Now You Shake Me: Towards Automatic 4D Cinema

Yuhao Zhou\textsuperscript{1} Makarand Tapaswi\textsuperscript{1,2} Sanja Fidler\textsuperscript{1,2}
\textsuperscript{1}University of Toronto \textsuperscript{2}Vector Institute
\{henryzhou, makarand, fidler\}@cs.toronto.edu
http://www.cs.toronto.edu/~henryzhou/movie4d/

Abstract

We are interested in enabling automatic 4D cinema by parsing physical and special effects from untrimmed movies. These include effects such as physical interactions, water splashing, light, and shaking, and are grounded to either a character in the scene or the camera. We collect a new dataset referred to as the Movie4D dataset which annotates over 9K effects in 63 movies. We propose a Conditional Random Field model atop a neural network that brings together visual and audio information, as well as semantics in the form of person tracks. Our model further exploits correlations of effects between different characters in the clip as well as across movie threads. We propose effect detection and classification as two tasks, and present results along with ablation studies on our dataset, paving the way towards 4D cinema in everyone’s homes.

1. Introduction

Fast progress in deep learning together with large amounts of labeled data has enabled significant progress in tasks such as image tagging [16], object detection [14], action recognition [10], and image captioning [43]. Neural networks have also proven themselves as surprisingly good artists by repainting images in different styles [12], writing poems [18], and synthesizing music [5, 42]. With the emerging market of virtual reality, simulated roller-coasters, and infotainment, machines might also help us reach a new level of the entertainment experience.

In 4D cinema, the audience is taken on a wild ride through the movie: their seats shake when a high speed car chase unrolls on the screen, water splashes on their faces when a boat cuts through the Perfect Storm, and smoke veils around them when Clint Eastwood lights up yet another cigarette. While entertaining for the audience, such effects are not so fun to annotate for the movie creators. They are time consuming and require careful annotation of what physical phenomena is occurring at every time instant in the film [23]. The strength of the effect, and possibly direction is also important to faithfully recreate the fast-paced dynamic world for the audience.

In this work, we take a step towards promoting creation of 4D cinema by automatically parsing detailed physical effects in movies. In particular, given a streaming video, we aim to detect both which effect is being applied to each of the characters in the scene (or camera), as well as to predict accompanying details such as the intensity of each effect, its duration and possibly direction. While inferring effects from videos has clear significance for the entertainment industry, we believe it also has value for building intelligent robots in the future. When faced with the real world, robots will need to foresee physical forces based on current visual or audio information in order to cope with them.

Due to unavailability of an existing dataset of this form, we first collect the Movie4D dataset containing rich annotations of physical effects in feature films (Fig. 1). Our dataset contains 9286 effect annotations with time stamps and accompanying details. The effects are also grounded to either the camera’s point of view, or to a particular character in the clip. These effects take place in various scenes, ranging from heroic battlefronts to everyday lives.

We propose a model to parse effects from untrimmed videos and ground them to characters’ tracks. We formalize the task as performing inference in a Conditional Ran-
dom Field, that exploits potentials extracted from multiple modalities (visual, audio, and semantic) via neural networks. Our model further profits from correlations between effects applied to different characters in the same clip (such as one character being exposed to a shake likely means that the other character experiences the same effect), as well as across clips (some effects like water splashing are long in duration). We showcase the model through ablation studies, and point to challenges of the task.

Our code and data will be released\(^1\) in order to inspire more research in bringing 4D cinema to everyone’s homes.

2. Related Work

Video analysis, especially movies and TV series have several research directions. Among them, some of the most popular tasks are automatic person identification [4, 7, 9, 27], pose estimation [8], describing short video clips using one sentence [29] or aligning videos with plots [38] and books [39, 47]. Video-based question-answering is also growing in popularity, and among these MovieQA [40] and PororoQA [19] are based on movies and TV series.

Actions and interactions have also been studied in the context of movies. Hollywood2 [21] aims at predicting a few action classes given short clips, while human interactions such as hugging, kissing, are studied in [26]. A few approaches aim at finding people looking at each other [24]. With deep learning, and the need for larger datasets, action recognition (not necessarily in movies) has grown via ActivityNet [17], the THUMOS challenge and related UCF dataset [36]. Moving away from classifying actions given a segmented clip, there is a drive to detect and classify actions in longer untrimmed videos.

