrap the above components into a user interface. The development of this system is described in five stages as follows:

3.1. Image Analysis

Many image features could be analysed, but which ones? Maybe the most complete set of features are those which cinematographers consider when planning and executing shots. Clearly subject framing is important since without the subject in frame, the intent of the shot is lost. Many other elements are considered, and are the subject of extensive analysis in the field of film studies which are not covered here. Another set of features are ones which have been shown to be related to induced emotion in viewers. According to existing work in image feature analysis as an affective cue in film, table 1 shows the features most associated with providing affective emotion. Although these cannot be assumed to correlate to how cinematographic style is encoded within film, it nonetheless provides a guide to important features when considering image composition. Yet another set could be features shown to show promise when applied to clustering video, [MZBB16], or image Content-Based Image Retrieval (CBIR) tasks whereby rapid and robust comparison is required [SC97; CWK03]. Considering these options and figure 2, it was decided to progress with face, colour, and lighting key (brightness) analysis, with acknowledgment that additional features would present interesting lines of research for this work. The following describes the implementation of such analysis tools.

Face: The requirements for a face detector appropriate for this application required speed, accuracy on large and small faces, and the ability to feed downstream face angle detection models. Several models were evaluated for speed and accuracy. Inference durations for single face detection on 960 × 540. images were discovered as follows: MTCNN (60mS), RetinaFace ResNet-50 (33mS), RetinaFace MobileNet (28mS), YOLOv5 nano (18mS), MediaPipe BlazeFace (21mS for 16 images). Ultimately a batched and multi-threaded YOLOv5 model was used, giving a good balance of performance speed and accuracy on small faces, resulting in 16 images being inferred in 86mS. **Angle:** Face angle estimation (yaw, pitch, roll) was implemented using an FSA-Net [YCLC19] model, utilising the bounding box and keypoints detected in the YOLOv5 detector. **Colour:** Basic statistics and histograms were gathered from RGB and HSV colour space of both face and background elements. Downsampling and masking were employed for speed improvements. An estimated 'face+body' mask was implemented to avoid corrupting background measurements with the body of an actor.

Visual Feature	Rank
Camera movement	1
Colour energy, actor movement	2
Brightness, shot length	3
Shot scale, colour separation	4
Lighting key, shot coherence	5
Rule of thirds , focus, tonal range, horizon position, nose / head room, background clutter, face angle, lens	6
Vanishing point, blocking, texture map, aerial diffusion, line distribution	not scored

Table 1: *Summed and ranked visual features for affective computing, from [XJLD08; CL13; LBS*20; TLT20; RSS05; BSB*19; Wan06; Kan03; HX05]*

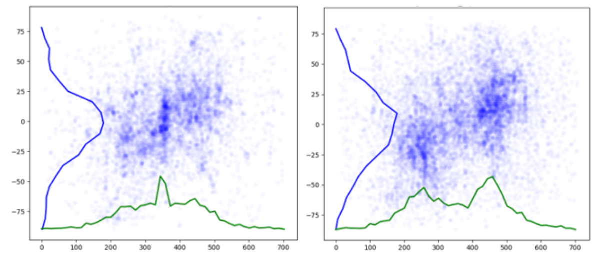


Figure 3: *Scatter plots showing distribution of face centre horizontal pixel position in frame (x) against face yaw angle degrees (y) throughout 2 films: (L) Ouija: Origin of Evil and (R) Murder on the Orient Express*

3.2. Unsupervised Clustering

Following the desire to not require human annotation and guidance for style identification, an unsupervised approach was taken with the aim of identifying characteristic shot types within the source material. The image analysis previously described was applied to every 10th frame from 213 movies. The following work describes the exploration of that feature data-set in an unsupervised fashion.

General distributions of individual films analysed with particular face features demonstrated that distributions of individual features can be unique between films, as in figure 3. This holds promise for style imitation in new filming tasks, but these distributions alone do not tie correlations between image features or allow individual shots to be represented.

User defined feature filters may offer the ability to drill down into finer resolution of shot types, and after implementing a simple user interface it became apparent the limitations of single feature distribution analysis still apply. Additionally, this process required additional human intervention, and also knowledge about what and how to apply the filters for interesting results, both of which are not aligned with the goal of reducing burden on the user.

