DIRECTOR IN A BOX: LEARNING CINEMATIC RHETORIC FOR CAMERA SHOT SELECTION

A Thesis
Presented to
The Academic Faculty

by

John B. Munro IV

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science with Research Option in the
School of Computer Science

Georgia Institute of Technology
December 2010

DIRECTOR IN A BOX: LEARNING CINEMATIC RHETORIC FOR CAMERA SHOT SELECTION

Approved by:

Dr. Mark O. Riedl, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Charles L. Isbell, Jr.
School of Interactive Computing
Georgia Institute of Technology

Date Approved: December 17, 2010

ACKNOWLEDGEMENTS

I wish to thank Dr. Mark Riedl and Dr. Charles Isbell for their help and support in this effort as well as Dr. Irene Middleton and Dr. Melissa Meeks for their help in writing and editing this thesis. I would also like to think my friends and family for their tremendous support.

TABLE OF CONTENTS

SUMMARY

Automatic generation of cinematic content has been a goal for both the military and the entertainment industry to allow more diverse plot structures so that a trainee or player may have a scenario tailored to their personal needs and desires. We approach this problem from a traditional view of story as being appropriately broken into two parts: plot and discourse. We focus on the rhetorical aspects of discourse, specifically selecting coherent and aesthetically pleasing shot and blocking constraints for a virtual cinematographer. In the past, selection has been solved using a decompositional planning approach. Unfortunately, each decompositional unit corresponding to a single film idiom must be hand-authored by an expert cinematographer, resulting in an intractable knowledge acquisition problem, prone to error and subjectivity. We show that this problem can instead be solved by reinforcement learning techniques, which train on features from existing sitcom and movie scenes. We will evaluate the precision of our method by running 10-fold cross validation on our training sets.

CHAPTER 1

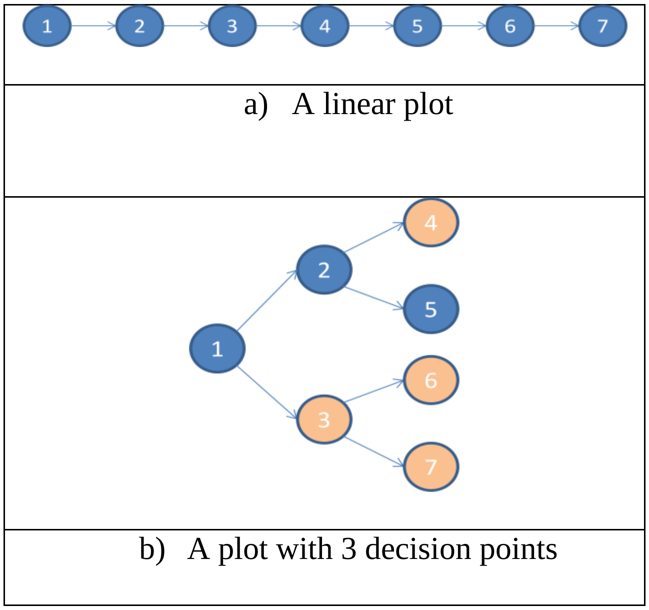
INTRODUCTION

Since the advent of the computer, the entertainment industry and military have had an interest in automating the authoring process of story content. Training systems and video games would both benefit from being able to generate video content on the fly. Being able to create new content without the need for human authorship will allow for scenarios tailored to the needs or interests of a user on an individual basis.

The military and many companies now use non-linear training videos for interactively teaching employees soft skills, such as how to make ethical decisions. The videos stop frequently, asking viewers to make decisions by choosing the next video segment from a list. Unfortunately, it costs approximately $2000 per minute to produce a moderately complex training video. Due to such high production costs per minute, there are few decisions for a viewer to make and each decision is evaluated as either correct or incorrect. If correct, the selected segment will display the next video in the sequence, but if incorrect a failure video explains what was wrong with that decision and will give the trainee another opportunity to choose differently. Often, however, decisions are not binary and require further development of the scenario before a single action may be analyzed.

For example, consider a time-sensitive combat scenario in which a commander must order four different squads to perform integral aspects of a mission. The speed and accuracy of his or her decisions may affect the final death count of the mission. This sort of complex decision is highly dependent on the accuracy and speed of related decisions that the commander makes. Only a highly adaptive system would be able to effectively administer such a training scenario. The current technology forces these sorts of scenarios to be handled by role-playing skits that require numerous personal to participate in, at least one of which must be a high ranking officer to evaluate performance.

Similarly, many in the entertainment industry would like to produce video games and interactive movies that tailor to the goals and interests of individual consumers. Unfortunately, adaptability requires exponentially more story content to be authored and produced for every point in which a user may change the course of the story. This forces each plot path to be quite short. Thus producers must currently deal with a tradeoff



a) A linear plot

b) A plot with 3 decision points

Figure 1: Compare two plot configurations, both with 7 events. (a) A linear plot sequence will always be longer than any single plot sequence in a (b) non-linear plot if there are an equal number of events in both. Also notice that if an audience where to watch the non-linear plot, they would only see 3 events out of the 7 for any given viewing.

between long but linear sequences that remove most autonomy from the player (Figure 1.a) and short but diverse sequences allowing high user self-agency (Figure 1.b). Currently they almost always choose the former, though often times giving the illusion of the latter (Fable 1, Knights of the Old Republic 1 & 2, etc.). Current video games are very linear, largely because they cannot justify creating a large number of plot events when a single player only sees a small percentage. If part of this production process was automated, it would potentially be much more reasonable for a producer to give players more choices throughout the plot.