In the related domain of audio analysis, AudioSet [13] is a large collection of audio events that range from human and animal sounds, musical instruments, to everyday environmental sounds. A recent audio-video dataset FlickrSoundNet [3] has enabled training audio-visual models in an unsupervised manner [2, 3]. Movie effects are audio-visual too, and we exploit these modalities in our models.

**4D effects.** Over the years, movie budgets have increased and facilitated use of dazzling special effects [1]. In this paper, we propose classification and detection of such effects and physical interactions in movies, as experienced by the camera and characters. Inspired by semantic role labeling for images that requires to predict the verb and the corresponding role-noun pairs (imSitu [44]), our effect annotations (e.g. wind) come with a variety of details that determine the intensity (e.g. strong), direction (e.g. from left) and even sub-types (e.g. cold wind).

In the past, several attempts have been made to predict a similar range of effects, however, in different, and importantly, isolated contexts. Classification of weather conditions has been analyzed for driver-assistance [30], while detecting water [25], and especially rain [11] is of special interest. In the context of fire safety, work by [6] aims at detecting smoke.

There is work on real rendering of audio-visual signals to sensory devices and chairs. [22] aims to translate audio signals (movies, hand-held games, etc.) onto a vibro-tactile sensory device, with [34] focusing on rendering gunshots. Probably the most related to our work, [23] analyzes 10 real 4D films with about 2.2K effects and manually groups them based on viewer experience (e.g. motion, vibration). As motion forms a large chunk of 4D effects, [23] employs optical flow along with Kalman filtering to use video motion to control the chair. Our work is different on two key fronts: (i) we use audio-visual information to detect and classify effects in movies, and even those that were not originally made for 4D; and (ii) our dataset annotations and model reason about which characters experience the effects. We also collect a significantly larger dataset with 63 movies and over 9.2K effect annotations.

### 3. Movie4D Dataset

We first introduce our dataset, by describing the annotation process, statistics, and proposed tasks. In the next section, we propose a model that aims to solve these tasks.

We build the Movie4D dataset to analyze the detection and detailed parsing of effects in films. Our dataset consists of 654 five-minute clips obtained from 63 movies. Most of our movie genres are action/venture, and sci-fi as they typically contain the highest number as well as diversity of effects. However, Movie4D also features films from drama, comedy, and romance, that could be used as a proxy to understanding effects in the real world.

#### 3.1. Annotating Effects

We are interested in annotating effects in video clips, including the start and end timestamps, effect type, and effect-specific details such as intensity, direction, and sub-type.

<table>
<thead>
<tr>
<th></th>
<th>TRAIN</th>
<th>VAL</th>
<th>TEST</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movies</td>
<td>50</td>
<td>7</td>
<td>6</td>
<td>63</td>
</tr>
<tr>
<td># Effects</td>
<td>7283</td>
<td>816</td>
<td>1187</td>
<td>9286</td>
</tr>
<tr>
<td># Instantaneous effects</td>
<td>1492</td>
<td>115</td>
<td>268</td>
<td>1875</td>
</tr>
<tr>
<td>Effect avg. duration</td>
<td>5.9</td>
<td>6.8</td>
<td>6.6</td>
<td>6.1</td>
</tr>
</tbody>
</table>

Table 1: Summary of the Movie 4D dataset.
We develop a web annotation interface, where the annotator is allowed to select a movie, and then browse through a set of clips. As preprocessing, we split the movie into 5-min clips and provide a few to the annotator. After selecting a clip, the annotators are presented with the video, and a dynamic questionnaire interface that allows them to add new effects (details in supp. material). While browsing (watching) the clip, the annotator first adds the type of the effect, such as shake or wind, and a few mandatory fields common to all effects: (i) start and end time; (ii) intensity; and (iii) point of view (POV). Effects that have a duration less than one second are referred to as instantaneous. For each effect type, we present additional effect-specific fields which the annotator can fill. Fig. 2 provides the list of effect types, details as well as examples. We explain our effect annotations in more detail below.

Intensity. We provide three options: mild, medium, and strong. Mild effects are common on a daily basis, such as a light blow of wind. Medium effects are significantly more noticeable, but are still acceptable to most people. An example would be stronger wind or a pouring rain. Strong effects are not common in normal life and may cause pain or discomfort, such as severe shaking due to explosion or earthquakes, or strong winds due to hurricanes.

