MIT Media Laboratory Technical Report #383
Also Appears as MIT Media Laboratory Masters Thesis
Supervised by Rosalind W. Picard

IMAGES AND TEXT OF *SEINFELD* OMITTED FROM THIS VERSION

A Video Browser that Learns by Example

Joshua S. Wachman
Vision and Modeling Group
MIT Media Laboratory
20 Ames Street; Cambridge, MA 02139

Abstract:

This thesis investigates the role of learning in a video browsing venue. Regions of significant change are preselected automatically from video sequences in a television situation comedy. These regions, which often depict portions of actors, are presented to the user for interactive labeling. The user specifies regions which are positive and negative examples of the actors and the computer trains by analyzing the regions with respect to a bank of signal models. Other regions in the video database, similar to the positive training examples, are found automatically. A feature of this work is the integration of high-level information, as encapsulated by the show's script and closed captions, and low-level signal feature analysis, as derived from similarity measures. The pooling of these descriptors constrains the search. Results of a database query are presented to the user during an interactive session. Given sufficient training data and user feedback, the computer learns the pattern of video which corresponds to a particular actor. By these means, a tool which can intelligently assist a human at indexing, browsing and searching through video is constructed.

Rosalind W. Picard
NEC Development Professor of Computers and Communications
Associate Professor of Media Technology
Thesis Supervisor

This work was supported in part by British Telecom and IBM.

PAGE INTENTIONALLY LEFT BLANK

PAGE INTENTIONALLY LEFT BLANK

SORRY, LICENSE TO USE *SEINFELD* IMAGERY AND SCRIPT EXCERPTS NOT GRANTED FOR WEB PUBLICATION. IMAGES AND TEXT FROM THE *SEINFELD* EPISODE NOT INCLUDED IN THIS VERSION.

FOR A HARD COPY OF THE TECHNICAL REPORT *WITH* IMAGES AND TEXT PLEASE SUBMIT REQUESTS IN WRITING TO:

TECH REPORT # 383
VISION AND MODELING GROUP
MIT MEDIA LABORATORY
20 AMES ST. CAMBRIDGE, MA 02139

Contents

Research Contributions of Thomas P. Minka

Design and evaluation of benchmark tests with FourEyes was a collaborative effort between Tom P. Minka and the author. Minka, who developed the FourEyes learning system with Picard, performed the benchmark tests reported. Minka also adapted the FourEyes system to communicate with the video browser presented.

Chapter 1

INTRODUCTION

Motivation

Techniques for sifting through on-line text documents based on keywords, concepts, or measures of word proximity have spawned the on-line information retrieval business. Methods for searching large databases of still-images are being developed to assist stock photography houses, image archivists and World Wide Web search agents. The consumer of the future may have automatic tools to organize a digital shoe-box full of family photos based on an array of content specific characteristics (images captured in the forest, images *sans* mother-in-law, etc..,). The impending proliferation of digital video databases portends the need for semi-automatic tools which can intelligently browse, index, annotate and navigate through video data with the ease associated with traditional document retrieval. Nascent research in automatic retrieval of still-images has yielded promising results [Nibla, 93] [Minka, 96], and is, in part, the basis for this thesis. In the near future it is expected that the largest databases of image data will be composed of video. The issue of how to get a handle on the content of a video stream is the primary motivation for this research. Specifically addressed is the problem of how to find, distinguish and learn the representation of a set of actors in a TV situation comedy. This work is demonstrated in the form of a video browsing venue which extends the FourEyes still-image browser and learning system into the video domain. Additionally, the thesis evaluates the effect of high-level script and closed caption information on learning performance provided by the low-level models in FourEyes [Picar, 95a][Minka, 95].

The Role of Learning

Humans are adept at learning visual patterns and at navigating through the world relying on visual perception. However, it is quite difficult for a computer to simply parse a video sequence into perceptually salient constituents, the problem is ill-posed [Hadam, 23], [Marro, 87], in the formal sense. The problem is complex because observations can be perceived in multiple ways.

Video pattern recognition and computer vision applications, more specifically, have traditionally been relegated to very simple problem domains (OCR, part alignment, industrial inspection). Learning, as a general computational tool, may offer sufficient leverage to pry some of the practical applications of pattern recognition out of the brittle confines of industrial expert systems. *Learning by example* is one method which offers significant promise.

Learning by Example

Both human and computer observers may be taught by example. The paradigm employed in this research is one where the human teaches by example and corrects the computer's response interactively. Positive and negative instances of an object in a video are labeled by a user. The computer forms a hypothesis about the object which subsumes experience across multiple interactive sessions. In this scenario, the strengths of the computer and human are complementary. With more human interaction and feedback, the computer, like any astute pupil, becomes wiser, if not smarter.