Numerous applications would greatly benefit from advances in the automation of any part of the story production process. If instead production costs were mitigated through the use of an automated production system, training scenario developers would be able to put more decision points throughout a story. Students would then be able to train for more complex skills without the aid of extra personnel. Technologies of this type will potentially be more effective instructors by implicitly identifying and correcting students' weaknesses. Academia has had many ideas to reduce the authoring burden. These ideas include automatic creation of plot sequences and artificially intelligent ways to present said plot sequences to a viewer in a coherent and engaging way. We will focus on the latter.

Our goal is to give a system the ability to make intelligent cinematographic decisions in lieu of a human director, giving it the expressive power necessary to motivate and engage an audience. Our approach builds off of narratological formalisms that motivate a bipartite software architecture, which assumes that we can worry about details of the plot separately from making rhetorical decisions of presentation. This research will focus specifically on the problem of learning proper blocking and shot constraint selections, which specify where characters are positioned within a scene and how the camera will display the scene, respectively, for use with an artificial cinematography system such as Cambot(Mark O. Riedl, Rowe, & Elson, 2007), which takes a movie script and produces a machinima based on its specifications.

CHAPTER 2

RELATED WORK

Narratological Background

According to Russian formalism, stories can be thought of as being broken into two layers, a *fabula* and a *syuzhet(Bal, 2009)*; these layers are independent of the media mode, be it text, video, live action, etc. A *fabula* describes all events that occur in a narrative world, including all the “behind the scenes” actions that take place outside of the final story. The *syuzhet*, on the other hand, refers to the rhetorical choices of a story. It defines which events in the *fabula* should be revealed, the order in which they should be shown, and the way in which emotions should be conveyed or provoked. Previous research has used the concepts of *fabula* and *syuzhet* as the basis for thinking about stories in a bipartite fashion.

The generation of a story’s *fabula* and *syuzhet* can be effectively divided into a bipartite software architecture(Young, 2007). Under this structure, the plot and character actions, corresponding to a *fabula*, are generated by the first partition of the architecture. Then the second partition, representing a *syuzhet* generator, takes this representation of a *fabula* and makes decisions about the way in which the story should be presented to an audience. The module that creates the *syuzhet* is specific to the mode of representation. For instance, one *syuzhet* generation module might make decisions about how a *fabula* should be presented textually, while another would be used to prepare a cinematic production. This specificity is necessary because each type of media has its own set of constraints and conventions, giving it specific communicative powers. For example, in cinematography, a *scene* has been defined as a single, continuous series of actions and dialogue that take place in one location; a *beat* has been defined as the smallest divisible unit of a scene, comprised of a single moment of action or dialogue(McKee, 1997). Previous research has made strides in generating a coherent story by thinking about the problem with this bipartite architecture.

Automatic Plot Generation - Fabula

Plot graphs are valid representations of *fabula*. Trabasso (Trabasso, Secco, & Van Den Broek, 1982) and Graesser(Graesser & Hemphill, 1991) developed detailed plot graphs that are useful in answering questions about a story, such as, "why did Bob look at the cake?" These graphs have proven helpful in the foundations of computational story understanding and generation. Graesser's conceptual graphs have been simplified and are now frequently used as the basic plot model when conveying a *fabula*. The most basic version consists of character-action sequences connected by causal links that specify that a previous action was necessary for the next to occur.

There are two conflicting philosophies of how plot graphs should be generated. The first approach to creating story was through the simulation of goal-centric characters. Tale-Spin(Meehan, 1976) produces plots that are wholly determined by character goals and how characters interact. This approach is character-centric since the story is generated by the characters' local decisions without any respect for the overall story. This style of simulated story creation is called emergent narrative. It often leads to stories that are either dull, due to non-interaction between characters, or so called "miss-spun" tales that result from flaws in the knowledge base.

The next approach was to give all control over the plot to an authorial perspective instead. Lebowitz's Universe(Lebowitz, 1984, 1985) generates soap-opera-style episodes based on hand-authored scripts called plot fragments. Hand-authored plot fragments define skeleton action sequences to be filled in by characters according to relationship and personality constraints. These fragments may also call for other fragments as part of their sequence, building a hierarchical structure. Although this approach may be acceptable in a soap-opera domain dominated by stereotypes and dramatic characters, it often results in feeling like a mad-lib, or fill-in-the-blank plot line, due to its lack of character believability or plot coherence.