POV denotes the subject that experiences the interaction within the video. We allow annotators to choose from a cast of main characters that we provide (seven per movie on average), as well as the camera. A camera POV indicates that the effect is applied to the cameraman or the observer of a scene from a first-person perspective.

Our annotators were hired from the freelance website Upwork, that facilitates interaction through a message board. We trained the annotators for roughly two hours and gave them constant feedback for the first two movies they annotated, in order to ensure consistency. Each annotator was asked to annotate a full movie. The annotators were paid by the hour.

1. Shake. The POV experiences continuous or sudden spatial motion. Shake has detail direction with options: left-right, front-back, up-down, all-around, and other.

2. Splash is caused by water (or any liquid) splashing onto the subject. Splash has detail direction with options: front, back, left, right, top, bottom, all-around, and other.

3. Wind is a result of natural weather phenomenon or artificial manipulations by machines (such as standing on a fast moving boat). The detail direction for wind has the same options as that of splash. Wind also has detail type with the following options: hot, cold, and normal.

4. Physical Interactions are effects defined between two characters, such as fighting. Physical interactions have detail type with the following options: hit, pinch, twist, string, rub, drag, massage, impact, gunshot, and other. We also ask our annotators to select the source and target of the physical interaction (e.g. ‘A drags B’, where A and B are characters from the cast).

5. Light effects are defined only for those that involve an artificial light source. The detail direction has the same options as splash and wind. Light effects also have detail type with different colors: white, red, yellow, orange, green, blue, purple, and other.

6. Weather effects are usually subject to both the camera and all characters in the scene. Weather has detail type and comprises: extremely-sunny, rain, snow, fog, wind, snowstorm, other.

7. Temperature is annotated as an effect when the ambient temperature is not normal. It has type high and low.

8. Liquid Surrounding indicates that a large portion of the character/camera is submerged in water or other liquids. The detail type tells us the type of liquid: water, or other.

9. Gravity effects are annotated only when POV experiences unnatural low/high gravity forces. They have the detail type with options: high gravity (acceleration), low gravity, and zero gravity.

Person tracks. As our effect reasoning requires determining POV, we ground this information to character tracks in the clips. In particular, we perform person tracking in every
shot of the clip, and ask the annotators to assign a character name from the cast list to each track. We use person detections from the YOLO9000 object detector [28], and combine subsequent detections into person tracks based on spatial overlap. 41.4% of our tracks correspond to main characters and 30.3% to background characters. We obtain several false positives due to detections spaced at 3 frames per second.

3.2. Dataset Statistics

We collected a total of 9286 annotations from 654 clips each 5-min long. Table 1 provides a summary of different features about our data, along with train-val-test splits. We create splits with disjoint movies and achieve a balance between movie metadata and effect annotations.

The distribution of the number of effect classes and their durations is presented in Fig. 3, and we see that light and wind are dominant effects. Fig. 4 presents the density of effects in a few example movies. We select 15 movies from various genres and compute the effect duration within them. Dramatic effects such as shake, liquid surrounding, and wind are more pronounced in action-packed movies such as adventure and sci-fi.

In Fig. 5, we show the t-SNE [41] visualization of the top 40 characters based on amount of time spent with effects. The effect duration is used as a feature for clustering. We observe that characters from sci-fi movies such as Interstellar, Gravity, and Iron Man are grouped together at the bottom (due to the zero-gravity floating and flying). In contrast, characters from adventure/action films such as Lord of the Rings and Hunger Games that experience natural phenomenon (wind, weather) are grouped at the top.

4. Effect Parsing in Videos

We propose models for effect detection and recognition. We first describe some preprocessing and introduce notation. Then, we present the neural architecture that is used to extract various features and perform classification. This acts as a baseline for our tasks. The classifier outputs are exploited as unary potentials in a Conditional Random Field (CRF) that performs joint reasoning about the effects within and across movie shots and threads. We address both trimmed effect recognition, and the more challenging untrimmed effect detection and parsing.

4.1. Video preprocessing

Careful consideration of shot boundaries and threading is important in order to compute features that are meaningful, as well as to exploit scene and filming priors. For example, wind effect typically applies to all characters as well as the camera in a shot, and possibly spans multiple neighboring shots. Similarly, if one character experiences a physical interaction it is highly likely that another character should exist and also undergo a physical interaction within the same shot or even thread.

