High-Level Features for Movie Style Understanding

Robin Courant[1] Christophe Lino[1] Marc Christie[2] Vicky Kalogeiton[1]

[1]LIX, Ecole Polytechnique, CNRS, IP Paris [2]Univ Rennes, CNRS, IRISA, INRIA

Abstract

*Automatically analysing stylistic features in movies is a challenging task, as it requires an in-depth knowledge of cinematography. In the literature, only a handful of methods explore stylistic feature extraction, and they typically focus on limited low-level image and shot features (colour histograms, average shot lengths or shot types, amount of camera motion). These, however, only capture a subset of the stylistic features which help to characterise a movie (*e.g. *black and white* vs. *coloured, or film editing). To this end, in this work, we systematically explore seven high-level features for movie style analysis: character segmentation, pose estimation, depth maps, focus maps, frame layering, camera motion type and camera pose. Our findings show that low-level features remain insufficient for movie style analysis, while high-level features seem promising.*

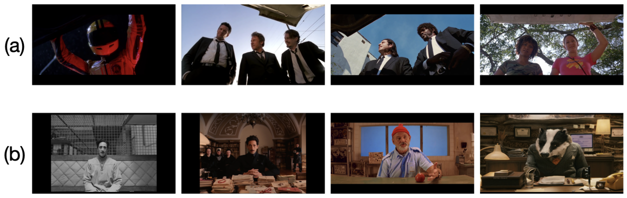

Figure 1. **Recognisable director movie styles.** Eight frames from eight movies from two directors: (a) Quentin Tarantino and (b) Wes Anderson. Even movies from different years (*Reservoir Dogs* in 1992 to *Kill Bill Vol.1* in 2003) and different types (*The Grand Budapest Hotel* with live action or *Fantastic Mr. Fox* with animation) from the same director encompass a common style, i.e., trunk-style looking-up point-of-view shot for (a); symetric and detailed scenes for (b). From the top-left to the bottom-right: *Kill Bill Vol.1* (2003), *Reservoir Dogs* (1992), *Pulp Fiction* (1994), *Death Proof* (2007), *The French Dispatch* (2021), *The Grand Budapest Hotel* (2014), *The Life Aquatic with Steve Zissou* (2004) and *Fantastic Mr. Fox* (2009).

1. Introduction

While the first film was released over a century ago, the rise of streaming platforms has led to two important phenomena: an acceleration in the number of films produced, and a very wide availability of these films (recent or old).

Movies are complex audio-visual contents to analyse and understand. They operate on multiple sensory and cognitive levels, built on a rich history of techniques. They are also difficult to formalize and offer a wide variety of dimensions to study (*e.g.* genre, artistic intentions, aesthetics, narrative arcs, and style). In turn, they offer a rich, yet challenging, source of data analysis. However, even today, machines remain unable to capture high-level information like the style or intention of directors, *e.g.* emotions or long-term interactions between characters. In fact, understanding all the details that make up a movie remain challenging. It requires great prior knowledge of cinematography. The analysis of director's intentions also remain subjective.

Only a few learning-based approaches rely on cinematographic data, director [5] or genre and style classification [25]. These methods share a commonality: they all extract low-level features (*e.g.*: dominant colour, number of frames per shot) to guide their decisions. However, we are convinced that director's secrets are deeply hidden in their frames and audio tracks, and such low-level features remove a large amount of information, crucial to unveil them.

Hence, as low-level features are too limited and raw frames are too general for the current methods, we propose to explore the extraction of higher-level cinematographic features. They should be both general enough, *i.e.* able to encapsulate a maximum of information, and elaborate enough, *i.e.* able to focus on particularities. Hence, as cinematography experts proceed, we propose to decompose film analysis into different axes (*e.g.*: frame composition, or camera behaviour), and to build high-level features, specific for each axis. Our goal is to keep the information from a particular axis, and remove the rest. Finally, combining these different specific features makes it possible to retrieve global information about the cinematic content.