Computer learning, in the context of the video analysis performed here, is in many ways analogous to the process of human cognitive learning. Consider for instance, an individual introduced to a group of strangers at a social event. Through interactions and observations, the individual gathers information about each of the strangers at the event. This information may pertain to the stranger's age, occupation, dietary habits, appearance, behavior, size, gait, emotional disposition, voice, etc. Each observation contributes to the cognitive model of that person and is associated with the individual's name, face or some distinguishing meta label. Each observation is affected by the observer's interpretation and determines the perception of that observation. The process of interpretation is governed by bias. Further interactions with the person result in more observations and more perceptions. Each observation reinforces or adjusts perceptions created during previous encounters. Collectively these experiences form an impression of the stranger. Given enough information, the stranger may become recognizable, if not familiar. Similarly, a computer system may be constructed to make observations and representations of people's features in a video stream so that the pattern that constitutes the people in the video can be learned and recognized. Bias in this context is a set of numerical weights that can influence the decision whether to include a set of elements in the representation or to exclude them. Experience from feedback with several human users offers an opportunity to adjust biases.

Identifying Features for Modeling

What does it mean for someone to 'move', 'look' or 'act' a certain way? What should the nature of the analysis be in order to determine these attributes? What are suitable training examples that would characterize such traits when teaching by example? Will these examples be representative enough so that the computer can uniquely abstract the representation and form a generalization? What are the best measures of similarity to choose in order to determine if two items are similar? How can example regions that exhibit characteristics be selected and analyzed automatically? The selection of models which best characterize the data is critical to preparing to address these questions.

A person skilled at performing comic impressions can distill the definitive characteristics of a well known person in order to present a convincing imitation. The mannerisms, when adopted by the performer, project an alter-ego. With feedback through experience in front of many audiences, the impressionist selects from all observable features, the grouping which best characterizes the mimicked individual to a substantial fraction of the audience. In a similar way, this research seeks to model the data in such a way that its distinguishing features can be used to consistently recognize a particular object or person in a set of video data. User-computer interaction provides one means of forming groupings of models by providing positive and negative examples; high-level description, or *context*, provides another.

The Role of Context

Observations made within a formal context help focus interpretations and thus provide an implicit bias. In this work high-level information embedded in the script and closed captions, in combination with low-level image processing filters, constrains the ill-posed vision problem, making it tractable for certain recognition tasks. The structure of the situation comedy is encapsulated in the script and echoed in the closed captions. These elements which include setting (*where*), characters (*who*), actions (*what*), and order (*when*), can be effectively used to analyze and interpret the low-level image features.

Why Situation Comedies?

Within the context of a TV situation comedy, each character has characteristic movements, clothing, expressions, speech patterns, vocabulary, preferences, behaviors, dietary habits, hair styles, sizes, shapes, gaits, etc.,. These are attributes by which many people may be distinguished. However, in a TV situation comedy, the actors play fictitious people who are *caricatures* of real people and therefore their mannerisms may be more pronounced than average. Such hyperbole makes situation comedies, as a genre, a rich test bed for recognizing patterns of people.

Sit-coms are driven by characters, actions, situations and dialog. Arguably, the most salient constituents of situation comedies are people. A library of situation comedy episodes contains multiple representations of the same cast of characters. Over the course of a single episode, the actors vary their costume, pose, expression and proximity to the camera. Through analysis of a library of episodes, one would expect the set of distinguishing characteristics of each actor to converge toward a singular unambiguous representation.

Benefits of using situation comedies as a test bed for video pattern recognition are numerous. They are professionally illuminated which lends a consistency to the image signal. They constitute a class of data ripe for applications based on repurposing of content. Their popularity makes them a real database about which someone might want to query. Their highly stylized, often formulaic structures have properties which can be measured and subsequently used to constrain the pattern recognition problem. And vast libraries of content exist. In the 1994 American television season for instance, situation comedies constituted a significant 22.7% of the Fall prime time line-up [Brook, 95]. Sit-coms also have scripts which are formal representations of the essence or *meaning* of the video in a compact form. These textual distillations can be used to produce meta-representations of the original video in the form of textual descriptions, semantic networks, indices or labels. Such handles on the content may assist a browser at finding perceptually meaningful points during a directed search or browsing session.

Why Look for People?

Situations are a function of context, and actions are often a function of people. People are one obvious class of objects by which video can be indexed. Although no statistical measurements exist, it is arguable that people are among the most common subjects in professionally produced video. Situation comedies, news, cinema, music videos, soap operas, talk shows, home videos, industrial videos and commercials, collectively constitute the majority of video recorded. Generally, these genres are motivated by people. For instance, people appeared in 98% of the 182 shots of Seinfeld's episode *The Beard* analyzed for this work. So finding the patterns which constitute people is a legitimate starting point for understanding the video itself.