Some have also attempted to unify the author-centric and character-centric approaches, claiming that both are important for a successful story. IPOCL(M. O. Riedl & Young, 2004)--an Intentional, Partially Ordered, Causal Link Planner--does just that. It was built on top of the Longbow planner(Young, Pollack, & Moore, 1994) and works like any other partially ordered, causal link planner, but adds the constraint that all actions a character takes must be explainable by that character's intentions to fulfill some personal goal. This additional constraint gives characters the illusion of having self-agency, similar to what would be achieved from a forward processing simulation, but more constrained so that output is of a more consistent quality. The system produces a causally linked plot graph, a subset of the types of graphs that Trabasso and Graesser developed. The general form of this plot output is assumed to be the input that this work will use as the basis for planning further cinematographic rhetoric.

These systems all use graphs as a representation of story plot. Plot graphs have become the traditional representation of a *fabula*. Thus it is logical that a system that expects to generate the *syuzhet* of a story should expect the *fabula* to be in this simple, graphical form. Our system will indeed assume the plot input to be in the form of a causally linked graph. The next part of the bipartite architecture will be responsible for making decisions about how this plot graph should be presented to an audience.

Automatic Discourse Generation - Syuzhet

The second part of the bipartite architecture generates the *syuzhet* from some plot-graph representation of a *fabula*. The *syuzhet* is a rhetorical transformation of the *fabula* graph into a coherent and engaging form. It will be responsible for making media-specific choices that constrain the ways of presenting the story to an audience. This layer may also select a subset of the actions to show, ordering them according to some aesthetic. For example, it may remove events that do not follow the main character's point of view and then decide that each remaining event will be shown in chronological order. Previous research has focused on both of these aspects of rhetoric, but only the first is relevant here as our research will assume the second part is done for us.

Constraint selection of appropriate cinematographical rhetoric acts has previously been done by using a decompositional planning approach. Decompositional planners chain simple actions to fulfill sub-goals that motivate higher-order goals in order to achieve some desired final state. Jhala(Jhala & Young, 2005) has done this with the Longbow planner. His system takes in a specification of a scene and uses film idioms in the form of plan operators to determine how the scene should be shot. These idioms are decompositional and hierarchical, giving the planner a large state space of possibilities with only a limited number of idioms. That said, each idiom must be meticulously coded by an expert and added to the idiom library. The system will only ever choose a sequence of shots that can be specified by a valid combination of idioms. Because this method uses a planner to make decisions, the possible rhetorical choices will always be limited by the size and versatility of the idiom library.

An expert film director would no longer be required to specify each possible idiom wanted in a story design if, instead, film idioms were learned by the program from pre-existing cinematic decisions. Our research builds on Jhala's work in this field, but uses reinforcement learning techniques instead of planning to have our program make similar cinematic decisions.

Synthesis

In order for movies and video games to give audiences the desired level of control over plot, aspects of the video must be generated computationally. The story, or *fabula*, may be created by a variety of methods such as simulations, templates, or plans. Each method can generate a plot graph representation of the story to be used by a second phase of processing, that of making rhetorical decisions. Previous approaches to discourse are intractable and initially require expert knowledge engineering. If a system is instead taught to make correct decisions based on examples of good directorship, then a human expert will never be required to directly specify any film knowledge. We demonstrate here that such a system is possible and effective.

Intelligent Cinematic Generation

Cinema can be broken down into a three-step process. The first step is to write a plot for the story. The plot encompasses all of the characters and their actions. The second step is for a screenwriter to make decisions about how each scene of the story should be shot (i.e. which characters to show, which to hide, what angle the camera should be at etc.) He specifies a set of constraints for the director to follow. Finally, a director must make the final detailed decisions, specifying exactly how actors should stand and which camera shots to use. This is the model for how we think about automatic generation of cutscenes. A *fabula* is generated as a plot graph, and then a system must take that plot and specify the appropriate scene and shot constraints for conveying the story to an audience. Lastly, there is an intelligent cinematography system which makes the final decisions of exactly which shots are used in the scene, attempting to follow as many of the specified constraints as possible. This last part will produce the final video in the form of machinima.

Machinima

Machinima is an automatable approach to creating animated stories by using real-time, three-dimensional environments. Virtual cinematography systems are used to generate machinima from a script that includes scene constraints and an environment. They produce an animated film that follows the script and satisfies as many of the constraints as possible. The script must specify all characters and actions for each beat of every scene, much like any traditional movie script. Optionally, it may also include a set of constraints on location, character blocking, camera view, and scene choices to further enhance the quality of the video produced. These optional constraints may be used to convey a particular mood, point of view, or authorial aesthetic.

This paper will show that it is possible for a program to learn how to make professional, directorial decisions about how to shoot a scene based on a *fabula* graph that includes character moods and additional parameters such as genre. These decisions will be in the form of constraints that specify how the camera and characters should be arranged. The selected constraints will then be given to a virtual cinematographer for the final machinima production.

CHAPTER 3

APPROACH TO CINEMATOGRAPHIC RHETORIC

Given a plot graph that contains character actions and emotional states, and a set of authorial annotations, a system will select appropriate cinematographic choices for machinima creation.

Assumptions of Input

A *fabula* can be generated by any of the aforementioned methods of creation. The *fabula* will include information about which characters will be in the scene and what actions they will take. It is assumed that, without much additional work, these systems will also be able to present information about character relationships, characters' internal states of emotion, and contextual information that specifies scene intensity and genre. This should be enough information to determine the appropriate methods for shooting each beat of a scene.