**Shots.** Given a (5-min) clip from our dataset, we first detect shot boundaries using the motion-compensated difference between two consecutive frames [45].

Fig. 6 presents the ratio of effects applied to characters or camera. Effects due to environmental conditions apply more frequently to the camera. On the other hand, shake and splash are very human-centric.
Sub-Shots. Shots longer than 3s are further divided into sub-shots capped at a maximum duration of 3s. We adopt these as our primary unit of analysis. While the average shot duration is 3.65s, some shots in our movies are longer than 10s. Using sub-shots (instead of shots) reduces the noise from neighboring non-effect frames, while still providing enough audio-visual content to retain relevant effect information. Note that shots that are less than 3s are equivalent to their sub-shots. On average, our sub-shots are 2.49s long, and are assigned an effect label if they have a 10% or greater overlap with the effect annotation.

Threads. We thread shots taken from the same camera angle and viewpoint using SIFT-based matching [37].

4.2. Neural Architecture

Effects in movies are dominated by their audio-visual nature. In a video clip, we denote the audio content of a sub-shot as $u_t$ and visual content as $x_t$. We use $i = 1, \ldots, |x_t|$ to index frames within the sub-shot, i.e. $x^i_t$ denotes the visual feature of frame $i$ in the sub-shot. We extract several features using networks pre-trained on different tasks:

(i) **Visual features** are the core of our model and intuitively are useful to detect all effect types. Within each second, we sub-sample three frames, and extract features from the pool5 layer of the VGG19 model [35] pre-trained on ImageNet [31]. As a sub-shot is capped at 3s, it corresponds to a maximum of 9 visual frames and features. We denote each frame’s representation as $v^i_t = \phi_v(x^i_t)$ and mean pool across the spatial grid to obtain $v^i_t \in \mathbb{R}^{512}$. Optionally, we adopt a two-layer MLP (512-128-1) to compute self-attention weights to exploit spatial features (on the $7 \times 7$ grid) and replace the mean pool by a weighted average.

(ii) **Optical Flow features** form a visual representation that encode motion and are useful to detect effects such as wind, splash, shake, etc. We use the temporal stream of the two-stream network [10] trained for action recognition, and encode a stack of 10 optical flow images for each visual frame to obtain $f^i_t = \phi_f(x^i_t), f^i_t \in \mathbb{R}^{512}$. Similar to Visual features, self-attention weights are used.

(iii) **Audio features** are complementary to images, and are being used successfully for unsupervised audio-visual learning [2, 3]. We extract audio features for 1s raw audio samples using the SoundNet8 model [3] from the pool5 layer, $a^i_t = \phi_a(u^i_t), a^i_t \in \mathbb{R}^{256}$. In conjunction with the image, we expect audio to help detect effects such as swooshing winds, splashing water, or a mechanical shake.

(iv) **Object detections** can play a complementary role to image features. For example, presence of people may indicate physical interaction, while a car suggests shake. Based on predictions from the YOLO9000 object detector [28], we choose 550 most significant classes and form a sparse feature vector corresponding to the object detection probabilities $a^i_t = \phi_o(x^i_t), a^i_t \in \mathbb{R}^{550}$ for each frame.

Finally, we obtain a sub-shot representation by concatenating embedded features through linear layers:

$$\phi(x^i_t, u^i_t) = [W_vv^i_t, W_ff^i_t, W_oa^i_t, W_uo^i_t],$$

Each linear layer embeds features into a $D = 512$ dim space. Using pre-trained models as feature extractors was a crucial requirement to train good models on our dataset.

Classifiers. We build two-layer MLP classifiers on the sub-shot representations to predict: (i) **count** ($C^n$): number of effects present in the current sub-shot (0, 1, 2, 3); (ii) **effect** ($C^e$): the effect labels (9 classes); (iii) **effect intensities** ($C^i$): three classes; and (iv) **effect-specific details** ($C^{di}, \ldots, C^{dG}$) such as type of physical interaction or direction of wind (each with its own independent MLP).

