Learning to Film from Professional Human Motion Videos

Chong Huang[1], Chuan-En Lin[2], Zhenyu Yang[1], Yan Kong[1], Peng Chen[3], Xin Yang[4]*, Kwang-Ting Cheng[2]
[1]University of California, Santa Barbara,
[2]Hong Kong University of Science and Technology
[3]Zhejiang University of Technology
[4]Huazhong University of Science and Technology
chonghuang@umail.ucsb.edu, clinaf@connect.ust.hk, {zhenyuyang, yankong}@ucsb.edu
chenpeng@zjut.edu.cn, xinyang2014@hust.edu.cn, timcheng@ust.hk

Abstract

We investigate the problem of 6 degrees of freedom (DOF) camera planning for filming professional human motion videos using a camera drone. Existing methods [4, 3, 5] either plan motions for only a pan-tilt-zoom (PTZ) camera, or adopt ad-hoc solutions without carefully considering the impact of video contents and previous camera motions on the future camera motions. As a result, they can hardly achieve satisfactory results in our drone cinematography task. In this study, we propose a learning-based framework which incorporates the video contents and previous camera motions to predict the future camera motions that enable the capture of professional videos. Specifically, the inputs of our framework are video contents which are represented using subject-related feature based on 2D skeleton and scene-related features extracted from background RGB images, and camera motions which are represented using optical flows. The correlation between the inputs and output future camera motions are learned via a sequence-to-sequence convolutional long short-term memory (Seq2Seq ConvLSTM) network from a large set of video clips. We deploy our approach to a real drone system by first predicting the future camera motions, and then converting them to the drone's control commands via an odometer. Our experimental results on extensive datasets and showcases exhibit significant improvements in our approach over conventional baselines and our approach can successfully mimic the footage of a professional cameraman.

1. Introduction

Filming human motions with a camera drone is a very challenging task, because it requires the cameraman to take an overall consideration of the scenes, the moving subject and the previous camera motions to determine the next camera motion, and meanwhile precisely control the drone towards the target position and camera pose. In this study, we investigate the problem of autonomously planning 6DOF motions of a camera drone as an expert for filming a video with a moving subject (see Fig. 1).

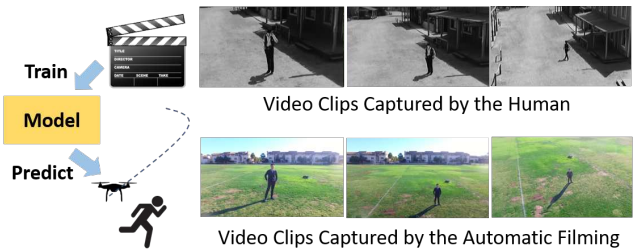

Figure 1. We learn a model to predict camera motions which imitate the operation of the professional cameraman.

Several studies have been conducted in the literature for autonomous camera planning. Existing solutions range from heuristic parametric methods to learning-based non-parametric approaches. For instance, the state-of-the-art autonomous filming application DJI QuickShot employs a parametric method by providing several filming modes, each of which utilizes a predefined path and camera poses for producing footages. Without considering the video contents, this simple solution usually suffers from problems including losing tracking of the subject and/or capturing repetitive unexciting patterns. Learning-based non-parametric methods, such as [4, 3, 5], are much more flexible as they can learn (almost) arbitrary camera motion predictors directly from training data. In [4, 3], Chen et al. learned a 3DOF camera pose predictor via recurrent decision trees to imitate professional filming basketball game events. In their method, a sequence of moving subjects' positions and camera poses were utilized to predict the next best camera pose. This method was also applied to automatic soccer game broadcasting [5]. However, existing

*corresponding author

learning-based methods only predict the rotation of a PTZ camera in a team sports, while in our drone cinematography task we demand prediction of both camera rotation and trajectories. In addition, the background scene in team sports typically has few impacts on camera planning while in drone cinematography the same subject's motion in different scenes could demand quite different camera motions. For instance, to film a subject walking along a cliff or walking on the grass, an expert may prefer to fly the camera drone towards the back and upward so as to capture a big perspective of the world around the subject for the former scene, while the expert may fly the camera drone forward and circle around to create a dramatic reveal of the subject for the latter scene. The increased degrees of freedom in camera motions and the diversity of the background scenes, significantly increase complexity of the relationship between the visual contents and future camera motions, while these issues have not been well studied.

Although several imitation learning methods, including behavioral cloning [48, 39, 44], policy learning [24] and inverse reinforcement learning [38, 1], have been developed and applied to robotics tasks, few of them are directly applicable to our automated drone cinematography task. On one hand, behavioral cloning [48, 39, 44] requires both observations and explicit control variables for training while direct control variables are not available from videos. On the other hand, policy learning [24] and inverse reinforcement learning [38, 1] require the access to the environment during training that provides the feedback between states and actions for training, while in our task there is no access to the environment during training.

