The AIIDE-17 Workshop on
Intelligent Narrative Technologies
WS-17-20

A Specialized Corpus for Film Understanding

David R. Winer,[1] **Joseph P. Magliano,**[2] **James A. Clinton,**[2] **Aidan Osterby,**[2]

Thomas Ackerman, ASC,[3] and **R. Michael Young**[1]

[1] School of Computing
University of Utah
Salt Lake City, UT, USA
{drwiner|young}@cs.utah.edu

[2] Department of Psychology
Northern Illinois University
DeKalb, IL, USA

[3] School of Filmmaking
University of North Carolina School of Arts
Winston Salem, NC, USA

Abstract

We have developed a cinematic annotation scheme and used that scheme to create a shot-by-shot annotated corpus of similar scenes to aid with research in film understanding. This paper introduces and describes the scheme and the created corpus, discusses some of its merits, and summarizes some basic findings.

Narrative films, like narrative texts, are useful for modeling the way we understand scenarios and communicate them. Recent work in computer vision for film will likely lead to new tools for film analysis (Suchan and Bhatt 2016), and many of the techniques used for narrative text understanding would be applicable, such as building narrative chains or scripts (Chambers and Jurafsky 2008; Hu et al. 2013; Pichotta and Mooney 2014). Films are valuable as an alternative or complementary source because films more directly interface with our perceptual and visual-spatial apparatuses than texts (Bordwell 2013). Since much of our interaction with the world is made in this way, research in film understanding will provide insights into the way we make meaning about our visual experiences which would be highly valuable for intelligent agents operating and communicating in virtual worlds or in the real world (Spranger, Suchan, and Bhatt 2016). Also, progress in film understanding will benefit the growing number of film-specific applications such as automated cinematography and other tools for the complex task of producing films and animations (Ronfard 2017).

To advance research in film understanding, we have assembled a shot-by-shot[1] annotated corpus of scenes all drawn from films of the same genre and conveying the same type of activity. There are three main features of this data set: the scenes are all similar, each shot is coded with cinematic features, and character actions are labeled using a declarative action scheme. The format of our annotation scheme reflects what an ideal computer vision system may soon be able to produce (e.g., estimation of camera movement (Suchan and Bhatt 2016) and/or action recognition (Laptev et al. 2008; Liu, Luo, and Shah 2009)). Each shot is annotated with the entities that are in that shot, where the entities are located in the story world, where the entities are located on the screen in that shot (composition), what actions the entities in the shot are performing, the type of camera shot being employed, and other useful details for extracting narrative structure. All scenes in the corpus are from a common genre and convey the same activity, in that they are all Western-style duel scenes where two or more gunmen have an escalating confrontation and face-off in a showdown. Previous research in film understanding suggests that viewer understanding is highly structured by character goals (Magliano and Radvansky 2001; Magliano, Taylor, and Kim 2005; Magliano and Zacks 2011), so our intention is to hold the kinds of character goals relatively constant and learn from the similarities and differences across scenes for communicating similar plot elements.

The actions performed by characters are tagged in each shot, and many of these action types are common across scenes. During coding, actions were mapped to a common dictionary of action types. We then used the most common action types to construct a planning domain, a library of STRIPS-style (Fikes and Nilsson 1972) action schemata, using a declarative knowledge representation where actions are explicitly annotated with their preconditions and effects. An action's preconditions describe every condition in the world that must obtain in order for an action to execute, and an action's effects enumerate every condition in the world that changes as a result of the action's execution. We describe a plot induction process we used to automatically construct a time line of actions in a scene and how we used the timing and precondition/effect information for actions to infer potential causal relationships between them.

Related Work

We use a specialized annotation scheme for describing shots similar to work others have done to provide formal or informal languages characterizing cinematic content. For instance, the Prose Storyboard Language (Ronfard, Gandhi, and Boiron 2013) is a formal language for annotating the cinematography of a shot, such as frame composition (spatial structure of objects on screen), shot transitions, and cam-

[1] A shot is the content recorded from continuous filming between intended cuts.