The count, effect, and intensity classifiers have hidden layers with size corresponding to the input, i.e. $h_n, h_e, h_i \in \mathbb{R}^{2048}$. As detail classifiers have much less training data, we set the hidden layer $h_d \in \mathbb{R}^{512}$. The output layer computes predictions in one-of-K classes:

$$\hat{y}^i_t = W_2 \cdot \text{ReLU}(W_1 \cdot \phi(x^i_t, u^i_t)) = C(\phi(x^i_t, u^i_t)).$$

Biases are ignored for notational brevity. We leverage the architecture (summarized in Fig. 7) to train task-specific models in the following.

4.3. Trimmed Video: Effect recognition

Our first task is to predict the effect type and the corresponding details when provided with a trimmed clip (known to contain an effect). We treat this prediction as tagging, i.e. we do not take into account POV information.

We first compute the set of sub-shots that overlap with the trimmed clip and use them to obtain predictions $\hat{y}^i_t$. For a given effect with sub-shots $x_1, \ldots, x_T$, we compute the final prediction $\hat{y}$ by

$$\hat{y} = \max_{t} \frac{1}{|x_t|} \sum_{t} \hat{y}^i_t.$$  

Averaging results within frames of a sub-shot ($\sum_{t} \hat{y}^i_t$) improves robustness, and selecting the highest scoring sub-shot ($\max_{t}$) reduces noise.
Learning and Inference details. While the pre-trained feature extractors are fixed, the feature embedding layers (Eq. 1) and effect \( C_e \), intensity \( C_i \), and detail \( C_d \) classifiers are trained. We adopt the cross-entropy loss for each classifier and optimize our model with Adam [20] using a constant learning rate \( 1e - 6 \). We found that training all classifiers jointly by accumulating losses worked well. Due to the large class imbalance at both levels: effects and detail predictions, we use an inverse frequency weight capped at 50 to avoid overfitting. Additionally, we use a dropout rate of 0.3 for MLP classifiers and set weight decay to 0.1. We choose a model checkpoint with the highest accuracy weighted on the validation set.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

4.4. Untrimmed video: Effect detection

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

4.4. Untrimmed video: Effect detection

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

4.4. Untrimmed video: Effect detection

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.

While trimmed video effect recognition focuses on classifying effects globally over a trimmed clip, we now aim to predict the effects experienced by both the characters and camera, and localize them in time. In this scenario, we are given an entire video along with sub-shot boundaries, threading information, and person tracks within each sub-shot. While tracking is performed within a shot, we divide the track across sub-shot boundaries if required. Our goal is to determine the effect type, start and end-time and POV.
CRF learning and inference. Our CRF may contain cycles due to thread edges (see Fig. 8), and thus inference is NP-hard. To perform inference we use distributed convex belief propagation [32], which has convergence guarantees. To learn the weights, we use the primal-dual method [15, 33], and use the typical 0-1 loss.

Sub-shot predictions to time intervals. Inspired by its success in untrimmed action recognition, we group sub-shot level predictions using the watershed transform [46] to obtain contiguous time intervals with effect predictions. Each of the 9 classes are processed separately to obtain effect detections of the form: \((t_{\text{start}}, t_{\text{end}}, e)\).

Details. The neural network is separately trained to predict unaries using cross-entropy loss. Similar to the trimmed model, we train our unaries prediction model with Adam [20]. Recall that we extract image features from the pool5 layer that provides 512-d vectors in a \(7 \times 7\) spatial grid. In contrast to the camera unaries, person track unaries average features within the region of interest based on the detection bounding box. We pick a model checkpoint that performs well at predicting effect existence (based on number of effects) and effect recognition.

5. Experiments

We first discuss the metrics for our new dataset and tasks.

Trimmed video: Effect recognition metrics. We propose several metrics to evaluate effect and details prediction when given a trimmed video. Our first metric \(E\) is effect classification accuracy. We propose intensity-weighted accuracy \(IE\), as a viewer experience metric that incorporates user annoyance when mild/strong effects are misclassified: \(1\times\) (mild), \(2\times\) (medium), and \(3\times\) (strong).

To evaluate detail prediction, we use a metric similar to [44]. We introduce \(D-GT\) that measures the fraction of details that are correct for each sample given GT effect label. Additionally, \(DA-GT\) measures the fraction of samples that have all details correct. Similarly, detail prediction using predicted effect is evaluated by \(D-PR\) and \(DA-PR\).