Required Output

The system described here is intended to feed information directly into a virtual cinematographer, so the system must be able to present each piece of information that the cinematographer requires. Scenes will consist of every consecutive action that takes place in a single location. The basic information of character presence and action for each scene will come directly from the given *fabula*.

Although this is sufficient information for the cinematography system to create a valid animation, it is also desired that we include optional constraints in order to impose

Table 1: Sample constraints that describe a single beat or scene.

Blocking	View	Scene
Approaching	Introduce	Quick_Cuts
Facing	See_Point_of_Gaze	Few_Cuts
Talking_To	Increase/Decrease_Intensity	Do_not_Cross_180_Line

aesthetic goals on the specifics of how the scene should be shot. These consist of Blocking Constraints, View Constraints, and Scene Constraints. Blocking Constraints inform the virtual cinematographer's choice of character location and movement within the environment. The View Constraints narrow down the camera angles to use when shooting a beat of a scene, reducing the number of choices for the cinematographer. Lastly, the cinematography system uses the Scene Constraints when making aesthetic choices for the entire scene in order to prevent unwanted sequences of camera views, for example preventing the camera from crossing the 180-degree line between characters.

Developing a Dataset

In creating a data set, we built a Graphical User Interface (GUI) to aid in quick and consistent data collection. A recorder used this tool to note scene and beat structure, which characters are in each beat, their actions and emotional states, and the blocking and shot used. Blockings and shots were recorded pictorially, representing relative character and camera position and orientation. We decided to record this information graphically

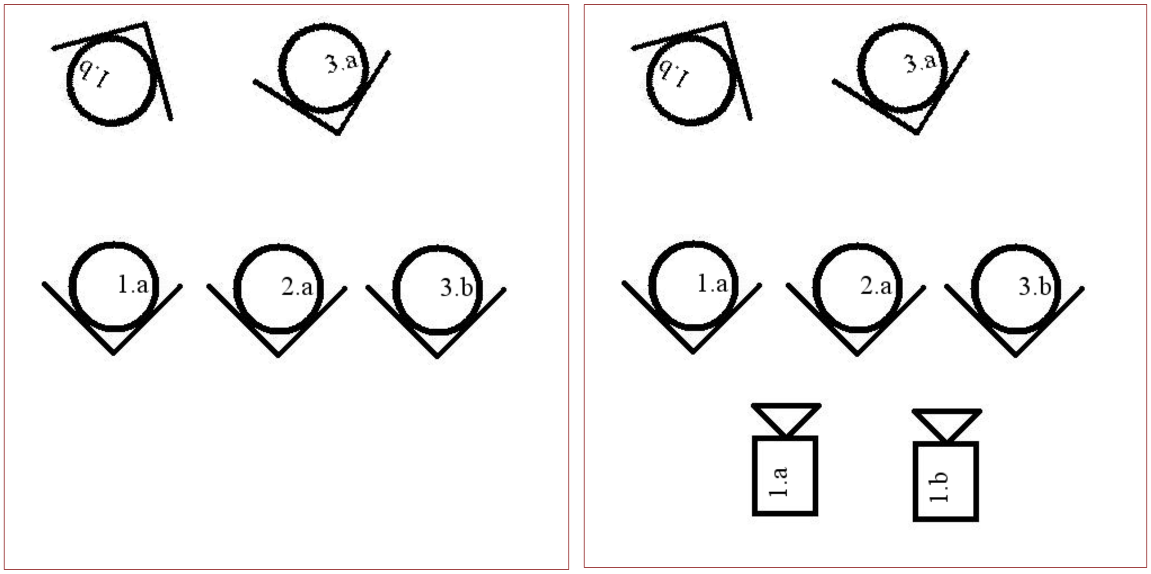


Figure 2: Example representations of Blockings (right) and Shots (left). Blockings denote relative character location and orientation as well as character movement. Characters and cameras are both numbered and lettered. Numbers distinguish between diferent characters or cameras while letters denote begin (a) and end (b) positions and orientations. In this example, you can see that character 3 has moved from the back to take up a position next to character 2. Shots are simply the addition of a camera for the specific blocking.

for a few reasons. First, this representation will make for quick, human recognition of the blocking or shot. Also, essential information can be inferred from a pictorial representation. Finally, a picture represents much more information than we use in this work, and thus with very little extra effort, a richer dataset has been built for further experimentation in the future.

A dataset has been defined as a corpus of scenes that are each governed by a single cinematographic style. For our purposes, this means that scenes from the popular television series *Scrubs* and *House* would comprise separate datasets. And so, each dataset represents the decisions made by a single director in a specific domain. Thus a learner that has been trained on a *Scrubs* dataset will make decisions consistent with the *Scrubs* style; where as a learner trained on a *House* dataset will produce a scene that fits well with *House*. Lastly, a dataset is comprised of independent scenes because we assume that a director makes decisions about beats based solely on factors within the specified scene.