Beyond People: Towards a General System

It should be noted that no explicit model of *people* as a class of objects was assumed by the underlying method used in this thesis. The analysis identifies regions of significant change in the video signal. It should be emphasized further that there is nothing inherent in the low-level feature analysis or learning performed that could not be applied to video sequences that do not include people and are not scripted or captioned. An attempt was made to construct a general purpose video learning and retrieval system that was not contingent on a specific video compression scheme or content. It is expected that the performance reported here on finding actors will not be as good as methods that incorporate explicit models using face detection, voice recognition or articulated geometric models. In order for these domain-specific models to succeed the data sets must be well-constrained. A strength of the present general purpose approach is that it should be flexible enough to find other classes of objects within the database aside from actors.

The Problem Statement

Can a computer be programmed to learn the representation of a cast of characters in a television situation comedy and distinguish among them sufficiently well to be able to classify each pattern and successfully retrieve relevant instances? What role can the high-level script and closed caption information play in optimizing the search? These issues are considered with respect to the goal of extending the Society of Models [Picar, 96a],[Minka, 96] approach of the FourEyes learning system into the video domain. A video browser tool serves as the test bed.

Thesis Outline

Chapter 2 reviews the FourEyes learning system, the Society of Models and the relation of the video browser to previous work. Chapter 3 outlines the specific approach and methods used in conducting experiments. Chapter 4 discusses the ground truth labeling and learning experiments. Chapter 5 summarizes the work and postulates directions for future research.

Chapter 2

BROWSING VIDEO AND LEARNING

Browsing Video is Difficult

Most current video cassette recorders are equipped with fast forward/fast rewind buttons that merely sample every *N* frames of the video. This may be an adequate tool for some video skimming tasks, but it is not an ideal tool for video retrieval or browsing. The major interface issues in constructing a useful video browser interface are how to solicit users queries on the database and how to effectively display related sequences to the user for feedback. But perhaps the most provocative issue related to video browsers, and the one motivating this work, is how to examine the underlying content in order to make intelligent responses to user queries.

Approaches to Video Browsing Research

Video signals contain a lot of redundant information. Video compression algorithms eliminate redundancy in order to capture the essential visual elements of the signal in the most compact form. Signal decompression techniques reconstruct the stored or transmitted signal into a perceptually salient approximation of the original video for viewing. Similarly, video browsing, annotation and retrieval systems attempt to extract the essence of the meaning of the video in a compact form so the compact representation can be used to index the video at meaningful points. In one extreme, a successful video annotation system would be able to construct a textual description of a video similar to a script, as scripts are one common representation that approximate the underlying content. A thorough discussion of how the structure of narrative knowledge can be represented is beyond the scope of this thesis; however, the in-depth treatment of [Schan, 95] is recommended to the interested reader.

Many researchers approach video retrieval as an outgrowth of video compression research where the goal is to decompose the edited video into its constituent signals. These components may be shot boundaries, camera motions, speech segments, or musical riffs. However, such structural elements do not necessarily imply meaning, that is, signal events are not necessarily plot events. One would expect such an approach to be most successful with action films. Events in this genre for instance, may include crashes, explosions, gunshots, chasing and punching. These events may be filmed quite differently from romantic comedies where the most salient plot events may be a kiss, dance or moonlit walk. Selecting features for such subtle signals may be quite difficult. However, if a computer could recognize individual people in a sequence and understand their geometric and temporal relationships then, with the added high-level information from the script, closed captions and accumulated experience, meaning might be inferred. Attempts at calculating the structural patterns of the editing, such as evaluating the average shot length may echo the underlying content [Salt, 83], but they are unlikely indicators of meaning by themselves. What is needed is a means of identifying and extracting patterns relating to attributes of the plot itself.

Related Techniques

In order for a user to browse a video database more effectively than simply fast forwarding every *N* frames, high-level tags on the underlying content have to be designated. These handles may, for example, take the form of textual annotations or of groupings of similar patterns and identification of events. Shot clustering methods are effective at partitioning the video and may assist a browser at reducing the search space [Yeung, 95]. Some methods exploit the existing structure of M/JPEG video compression schemes [Zhang, 95],[Meng, 96] and others unfold each sequence into a salient key frame for presentation [Karah, 95]. Methods which integrate multiple features such as texture, color and shape [Chang, 95], have shown promising results. Still, other methods attempt to integrate additional high-level data such as audio transcriptions with traditional low-level features like optical flow and color histograms. Such an event oriented distillation of the video, forms an index which can be used to generate a video 'skim'. These are automatically reedited versions of the original video which edit out that information considered to be statistically insignificant [Haupt, 95]. This method can expedite browsing large video databases by removing apparently redundant information in order to reveal statistically outlying events. But it is not clear how well these methods distill the meaning of their source video.