Finally, we present a slightly modified form of a confusion matrix. As multiple effects can co-occur, one trimmed clip could correspond to more than one effect label. In such a case, if we are able to correctly classify one of the effects, we say that the other effects are “missed”, but not “misclassified”. For example, during an explosion clip with \textit{light} and \textit{shake}, if our model only predicts \textit{light}, we count light as correct, and shake as missed.

Untrimmed video: Effect detection metrics. We consider two POV paradigms: (i) all effects are mapped to the camera (similar to trimmed); or (ii) both camera and characters experience effects. Nevertheless, given the entire video we are required to predict effect start- and end-times along with effect labels and POV.

<table>
<thead>
<tr>
<th>Model</th>
<th>E</th>
<th>IE</th>
<th>D-GT</th>
<th>DA-GT</th>
<th>D-PR</th>
<th>DA-PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>11.1</td>
<td>11.1</td>
<td>23.6</td>
<td>3.8</td>
<td>2.6</td>
<td>0.4</td>
</tr>
<tr>
<td>Ours</td>
<td>41.5</td>
<td>42.0</td>
<td>35.5</td>
<td>9.9</td>
<td>15.7</td>
<td>5.3</td>
</tr>
<tr>
<td>Ours + Attn.</td>
<td>43.7</td>
<td>45.9</td>
<td>35.8</td>
<td>9.9</td>
<td>15.8</td>
<td>4.6</td>
</tr>
<tr>
<td>All - visual</td>
<td>26.9</td>
<td>28.5</td>
<td>36.2</td>
<td>10.4</td>
<td>10.2</td>
<td>3.0</td>
</tr>
<tr>
<td>All - optflow</td>
<td>32.1</td>
<td>32.4</td>
<td>33.0</td>
<td>8.4</td>
<td>10.9</td>
<td>3.0</td>
</tr>
<tr>
<td>All - audio</td>
<td>40.7</td>
<td>41.9</td>
<td>35.0</td>
<td>9.4</td>
<td>15.2</td>
<td>4.5</td>
</tr>
<tr>
<td>All - objdet</td>
<td>36.7</td>
<td>37.4</td>
<td>37.8</td>
<td>11.3</td>
<td>15.5</td>
<td>4.5</td>
</tr>
</tbody>
</table>

Table 2: Results on the test set for trimmed effect recognition task with ablation study for different features (no attn.). Random indicates the performance when treating all classes as equally likely. \(E, D-PR\) and \(DA-PR\) are important metrics for future comparison.

5.1. Dataset Quality

For 15 clips (5 min each) from 5 movies, we gathered 3 sets of labels from different annotators. We evaluate human effect detection performance by comparing all pairs of annotations (one as GT other as pred.), using the F1 metric. Temporal detection and localization is a hard task even for humans (also seen in action localization), and we obtain an average F1 score of 62.6% (54% - 67% for each movie).

We also analyze effect classification agreement in trimmed clips among humans using Amazon Mechanical Turk (AMT). Each clip was shown to 5 workers. At least one worker agreed with our label for 90% of the clips. When 4 of 5 workers provide the same label, this corresponds to an accuracy of 88.4%. As each clip can exhibit multiple effects (e.g. \textit{shake} and \textit{wind}), it is not necessary to obtain a clear majority.

5.2. Trimmed video: Effect recognition

We present the effect recognition results on the test set in Table 2. A baseline (row 1) that chooses 1 of \(K\) classes with equal chance has 11.1% accuracy, however, rarely gets the effect and details all correct (0.4%). Selecting the most likely label (\textit{light}) can achieve 25.8% effect accuracy.

In comparison, our neural model performs much better with an accuracy of 41.5%. With attention, we obtain 43.7%, and intensity-weighted accuracy of 45.9%. We believe that IE does not differ much from accuracy \(E\) partially due to inconsistencies in intensity annotations. Finally, our
model is able to predict the effect label and all details correctly for 5.3% of all annotations, hinting towards the difficulty of the task.

We also present an ablation study evaluating the importance of each feature stream by leaving one out at a time. The visual features play an important role, followed by motion (optical flow) features. This makes sense as light, physical interaction and shake are among the dominant classes. Objects are also quite important (e.g., cars shake) validated by the approximate 5% drop in effect prediction accuracy when ignoring them. Finally, audio features seem to have smallest contribution, however, do affect DA-PR.