In this work, we aim at an autonomous drone cinematography system which imitates experts to plan camera poses and trajectories. To this end, we propose a novel imitation learning framework which takes previous subjects' and camera motions and background scenes as input and predicts subjects' and camera motions in the following moments. We design 2D skeleton-based features to represent subjects' motions, leverage CNN features to represent background scenes and utilize dense optical flows to represent camera motions rather than 6DOF motion parameters for greatly reducing the difficulties in acquiring training data. We apply a sequence-to-sequence convolutional long short-term memory (Seq2Seq ConvLSTM) network to combine the temporal and spatial information of different inputs, and to predict the following subject's and camera motions. Since there is no such data for learning to imitate professional filming, we collect a new dataset which contains 92 video clips and is group into 6 categories according to the camera motion styles for filming a video. We analyze the impact of different inputs and compare the proposed method with several baselines. Experimental results demonstrate the superiority of our method to conventional baselines. We also deploy our model to a real drone platform and the real demo shows that our drone cinematography system can successfully mimic the footage of a professional cameraman.

In summary, our contributions are four-fold:

- A novel autonomous drone cinematography system which could well imitate a professional cameraman to film videos of human motions.
- An imitation learning framework which takes scenes, subjects' poses and camera motions as inputs and learns to predict camera motions from a large set of professional human motion videos.
- A new dataset consists of 92 video clips for imitation filming. The dataset will be released to benefit the community of learning-based filming.
- Comprehensive experiments and ablation studies to demonstrate the superiority of the proposed method over the state-of-the-arts.

We discuss related work in Sec. II, and describe our methods in Sec. III. In Sec. IV, we present the experimental results to evaluate our system. Finally, we give the conclusion in Sec. V.

2. Related Work

Autonomous Aerial Filming: Commercially available applications are often limited to watching a specified target from a fixed viewpoint, e.g. ActiveTrack [8], or a predefined path, e.g., Quickshot [9]. These solutions can hardly provide cinematic footage for various dynamic scenarios.

Recent research works [36, 37, 12, 13, 23, 16, 25, 18, 17] enable more flexibility in terms of human-drone interactions in an aerial filming task. For instance, in [36, 37, 12, 13], the users are allowed to specify the subject size, viewing angle and position on the screen to generate quadrotor motion plans automatically. However, their methods consider the camera and the drone as the same rigid body in the control optimization process. As a result, these methods usually suffer from the severe shaking problem in the captured footages. The systems in [23, 18, 25] model a camera on a gimbal attached to a quadrotor and apply two independent controllers to guarantee smooth videos. These techniques are used essentially to move the camera to the target pose specified by the user; therefore, the aesthetic quality of the video highly relies on the user's input. Huang et al. [17] designed an automatic drone filming system that estimate the next optimal camera pose which maximizes the visibility of the subject in an action scene. But only considering the subject's visibility is still too simplified to ensure a high-aesthetic quality of the captured video in various complex real-world scenarios.

Imitation Filming: Imitation filming is essentially a data-driven autonomous camera planning solution. In [4] and [3], the authors utilized video clips of basketball games to imitate professional filming for team sports. In [15], the authors learned a model based on images labeled with the object's positions for tracking the most salient object in a 360° panoramic video.

Video Aesthetic Quality Assessment: Many studies have been conducted to imitate human's way for evaluating the aesthetic quality of videos. Conventional methods mainly employ handcrafted features including color distribution [19, 21, 20], the rule of thirds [6], simplicity [33, 32], composition [7] and motion [35] for describing the aesthetic quality of a video. Recent works focus on learning deep convolutional neural networks (CNNs) features [28, 50] for the aesthetic quality assessment task [30, 26, 31, 34, 47].

3. Method

3.1. Problem Formulation

We aim at a learner that can imitate human experts' policy for camera planning by "watching" a large collection of professional videos from experts. Let $\dot{I} = (I_{t-M+1}, ..., I_t)$ and $\dot{c} = (c_{t+1}, c_{t+2}, ..., c_{t+N})$ denote a sequence of input video frames and a stream of the subsequent camera motion, where t denotes a temporal point. The control policy of human experts is defined as a prediction function: $\dot{c} = \pi^*(\dot{I})$. The goal of the learner is to find a policy $\hat{\pi}$ that best imitates the human experts π^*.

In particular, three key factors that affect the prediction performance are: 1) the feature design of the input images, 2) the accuracy of the groundtruth (i.e., camera motion), and 3) the learning ability of the model.

Input feature: Three components are highly related to imitation-oriented camera planning: 1) The motions (e.g. velocity, position and pose) of the subject, which would determine the camera's moving velocity, trajectory and viewpoint to provide the best view of the subject; 2) The background scene which would affect the composition of a frame, e.g. it is better to include both the subject and flowering shrubs in a frame, and exclude disordered clutters from the frame; 3) The previous camera motions which could ensure a smooth footage. Imitating an expert to film is a highly complex task and we think it is necessary to jointly consider all the three components and their impacts on the final captured footage. Therefore, in this work we design novel features to effectively represent the three components. Details will be presented in Sec. 3.2.1.

Camera motion: Just like beginners always learn to paint by first copying every stroke of masters' work, we think that training the learner to duplicate the flying trajectories and camera poses provided by the experts should be an effective scheme for imitating filming. It is feasible to obtain camera motions via visual-inertial navigation on a real camera drone platform equipped with inertial sensors [41]; however, it is impossible to derive absolute camera motions directly from training videos which do not contain inertial sensor data collected from the internet. As a result, directly utilizing camera motions as the output prohibits the usage of enormous professional videos publicly available online for training, and in turn impose great difficulties in collecting sufficient training data.