What distinguishes the present approach from those cited is that it exploits the human operator in the loop and uses learning as a mechanism for improving performance. It is anticipated that these means, in combination with high-level contextual information, can facilitate more effective video browsing and retrieval tools.

Video Annotation

Video annotations are descriptions which can be used to reference or index the associated video. Perhaps the simplest method of annotating video is for a viewer to record a description of the content. Subsequent searches on the video may access the description in order to index the video. This is how broadcasters index their archives of video material [Bouch, 96]. It is a labor intensive and imprecise approach which relies on the objectivity and thoroughness of the transcriber. Also, the textual representation of the video is static and is therefore immutable without further operator interaction. Public on-line archives may be communally annotated which would make the textual descriptions richer. Davis's Media Streams utilizes a semantically based hierarchical iconic language to create multi-layered, temporally indexed annotations of video content [Davis, 95]. In this work, the icons are associated with video during logging and there is no notion of automatically generated annotations. Chakravarthy [Chakr, 94] attempted to augment static annotations using semantic knowledge networks. While existing annotations may be enhanced by these means, they are incapable of initiating a description, given only the image or video. What is needed is video that annotates itself and can adjust its annotation based on each inquiry. In order to facilitate such flexibility the annotation has to contain a representation of the video's fundamental components. In the present work, these components are actors. Further flexibility, given the designation of the fundamental components, could be achieved by incorporating Chakravarthy's expansion method.

Methods which exploit the co-occurrence of text and images to index images have been previously explored [Bove, 83],[Sriha, 94],[Sriha, 95]. These approaches parse the associated text for key words which may indicate some sort of geometric or temporal information about the imagery. In Srihari's work the caption information is used to identify proximity information within the imagery. Bove demonstrated indexing news broadcasts by parsing the closed caption information into speakers, topics and keywords. In a similar way the present work attempts to parse the closed caption information to establish geometric and temporal relationships with the occurrence of characters in the accompanying video. However, by including computer learning and a human operator in an interactive feedback loop, the present system has the potential to get better results than methods which simply exploit the co-occurrence of text and image.

Extension of Still-image Database work

One of the more promising approaches to still-image retrieval was demonstrated by Picard and Minka [Picar, 94] who developed a still-image database browser that learns about its contents through user interaction. This work, entitled, FourEyes (formerly "Photobook with Learning") successfully demonstrated user assisted content-based retrieval on various still-image databases. The foundation of this work is the Society of Models approach to learning.

Computer Assisted Learning

The Society of Models

Although the Society of Models approach to learning is general enough to be applied to many signal domains such as audio, stock market data, biological systems, it is described as follows with respect to its first implementation as the FourEyes still-image database browser. Basically the Society of Models approach is to provide for multiple notions of similarity among the same data.

In FourEyes, each image in the database is tessellated into blocks of the same size. Each model computes features on each block. These models may, for example, evaluate each image block with respect to color, texture, position or some specific high-level attribute such as photographer's name. Collectively, the 'society' of models characterizes the processed data with multiple measures of similarity among blocks. Distributions of similar blocks within a set of images may indicate similarity among the images they compose. The goal of the FourEyes system is to find more *stuff* like the user's example *stuff* where *stuff* is a region of relatively homogeneous color or texture such as foliage, brick, straw, carpet etc.,. To achieve this, the system takes user feedback and tries to infer which of the kinds of similarity it knows can best approximate the users notion of similarity. The elegance of the FourEyes approach is that the role of computer and human complement each other. By labeling samples of the database, the human corrals the samples into perceptually similar bins. These may or may not correspond to statistically similar bins, but the system is able to accommodate any and all user specified groupings.

Details of Learning with FourEyes Method

Initially the computer builds a representation of the database by hierarchically grouping statistically similar regions together. This data representation is referred to as a similarity tree. The leaves on a given branch of the tree are pointers to image regions which have, for instance, similar color histograms, similar texture metrics, or similar positions in the source image. A 'forest' of content dependent trees may be generated from the data set. Some similarity measures may be better than others at forming groupings of certain data.