Multi variate clustering was considered to accommodate multiple input features, separate out unique clusters, and probabilistically synthesise parameters according to the discovered clusters to be used in downstream pose solving tasks as in section 3.4. By considering various types of clustering from [Xu15] ,the distribution / model based method of multi-variate Gaussian Mixture Model (GMM) [DHS73] analysis was implemented , such that a nominated subset of features could be analysed for the purpose of cluster representation. With the decision about which features to analyse in the first place, the decision about which features to present to GMM also needed consideration, which will be discussed in section 3.4.

For the purposes of development however, 5 scalar features relevant to face pose (x, y, height and pan and yaw rotations) were used to demonstrate the complete GMM process (see figure 4), as applied to a single film. GMMs with varying number of components (1,2,3,5,8,40) were produced. The Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) calculations provided guidance for optimum component count. Increasing component count eventually provides diminishing returns, due to the compute cost of calculating multiple GMMs , and potential overfitting of the data. Too few components may group clusters which otherwise be useful to separate.

Once a GMM was constructed, it was possible to synthesise multi dimensional data by either sampling from the entire GMM, or by sampling from a nominated single Gaussian component. The latter provided the ability for parameters representing individual shot-types to be generated.

Several component importance metrics were calculated, including component weight, number of original data points covered by the component and a variance-normalised point count. Additionally, inspired by [HO07] and [JECJ07], a pair-wise distance metric between all multi-variate Gaussian components using a Kullback–Leibler divergence (KL) was calculated. The resulting confusion matrix column and row means could be used as a guide for particularly separate clusters.

The result of GMM analysis was twofold. Firstly as already discussed, sampled data and component parameters were stored for use later in the downstream pose solver as described in 3.4. Secondly, the user was presented with a representations of the clusters in either the form of still images representing maximum probablity for each GMM component (i.e. the 'best' frame for each component), or as a time-series (figure 4), allowing the user to understand how clustered shot-types were used throughout the duration of a film section. It should be noted these frames were only of shots containing 1 face, and at a frame skip rate of 30 frames, so only present a subset of the movie throughout time.

Discussed so far was the application of fitting a GMM to multivariate *scalar* parameters. As mentioned, some feature analysis produces a *histogram* per frame (e.g. colour), and although each of the bins (32) could be presented as a scalar dimension to the GMM fitting process, we were conscious of the increased computation and stability risk with this approach. As such, a nested GMM approach was taken to represent colour histograms, in a technique to reduce dimensionality down in a 2 step process. Each major GMM component was instead presented with mean and variance scalars representing 3 Gaussian distributions for each colour channel, such that an original colour distribution could be re-sythesised in the optimization stage to optimise against. This offered a convenient representation ($3 \times 2 \times 3$) of the otherwise high dimensional (32×3) colour histograms for colour channels.

		Streams retrieved					
# cams	fps	1	2	4	8	16	32
1	80	53 (17)					
2	80	54 (17)	51 (17)				
4	52	87 (17)	89 (17)	85 (17)			
8	30	137 (17)	134 (17)	136 (17)	157 (32)		
16	17	197 (16)	217 (16)	218 (16)	250 (32)	316 (62)	
32	8.5				511 (32)	496 (60)	650 (130)

Table 2: *Total duration in mS for moving cameras and retrieving the new images (parenthesis) from Unreal Engine, using the HTTP API and Spout plugin*

3.3. Pose Control

To build a system in which exploration can be automated, the following objectives were set: **a)** - use a well-known 3D virtual environment that allows potential for future projects to be easily integrated, **b)** - enable external control of key components in the scene - camera, actors and lighting parameters, **c)** - provide efficient control and image I/O, **d)** - allow solving routines to be integrated into the external control component. Each was approached as follows:

Considering previous work and available applications, Unreal Engine 4 (UE4) was selected - being well supported, allowing multiple cameras and timeline control, external API control, and can render camera images to external programs. It is also familiar in the virtual production community, lending itself well to workflow integration. A camera control layer has been implemented for Unreal Engine in "CineAirSim" [PCM20], however greater flexibility was required for our task. The communication mechanisms implemented were the UE4 API which includes HTTP and websocket control of engine parameters, and a Spout [SPO23; OFF23] camera renderer plugin which allows shared GPU memory access to camera images from UE4 to Python. All camera translation, rotation and intrinsic parameters of focal length, aperture and focus distance were controlled through Python. Control of actor and lighting position and intensity was also implemented. Multiple cameras were setup in UE4 which gave rise to aggregate speed improvements as show in table 2, realised through a single batch camera move API call, and multiple Spout receivers. Due to the nondeterministic timing of the UE4 HTTP API, it was necessary to implement a watermark system for positive validation that the image retrieved came from correct camera in correct position.