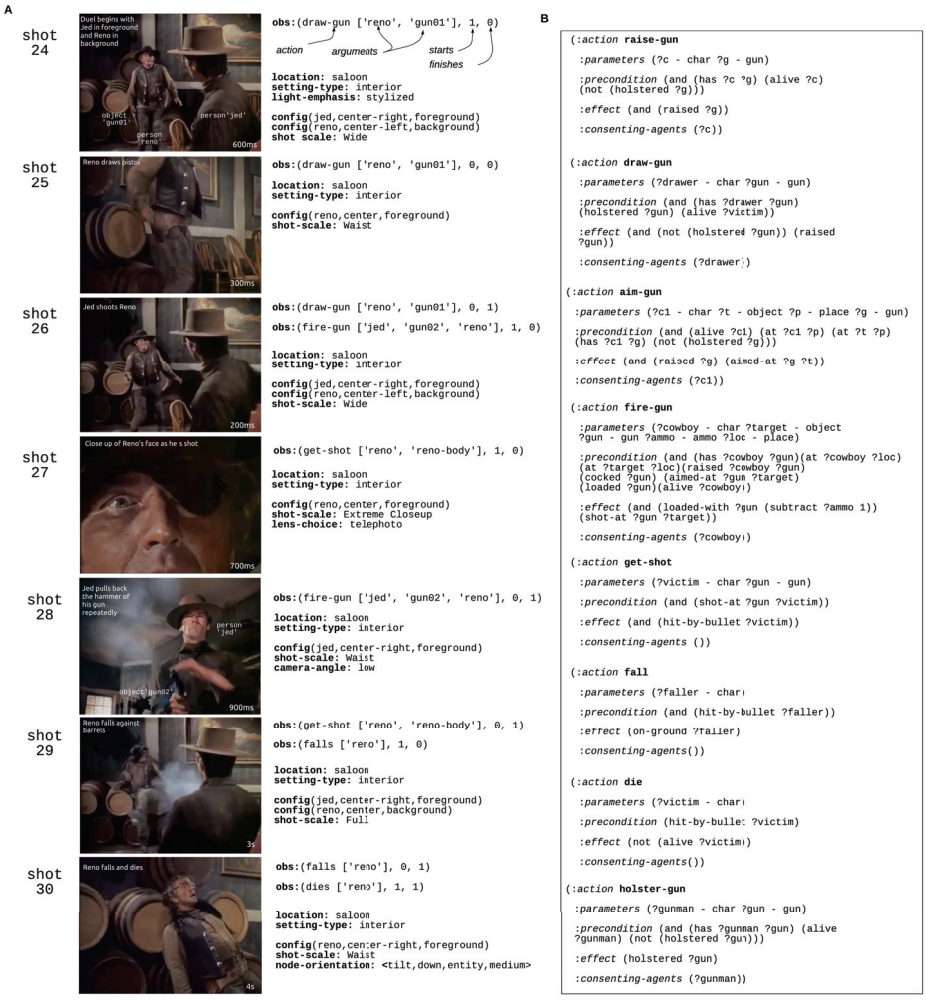

Figure 1: **[A]** An extracted snippet of scene from film *Hang 'Em High* with sample coding attributes **[B]** Sample action types for *Western World* planning domain

era movement. Other declarative representations for cinematography define actionable camera control such as for automated cinematography (e.g., (Christianson et al. 1996; Halper and Olivier 2000; Courty et al. 2003; Amerson, Kime, and Young 2005)). The Movie Script Markup Language (MSML) (Van Rijsselbergen et al. 2009) was developed to computerize screenplays to facilitate interactive collaboration among the various parties involved with producing film; it is a language for creating screenplays which handles the story logic and other internal pragmatics of shooting film.

One of the central contributions of our annotation scheme is action labeling – that is, identifying and naming actions as they occur in a stream of events depicted in a film. Previous projects on film understanding have addressed the problem of action recognition by finding action verbs in corresponding movie script or subtitles and creating a database of movie clips which map to each verb (Cour et al. 2008; Laptev et al. 2008). Rohrbach and colleagues (Rohrbach et al. 2015) created a corpus aligning 68k sentences and video snippets from 94 movies which could also be used for this purpose.

Suchan and Bhatt (Suchan and Bhatt 2016) demonstrate a visual processing framework to extract the scene structure and the geometry of a scene from a cinematographic viewpoint (e.g., camera movement, composition, spatial-temporal dynamics of objects, areas of attention, visual-perceptual saliency, etc.). Our annotations were created manually, and it is expected that progress in computer vision will make the task easier in the future.