To alleviate the problem of collecting training data, we utilize dense optical flow of the static regions in a video as the output label of the learner to reflect the camera motion. The camera motions in the testing phase are recovered via a VIN system. Details of converting optical flows to camera motions are presented in Sec. 3.2.3.

Learning method: A desired learning model should effectively fuse information from multiple inputs and meanwhile spatial and temporal information from the inputs and their correlations with output predictions. To this end, we design on learning model based on the convolution long short-term memory (ConvLSTM) [49] to mine spatial and temporal information of each type of input and to fuse multiple input types in the network.

In addition, the length of the input and output sequences may be not equal. Therefore, we apply sequence-to-sequence architecture (Seq2Seq) [46] to map the input observation to the future camera motion. Details of our learning method is presented in Sec.3.2.2.

3.2. Imitation Learning Framework

This section describes the framework of our imitation learning algorithm (see Fig. 2), including feature extraction (Sec.3.2.1), prediction network (Sec.3.2.2) and camera motion estimation (Sec.3.2.3).

3.2.1 Feature extraction

Subject motion feature: As discussed in the previous section, we desire to encode the subject's motion in the feature representation. To this end, we design the subject motion feature in which the pose of the subject are represented using 13 keypoints of a skeleton extracted from OpenPose [2] (see Fig. 3 3^{rd} column). The velocity of the subject could be partially reflected by consecutive subject motion maps. However, OpenPose detects only a single pixel for each keypoint, which could be very sensitive to small geometric changes. To address this problem, we convolve each of 13 keypoints using a Gaussian kennel independently to blur and dilate the keypoint, yielding 13 subject motion maps for a video frame (see Fig. 3 last column).

It is common that OpenPose could fail to detect joint keypoints when the size of the subject is too small. To alle-

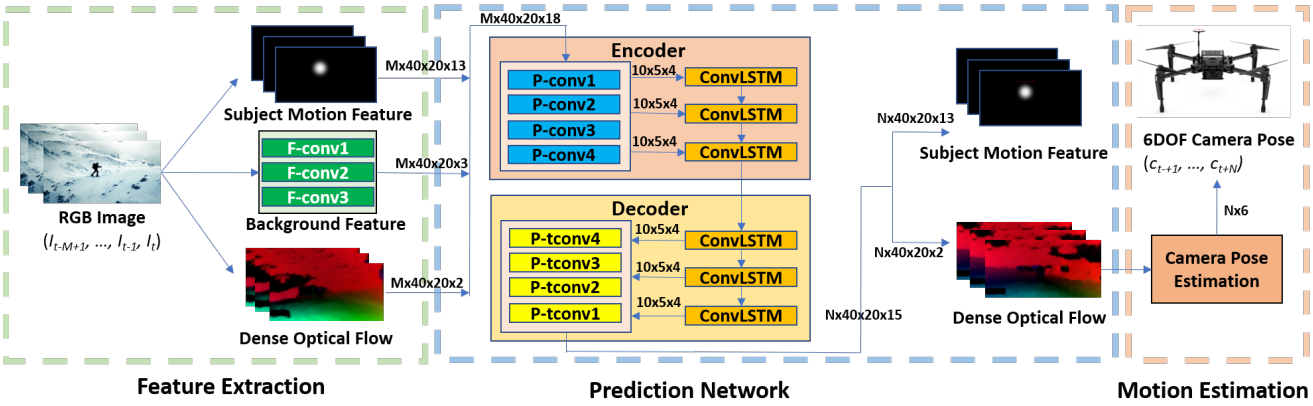

Figure 2. Overview of our imitation learning framework. The framework is consisted of three modules: 1) feature extraction, 2) prediction network and 3) camera motion estimation. We illustrate the dimension of the data flow as (time-step × width × height × depth).

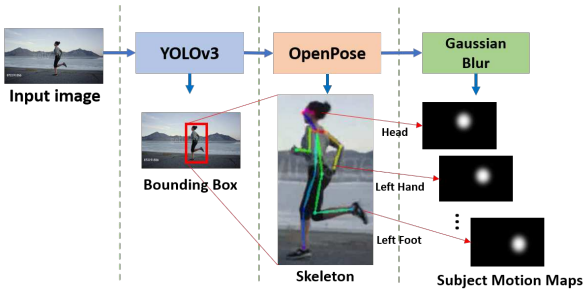

Figure 3. The extraction process of the subject motion feature.

viate this problem, rather than directly applying OpenPose to the entire image, we use subregions of the image containing human detected by the YOLOv3 [42] (see Fig. 3 2^{nd} column). Such preprocess step could greatly exclude background clutters and remove distractors for OpenPose. If the YOLOv3 detects the human but the OpenPose fails to detect the keypoints, we will copy the keypoints in the previous frame to this frame. We noticed that this scheme performs well on our benchmark videos despite the size of the subject is small.

The subject motion maps at temporal point t could effectively represent the pose of the moving subject. Concatenating subject motion maps of successive temporal points could reflect the relative motion between the subject and camera. In our experiment, we resize each feature map into 40×20 pixels.

Background Feature: To represent background scenes, we extract CNN features of the original RGB image using a 3-layer convolutional encoder as describe in Tab. 1. The final output feature maps are converted to three maps with size of 40×20 pixels.