Each scene is represented in the dataset as an ordered sequence of beats; each beat is comprised of character and camera information. Characters are described by their name, action, and emotion, as well as their relative position, orientation and movement within the scene. The camera in each beat is represented by its relative position and orientation within the scene, as well as the camera height and distance for that particular beat. Points of Interest (POI) may also be included in a beat's blocking and are defined as any important parts of the scene that are only referentially relevant to the beat and do not play an active role. For example, a POI may denote an important prop or a group of unimportant characters that is referenced by important characters within the beat and may have been considered when a director was selecting the appropriate camera shot. Each initial and final position and orientation of character and camera can be recorded in order to show movement within a given blocking. The camera height and distance may be defined as follows:

Camera Height:

- Top View – shown from the top of focus
- High – camera is above focus height
- Medium – camera is even with focus

- Low – camera is below focus height
- Very Low – camera is essentially on the ground looking up at focus

Camera Distance:

- Extreme Close Up - only part of the object is visible (e.g., just the mouth of a face)
- Close up - focus takes up entire frame (e.g., just the head)
- Medium Close Up - between close up and medium, (e.g., the head and shoulders)
- Medium - waist up shots to full body shots
- Long Shot - situate the character in scene, distance from front row to stage (10-20 ft.)
- Extreme Long Shot - very little detail, often outside, show setting, city, or building

Initial Dataset

As mentioned above, we have built an initial dataset based on the popular television show *Scrubs*. *Scrubs* is an American television show that was created by Bill Lawrence in 2001. Although most of the show conforms to a simple representation, it does occasionally include imaginative sequences that either cut to a separate location mid-scene or display unusual actions and timing, such as bricks falling on a character and then time reversing. Such scenes have only been coded up to the beginning of these odd sequences and coding was only resumed after the sequence if there were no timeline effects. These sequences have been omitted from the initial dataset because they either involve one-time actions that cannot be generalized or timeline reversals that are poorly represented by our coding scheme. Despite these omissions, we were able to code 237 beats from 15 scenes of a single episode.

Our initial dataset is based on fifteen scenes from *Scrubs*. The characters in most beats were generally happy, but there were also a good portion of instances where characters were either angry or fearful. We coded a total of 15 distinct character actions during these scenes. “Listen” was the most common character action with “Talk” and “Think” as the next most common. In this dataset, there is an average of 15.8 beats per scene with a standard deviation of 7.9. We have recorded 48 distinct blockings, with an average of 2.2 shots per blocking and a standard deviation of 1.9. There is a wide variety of distinct blockings within this dataset because our representation of blockings is quite specific. This specificity maintains a lot of information that can later be classified according to features and aggregated for generalized analysis, however. This dataset seems to have a fairly realistic distribution of actions and emotions, but there is a very high proportion of rarely used blockings and a larger corpus will certainly be required for a learner to account for atypical situations.

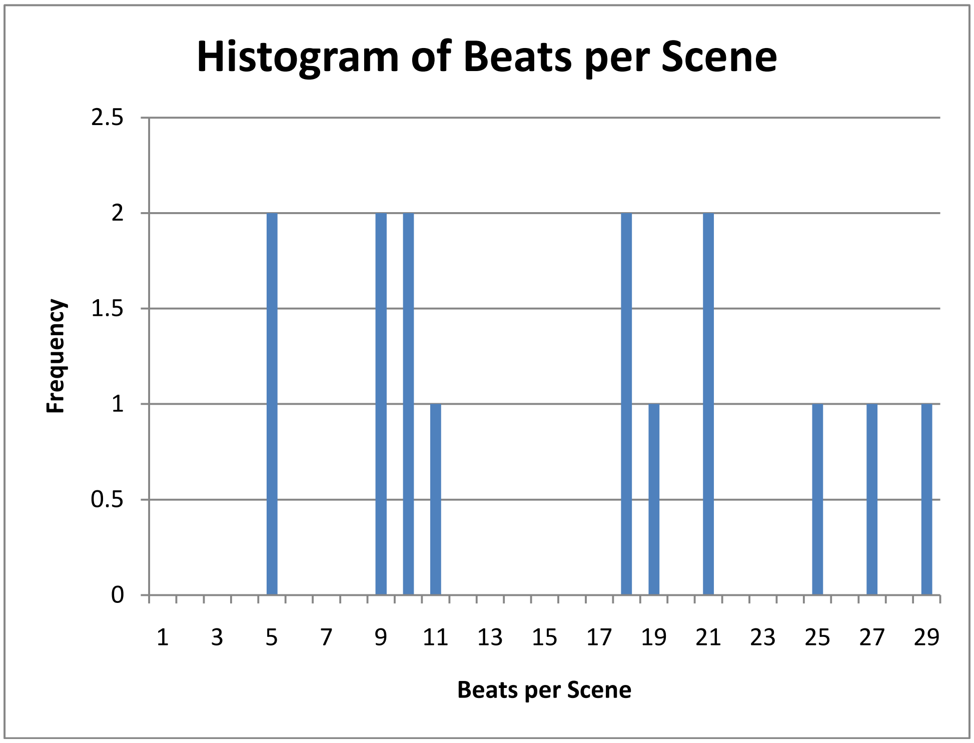


Figure 3: A histogram of beats per scene for the initial dataset based on the TV show *Scrubs*.