In this paper, we first present a set of straightforward experiments on directors' style classification. They show how complex this task is. By examining the resulted feature space, this also corroborates our initial claim that low-level features are not representative for this task (Section 3). Then, we analyse high-level features which remain unexplored and could improve movie style understanding, i.e., frame layering and camera motion type (Section 4). This work explores a promising avenue in movie style understanding, and paves the way for future research on this area.

2. Related Work

Cinematography is a well-studied field, both for analysis and synthesis purposes. As pointed by [19], many applications can benefit from automatic cinematic approaches (*e.g.* virtual production or interactive drama). Furthermore, movies and TV shows become more easily accessible, especially on the web, and just start to get exploited in computer vision.

Working with these video contents, [3] propose a multimodal person clustering algorithm, and [18] propose a method to generate textual description of a clip. These contents are also exploited in more general video analysis tasks, as they provide examples of daily scenes. For instance, [22] learn social interactions from movies and [14] propose a dataset for realistic human action recognition.

Some works also focus on automatic cinematography analysis. [5, 25, 20] extract low-level features from visual, audio and textual modalities to tackle different cinematic tasks, *e.g.* director classification, genre classification, film rating prediction or production year prediction. More recently, large-scale annotated movie datasets such as [9] open the perspective for better cinematic analysis. [16] propose a supervised framework to classify the shot type (camera movement and scale), [17] propose a multimodal scene segmentation method, and [10] propose to learn visual models from movie trailers. However, approaches focusing on the analysis of film genre or directorial styles remain limited, probably due to a lack of data. Indeed, creating large datasets of movies and TV shows requires to acquire all copyrights. Such approaches also rely solely on low-level visual or audio features (*e.g.* colour histograms, shot length distribution, voice spectrograms). To overcome these limitations, in this paper, we propose to explore the extraction and use of high-level features, enabling to retrieve more global information on the directorial style of a movie.

3. Low-Level Features in Director Recognition

Given a set of directors and clips from their respective movies, director classification consists in associating each clip with its director. In this section, we propose a supervised approach to director classification on a dataset (CMD8) specifically designed for this task.

Method. For classification, we rely on a 3D variant of the ResNet architecture [7]. We take as input 16 raw frames, selected by splitting the video clips into chunks of 32 frames, which are then randomly sampled.

Table 1. **Director classification results.**

Method	Precision	Recall	F1
Random	12.77	12.86	11.89
Weighted Random	12.42	12.42	12.42
ResNet-18 (scratch)	35.93	34.16	35.02
ResNet-18 (k700)	**51.17**	**47.34**	**49.18**
ResNet-50 (scratch)	38.14	38.55	38.34
ResNet-50 (k700)	48.08	41.91	44.78

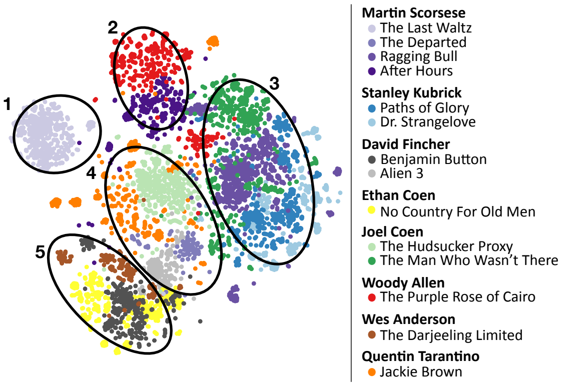

Figure 2. **t-SNE visualization of feature embedding on CMD8.** Clusters correspond to: (1) 1970s movies, (2) 1980s movies, (3) black and white movies, (4) 1990s movies and (5) 2000s movies.

3.1. Datasets

CondensedMovies [1] is a corpus gathering more than $1,270$ hours of around two minutes clips, taken from YouTube, from more than $3,600$ movies. For each movie, the dataset contains several clips. Each clip depicts a key scene of the movie and comes with semantic description of the scene, character face-tracks, and movie metadata.

CMD8 (*Condensed Movies Director 8*) contains 24 hours of clips from CondensedMovies. We pick all clips related to eight directors.

3.2. Experimental Results

Metrics. For evaluation, we use: Precision, Recall and F1 score as the harmonic mean between precision and recall.