3.4. Pose Optimization

Once external control over the environment was proven to be robust to fast consecutive operation, the next step was to implement optimized based convergence of pose. For this, several custom implementations were made for the key components as follows:

The objective function in this application was set to move objects as per new co-ordinate parameters, receive and analyse the

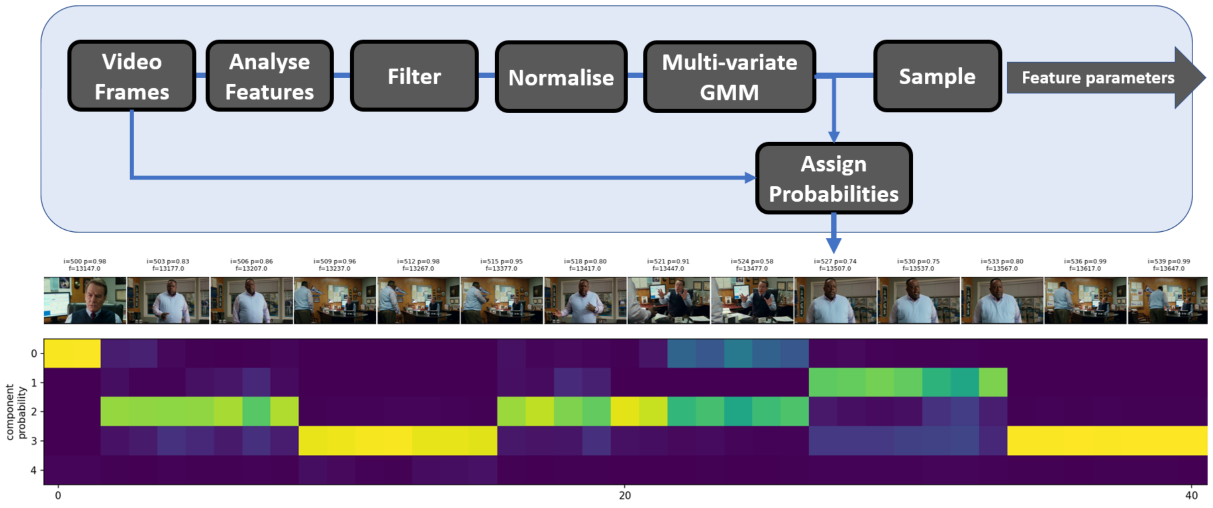
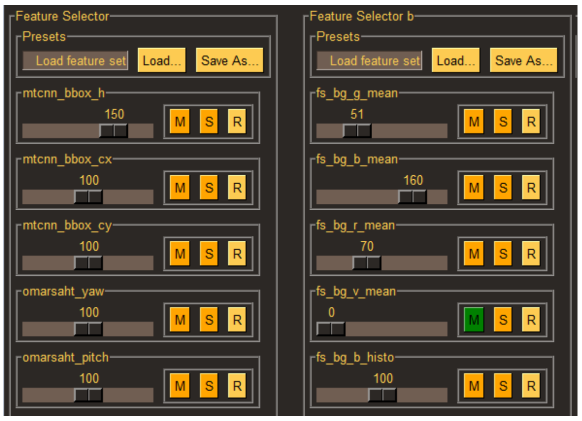


Figure 4: *(Upper) System architecture for multivariate GMM clustering and probability assignment from source film. GMM component sampling provides characteristic shot-type parameters which are used downstream in the virtual pose solving process. (Lower) shows sampled component probabilities through time from film Why Him? to visualise shot type progression. This example shows a 5 component GMM, fit to 6 face feature (5 for position and angle, 1 for brightness)*

new images according to nominated features, perform some comparison to the parameter goals, and return a combined loss score for that iteration. Since the system was based on a multi-camera setup, the optimization function was designed to allow multiple pose parameters to be proposed, and return back multiple loss scores. The objective function made use of the control methods already described in section 3.3 to move cameras, actors and lights.

The loss function , the next key component considered, needed to perform the image analysis (as per 3.1) of each new camera pose, compare the extracted feature parameters against the goal parameters, and combine to produce a single scalar loss value for each image. In general, the loss function could be setup to test for any nominated set of image features and weights , and compared against a reference set of parameters covering the same set of features. Combining individual feature scores accorded to either L_{inf}, L_1 or L_2 norms. In the case where features could not be detected (like a face in the frame), worse-case values were used in default for that feature. Figure 5 shows some feature and weight settings, as editable by the user for 2 of the optimization stages which are described in the next section.