Data Collection

In total, 30 scenes from Western films were annotated, shot by shot, with information about the events in the plot and about the way the shot is composed by the cinematographer. Scenes were chosen because they featured two rivaling persons or parties facing off in a gun duel where the action ended with gunfire. The scenes were drawn from the 30 films

listed in Table 1.

The scenes were collected and extracted from YouTube and are available online (Winer et al. 2017). In our initial version of the corpus, there are 1428 shots, 586 entities, and 3014 action observations. There are an average of 19.5 entities per scene, 100 action observations per scene, and 48 shots per scene. The average shot duration is 3634 milliseconds, and the average scene duration is close to 2 minutes 53 seconds.

For each scene, we tag the following properties about each shot:

- *Scene name*
- *Start time, duration* (milliseconds),
- *Shot number* (of scene)
- *Event description* (English sentence)
- *Entity tags*, sometimes typed as "character", "object", or "location".
- *Ordered list of observed action instances* (discussed in text)
- *Composition configurations of beginning and ending frames* (A *configuration* is a 3-ary literal of the form $config(e, x, z)\rangle$ where e is the name of an entity or world condition, $x \in$ {"left", "center-left", "center", "center-right", "right"} corresponds to the position of the entity from screen-left to screen-right, and $z \in$ {"foreground, background"} labels the depth of the entity in the frame relative to the orientation of the viewer.)
- *Location* name where entities in shot are located
- *Shot scale* $\in$ {Extreme wide (EW), wide (W), 3/4 Figure, 3/4 Figure, Full figure (FF), Waist, Close up (CU), Extreme close up (ECU)}
- *Camera node orientation*, either "None" or else a tuple of the form $\langle a \in$ {"pan", "tilt"}$, b \in$ {"up", "down", "left", "right"}$, c \in$ {"entity", "setting"}$, d \in$ {"slow, "medium", "fast"}$\rangle$
- *Camera movement*, either "fixed" (None), "moving" (right, left, down, up), or "tracking entity" with type in set {"parallel", "leading", "following", "3/4 angle", $direction \in$ {"right, "left", "front", "behind"}}
- *Camera angle* $\in$ {"level", "low", "high"}
- *Zoom* in set {"None", "In", "Out"}
- *Lens choice* in set {"normal", "wide angle", "medium tele", "telephoto" }
- *Light emphasis*, a Boolean, false for natural lighting.
- *Subject type* in set {"agents/entity", "scene", "object" }
- *Setting type* $\in$ {"interior", "exterior/outside"}
- *Continuity matching type* in set {"None", "action", "character"}
- *Point of view* in set {"none", "POV", "OTS"}

Coding for cinematography is complex and our categories reflect an iterative decision process for selecting attributes and splitting them into discrete categories. Sometimes, a single shot can start with one subject, move to another, then a third, etc., although more complex shots in this genre are rare. We attempted to strike a balance, such as coding only for the *beginning and ending configurations* of each shot, rather than reflecting all configurations that occur during the shot, and in some cases this means information about the shot composition is not reflected in the coding. *Camera node orientation* is distinguished from *camera movement*, where the former is the movement of the camera but is stationary in 3D space (e.g., pan and tilt), whereas the latter reflects movement of the camera through space such as moving up and down on a crane, or tracking on a dolly.

Cinematographers carefully craft the lighting for shots and work with lighting designers to create a highly nuanced image crafted with natural and artificial light (Gillette 1998); as a very preliminary placeholder for more expressive coding schemes (Barzel 1997; El-Nasr and Horswill 2003), we use the *light emphasis* attribute in our scheme to indicate whether or not lighting is used in the shot to emphasize something, and this is a judgment call. We are currently working on a coding scheme for lighting direction and shadows (hard vs soft). *Subject type* reflects the gist of whether the shot is focused on a character, on scenery, or on an object. *Continuity matching type* is the classification of whether there is some logic transitioning two shots based on an action or a character. *Point of view* can be a point-of-view (POV) shot if judged to be from the eyes of a character, or over-the-shoulder (OTS), and "none" otherwise.