Quantitative results. We test two approaches: (i) training from scratch, using random weight initialization; or (ii) using a model pre-trained on Kinetics-700 [4]. We also experiment with two ResNet depths: 18 and 50. For comparison, we also report results on two baselines: (a) a random classifier, and (b) a random weighted classifier (*i.e.*: 1 chance out of 8, weighted by the number of samples in the class.).

Table 1 shows the results on CMD8. We observe that ResNet outperforms both baselines, resulting in precision and recall of around 50%. However, we make the following two remarks. First, the shallowest architecture performs better than the deepest. Second, the performance margin be-

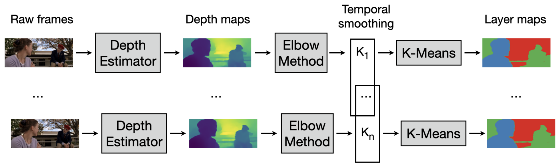

Figure 3. **Frame layering pipeline.** Raw frames are used to estimate depths maps, which are converted into a discrete set of layers.

tween scratch and pre-trained configurations is small. Both remarks are in contrast to the observations of the state of the art, where deeper models outperform shallow ones (e.g., as in [7] for HMDB-51 [13]) and using models pre-trained on Kinetics [4] doubles the performances. The first remark shows that probably there are not enough training data; the second shows that this type of models with pre-training on action datasets are not optimal for director classification, thus revealing the need for more elaborate methods.

Feature visualization. To have an idea of criteria learned by the network, Figure 2 displays the t-SNE [8] visualisation of the CMD8 test set. For each sample, we extract visual features learned by the ResNet-18 pre-trained on Kinetics 700. From this visualisation, we distinguish five clusters and characterize them by examining their movie characteristics: each cluster gathers movies from the same decade or in black and white. This would suggest that, using only raw frames, the network learns low-level visual features such as the amount of blur, that highly characterizes the technical evolution of the camera over the decades (film in contrast with digital). It requires further investigation.

Discussion. Our findings show that pre-training on large action datasets like Kinetics [4] is not necessarily suitable for style analysis. Possible solutions for this would be to either pre-train on movie datasets or to use other inputs. As the dataset is not large enough, models cannot extract easily all the information. While this lack of data is a possible reason of failure, our hypothesis is that the provision of higher-level features (which experts use to perform director classification) shall be explored.

4. High-Level Features in Movie Style Analysis

We propose the study of seven high-level features for film style understanding in cinematography, grouped into three categories. We describe our proposed pipelines for two of them: frame layering and camera motion detection.

(a) Character-based features enable tracking what the director is focusing on, as characters are very often the centre of the story and of the visual content. We extract two character-based features: **character segmentation** using Detectron2 [24] and **pose estimation** using DOPE [23]. Both are tracked along each shot with a Kalman filter combined with the Hungarian algorithm [2].

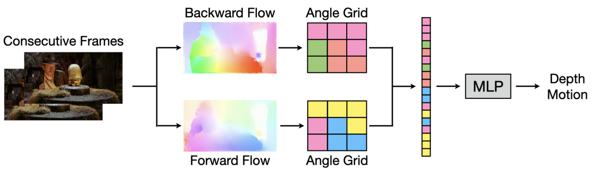

Figure 4. **Camera motion detection pipeline.** Consecutive raw frames are used to compute forward and backward optical flows that are converted to angle grids. Once flattened, angle grids are fed to a MLP that learns the camera motion.

(b) Composition-based features are essential to understand the aesthetics of directors. In particular, the frame composition is closely related to the complexity of a *mise-en-scene*. We extract three composition-based features: **depth estimation** using [15], **focus estimation** with [6] and **frame layering**. Further, we propose a frame layering method that splits a frame into depth layers to retrieve various frame composition levels, *e.g.* foreground, middleground or background.

(c) Camera-based features are key markers of cinematographic style. They define characteristic camera behaviour in relation with scene contents. Moreover, the camera is the eye of the audience. Therefore, we argue that understanding the camera behaviour helps to better understand the director's intentions. We extract two camera-based features: **camera pose estimation** in the toric space from [11] and **camera motions**. For the camera motion detection, we build our own model that learns six camera motion types.