Optimization methods. Recalling the set of features analysed being face, colour and brightness, the optimization was initially tasked with moving cameras around a fixed actor, in order to frame the faces as per the goal parameters. Initialization poses were calculated to have cameras at least pointing in the general direction of the actor, since with no face present on-screen, the optimizer has not hint about which direction should result in a lower loss score (e.g. to find the face again). Initially after testing minimization techniques (Nelder-Mead, BFGS, Powell, Newton-CG), it was discovered these suffered from sequentially searching each enabled camera dimensions, often failing to converge on a believable result. As such and as inspired by previous work [BDER08], a Particle Swarm Optimization approach was taken. This provided good coverage within the allowable search bounds, generally reducing a 48 particle swarm variance after 40 iterations using 8 virtual cameras,



mtcnn_bbox_h 150
mtcnn_bbox_cx 100
mtcnn_bbox_cy 100
omarsaht_yaw 100
omarsaht_pitch 100
fs_bg_g_mean 51
fs_bg_b_mean 160
fs_bg_r_mean 70
fs_bg_v_mean 0
fs_bg_b_histo 100

Figure 5: *VACE loss function feature adjustment user interface. Features can be muted, solo'd, and have their weights adjusted, affecting the loss fucntion summation.*

and providing believable results for the final camera pose image. Experimentation in adding more features to solve against like background colour highlighted the problem with a single static actor, in that it provided only a limited set of possible backgrounds. Therefore it was proposed that additional optimization steps could be used after the initial camera-to-actor (face framing) pose solving.

Multi-stage optimization provided a solution to the need of optimizing many features which otherwise often failed to converge at all, or within reasonable number of iterations. A 3-step PSO was implemented. The first solved face position and angle by only adjusting camera 6DoF pose with the target actor static as already mentioned. The second solved against background colour parameters, with the camera able to search in azimuth (x,y,Pan), but now moving the actor such that the previous relative standoff with respect to the camera was maintained. These first two steps could be considered the scouting part of the system, whereby the envi-

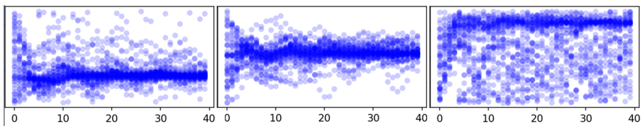


Figure 6: *VACE particle parameter convergence - 48 particles, 40 iterations and solving for 3 camera dimensions. L to R: X, Y, Yaw angle. Optimised against background rgb and v mean pixel values*

ronment is searched without prior knowledge or annotation. The third optimization was to control lighting intensity of a light at a static offset to the actor. Each PSO was controlled by a set of nominated image features and weights, along with lower / upper control bounds and a loss norm function. Initially all PSO stages accumulated feature scores in the default L_2 norm fashion. However, experimentation showed that an L_2 norm would produce unsatisfactory results for rgb colour histogram feature matching. Often, a beige background would score better than a solid block colour background. In order to alleviate this, the Linf norm was selected for the middle stage PSO solver, and this produced more credible results for colour histogram matching. This resulted in experimental best use of L_2 for face, L_{inf} for BG colour, L_2 for lighting. Figure 6 shows example particle convergence when tasked with optimizing for 4 background colour features by adjusting 3 control parameters of camera position and yaw in the second PSO.

Experimenting with PSO hyperparameters of c1, c2, and w, (cognitive, social, and inertia weights) confirmed the default selection of 0.5, 0.3, and 0.9 respectively performed the best for all steps in that in the majority of examples, the position could be closely aligned given at least 30 iterations (5 for step 3) . Both Local-best and Global-best methods were tested, with Global-best performing faster convergence. For step 2 (background), In fact, the default GlobalBest PSO parameters worked best given enough iterations (at least 30). It can be observed that there was generally less convergence between particles in step b, meaning although a cluster of particles did tend towards a good position, at least half of the other particles continued searching. Step 3 (lighting) could converge with the least number of iterations, generally 5 or less. Solver functionality as described above was implemented in software named Virtual Autonomous Cinematography Environment (VACE), and by which the user could interact using a user interface as developed below.

3.5. User Interface

The VACE system requires a goal for the pose solver to solve against. This section describes the step-wise approach taken to user interaction for setting this goal.