The data set is accompanied by an *action predicate dictionary* denoting the actions that are observed performed by characters across scenes. In total, there are 202 action predicates, each annotated with a category (i.e., navigational, bodily-movement, cognitive-emotional, verbal/communicative, and duel-related), a text definition, the key arguments and their types (character, object, location, etc.), and the scenes they are observed in. During the initial pass of coding, action predicates were formulated such that semantically similar actions were binned together and collapsed when the terms for that action were consistent. The level of granularity for action coding was motivated by psychological research about the way we segment continuous human activity into discrete events (Kurby and Zacks 2008). In the corpus, actions are also coded for whether they are observed to begin and or finished during the shot.

A group of 3 researchers worked as a team to create the action predicate dictionary and label the actions in the scene. The scenes are automatically split into shots using Transana software (Woods and Fassnacht 2009) to minimize errors. We plan to have to two independent individuals use the predicate dictionary to label the actions using this dictionary and we will use these labels to calculate interrater reliability. We may do the same for shot composition, shot scale, camera movement, and other features.

A JSON object is produced for each scene, composed as a list of shots. The corpus is available for download online (Winer et al. 2017).

Table 1: Films and scene details used in the Western Duel Corpus (Winer et al. 2017)

Film Title	abbrv	scene length (s)	shots	action observations	entities
Three Amigos	3a	129.2	68	119	28
Five Card Stud	5cs	204.7	39	85	12
A Gunfight	agf	197.2	75	148	18
The Big Country	bc	237.8	62	156	18
Blazing Saddles	bs	83.1	19	83	13
Cheyenne Social Club	csc	127	31	64	17
Duel in the Sun	dis	110	15	56	18
El Diablo	ed	199.9	55	94	15
For a Few Dollars More	fdm	324.5	90	210	24
A Fistful of Dollars	ffd	162.1	43	88	22
The Good, the Bad, and the Ugly	gbu	190.3	80	103	13
The Gunfighters	gfs	122.5	13	55	21
Hour of the Gun	hg	259.2	39	112	21
Hang'Em High	hmh	144.2	32	69	22
High Noon 2	hn2	173.4	32	75	18
High Plains Drifter	hpd	119.3	25	61	18
Jubal	ju	167	30	82	19
The Outlaw Josey Wales	jw	102	44	55	11
My Name is Nobody	mnn	176.1	44	71	13
The Man who Shot Liberty Valance	mslv	129.1	22	61	14
Once Upon a Time in the West	outw	195.3	35	121	22
Pale Rider	pr	201.3	78	117	22
The Quick and the Dead	qad	204.1	71	119	29
Quigley Down Under	qdu	155.7	36	90	16
Shane	sh	262.7	75	164	25
Silverado	sil	130.5	24	41	15
True Grit	tg	136.6	71	136	35
The Outsider	tout	161	57	123	28
Tombstone	ts	164.1	52	76	12
Young Guns	yg	219.4	71	180	27
	avg	173	47.6	100.5	19.5
	sum	5189	1428	3014	586

Plot Induction

We outline a procedure for reconstructing plot by using *action types*. An action type designates a STRIPS-style declarative action model (Fikes and Nilsson 1972) which includes typed variable parameters, non-ground literal preconditions designating which conditions would need to hold for an instance of the action type to occur, and non-ground effect conditions designating what conditions the action's execution would make hold. We created the action types as part of a planning domain where the actions were inspired by frequent action predicates used in the action predicate dictionary. We created 76 action types (available online (Winer et al. 2017)), where 32 of these types are the receiving end of a manual many-to-one mapping from action predicates to action types. The unused actions were created to enable a more comprehensive and usable planning domain. The caveat is that mapping of actions to plan-based actions is not perfect, as some interpretation of the mechanics of the world are needed to build a planning domain.

Actions may be observed over the course of multiple shots and are sometimes interleaved with other shots. In Figure 1, Jed begins to draw his gun in shot 24 and finishes in shot 26, and begins to fire in shot 26 and finishes in shot 28. This creates a problem for extracting the scenario underlying the discourse presentation. We construct a potential plot for each scene using the action observations through a process we call plot induction. This process involves using the types of actions observed being performed by characters. An instance of an action is one where its variable parameters are substituted by consistent-typed entities.