Frame layering has never been exploited before, even though it is a spontaneous process humans do when looking at an image. In this work, we propose a frame layering approach that extracts a *layer map* where pixels are grouped into depth layers (*e.g.* foreground, middleground, background). Figure 3 shows our extraction pipeline. Given a sequence of consecutive frames, we first compute their depth maps using [15]. We then cluster pixels through a K-means. The optimal number of clusters is computed with the elbow method [12]. To improve the temporal smoothness of computed maps (NN-based depth estimators are inherently noisy), we smooth the optimal number of clusters for consecutive depth maps using a max pooling sliding window. In the end, we compute a final cluster map using the smoothed optimal number of clusters.

Camera Motion is the way the camera moves in space and creates dynamics within consecutive frames. Typical examples encompass static shots, horizontal movements (pan and truck), vertical movements (boom and tilt), depth movements (zoom, pull-out and push-in) and rotational movement (roll). In this work, we propose a pre-processing

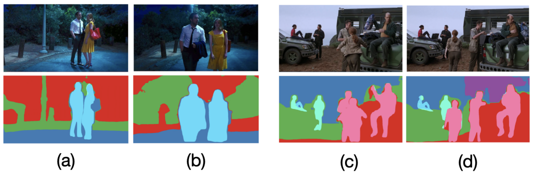

Figure 5. **Examples of frame layering.** (top) Raw frames, and (bottom) layer maps with overlayed character segmentation masks. Our pipeline (a), (b): correcly predicts the number of layers and is consistent over time. (c), (d): incorrectly predicts a new layer (purple) when a character moves towards the background.

and detection pipeline, learning to differentiate among them (Figure 4). Given two consecutive frames, we first compute their forward and backward optical flows using [21]. Then, we compute the flows' angles, and average pool them to obtain the *angle grids*. We finally flatten and concatenate the backward and forward angle grids and feed them to a two-layer MLP that learns to classify them. Behind this detection, we aim to estimate the extrinsic 6 degrees of freedom of the camera. Note that at training we merge some motions into more general groups (*e.g.* horizontal, vertical, depth motions). In practice, it is difficult to distinguish them.

5. Experiments

5.1. Experiments on Frame Layering

Quantitative results. Figure 5 displays some results when applying our pipeline on several video sequences. (a, b) show an example where layering is successful, i.e., with the right number of clusters and good temporal continuity. (c, d) show a failure case: while the central character is transitioning from foreground to background, our method incorrectly generates a new layer (in purple, top right of layer map (d)). We argue that this lack of robustness results from both the noisy depth map output and the way we choose the number of clusters (which seems suboptimal). Overall, both examples show the efficacy of our layering method for unseen and challenging (dark as in (a), cluttered in (c)) scenes.

5.2. Experiments on Camera Motion Detection

MotionSet is a dataset we created with camera motion clips from YouTube[1]. We split each shot into sub-clips with single camera motion, resulting in 75 clips with motions: static, horizontal (pan and trucking), vertical (boom, tilt), depth (zoom, pull out and push in) and rotational (roll).

Quantitative results. We train and test our camera motion detector on MotionSet. Table 2 reports the results. For comparison, we also evaluate the *Random* and *Weighted Random* baselines. Our model significantly outperforms both baselines, and it reaches high performances, i.e., 96.81% of F1 score and almost perfect precision 98.47%. This shows that for simple and short sequences, our model correctly recognizes the camera motions.

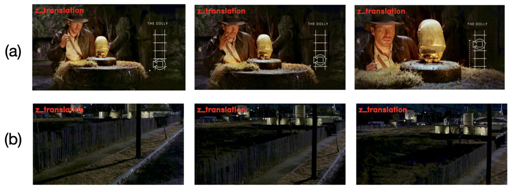

Figure 6. **Examples of camera motion detections.** (a) Correctly predicted zoom motion. (b) Failure case of a mixed horizontal and vertical motion incorrectly predicted as zoom.