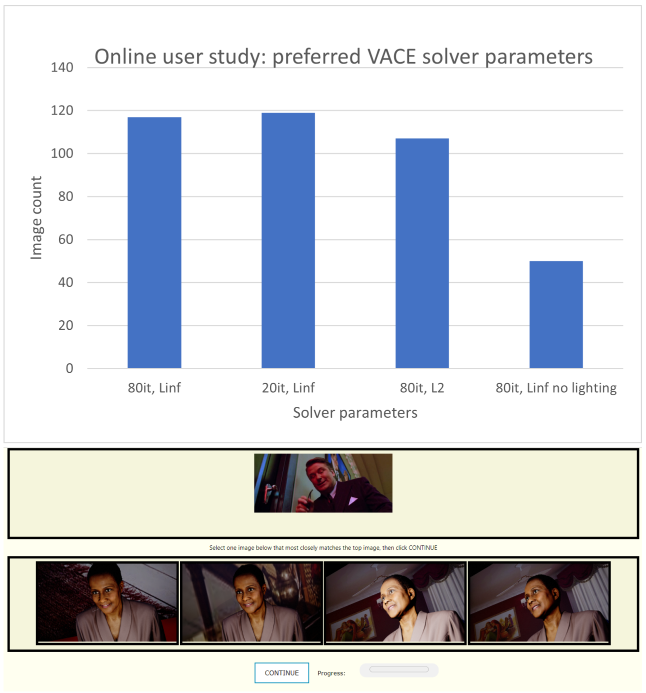
User defined image imitation was implemented first. Image feature parameters were extracted from a reference image nominated by the user, and VACE solves against those parameters. Feature selection and weights were controlled as per figure 5. The output quality of this method was observed in an online survey as described in 4, to understand how VACE loss and iteration hyperparameters performed. This method of interaction provided confidence that solving as per feature parameters could be achieved. **Automatic GMM component sampling** started to bring together the concept of autonomous unsupervised virtual scouting, in that the previously analysed film clusters are now used as goal parameters for the VACE solver. In this case, previously computed (section 3.2) feature clusters are provided as input to the VACE solver via samples taken from each of the multivariate Gaussian mean and covariance matrices. Each stage of the optimization (actor position, background, and lighting) required its own set of Gaussian sampled parameters appropriate to that stage (e.g. face features for stage 1, background features for stage 2, and face brightness for stage 3). GMM component nomination could be user defined to achieve a specific shot type, or all components could be solved sequentially. The latter option gave the user a full representation of film-specific characteristic shot-types, but applied to the environment and actors defined in the 3D space within UE4.

All the above modes of operation could be run through a command line interface to support batch processing of large datasets, or with the GUI presented to the user for manual control. The subsequent steps describe how VACE output was presented as recommendations for a storyboard creation task.

Storyboard creation . The above components provided a method for the virtual camera, actor and light to be adjusted in the virtual space via VACE as per the desired goal. The output of VACE was pose parameters and still images representing its convergence on the target parameters. As such, a user interface was developed with the aim of allowing a user to select a series of still camera shots into a storyboard. Shot selection was either by pre-computed VACE images, or by a live Spout connected virtual camera feed from the UE4 environment. The purpose was to allow the user to complete a short scouting and shot planning task, and to experiment with recommended shots derived from imitating images, films, or clustered shot-types from VACE, and from completely manually controlled cameras in UE4. **Hybrid motion tracked and VACE recommendation** allowed further experimentation that aimed to provide a more natural , familiar cinematographic method of manual exploration by holding and moving the motion-tracked camera, rather than interacting with mouse and keyboard. This hybrid approach aimed to combine the best of both worlds, and approach the contribution defined in 1.1 of supporting creativity and efficiency, which may otherwise be hampered if the user was only able to manually adjust virtual cameras with traditional peripherals of mouse and keyboard. A gamepad was also provided, to enable quicker movement within the environment, as the tracked camera only allowed a limited 6DoF, bounded by the physical room dimensions.

4. Evaluation

We provide evidence on the first claim put in 1, that professional level camera poses can be realised using the system developed in this paper. Here, we test this hypothesis by presenting output images to lay people, and asking their preference in an image imitation scenario. An online user study was conducted in order to measure how different optimization parameters performed. A set of 23 reference images was selected from [SKSB21], and provided as input to the VACE system, which was primed with a 3D environment, actor, and a single light source. The system was tasked with imitating the reference image composition through PSO, and output images recorded for 4 different configurations. The survey presented a random reference image and 4 candidate VACE images.