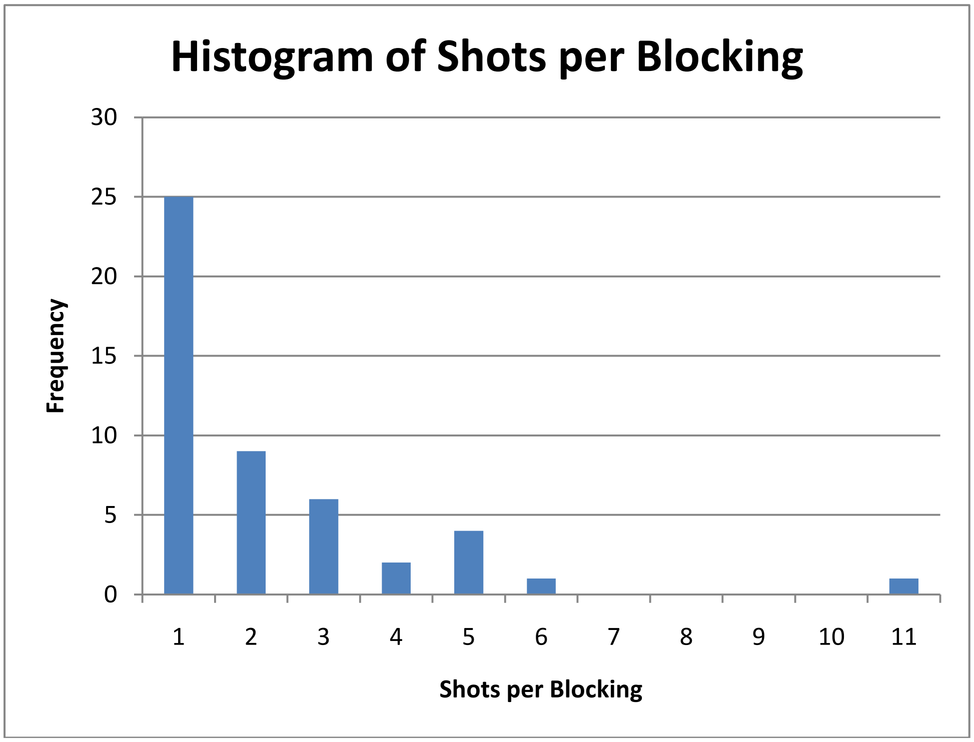


Figure 4: A histogram of shots per blocking for the initial dataset based on the TV show *Scrubs*.

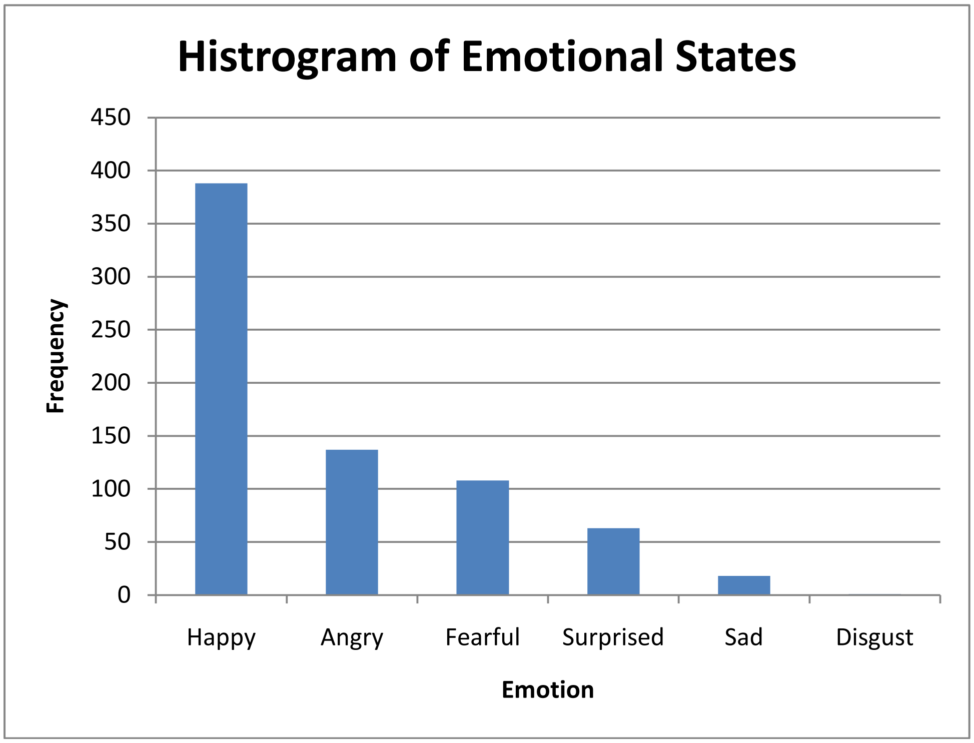


Figure 5: A histogram of character emotional states for the initial dataset based on the TV show *Scrubs*.