Fig. 9 presents the modified confusion matrix. The confusion between light - temperature, or splash - wind (possibly because of motion), however, max is able to predict correctly.

5.3. Untrimmed video: Effect detection

We present effect detection results in two parts. First, in Table 3 we show the performance of detecting the presence of and classifying effects for each processing unit (sub-shot camera and/or track). The top part (rows 1-5) display results when all effects are mapped to the camera POV. We see the impact of different pairwise potentials: sub-shot, shot, and thread (rows 2-4), while, row 5 corresponds to the best result when using all pairwise terms. A large 8.8% boost in effect accuracy and a substantial 1.9% increase on Exist AP is obtained over the unary outputs from the neural model.

The bottom part presents results when considering effects separately for camera and tracks POVs. Row 7 and 8 show the impact of the CRF, and the final pairwise potential connecting camera nodes with tracks \( P_p \). We observe a small 3.5% improvement in effect accuracy, however detection AP reduces a little.

We combine the unit predictions into time intervals, and show results for comparing ground-truth and predicted time-intervals in Table 4. Note that this task is considerably harder as we need to predict a contiguous set of sub-shots correctly in order to obtain good time boundaries. When assuming all effects apply to camera POV, the CRF provides a 8.4% boost in F1 measure. However, when analyzing camera and person tracks, the improvement is small at 0.6%.

6. Conclusion

We introduced the Movie4D dataset consisting of 63 movies and 9286 effect annotations that enlist physical and special effects in movies along with details such as duration, intensity, effect sub-types and direction. We presented a thorough exploration of the dataset showcasing its features. We proposed a CRF model that combines cues from a multimodal neural network while respecting shot boundaries and threading information in a video. We evaluated our approach through various ablation studies, pointing to exciting avenues going forward.

Acknowledgments. Supported by NSERC. We thank NVIDIA for GPU donation. We thank Relu Patrascu for infrastructure support, and UpWork annotators for data annotation.
References


Appendix

We present several details about our dataset, and few additional results. In addition to this document, please note that the supplementary material also includes: (i) several example video clips showing ground-truth effect annotations and predicted results; and (ii) a web page with GIF visualizations (effects are best understood by watching videos instead of static images) for all 9 effects spread across the three intensities. We link to and discuss them in the document below. To download the GIF visualization: www.cs.toronto.edu/~henryzhou/movie4d/CVPR2018_Movie4D.zip. To download the qualitative video result: www.cs.toronto.edu/~henryzhou/movie4d/example_qualitative_result.zip.

![Video and Annotation Interface](image)

Figure 11: Web annotation interface. The annotators watch the full untrimmed video shown on the left. Upon selecting the effect type, the questionnaire asks to fill in corresponding details related to that effect. The annotators are required to answer all questions and submit the annotation before proceeding.

A. Annotation interface and effects

Fig. 11 provides a glimpse into our annotation interface. The annotators switch between video clips by clicking on the list of videos (below the video), can see the list main characters in the movie (bottom left), and answer a dynamic questionnaire that chooses questions based on previously chosen answers. Also note that the start and end-timestamps are grabbed from the video by pressing buttons thus minimizing human error in annotating the duration in seconds.

Fig. 12 presents the effect annotations for the clip with highest number of effects (count). Note how multiple effects can co-occur, while other parts of the video have no effects.

Fig. 13 presents all effect types in our movies along with mild, medium, and strong intensities. We explain each sample from the effects in the figure caption in detail. Additionally, as effects are seen best as videos, we create short 2 second GIF visualizations for the same examples. Please open effect_intensity.html in your browser to view them.
Figure 12: Example annotations made within a clip from the movie: *Thor: The Dark World*. We plot time on the x-axis and create a small rectangle for each effect type (based on color) corresponding to one second in the video. As can be seen, multiple effects do occur concurrently during intense scenes. On the other hand, possibly during dialog, effect annotations are sparse.

B. Dataset analysis

We extend Fig. 5 of the main paper (t-SNE [41] plot for 40 characters) in Fig. 14 showing 60 characters grouped by effect duration. A similar semantic grouping is observed, where sci-fi, action, and adventure characters cluster together.