Select one image below that most closely matches the top image, then click CONTINUE

Figure 7: *(upper) User preference for VACE processing method, when tasked with imitating a single shot from a movie. Coloured by participant. (lower) The study interface.*

The user was asked to nominate the most closely matching VACE image to the reference image, with the test repeating 30 times for each participant. See figure 7, showing indication from users that methods including lighting adjustment are preferred, however none out of those 3 preferred methods particularly out perform the rest. Qualitative feedback from a questionnaire returned positive results for face positions (70% respondents indicating most images were *mostly the same*), but indicated room for improvement for background colour and lighting (80% indicating most images were *sometimes the same*) as the reference image. Out of the respondents experience levels were recorded as 3 'occasionally took photos', 4 'experienced amateur', 1 'professional', and 2 'never took photos'.

Shortcomings, some of which may be tackled in future work are identified as follows: The validation of clustering as a method of extracting characteristic shot types from film, when calculated in the method described in section 3.2. This needs verification through studies that identify style or characteristics of generated shots in the system. The ability for style to be *correctly* imitated when compared to other sources (e.g. compare clusters extracted from one film / genre, to clusters extracted from another film / genre), or compare extracted clusters from one film / genre against a large superset of clusters that covers all films analysed in the dataset. The ability to alter how 'unique' shot types are identified or recommended to the user - for instance how should the user know that some shot-type clusters are the 'generic' ones like RoT, and how some shot-types are particularly unique for that film? Consideration for a collaborative approach is given, in that we hypothesise that tools like the ones developed in this paper act as one part in the human workflow of film creation. However, the exact integration details of such tools into the workflow needs to be evaluated, and their 'ownership' of particular sub-tasks needs to evaluated for their merit or hinderance. We also desire to test the utility of tools like this in a variety of settings - for novice film-makers who may benefit from guidance, and for professionals who may require time-saving instruments in order to be more effective. Quality improvements are desired, and acknowledgement is given to the questionnaire feedback that colour and lighting do not score as well as face positions. From this, we believe additional features will help to achieve better imitation. Also, 2 main components of cinematography are omitted from this work, firstly the ability to consider dynamic environments of actor and camera movement, and secondly the ability to consider multiple actors in frame, or at the least in an over-the-shoulder type shot which is commonly used for conversations. Editing between shots is not considered, and although a requirement in subsequent film-making steps, this body of work does not expect to tackle that problem. Previous research has approached this problem.

Regarding the claim that recommended poses driven from film analysis provides efficient, creative and appropriate images for storyboard creation — this work is yet to be completed, but will test the utility of producing a storyboard using the tools in section 3.5.

5. Conclusion

This work demonstrates automating recommendations for the task of creating storyboards, with elements of location scouting, for film pre-production. System development and evaluation has been discussed, which currently demonstrates unsupervised guidance of virtual camera pose. A small scale user study of the output images shows generally good levels of compositional similarity for face positioning, with further work required to improve colour and brightness matching. Other features are also expected to be introduced for future work. Proposals for hybrid workflows that combine both autonomous recommendations and a tactile tracked cameras have been suggested, as future work which will extend the current proof of concept towards a valuable industry tool. Additional investigation including ablation studies is also expected to improve understanding of system component performance.

6. Acknowledgement

This work was supported by EPSRC Centre for Digital Entertainment with the grant number EP/L016540/1.

References

[AMJ*20] Achary, S., Moorthy, K. L. B., Javed, A., et al. "CineFilter: Unsupervised Filtering for Real Time Autonomous Camera Systems". *Workshop on Intelligent Cinematography and Editing*. 2020 2.

[AXY*23] Anyi, R., Xuekun, J., Yuwei, G., et al. "Dynamic Storyboard Generation in an Engine-based Virtual Environment for Video Production". *arXiv:2301.12688* (2023) 2.

[BDER08] Burelli, P., Di Gaspero, L., Ermetici, A., and Ranon, R. "Virtual camera composition with particle swarm optimization". *Smart Graphics: 9th Int. Symp.,* Springer. 2008, 130–141 2, 5.

[BR14] Barry, W and Ross, B.J. "Virtual photography using multi-objective particle swarm optimization". *Annual Conference on Genetic and Evolutionary Computation*. 2014, 285–292 2.