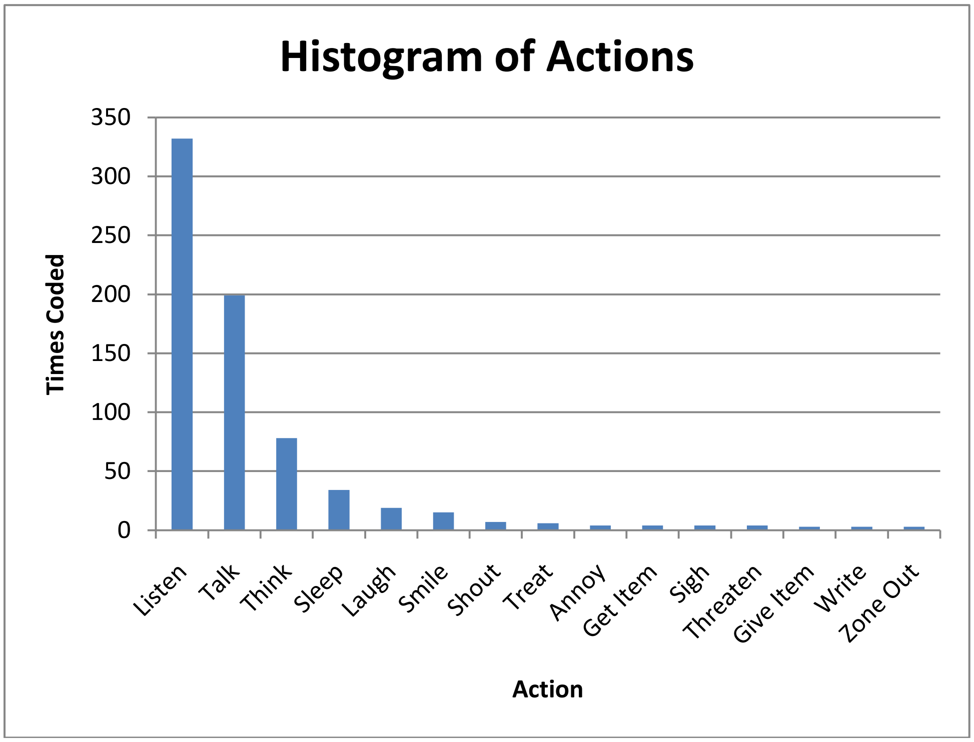


Figure 6: A histogram of character actions for the initial dataset based on the TV show *Scrubs*.

Inferences on the Dataset toward Constraint Selection

Further features may be deduced from this dataset in order to classify, or generalize, the shots and blockings. The pictorial blocking data allows us to infer that a character is approaching, leaving, facing, turning toward, or turning away from others, determining the Blocking Constraints to the beat. This representation also allows us to categorize the camera as still, moving in, moving out, moving up, or moving down by examining the transition state so that we may specify the Shot Constraints that apply. We can also label some characters as seen by the camera and others as out of view by assuming a typical focal angle of 54.4 degrees, which corresponds to a 35mm focal length. The camera may also be seen as looking from a specific character's perspective if it happens to be within some small distance of the center point of a character icon. A scene may then be determined to have many or few cuts approximated by the ratio of camera location changes (but not transitions) to the total number of beats within the scene. A 180-degree line may be found that goes through characters, and if the camera does not cross this line, then the scene will be labeled as respecting the 180-degree line. All of these features may be used as constraints to include in the produced script for Cambot's use. These features will also allow for a distance metric to be computed between blockings and shots.

A distance metric will describe the relative difference between two selections. It is a key component to many machine learning techniques and will likely be required for the completion of this project. A reasonable metric would likely weight features such as the number of characters, the number of characters that are moving, the movement of the camera, the number of characters approaching another, etc. Without further analysis, however, it is difficult to say exactly what a good metric will be.

CHAPTER 4

CONCLUSION AND FUTURE WORK

We have proposed a method for selecting correct and aesthetically pleasing camera shots and character blockings. Future endeavors will complete the automatic generation of full scripts for Cambot. This work will allow for an end-to-end connection between a *fabula* generator and virtual cinematographer, assuming that the *fabula* generator is capable of providing a plot graph that includes emotional states, character relationships, and scene/genre tags. In the future, this work will give researchers the ability to begin testing such *fabula* generation systems through user studies. Others will also be able to extend the ideas laid here to other media such as text or comic book generation.

Further automations of this process will be important for faster and better results. The dataset has been engineered to include an exceptional amount of information thus far unused. The pictorial representations of the shots and blockings may be used in the future for automatic generation of shots and blockings within Cambot. This would further decrease the authoring burden, as current Cambot users must build each new blocking and shot by hand. Further, it may be possible to eventually automate the data collection process using a combination of face and voice recognition to sense character action, emotion, placement and orientation within the scene, greatly enhancing the size and number of datasets with very little human effort.

Future Experiments

Constraint Selection

Specifying the details of how to shoot and orient each part of a scene requires information about cinematography that only an experienced movie director knows. In order for the system to make good choices on its own, we must find a way to teach it the rules and aesthetics that a professional considers. Machine learning, specifically reinforcement learning, provides a way to model the rules and aesthetic information in such a way that it may later be used to make decisions of the same type according to the learned model.