Table 2. **Motion classification results.**

Method	**Precision**	**Recall**	**F1**
Random	21.26	21.70	19.62
Weighted Random	18.62	18.49	18.52
Ours	98.47	95.50	96.81

Qualitative results. Figure 6 displays some results when applying our detector on several video sequences. (a) shows an example of zoom camera motion correctly predicted; (b) shows a failure case, where a mixed horizontal and vertical motion is incorrectly predicted as a zoom. We observe that in most cases, the detector recognizes the motion correctly. However, we are aware that our dataset is probably not diverse enough. In addition, when using our motion detection in the wild, we observe that it is not robust to combined motion (*e.g.* mixing vertical with horizontal camera motion). In this case, our model typically fails, most likely because it is not trained with such challenging samples.

6. Conclusion

In this paper, we perform straightforward experiments on director style classification, and show that the performances are not satisfying when solely relying on raw frames. We then propose and analyse a non-exhaustive list of high-level features that we believe could improve such classification tasks. We finally show the first results for frame layering and camera motion detection, which seem promising for cinematographic applications.

In the future, we plan to consider more features. For instance, we could use audio-based features (*e.g.* active speaker) or exploit soundtrack analysis. Finally, a longer-term objective would be to explore latent representations learnt by our models (i) to understand what they learnt and which features are important; and (ii) to exploit these representations for various applications to help filmmakers.

[1] `https://youtu.be/GbnYBmqBbKA`

References

[1] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings. In *Proc. Asian Conf. on Computer Vision*, 2020.

[2] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. *CoRR*, 2016.

[3] Andrew Brown, Vicky Kalogeiton, and Andrew Zisserman. Face, body, voice: Video person-clustering with multiple modalities. *arXiv preprint arXiv:2105.09939*, 2021.

[4] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

[5] Priyankar Choudhary, Neeraj Goel, and Mukesh Saini. A multimedia based movie style model. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2019.

[6] Xiaodong Cun and Chi-Man Pun. Defocus blur detection via depth distillation. In *ECCV*, 2020.

[7] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018.

[8] Geoffrey Hinton and Sam Roweis. Stochastic neighbor embedding. In *NeurIPS*, 2002.

[9] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *ECCV*, 2020.

[10] Qingqiu Huang, Yuanjun Xiong, Yu Xiong, Yuqi Zhang, and Dahua Lin. From trailers to storylines: An efficient way to learn from movies. *arXiv preprint arXiv:1806.05341*, 2018.

[11] Hongda Jiang, Bin Wang, Xi Wang, Marc Christie, and Baoquan Chen. Example-driven virtual cinematography by learning camera behaviors. *ACM Transactions on Graphics (TOG)*, 2020.

[12] Trupti M Kodinariya and Prashant R Makwana. Review on determining number of cluster in k-means clustering. *International Journal*, 2013.

[13] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

[14] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[15] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, 2019.

[16] Anyi Rao, Jiaze Wang, Linning Xu, Xuekun Jiang, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A unified framework for shot type classification based on subject centric lens. In *ECCV*, 2020.

[17] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A local-to-global approach to multi-modal movie scene segmentation. In *CVPR*, 2020.

[18] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 2017.

[19] Rémi Ronfard. Film directing for computer games and animation. In *Computer Graphics Forum*. Wiley Online Library, 2021.

[20] Jussi Tarvainen, Mats Sjöberg, Stina Westman, Jorma Laaksonen, and Pirkko Oittinen. Content-based prediction of movie style, aesthetics, and affect: Data set and baseline experiments. *IEEE Transactions on Multimedia*, 2014.

[21] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[22] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *CVPR*, 2018.

[23] Philippe Weinzaepfel, Romain Brégier, Hadrien Combaluzier, Vincent Leroy, and Grégory Rogez. Dope: Distillation of part experts for whole-body 3d pose estimation in the wild. In *ECCV*, 2020.

[24] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[25] Federico Álvarez, Faustino Sánchez, Gustavo Hernández-Peñaloza, David Jiménez, José Manuel Menéndez, and Guillermo Cisneros. On the influence of low-level visual features in film classification. *PLOS ONE*, 2019.