[BSB*19] Benini, S., Savardi, M., Balint, K., et al. "On the influence of shot scale on film mood and narrative engagement in film viewers". *IEEE Trans. on Affective Computing* c (2019), 1–1 3.

[CL13] CANINI L.and Benini, S. and LEONARDI, R. "Affective recommendation of movies based on selected connotative features". *IEEE Trans. on Circuits and Systems for Video Technology* 23.4 (2013), 636–647 3.

[CON08] CHRISTIE, M., OLIVIER, P., and NORMAND, JM. "Camera control in computer graphics". *Computer Graphics Forum* 27.8 (2008), 2197–2218 2.

[CWK03] CHEN, Y., WANG, J. Z, and KROVETZ, R. "Content-based image retrieval by clustering". *Proceedings of the 5th ACM SIGMM int. workshop on Multimedia information retrieval*. 2003, 193–200 3.

[DHS73] DUDA, R. O., HART, P. E., and STORK, D. G. *Pattern classification and scene analysis*. Vol. 3. Wiley New York, 1973 4.

[EG22] EVIN I.and Hämäläinen, P. and GUCKELSBERGER, C. "Cine-AI: Generating Video Game Cutscenes in the Style of Human Directors". *Proc. ACM Hum.-Comput. Interact.* 6.CHI PLAY (2022) 2.

[Gal15] GALVANE, Q. "Automatic Cinematography and Editing in Virtual Environments." PhD thesis. Université Grenoble Alpes (ComUE), 2015 2.

[GR17] GALVANE, Q. and RONFARD, R. "Implementing Hitchcock - the Role of Focalization and Viewpoint". *Workshop on Intelligent Cinematography and Editing*. 2017 2.

[HAB17] HOESL, A., ARAGON BARTSCH, S., and BUTZ, A. "TrackLine: Refining touch-to-track Interaction for Camera Motion Control on Mobile Devices". *Lecture Notes in Computer Science* (2017), 283–292 2.

[HCS96] HE, L.W., COHEN, M. F., and SALESIN, D. H. "The virtual cinematographer: A paradigm for automatic real-time camera control and directing". *23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH* (1996), 217–224 2.

[HO07] HERSHEY, J. RR. and OLSEN, R. A. "Approximating the Kullback Leibler divergence between Gaussian mixture models". *Int. Conference on Acoustics, Speech and Signal Processing*. Vol. 4. IEEE. 2007, IV–317 4.

[HX05] HANJALIC, A. and XU, LQ. "Affective video content representation and modeling". *IEEE Trans. on multimedia* 7.1 (2005), 143–154 3.

[JECJ07] JENSEN, J.H., ELLIS, D.PW., CHRISTENSEN, M. GRÆSBØLL, and JENSEN, S.H. "Evaluation of distance measures between Gaussian mixture models of MFCCs". *Proceedings of the 8th Int. Conf. on Music Information Retrieval*. 2007, 107–108 4.

[JWW*20] JIANG, H., WANG, B., WANG, X., et al. "Example-driven virtual cinematography by learning camera behaviors". *ACM (TOG)* 39.4 (2020), 45–1 2.

[Kan03] KANG, H.B. "Affective content detection using HMMs". *Proceedings of the ACM int. Multimedia Conference and Exhibition* (2003), 259–262 3.

[KE95] KENNEDY, J. and EBERHART, R. "Particle swarm optimization". *Proceedings of ICNN'95-Int. conference on neural networks*. IEEE. 1995, 1942–1948 2.

[KRE*14] KURZ, C., RITSCHEL, T., EISEMANN, E., et al. "Generating Realistic Camera Shake for Virtual Scenes". *Journal of Virtual Reality and Broadcasting* 7 (2014) 2.

[LBS*20] LANKHUIZEN, T., BÁLINT, K. E., SAVARDI, M., et al. "Shaping Film: A Quantitative Formal Analysis of Contemporary Empathy-Eliciting Hollywood Cinema". *Psychology of Aesthetics, Creativity, and the Arts* (2020) 3.

[LCL18] LOUARN, A., CHRISTIE, M., and LAMARCHE, F. "Automated staging for virtual cinematography". *Proceedings - MIG 2018: ACM SIGGRAPH Conference on Motion, Interaction, and Games* (2018) 2.

[MBC14] MERABTI, B., BOUATOUCH, K., and CHRISTIE, M. "A virtual director inspired by real directors". *AAAI Workshop - Technical Report* WS-14-06 (2014), 40–48 2.