The Users' Role in FourEyes Learning

In FourEyes, the user need not have any knowledge of the internal representation of the image, this is unlike current commercial systems such as [Virage],[Nibla, 93]. A user simply interacts with the database by labeling regions and groups of regions in an image. A query into the database is initiated when a user submits a positively (and optionally negatively) labeled grouping. The computer then traverses each tree looking for nodes that satisfy the constraints of the query. Results are returned to the user in a ranked order of similarity. The user has the opportunity to correct the returned set of data, thereby converging on a refined set that is perceptually similar to the user's notion of the region of interest being sought. This feedback prompts the computer to effectively rearrange the branches of the similarity trees to accommodate the new input. So during a subsequent query for similar data, the bias formed on each interaction will be retained in the form of new groupings. How much the system learned is measured by how fast the system is at finding suitable responses to the user's query. If learning occurred, the computer should be quicker at retrieving similar data. For a thorough explanation of FourEyes, the reader is referred to [Minka, 95] and [Minka, 96].

Difference Between Analyzing and Browsing Still and Video Imagery

Extending this still-image database work into the video domain is non-trivial because of the volume of data. It is impractical and inefficient to simply build a still-image database from a set of sampled frames from the video. A more prudent method would select regions which were representative of the type of data in the database. But, which regions may be considered representative varies. The extension of FourEyes into the video domain is an exercise in picking representative regions to process, given the type of data being queried. The temporal cohesiveness and continuity of video as a database provides means for somewhat intelligent region selection by looking for regions of significant change. As mentioned above, attributes of situation comedies as a venue offer high-level features which help constrain the selection problem.

Research Approach

The classic approach to problems in pattern recognition is to 1) pick a set of features which best characterize the data 2) train by analyzing the data according to some metric 3) classify the results of analysis according to some classifier 4) test the classification of new data against the classifier. What distinguishes the present approach is the ability of the system to pre-select regions for the analysis and to adjust and learn classification about the data based on human input.

Chapter 3

PREPROCESSING METHODS

Bootstrapping the Learning by Example

The goal of gathering the script and closed caption information is to explore to what extent this high-level information can be used to assist the system in learning the representation of each character. As will be demonstrated, this integrated knowledge can be used to initiate the learning if the media are properly aligned. A detailed description of the alignment procedure is described below. As an overview however, the learning process can be bootstrapped by taking advantage of the correlations between script and captions on a shot-by-shot basis. As prescribed by the script, each scene, which is merely a sequence of shots, may be labeled with the characters who appear in it. The knowledge of who could be in the scene limits the number of classes to which the patterns could be assigned by the FourEyes learning system. The objective then becomes to extract regions in each shot which are likely portions of the characters' representation. Reasonable assumptions about the way television situation comedies are filmed provide one form of implicit context and makes the task of harvesting regions relatively straightforward.

High-Level Feature Extraction

High-level information is derived from the closed captions and script. Although contextual information of various kinds could be entered manually by a user as mandatory in FourEyes, the following method outlines a means of automatically extracting it from the accompanying script and captions, for the purpose of constraining the pattern recognition and analysis problem.

Correlation of the Script and Closed Captions

Although, the script discloses *who* says *what*, it does not specify precisely *when* the snippet was spoken. In contrast, the closed captions disclose *what* was said and approximately *when*, but do not always indicate by *whom* the captioned words were spoken. However, the combination of the script and closed captions helps indicate *who* said *what when*. Often the position of the caption on the screen is indicative of the position of the person who spoke it. This is especially true for the Seinfeld database, when more than one character is speaking. Such information can assist the system at getting a handle on *where* the speaker is in the frame. The proximity of the character to the left margin of the caption indicates which character is speaking the particular caption. Figure 1 illustrates a pair of frames in sequence where the position of the left margin of the caption indicates the position of the speaker in the frame. While this may be an indication, it does not guarantee precision. For instance, if the speaker is off screen, the captions can be closer to a non-speaking actor. This information was not ultimately incorporated into the experiments performed.

Images from *Seinfeld The Beard* Scene 2, Shot 62. Images Appear Courtesy of Castle Rock Entertainment