Camera motion: As discussed in Sec.3.1, we use optical flow to represent the camera motion. In particular, we adopt the dense optical-flow method because it outputs the fixed amount of motion vectors, each of which corresponds to a pixel with the same spatial position in RGB image. This

Table 1. Layer parameters of background feature extraction network. The output dimension is given by (width × height × depth). PS: patch size for convolutional and transposed convolutional layers; S: stride. Layer types: C: convolutional

Name	Type	Output Dim	PS	S
F-conv1	C	160x80x32	3x3	2
F-conv2	C	80x40x64	3x3	2
F-conv3	C	40x20x3	3x3	2

design facilitates learning the spatiotemporal relationship between the subject and the background. Many advanced dense optical flow extraction methods could be used, e.g. FlowNet1.0 [10] and FlowNet2.0 [22]. In this work, we utilize the method proposed by Liu et al. [29] for its high efficiency on a drone platform and sufficient robustness in our task. For each frame, two dense optical flow maps are outputted by [29], representing the horizontal and vertical components of the optical flow for every pixel respectively. The dense optical flow maps are also resized to images of 40×20 pixels for the subsequent processing.

For each frame, we stack the 13-channel subject motion maps, the 3-channel scene maps and the 2-channel optical flow maps to form an 18-channel representation. We concatenate feature representation of M consecutive frames as the input of the prediction network.

3.2.2 Prediction Network

The prediction network is based on a Seq2Seq ConvLSTM model (see Fig. 2), including an encoder and a decoder. All the ConvLSTM [49] cells in the encoder and decoder share the same weights.

The encoder first processes each input of M feature maps ($40 \times 20 \times 18$) using 4 convolutional layers, and then feeds the output of the last convolutional layer to the ConvLSTM recurrently. The decoder receives the state vector of encoder conditioned on M inputs and produces predictions for the following N steps. The outputs of the ConvLSTM

are further processed using 4 transposed convolutional layers [11] to predict the subject's motion and camera motion. Each subject's motion is represented using 13 subject motion maps and the corresponding camera motion is described using 2 dense optical maps. Thus we split the output of the last transposed convolutional layer to two groups, the first group consists of $N\times40\times20\times13$ maps representing the subject's motions of N temporal points and the other group consists of $N\times40\times20\times2$ maps representing the camera motions of N temporal points. Details of the prediction network are shown in Tab. 2. The selection of M and N is experimentally evaluated in Sec.4.4.

Table 2. Layer parameters of prediction network. The output dimension is given by (width × height × depth). PS: patch size for convolutional and transposed convolutional layers; S: stride. Layer types: C: convolutional, TC: transposed convolutional, CL: convolutional LSTM cell.

Name	Type	Output Dim	PS	S
P-conv1	C	40x20x8	3x3	2
P-conv2	C	20x10x8	3x3	1
P-conv3	C	20x10x8	3x3	2
P-conv4	C	10x5x4	1x1	1
convLSTM	CL	10x5x4	3x3	1
P-tconv4	TC	10x5x4	1x1	1
P-tconv3	TC	20x10x8	3x3	2
P-tconv2	TC	20x10x8	3x3	1
P-tconv1	TC	40x20x15	1x1	2

We train our prediction network using a combination of two L2-norm losses: 1) the pixel-wise mean square errors (MSE) between the predicted optical flow and the corresponding ground truth, i.e. $L2(\dot{f}, \dot{f}^*)$, and 2) the pixel-wise MSE between the predicted subject motion feature and the corresponding ground truth, i.e. $L2(\dot{p}, \dot{p}^*)$, as shown in Eq. 1.

$$min \quad \alpha * L2(\dot{f}, \dot{f}^*) + \beta * L2(\dot{p}, \dot{p}^*) \tag{1}$$

where $\dot{f}$ and $\dot{p}$ refer to the future dense optical flows $(f_{t+1}, ..., f_{t+N})$ and subject motions $(p_{t+1}, ..., p_{t+N})$, respectively. $()^*$ is used to distinguish the ground-truth from the prediction. We use α and β to balance the weight of the prediction of the optical flow and human pose. In our experiments, α and β are set as 1 and 0.3.

The first loss $L2(\dot{f}, \dot{f}^*)$ ensures a high-fidelity imitation of the planned camera's trajectories and poses. The second loss $L2(\dot{p}, \dot{p}^*)$ ensures the proper composition of the picture and the view of the moving subject.

3.2.3 Camera Motion Estimation

This section describes how to estimate the camera motion from optical flow during online filming. We apply a two-stage strategy to generate a camera trajectory: First, we use the learned model to predict the future optical flow $\{f_{t+1}, .., f_{t+N}\}$ for the new input images $\{I_t, .., I_{t-M+1}\}$. Second, according to the optical flow maps at time $[t+1t+N]$, we identify 800 matching points, based on which we can derive an essential matrix E [14]. We decompose E as Eq. 2 to obtain the 3DOF rotation and 3DOF translation of the camera at time $t+i$:

$$\begin{aligned} (p')^T K^{-T} E K^{-1}(p) = 0 \\ E = R[t]_\times \end{aligned} \tag{2}$$

where p and p' are homogeneous image coordinates of the start point and end point of the optical flow vectors. K, E, R and $[t]_\times$ refer to the camera intrinsic matrix, the essential matrix, the rotation matrix and the matrix representation of the cross product with the translation vector, respectively.