[MVWM19] MUENDER, T., VOLKMAR, G., WENIG, D., and MALAKA, R. "Analysis of previsualization tasks for animation, film and theater". *CHI Conference on Human Factors in Computing Systems*. 2019, 1–6 2.

[MZBB16] MOSS, F.M., ZHANG, F., BADDELEY, R., and BULL, D. R. "What's on TV: A large scale quantitative characterisation of modern broadcast video content". *IEEE Int. Conference on Image Processing*. IEEE. 2016, 2425–2429 3.

[OFF23] OFFWORLDLIVE. *Live-streaming technology for games engines*. 2023. URL: https://offworld.live/ (visited on 03/17/2023) 4.

[PCM20] PUEYO, P., CRISTOFALO, E., and MONTIJANO E.and Schwager, M. "CinemAirSim: A camera-realistic robotics simulator for cinematographic purposes". *IEEE Int. Conference on Intelligent Robots and Systems* (2020), 1186–1191 4.

[PHPU16] PRIMA, D.A., HARIADI, M., PURNAMA, I.K.V., and USAGAWA, T. "Virtual camera movement with particle swarm optimization and local regression". *Int. Review on Computers and Software* 11.9 (2016), 773–793 2.

[RGBM22] RONFARD, R., GANDHI, V., BOIRON, L., and MURUKUTLA, V.A. "The Prose Storyboard Language: A Tool for Annotating and Directing Movies". *Workshop on Intelligent Cinematography and Editing*. 2022, 1–16 2.

[Ron21] RONFARD, R. "Film Directing for Computer Games and Animation". *Computer Graphics Forum* 40.2 (2021). Eurographics State of the Art Report, 713–730 2.

[RSS05] RASHEED, Z., SHEIKH, Y., and SHAH, M. "On the use of computable features for film classification". *IEEE Trans. on Circuits and Systems for Video Technology* 15.1 (2005), 52–63 3.

[SC97] SMITH, J. R. and CHANG, S.F. "VisualSEEk: a fully automated content-based image query system". *Proceedings of the fourth ACM int. conference on Multimedia*. 1997, 87–98 3.

[SDM*14] SANOKHO, C., DESOCHE, C., MERABTI, B., et al. "Camera Motion Graphs". *SCA 2014 - Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2014), 177–188 2.

[SKSB21] SAVARDI, M., KOVÁCS, ANDRÁS BÁLINT, SIGNORONI, A., and BENINI, S. "CineScale: A dataset of cinematic shot scale in movies". *Data in Brief* 36 (2021) 1, 6.

[SPO23] SPOUT. *realtime video routing for Windows*. 2023. URL: https://spout.zeal.co/ (visited on 03/17/2023) 4.

[TLT20] TARVAINEN, J., LAAKSONEN, J., and TAKALA, T. "Film Mood and Its Quantitative Determinants in Different Types of Scenes". *IEEE Trans. on Affective Computing* 11.2 (2020), 313–326 3.

[WAE*20] WRIGHT, C., ALLNUTT J.and Campbell, R., EVANS, M., et al. "AI in production: video analysis and machine learning for expanded live events coverage". *SMPTE Motion Imaging Journal* 129.2 (2020), 36–45 2.

[Wan06] WANG H. L.and Cheong, L.F. "Affective understanding in film". *IEEE Trans. on Circuits and Systems for Video Technology* 16.6 (2006), 689–704 3.

[XJLD08] XU, M., JIN, J. S., LUO, S., and DUAN, L. "Hierarchical movie affective content analysis based on arousal and valence features". *ACM int. Conference on Multimedia* (2008), 677–680 3.

[Xu15] XU D.and Tian, Y. "A comprehensive survey of clustering algorithms". *Annals of Data Science* (2015), 165–193 4.

[YCLC19] YANG, T. Y., CHEN, Y. T, LIN, Y.Y., and CHUANG, Y.Y. "Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image". *Proceedings of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* (2019), 1087–1096 3.

[YSP*21] YOO, J. E., SEO, K., PARK, S., et al. "Virtual Camera Layout Generation using a Reference Video". *CHI Conference on Human Factors in Computing Systems*. 2021, 1–11 2.

[YYWR22] YU, Z., YU, C., WANG, H., and REN, J. "Enabling Automatic Cinematography with Reinforcement Learning". *Int. Conference on Multimedia Information Processing and Retrieval*. IEEE. 2022, 103–108 2.