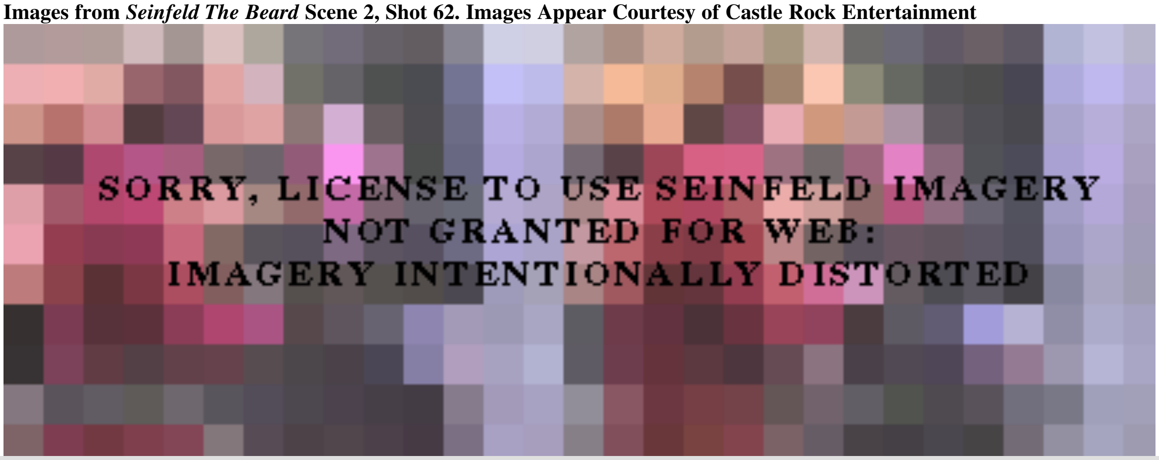

Figure 1 Position of caption indicates position of speaker.

Analyzing the Closed Captions

A Data Recovery Unit (EEG Model DE 152) converted the closed captions into a serial stream of data which the computer received. Custom software decoded this data stream into a file of captions. In addition, the software also decoded the location of the left hand corner and the onset and duration times of each caption.

Assumptions about Video of People in Situation Comedies

If the closed captions consistently broadcast the name of the character who spoke the caption, then the name could be used to label the shot with some degree of reliability. The following hypotheses are considered in this regard: 1) a person who speaks in a given shot is in the shot, or at least very relevant to the shot and 2) the person speaking is the one moving the most.

Hypothesis 1: Character speaking is in the shot.
This is a generalization which holds with few exceptions. For instance, in Shot 20 of *The Beard*, the script states:

NOT PUBLISHED ON WEB VERSION OF TECHNICAL REPORT

Thus, the script explicitly states "(V.O)" for voice over. However, GEORGE is speaking on the intercom and is not visible on screen, his caption:

"..GET IT ON ..WITH YOUR BAD SELF"

does not indicate that it is a voice over, although one might be able to infer it from the previous caption "[BUZZER SOUNDS]". But these type of exceptions are few. More common are the instances where a caption is being spoken by a character either peripherally on camera or entirely off camera. In these scenarios during a conversation, the camera will frequently linger on the person who reacts to the line being spoken.

Hypothesis 2: Character speaking is the one moving the most:
This assumption is guided by the observation that when characters speaks, they move their heads, gesture and are the focus of the camera's attention. If there are other actors in the shot, they are generally less active as to not upstage the one speaking. Of course there are violations of this premise too as in Shot 25 where Elaine talks and both Jerry and Kramer are moving around her. Figure 2 illustrates the duration of the caption in frames for Shot 25. The circles indicate frames during which at least one region was attributed to that

character. The graph illustrates a violation of the hypothesis that the person speaking is the one moving the most.

The methods of parsing the captions are imprecise. They can only be used to seed a label and are not robust. In combination with other methods of signal analysis, however, they can be used to identify the people likely present in a given shot or scene.