Because the essential matrix is up to scale (i.e. the scale of the translation is ambiguous), we apply a simple and efficient method to get the scale parameter before autonomous filming: the drone automatically moves backward to collect 10 images of a subject within 2 meters. We estimate the translation (up to scale) from the collected images based on the decomposition of the essential matrix. We calculate the scale parameter by dividing the camera trajectory from the drone's navigation system and the estimated translation. After initialization, the scale, as a constant factor, is multiplied to the camera translation estimated from optical flow as the final translation.

Once N steps of camera motions are obtained, we generate a smooth and feasible trajectory with a min-snap polynomial trajectory planning algorithm [43]. This trajectory will be sent to the actuator to control the drone.

3.3. Dataset Collection

We collect the professional-level outdoor video clips from a website *www.gettyimages.com*, which offers professional photography and videography. Specifically, we used three keywords "*aerial view, one man only, sport*" to initialize our search. This study focuses on imitate filming videos of human motions, thus the collected videos mainly contain three types of activities, i.e. walking, jogging, stretching and rotating. We excluded the searched video results which contain extremely poor lighting conditions, subjects taking up too small regions and/or being occluded for too long time during the video. As a result, we obtained 92 qualified videos, each of which is around 15-30 seconds long, yielding videos of totally 2284 seconds. We resized each video frame to 320x160 and down-sampled the video to frame rate of 3fps to adapt to the actual computation speed. In addition, we provide the ground-truth of optical flow and subject motion feature. More specifically, we apply the state-of-the-art optical flow method Flownet2.0 [22] to extract the ground-truth of optical flow. The ground-truth of subject

pose is based on the result of OpenPose, while we manually correct the misidentified skeleton joints to replace the original result.

Figure 4. The examples of the videos labeled with the style.

To learn different filming styles, we recruit 3 human annotators and asked them to manually group the videos based on the "filming styles". As the "filming styles" are actually subjective and are not explicitly provided, in this study the annotators determine the style subjectively according to an overall consideration of 1) the relative movement between the background scene and subject in a video and 2) the content of the scene.

Each video was labeled by 1 annotator and verified and corrected by the other 2 annotators. Each annotator could add a new style group if he/she thought a current video should not belong to any of the existing style groups. Eventually, the dataset is grouped into 6 styles (see Fig. 4) and Tab. 3 shows the statistics of the style annotations.

Table 3. statistics of the style annotations in our data

Style	A	B	C	D	E	F
Total Video	21	18	7	18	15	13
Length (seconds)	345	314	75	414	271	249

We analyze the distribution of different styles in terms of different features. We represent each video as a set of segments, each of which includes 9 successive frames. Each frame is represented by dense optical flow, the screen coordinates of the skeleton and the RGB images. We visualize the distribution of the videos with different styles in terms of three features after PCA (see Fig. 5). We can see that the optical flow, skeleton and RGB image have complementary relationship for filming styles segmentation, which verifies our intuition to fuse these feature to predict camera motion.

4. Experiments

In this section, we describe the experimental setup and the measurement metrics used for evaluation, followed by experimental results.

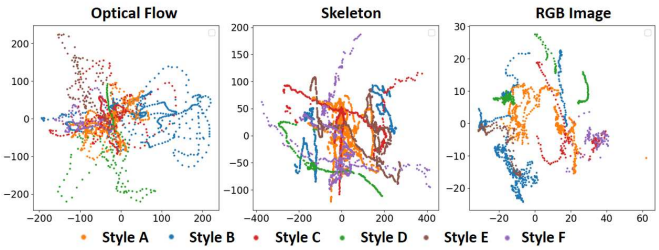

Figure 5. The distribution of 6 filming styles in terms of optical flow, skeleton and RGB image after PCA.

4.1. Experimental Setup

We split our dataset into 60 training videos and 32 test videos. The number of videos from style A, B, C, D, E and F is 14, 12, 5, 12, 10, 9 for the training set and 7, 6, 2, 6, 5, 4 for the testing set. For each training and testing video, we applied an overlapping sliding window with a length of 27 to generate a set of clips. The stride of the overlapping sliding window is 1. Accordingly, we generate a total of 5827 training clips and 3108 testing clips. We further augmented the training data by flipping each video clip along the horizontal axis, yielding 11654 training clips. We train our network on the Nvidia Tesla K50c and utilize Adamax [27] to perform the optimization, with a learning rate of 0.001.

We evaluate our method using two types of metrics:

1) The average endpoint error (AEE) [45] of the predicted optical flow which aims at examining the differences between the predicted camera motions and the true camera motions from experts. Specifically, each pixel (i, j) in the predicted optical flow map contains the horizontal and vertical components $(u_{i,j}, v_{i,j})$. The AEE is defined as the means of an absolute error between the estimated flow map (u, v) and ground truth optical flow map (u_{GT}, v_{GT}):

$$AEE = \sqrt{(u - u_{GT})^2 + (v - v_{GT})^2} \tag{3}$$

2) A subjective quality score obtained from a user study. We recruited 10 volunteers and each volunteer was asked to score the similarity between the recorded video with training videos (from 1: worst to 5: best). We calculate the average score of the 10 volunteers for each testing video and then average the score on all testing videos.