Similarly, in Fig. 15, we show a t-SNE visualization of the movies themselves based on their effect density (fraction of time spent with each effect type).

Finally, we explore correlations between the number of effects in a movie and its “coolness”. We evaluate this based on the movie ratings, gross revenue, and budget (all obtained from IMDb). Fig. 16 presents scatter plots for each of the above metrics vs. the amount of time for which a movie has effects (density). While movies with more effects typically require higher budgets, good storytelling does not seem to depend entirely on the density of effects (*e.g.* *Dark Knight, Pulp Fiction*). However, having more effects does not hurt in general.

C. Video results

Results for effect detection and classification are best seen in video form. We present several example video clips depicting the ground-truth and predicted effect annotations. For simplicity of video creation and to see the effects clearly, effect and prediction labels change only at each second. Due to the size of the videos, they are included on our website instead of supplementary materials.

D. Effect correlation

We end with confusion matrices for a study of effect correlation. Fig. 17 presents the correlation between a pair of sub-shots with a gap of $n$ in between. We drop sub-shots pairs if either does not have any live effects. We see a strong diagonal indicating effects last for multiple sub-shots and/or are repeated within a period of time. Additionally natural effects (*Shake, Wind, Temperature, Light*) seem to be more correlated as compared to others like *Gravity*. 
<table>
<thead>
<tr>
<th></th>
<th>a. Mild</th>
<th>b. Medium</th>
<th>c. Strong</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Shake</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Splash</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Gravity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Light</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Wind</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Weather</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Temperature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Liquid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>Physical Interaction</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 13: Example frames for each effect at the three different intensity levels. Below we describe in one sentence the scenario for each example. **Shake** 1a. Characters are driving a car on a bumpy road. 1b. A character is sitting in a spacecraft traveling at extremely high speed. 1c. A missile hit the target, resulting in a strong explosion. **Splash** 2a. Dogs are playing in the water at the beach. 2b. Characters are fighting on a motorboat sailing in the sea. 2c. A pirate ship comes up from under the sea. **Gravity** 3a. A character jumps off from a high ground (zero-gravity for a short period of time). 3b. A character accelerates on his motorcycle at high speed. 3c. Multiple characters are floating at zero-G inside the space capsule. **Light** 4a. A flashlight flaring towards the camera. 4b. A projector shines at the back of the room. 4c. A character travels through a portal to another world. **Wind** 5a. A gentle breeze touches character’s face on a cliff. 5b. The sea wind blows away sailor’s hat. 5c. A character rides a flying create hunting from the sky (wind in his face due to motion). **Weather** 6a. Light snow at a party. 6b. Rain in London. 6c. Soldiers are crossing a jungle in the rain. **Temperature** 7a. Characters gather around at a campfire. 7b. A character visits a village on a plateau in Tibet. 7c. A character was surrounded by burning woods. **Liquid** 8a. Two characters are trapped on a ship which is about to sink. 8b. A character finds herself awoken in the sea. 8c. A character sinks into a lake. **Physical Interaction** 9a. Two characters are shaking hands. 9b. The middle character is being forced out of a carriage as a hostage. 9c. A character (on the right) is about to counter a gorilla attack.
Figure 14: Character t-SNE based on effect duration. Using the top 60 characters and plot them. The same grouping behavior still exists: Sci-Fi movies cluster on the right, action movies cluster on the left, adventure movies cluster in the bottom.
Figure 15: Movies are embedded using t-SNE. We computed the density of each effect for each movie and select the top 50 movies containing the most effects. The cluster shown in the t-SNE visualization reveals genre and even movie content. Sci-Fi movies form a cluster in the top-left of the figure. Drama and comedy are mostly seen on the right. And adventure/action movies are mainly clustered in the bottom.

Figure 16: Movie performance as measured by rating, budget, and gross revenue vs. density of effects. More effects seem to require higher budgets, but also have more revenue. However, the movie rating depends more on the storytelling rather than density of effects.
Figure 17: Effect correlation graph: we examine the causal effect of effects. For each sub-shot, we look ahead the next $n$ sub-shots and see if the current effect has an impact on the next sub-shots. From left to right of the figure, $n$ is 1, 5, 10 sub-shots ahead. We see that a strong diagonal for all plots indicating that effects of the same type tend to last for a long period of time, or occur again soon.