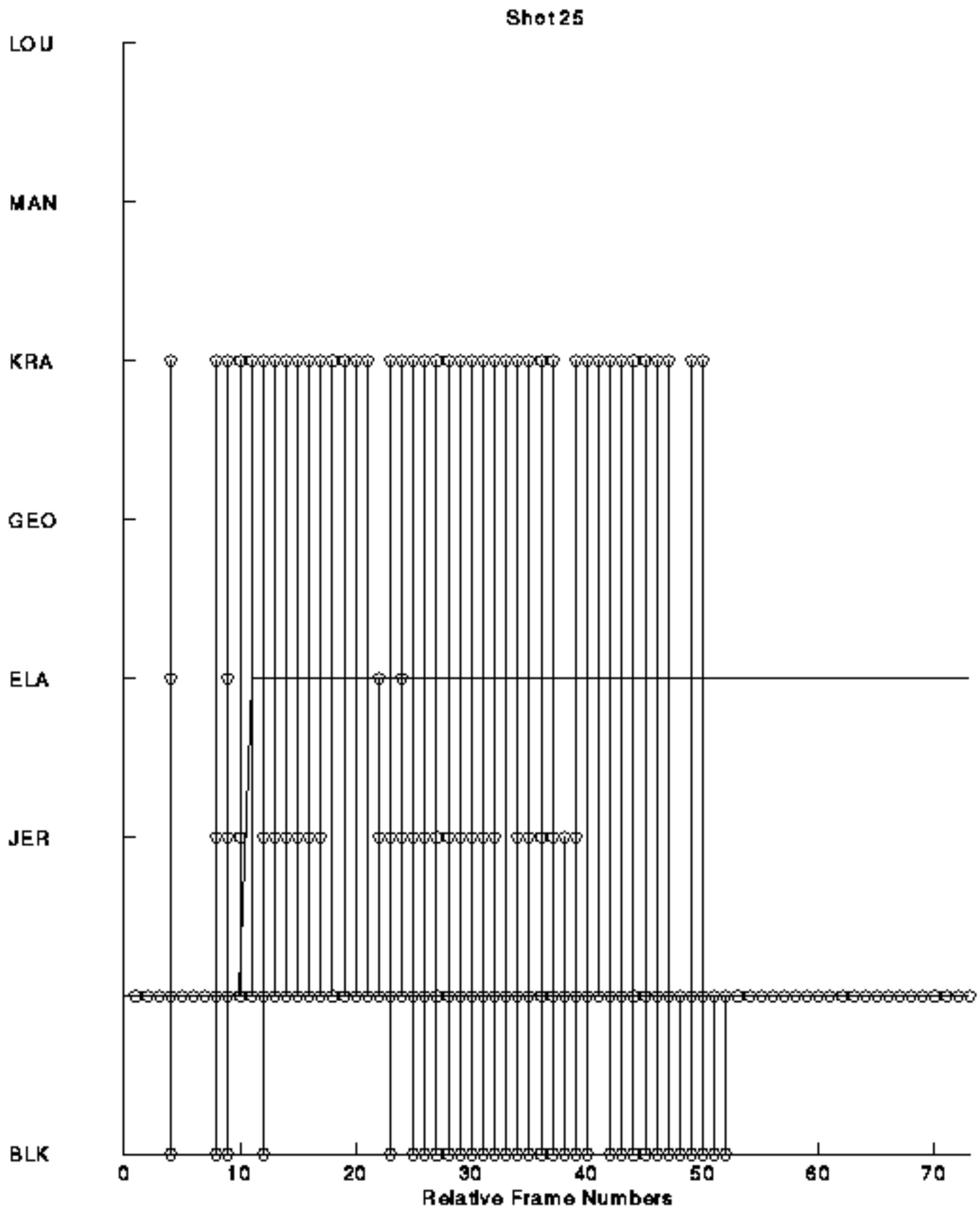

Figure 2 The person who speaks is not always the one moving the most. Circles indicate that during that frame number a portion of the frame was associated with a given character. The continuous plot illustrates the frames during which Elaine's caption was displayed.

Analyzing the Script

Scripts are textual formats which are approximated by a narrative theatrical or cinematic production. For this work, the actual script of *The Beard* was acquired from the *SEINFELD* production company and typed into the computer with all of the script formatting maintained. Although their format may vary, there are some constant script features. Scripts contain key words and punctuation styles which set off their constituents. The speaking character's name is often capitalized and the stage directions set off with parentheses. In William Shakespeare's Hamlet for instance, acts, and scenes are demarcated thusly:

ACT III

SCENE I A room in the castle.

[Enter KING CLAUDIUS, QUEEN GERTRUDE, POLONIUS, OPHELIA, ROSENCRANTZ, and GUILDENSTERN]

KING CLAUDIUS   And can you, by no drift of circumstance,
Get from him why he puts on this confusion,
Grating so harshly all his days of quiet
With turbulent and dangerous lunacy?
...

And in Seinfeld *The Beard* as follows:

NOT PUBLISHED ON WEB VERSION OF TECHNICAL REPORT
DUE TO LIMITATIONS OF LICENSE AGREEMENT

...

With the format of the scripts so regular, it is straightforward to parse them into snippets of dialog labeled with the name of the speaker. Only the following keywords were used to successfully parse *The Beard*. {"ACT", "SCENE", "SHOW", "OPEN", CLOSE", "CUT", "FADE", "ONE", "TWO", "THREE", "END" "ENTERS", "EXITS"}.

Images Appear Courtesy of Castle Rock Entertainment

Figure 3 A screen dump of the tool which correlates the script and closed captions and partitions the video into shots. The image in the bottom left is the first frame in the shot, the key frame. The image in the bottom right is a sample of the video as it plays.

Results of Automatic Correlation of Script and Captions

For the entire episode there were 516 captions and 417 script snippets. In general, it is reasonable to expect the number of captions to exceed the number of snippets because the caption length is limited to what can legibly fit on a TV screen whereas the snippet length is governed by the character's exposition. The top graph in figure 4 shows the distribution of snippets for the first 82 shots of *The Beard*. The bottom graph shows the distribution of captions for the same dialog. These distributions indicate how often a character speaks during the episode segment analyzed. It is one indication of a characters' relative presence, and could be used over large sets of data to learn priors on characters’ speaking patterns.