4.2. Ablation Study

In this subsection, we design the experiments based on our dataset to carefully analyze the impact of three factors on the final imitation filming results: 1) parameter selection, 2) inputs selection, and 3) the learning model.

4.2.1 Selection of M and N

We experimentally evaluate the combination of several different (m, n) values, where m and n denote the length of

input and output sequences. The psychological literature [40] indicated that the human brain system has a specific mechanism to split the process of perception into successive stimuli to reduce the complexity, and the temporal grouping of stimuli roughly has a temporal limit around 3 seconds. Therefore, we focused on analyzing video clips of 3 seconds, which is 9 frames based on the frame-rate (i.e. 3fps) of our videos. Accordingly, the potential parameter combinations investigated in this study are under the constraint that the summation of m and n is 9. . For each combination, we train a prediction model, denoted as Mm_Nn. For each training clip, we select the first m frames (i.e. I_1 ... I_m) as input and the sub-sequence $\{I_{m+1}$ ... $I_{m+n}\}$ as output (ground-truth). For each testing clip, we select the sub-sequence $\{I_{18-m+1}$ ... $I_{18}\}$ as input and the last 9 frames (i.e. I_{19} ... I_{27}) as output, because this scheme can guarantee the same ground-truth for different models.

Fig. 6 displays the AEE (unit: pixel) as a function of the number of predicted frames for different (m, n) values. It is noted that the length of prediction is not limited by the selection of N because the decoder can continuously predict the next frame by feeding itself with the current state vector. Three observations can be made from the results:

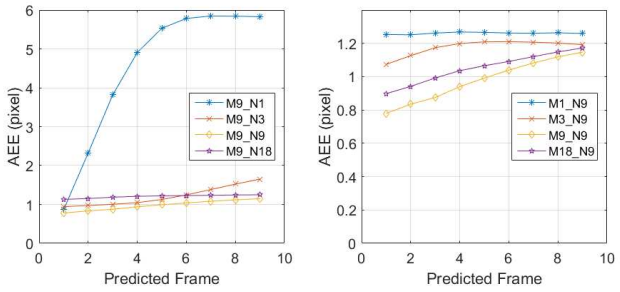

Figure 6. Comparison of the prediction models with varying settings (i.e., M and N) on the training sequence.

1) The prediction accuracy decreases as the length of the horizon increases.

2) There exists an optimal N for a given input length, i.e. $AEE(M9\_N9) < AEE(M9\_N3) \approx AEE(M9\_N18) < AEE(M9\_N1)$. Large N increases the difficulty of training even if it contributes to extending the prediction horizon.

3) The M is not always positively correlated with the prediction accuracy, i.e. $AEE(M9\_N9) < AEE(M18\_N9) < AEE(M3\_N9) < AEE(M1\_N9)$. Although larger M provides more context for prediction, it increases the probability of overfitting during training.

According to Fig. 6, we set M = 9 and N = 9 in the following experiments.

4.2.2 Impacts of Three Inputs

We exam the impacts of three types of inputs: optical flow (F), background feature (B) and subject motion feature (S). We train seven models with different combinations and Fig. 7 shows performance of all the models. When comparing the three models using only F, S and B respectively as the input, we observe that optical flow plays a greater role in predicting the future camera motions than background and subject motion, i.e. $AEE(F) < AEE(B) < AEE(S)$. Integrating either subject motion or scene features into the model F, denoted as S+F and B+F, could further reducing the prediction error. Combing all the three inputs, denoted as S+B+F, achieves the smallest AEE, indicating the complementary information from three types of inputs.

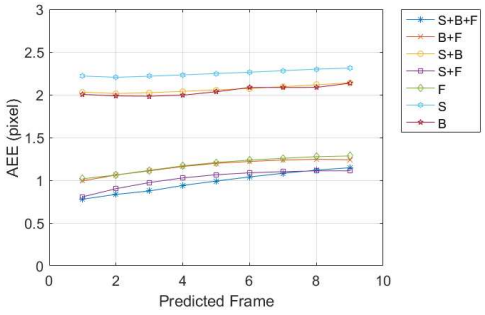

Figure 7. Comparison of different input combinations.

4.2.3 Impact of Human Motion Prediction Loss

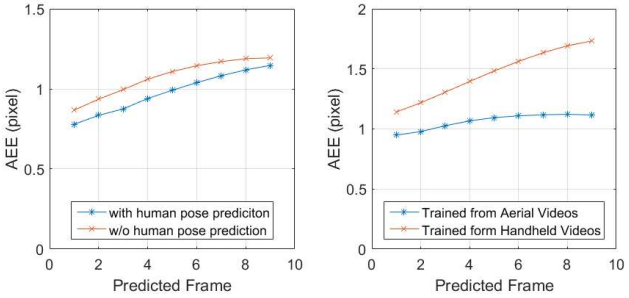

Figure 8. Left: comparison of the prediction models with/without the human motion prediction loss. Right: comparison of two models trained from aerial videos and handheld videos.