Reinforcement learning requires that we map our problem onto some set of states and that we also provide some way to traverse between states. For this problem, each state will represent a single way of shooting a beat of the scene and will be defined by:

A: a set of actions that the characters will take

S: authorial scene tags such as "confrontation"

C: a set of Blocking, View, and Scene constraints

Also, the traversal function, T, will select a set of constraints to use, transitioning to a new state in the space.

Framed as a reinforcement learning problem, there will need to be a reward function, R, which specifies the goodness of each state. This function will take in all of the state information and return the utility of it. Since this is a very difficult mapping to specify, as it incorporates all knowledge about cinematography as well as the aesthetics of each choice, this function will be learned programmatically from a number of example shot selections as training data.

In order to get training data, we have examined scenes from popular sitcoms and movies. At each beat, we have recorded the actions taken, appropriate contextual information corresponding to the authorial scene tags previously mentioned, and a set of Blocking, View, and Scene constraints that classify the beat most specifically. We will use this information to classify states of our space as being good and any unseen state as unknown by assigning a low, positive weight.

Testing Procedure

We plan to test this method by creating two distinct reward functions for two separate genres. The first training set will only include data about sitcom scenes while the second will include only data about fight scenes. We will train our learner on 90% of the data and use the other 10% to evaluate accuracy. This 90/10 ratio will be randomly selected 10 separate times and the overall accuracy will be determined by the average of these trials. This method is called 10-fold cross validation and will be used to assure that our learner generalizes to novel queries.

We will use our learner to generate a full script for Cambot(Mark O. Riedl, et al., 2007). Cambot was designed to be a lightweight virtual cinematographer that takes a script, which includes scene constraints, and an environment as input in order to produce an animated film that follows the script and satisfies as many of the constraints as possible. We believe that the film decisions of our data sets should be distinct enough to provide noticeably different information for a virtual cinematographer. If we find that using these two reward functions do not produce a noticeably different movie, then we will be forced to rethink the features we are considering in our data sets.

As our dataset grows in the future, there are a number of statistical experiments that we will need to run on it before determining the exact nature of the proposed learning agent. Because our feature set is so rich, it is expected that a much larger dataset will be required than what has been completed thus far. Future exploration of the data is difficult to predict, but there are a number of questions that will need to be asked: Can any of the Blocking or Shot constraints be predicted independently by the set of action-emotion pairs for a single beat? Can a sequence of character actions and emotions predict Blocking or Shot constraints? How large does that sequence need to be? Does it help to include the Blocking or Shot constraints of previous beats? What is the sequence size to accuracy ratio? Extra features may need to be taken into account in order to produce better results. For example, a rich web of character relationships may be important to determine the dominance structure of a beat. That dominance structure might determine which character stands in which position, as well as their relation to the camera. Future research will determine what, if any, extra information will be necessary to accurately predict a director's decisions.

REFERENCES

Bal, M. (2009). *Narratology: Introduction to the Theory of Narrative* (Third ed.): University of Toronty Press, Scolarly Publishing Division.

Graesser, A. C., & Hemphill, D. (1991). Question answering in the context of scientific mechanisms. [doi: DOI: 10.1016/0749-596X(91)90003-3]. *Journal of Memory and Language, 30*(2), 186-209.

Jhala, A., & Young, R. M. (2005). *A discourse planning approach to cinematic camera control for narratives in virtual environments*. Paper presented at the Proceedings of the 20th national conference on Artificial intelligence - Volume 1.

Lebowitz, M. (1984). Creating characters in a story-telling universe. [doi: DOI: 10.1016/0304-422X(84)90001-9]. *Poetics, 13*(3), 171-194.

Lebowitz, M. (1985). Story-telling as planning and learning. [doi: DOI: 10.1016/0304-422X(85)90015-4]. *Poetics, 14*(6), 483-502.

McKee, R. (1997). *Story: Substance, Structure, Style, and the Principles of Screenwriting.* New York: HarperCollinss.

Meehan, J. R. (1976). *The metanovel: writing stories by computer.* Yale University.

Riedl, M. O., Rowe, J. P., & Elson, D. K. (2007). *Toward intelligent support of authoring machinima media content: story and visualization*. Paper presented at the Proceedings of the 2nd International Conference on Intelligent Technologies for Interactive Entertainment.

Riedl, M. O., & Young, R. M. (2004, 2004). *An intent-driven planner for multi-agent story generation.* Paper presented at the Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004. Proceedings of the Third International Joint Conference on.

Trabasso, T., Secco, T., & Van Den Broek, P. (1982). *Causal Cohesion and Story Coherence*. Paper presented at the Annual Meeting of the American Educational Research Association.

Young, R. M. (2007). Story and discourse: A bipartite model of narrative generation in virtual worlds. [doi:10.1075/is.8.2.02you]. *Interaction Studies, 8*, 177-208.

Young, R. M., Pollack, M. E., & Moore, J. D. (1994). *Decomposition and Causality in Partial-Order Planning.* Paper presented at the Second International Conference on Artificial Intelligence and Planning Systems, Chicago, IL.