Definition 1 (Action) *An **action instance** is a tuple $\langle t, V, a, P, E\rangle$ where t is an action type, V is an ordered list of entities, $a \in V \cup \emptyset$ is an agent which performs the action, P is a set of ground function-free literal preconditions, and E is a set of ground function-free literal effects. If s is an action instance of the form $\langle t, V, a, P, E\rangle$, $eff(s) = E$, $pre(s) = P$, $ag(s) = a$, and $args(s) = V$.*

Action instances are observed via cinematic discourse in the format of camera shots. The same action instance can be observed in multiple shots. Intervals are represented as logical variables (Allen and Ferguson 1994) (whose endpoints are millisecond values, *moments*, in $\mathbb{Q}$). Let $\min(i)$ indicate the starting moment of i and $\max(i)$ denote the ending moment.

Definition 2 (Observation) *An **observed action** is literal of the form $obs(a, s, f, \tau)$ where a is an action instance, s is a boolean tag indicating whether the action begins at the observation, f is a boolean tag indicating whether the action finishes in the observation, and τ is an interval. If o is an observation of the form $obs(a, s, f, \tau)$ let $act(o)$ denote a, $ent(o) = args(act(o))$, $st(o) = s$, $fin(o) = f$, $\min(o)$ denotes $\min(\tau)$, and $\max(o)$ denotes $\max(\tau)$.*

Each scene is composed of shots, ordered chronologically, and each shot has a list of observed actions (see Figure 1A).

An action instance can begin before the first time the action is observed, meaning the camera does not show the beginning of the action (or similarly does not show when the action finishes). Thus, we find the nearest observations with the same performing agent and infer that if the agent was observed and wasn't performing the action, then the action must have stopped/started at least at the beginning/end of this recent observation. Algorithm 1 steps through the inference procedure, which defines the **interval span** of an action instance.

Algorithm 1 `find-interval-span`

Input: Action observations A_{obs} and action instance a
Output: Interval span of a

$o_s := \arg\min_{o \in A_{obs}}(act(o) = a)$
$o_f := \arg\max_{o \in A_{obs}}(act(o) = a)$
$s, f = (\min(o_s), \max(o_f))$
$\Phi, \Omega = (st(o_s), fin(o_f))$
if $\neg\Phi$ **then**
  $s := \max_{o \in A_{obs}}$ s.t. $\max(o) < s, ag(act(o_s)) \in ent(o)$
end if
if $\neg\Omega$ **then**
  $f := \min_{o \in A_{obs}}$ s.t. $\min(o) > f, ag(act(o_f)) \in ent(o)$
end if
return (s, f) object with attributes $.s$ and $.f$

Definition 3 (Plot-Reconstruction) *A **plot-reconstruction** of a scene with action observations A_{obs} is a tuple of the form $\langle A, I, \prec, L, \rangle$ where A is a set of action instances for each unique action instance in A_{obs}, I is the* `find-interval-span` *function $I : A \rightarrow \mathbb{R}^2$ mapping action instances in A to their interval span, $\prec$ is an ordering over actions in A s.t. $a_i \prec a_j$ where $a_i, a_j \in A$ indicates that $I(a_i).f \leq I(a_j).s$, and L is a set of potential causal-links of the form $a_i \xrightarrow{p} a_j$ where $a_i, a_j \in A$, $a_i \prec a_j$, $p \in eff(a_i) \cap pre(a_j)$, and $\neg\exists a_{threat} \in A$ s.t. $I(a_i).f \leq I(a_{threat}).f \leq I(a_j).f$) and $\neg p \in eff(a_{threat})$*

Previous research suggests that causal relationships between events are central to narrative comprehension (Lehnert 1981; Trabasso and Van Den Broek 1985; McNamara and Magliano 2009) and may serve as key features for tasks such as automated narrative summarizing (Cheong et al. 2008) or narrative script learning (Chambers and Jurafsky 2008). The potential causal-links can also be used to extend the plot reconstruction to induce slot-filling inferences: actions which are not shown but are necessitated and enabled (Niehaus and Young 2010), or to recognize character plans (Cardona-Rivera and Young 2017).

Action Clustering

We clustered action types using the potential causal-link feature of the reconstructed plot. Action types which are clustered may form meaningful story chunks reflecting different aspects of the scene type. Some action types are inherently similar because they are variations of the same meaning sense but with different parameter configurations (e.g., arrive vs. walk-from-to).