Fig. 8 (Left) compares the two models trained with and without the human motion prediction loss, i.e. $L2(\dot{p}, \dot{p}^*)$ in Eq. 1. Results show that including $L2(\dot{p}, \dot{p}^*)$ in the training process could further reduce the camera motion prediction error by 3.93%$\sim$10.20%. This is because the evolution of the subject's appearance is also related to the optical flow prediction. The loss function poses a constraint for learning filming skills, which reduces search space and enables an increase in efficiency of solving optimization problem.

4.2.4 Learning Ability of the Imitation Model

We evaluate learning ability of our imitation model by training two networks using two different types datasets: the collected aerial videos as described in Sec.3.3 and the human

motion videos captured using handheld devices downloaded from Youtube. Both training sets contain 60 videos, which generate 11654 and 12032 training clips, respectively. We evaluate both models on the 32 test aerial videos (Sec.4.1).

Fig. 8 (Right) illustrates that the model trained from the aerial videos can predict more accurate optical flow than the model from handheld videos because the handheld videos are different from aerial videos in terms of filming style. We can draw the conclusion that the imitation model can distinguish filming styles from different training videos.

4.3. Application to Drone Cinematography System

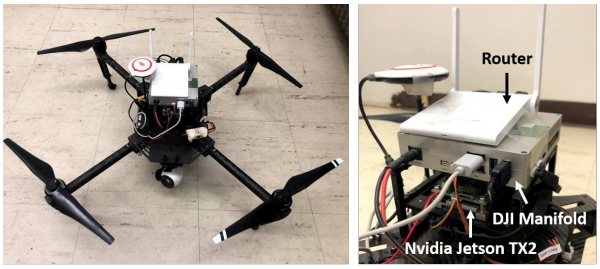

Figure 9. Left: top view of our prototype drone based on DJI Matrix 100. Right: the onboard processor module contains Nvidia Jetson TX2 and DJIManifold. The communication between onboard systems and ground station is bridged by a router.

In this subsection, we deploy our imitation filming method to a real drone platform for the autonomous cinematography task. Specifically, we build our drone cinematography system on the DJI Matrix 100 (see Fig. 9 (left)). We deploy the algorithm on two onboard embedded systems (Nvidia Jetson TX2 and DJI Manifold) (see Fig. 9 (right)) and a ground station PC (ThinkPad Intel i7). We mount a USB-powered router on the drone to enable wireless communication between the drone and the ground station.

Table 4. Comparison of imitation performance of different styles

A	B	C
4.14±0.73	4.42±0.49	4.57±0.29
D	E	F
4.29±0.61	4.14±0.24	4.14±0.49

We conduct a user study (as described in Sec. 4.1) to evaluate the quality of the autonomous drone cinematography system. Prior to the experiments, we trained the volunteers by showing them a collection of video pairs and the corresponding predefined score from 1 to 5 so that every volunteer follows the same criterion for their assessment. In the experiments, we use the drone system to automatically capture 30 videos, 5 videos for each filming style. Tab. 4 shows the mean score and its standard deviation for each style. In general, the average score for each style is above 4.0, indicating an overall satisfactory quality of the captured videos. Fig. 10 shows several snapshots of two videos autonomously captured by the model with style A, which share the similar style of zooming out.

Figure 10. The snapshots of two autonomously captured video.

Table 5. Subjective quality of videos captured by our autonomous drone cinematography system.

Proposed	Huang et al. [17]	ActiveTrack [9]
4.29±0.54	3.15±0.24	2.64±0.37

We also compare our system with two state-of-the-art autonomous drone cinematography systems, i.e. Huang et al. [17] and commercial product DJI drone *Spark*'s mode "ActiveTrack" [9]. Huang et al. [17] defined the viewpoint that maximizes the visibility of the human pose as the "best" viewpoint. DJI Active Track [8] locks the subject on the image center and keep the fixed viewpoint. We start at a fixed relative drone pose with respect to the subject and capture 10 video clips for each system. The results in Tab. 5 show that our work can capture more visual-pleasing videos than those from Huang et al. [17] and commercial product DJI drone *Spark*'s mode "ActiveTrack" [9]. Huang et al. [17] will capture jittering videos when the fast human motion generates dramatic change of the best viewpoint. The fixed viewpoint in ActiveTrack makes the viewer feel unexcited.

5. Conclusions

This work studies how to use imitation-based method to plan the camera motion for capturing professional human motion videos. We formulate and analyze this problem from three aspects: 1) the input image feature 2) the camera motion and 3) the learning model. In addition, we collect a dataset including professional human motion videos and manually group these video into six styles based on the camera trajectory. Based on the analysis on this dataset, we propose a learning-based framework which directly incorporates the video contents and previous camera motions to predict the future camera motions. Our experimental results on the datasets and showcases exhibit significant improvements of our approach over existing methods and our approach can successfully mimic the professional footage.

6. Acknowledgement

The work described in this paper was partially supported by a grant from Research Grants Council of Hong Kong.

References

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[2] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016.

[3] J. Chen and P. Carr. Mimicking human camera operators. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 215–222. IEEE, 2015.

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

[5] J. Chen and J. J. Little. Where should cameras look at soccer games: Improving smoothness using the overlapped hidden markov model. *Computer Vision and Image Understanding*, 159:59–73, 2017.

[6] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Studying aesthetics in photographic images using a computational approach. In *European Conference on Computer Vision*, pages 288–301. Springer, 2006.