Similarity between two action types s, t, written $sim(s, t)$ is defined as

$$log_2 \frac{P(\exists p, s \xrightarrow{p} t | t \xrightarrow{p} s)}{P(\exists p', t', s \xrightarrow{p'} t' | t' \xrightarrow{p'} s) P(\exists p', s', t \xrightarrow{p'} s' | s' \xrightarrow{p'} t)}$$

where $P(s \xrightarrow{p} t | t \xrightarrow{p} s)$ is the probability that two action instances of types s, t are in a potential causal-link, either $s \xrightarrow{p} t$ or $t \xrightarrow{p} s$ for some literal p.

We performed hierarchical clustering using a distance measure between hierarchical clusters S, T defined as:

$$dist(S, T) = \max_{s,t \in S \times T} sim(s, t)$$

for complete-link and with $\min$ for single-link. The clustering procedure is to iteratively merge the two clusters with the largest minimum sim between any two contained action types (single-link) or the largest maximum sim between any two contained action types (complete-link).

Sample clustering results for $k = 5$, complete-link:
$c_0 = \{$drop-gun, walk, die, raise-gun, fall$\}$
$c_1 = \{$identify, dismount, arrive, wince, give, pick-up, draw-gun, cheer$\}$
$c_2 = \{$drink, drop, carry-from-to, reveal, lower-gun, cock-gun, stand-up, side-step, adjust-clothing, face-at, look-at$\}$
$c_3 = \{$fall-from-to, face-from-to$\}$
$c_4 = \{$run, holster-gun, taunt, de-escalate, leave, get-shot$\}$

Conclusion

We introduced an annotation scheme for cinematics and used that scheme to hand-annotate a small corpus of scenes from existing Hollywood films. This corpus is unique in a way we predict will be beneficial for film understanding research because it combines 3 things:

1. It features 30 instances of the Western gun duel scene archetype, and thus each scene has similar discourse goals and underlying story actions.
2. Each scene is manually coded shot-by-shot with a specialized cinematic annotation scheme.
3. Character actions observed in shots are mapped to declarative action types which can be used for plot induction and other intelligent narrative tasks.

In future work we hope that by using item 1, we can automatically learn a script representing the gun duel scene archetype and that this will lead to insights for a domain-independent approach for learning other scene archetypes; however, until other corpora similar to this one are created, it will be difficult to evaluate the effectiveness of this work. Also, the corpus will likely be useful for narrative processing tasks in computer vision, either for training or as a benchmark test. Finally, the corpus may be useful for psychologists studying narrative discourse comprehension; experiments in this field often involve showing movie clips to participants which vary only on experimental conditions (Magliano, Taylor, and Kim 2005;

Magliano and Zacks 2011), but it is typically difficult to either create or find clips which vary systematically and in well-defined ways.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1654651, for which the authors are thankful.

References

Allen, J. F., and Ferguson, G. 1994. Actions and events in interval temporal logic. *Journal of Logic and Computation* 4(5):531–579.

Amerson, D.; Kime, S.; and Young, R. M. 2005. Cinematic camera control for interactive narratives. In *Proceedings of the ACM International Conference on Advances in Computer Entertainment*, 365–370.

Barzel, R. 1997. Lighting controls for computer cinematography. *Journal of Graphics Tools* 2(1):1–20.

Bordwell, D. 2013. *Narration in the fiction film*. Routledge.

Cardona-Rivera, R. E., and Young, R. M. 2017. Toward combining domain theory and recipes in plan recognition. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*.

Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *Association for Computational Linguistics*, volume 94305, 789–797.

Cheong, Y.; Jhala, A.; Bae, B.; and Young, R. 2008. Automatically generating summaries from game logs. In *Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference*, 167–172.

Christianson, D. B.; Anderson, S. E.; He, L.-w.; Salesin, D. H.; Weld, D. S.; and Cohen, M. F. 1996. Declarative camera control for automatic cinematography. In *AAAI/IAAI, Vol. 1*, 148–155.

Cour, T.; Jordan, C.; Miltsakaki, E.; and Taskar, B. 2008. Movie/script: Alignment and parsing of video and text transcription. *Computer Vision–ECCV 2008* 158–171.

Courty, N.; Lamarche, F.; Donikian, S.; and Marchand, É. 2003. A cinematography system for virtual storytelling. In *Virtual Storytelling. Using Virtual Reality Technologies for Storytelling*. Springer. 30–34.

El-Nasr, M. S., and Horswill, I. 2003. Expressive lighting for interactive entertainment. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 1, I–425. IEEE.

Fikes, R. E., and Nilsson, N. J. 1972. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2(3):189–208.

Gillette, M. J. 1998. Designing with light: an introduction to stage lighting. mountain view.

Halper, N., and Olivier, P. 2000. Camplan: A camera planning agent. In *Smart Graphics 2000 AAAI Spring Symposium*, 92–100.

Hu, Z.; Rahimtoroghi, E.; Munishkina, L.; Swanson, R.; and Walker, M. A. 2013. Unsupervised induction of contingent event pairs from film scenes. In *The 2013 Conference on Empirical Methods on Natural Lanugage Processing*, 369–379.

Kurby, C. A., and Zacks, J. M. 2008. Segmentation in the perception and memory of events. *Trends in cognitive sciences* 12(2):72–79.

Laptev, I.; Marszalek, M.; Schmid, C.; and Rozenfeld, B. 2008. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1–8. IEEE.

Lehnert, W. G. 1981. Plot units and narrative summarization. *Cognitive Science* 5(4):293–331.

Liu, J.; Luo, J.; and Shah, M. 2009. Recognizing realistic actions from videos in the wild. In *Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on*, 1996–2003. IEEE.

Magliano, J. P., and Radvansky, G. A. 2001. Goal coordination in narrative comprehension. *Psychonomic Bulletin & Review* 8(2):372–376.

Magliano, J. P., and Zacks, J. M. 2011. The impact of continuity editing in narrative film on event segmentation. *Cognitive Science* 35(8):1489–1517.

Magliano, J. P.; Taylor, H. A.; and Kim, H.-J. J. 2005. When goals collide: Monitoring the goals of multiple characters. *Memory & Cognition* 33(8):1357–1367.

McNamara, D. S., and Magliano, J. 2009. Toward a comprehensive model of comprehension. *Psychology of learning and motivation* 51:297–384.

Niehaus, J., and Young, R. M. 2010. A method for generating narrative discourse to prompt inferences. In *Proceedings of the Intelligent Narrative Technologies III Workshop*, 7. ACM.

Pichotta, K., and Mooney, R. J. 2014. Statistical script learning with multi-argument events. In *European Chapter of the Association for Computational Linguistics*, volume 14, 220–229.

Rohrbach, A.; Rohrbach, M.; Tandon, N.; and Schiele, B. 2015. A dataset for movie description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3202–3212.

Ronfard, R.; Gandhi, V.; and Boiron, L. 2013. The prose storyboard language: A tool for annotating and directing movies. In *2nd Workshop on Intelligent Cinematography and Editing part of Foundations of Digital Games-FDG 2013*.

Ronfard, R. 2017. Five challenges for intelligent cinematography and editing. In *Eurographics Workshop on Intelligent Cinematography and Editing*.

Spranger, M.; Suchan, J.; and Bhatt, M. 2016. Robust natural language processing-combining reasoning, cognitive semantics and construction grammar for spatial language. *arXiv preprint arXiv:1607.05968*.

Suchan, J., and Bhatt, M. 2016. Semantic question-answering with video and eye-tracking data: Ai foundations

for human visual perception driven cognitive film studies. In *IJCAI*, 2633–2639.

Trabasso, T., and Van Den Broek, P. 1985. Causal thinking and the representation of narrative events. *Journal of memory and language* 24(5):612–630.

Van Rijsselbergen, D.; Van De Keer, B.; Verwaest, M.; Mannens, E.; and Van de Walle, R. 2009. Movie script markup language. In *Proceedings of the 9th ACM symposium on Document engineering*, 161–170. ACM.

Winer, D. R.; Magliano, J.; Clinton, J.; Osterby, A.; Ackerman, T.; and Young, R. M. 2017. Western duel film corpus. https://github.com/drwiner/WesternDuelsFilmCorpus.

Woods, D., and Fassnacht, C. 2009. Transana v2. 41. *[Computer Software]. Madison. WI: The Board of Regents of the University of Wisconsin System*.