[7] S. Dhar, V. Ordonez, and T. L. Berg. High level describable attributes for predicting aesthetics and interestingness. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1657–1664. IEEE, 2011.

[8] dji. `https://store.dji.com/guides/film-like-a-pro-with-activetrack/`, 2015.

[9] dji. `https://www.drone-world.com/dji-mavic-air-quickshot-modes/`, 2018.

[10] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[11] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[12] Q. Galvane, J. Fleureau, F.-L. Tariolle, and P. Guillotel. Automated cinematography with unmanned aerial vehicles. *In Proceedings of the Eurographics Workshop on Intelligent Cinematography and Editing*, 2016.

[13] Q. Galvane, C. Lino, M. Christie, J. Fleureau, F. Servant, F. Tariolle, P. Guillotel, et al. Directing cinematographic drones. *ACM Transactions on Graphics (TOG)*, 37(3):34, 2018.

[14] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[15] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun. Deep 360 pilot: Learning a deep agent for piloting through 360 sports video. In *CVPR*, volume 1, page 3, 2017.

[16] C. Huang, P. Chen, X. Yang, and K.-T. T. Cheng. Redbee: A visual-inertial drone system for real-time moving object detection. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1725–1731. IEEE, 2017.

[17] C. Huang, F. Gao, J. Pan, Z. Yang, W. Qiu, P. Chen, X. Yang, S. Shen, and K.-T. T. Cheng. Act: An autonomous drone cinematography system for action scenes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7039–7046. IEEE, 2018.

[18] C. Huang, Z. Yang, Y. Kong, P. Chen, X. Yang, and K.-T. T. Cheng. Through-the-lens drone filming. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4692–4699. IEEE, 2018.

[19] K. Huang, Q. Wang, and Z. Wu. Color image enhancement and evaluation algorithm based on human visual system. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 3, pages iii–721. IEEE, 2004.

[20] K.-Q. Huang, Q. Wang, and Z.-Y. Wu. Natural color image enhancement and evaluation algorithm based on human visual system. *Computer Vision and Image Understanding*, 103(1):52–63, 2006.

[21] K.-q. Huang, Z.-y. Wu, G. S. Fung, and F. H. Chan. Color image denoising with wavelet thresholding based on human visual system model. *Signal Processing: Image Communication*, 20(2):115–127, 2005.

[22] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, volume 2, page 6, 2017.

[23] N. Joubert, D. B. Goldman, F. Berthouzoz, M. Roberts, J. A. Landay, P. Hanrahan, et al. Towards a drone cinematographer: Guiding quadrotor cameras using visual composition principles. *arXiv preprint arXiv:1610.01691*, 2016.

[24] G. Kahn, T. Zhang, S. Levine, and P. Abbeel. Plato: Policy learning using adaptive trajectory optimization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3342–3349. IEEE, 2017.

[25] H. Kang, H. Li, J. Zhang, X. Lu, and B. Benes. Flycam: Multi-touch gesture controlled drone gimbal photography. *IEEE Robotics and Automation Letters*, 2018.

[26] Y. Kao, C. Wang, and K. Huang. Visual aesthetic quality assessment with a regression model. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 1583–1587. IEEE, 2015.

[27] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[29] C. Liu et al. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[30] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang. Rapid: Rating pictorial aesthetics using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 457–466. ACM, 2014.

[31] X. Lu, Z. Lin, X. Shen, R. Mech, and J. Z. Wang. Deep multi-patch aggregation network for image style, aesthetics, and quality estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 990–998, 2015.

[32] W. Luo, X. Wang, and X. Tang. Content-based photo quality assessment. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2206–2213. IEEE, 2011.

[33] Y. Luo and X. Tang. Photo and video quality evaluation: Focusing on the subject. In *European Conference on Computer Vision*, pages 386–399. Springer, 2008.

[34] L. Mai, H. Jin, and F. Liu. Composition-preserving deep photo aesthetics assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 497–506, 2016.

[35] A. K. Moorthy, P. Obrador, and N. Oliver. Towards computational models of the visual aesthetic appeal of consumer videos. In *European Conference on Computer Vision*, pages 1–14. Springer, 2010.

[36] T. Nägeli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges. Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization. 2(3):1696–1703, 2017.

[37] T. Nägeli, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges. Real-time planning for automated multi-view drone cinematography. *ACM Transactions on Graphics (TOG)*, 36(4):132, 2017.

[38] A. Y. Ng, S. J. Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[39] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

[40] E. Poppel. Lost in time: a historical frame, elementary processing units and the 3-second window. *Acta neurobiologiae experimentalis*, 64(3):295–302, 2004.

[41] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *arXiv preprint arXiv:1708.03852*, 2017.

[42] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[43] C. Richter, A. Bry, and N. Roy. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *Robotics Research*, pages 649–666. Springer, 2016.

[44] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

[45] N. Sharmin and R. Brad. Optimal filter estimation for lucas-kanade optical flow. *Sensors*, 12(9):12694–12709, 2012.

[46] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[47] J. Tang, X. Du, X. He, F. Yuan, Q. Tian, and T.-S. Chua. Adversarial training towards robust multimedia recommender system. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[48] F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

[49] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[50] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.