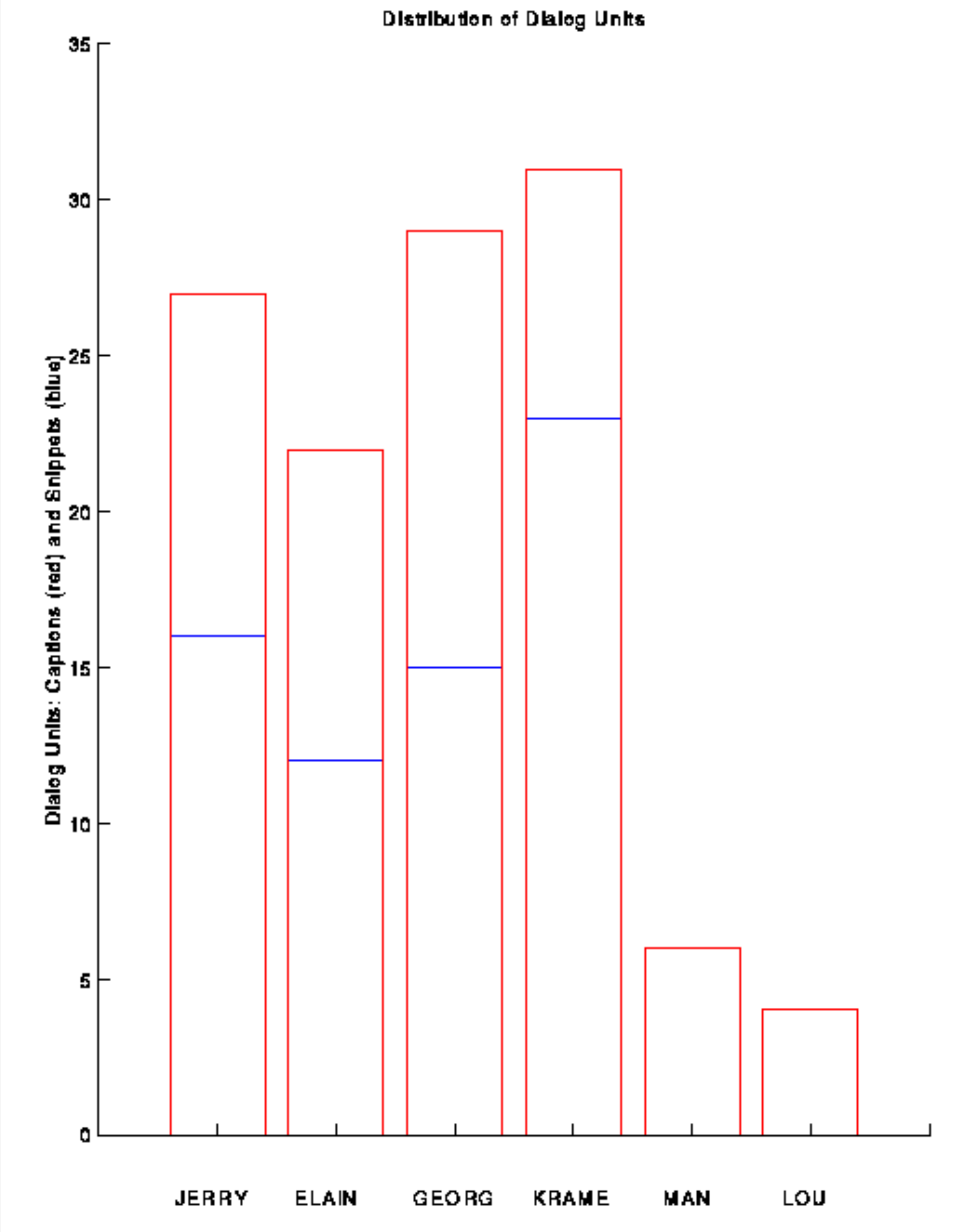

Figure 4 How often a character speaks is an indication of the character's frequency of appearance in the episode. Shown are histograms of the same text (first 82 shots of *The Beard*) broken down by character for the script snippets and closed captions. Blue bars indicate the 76 dialog snippets from script; red bars, from the 114 closed captions.

Video Preprocessing

The Data Set

The Beard is the 22 minute (*sans* commercials) episode of *Seinfeld* which was used for the analysis performed (See Appendix A. for Episode Summary). The episode was video taped from cable TV using a commercially available S-VHS VCR. It was dubbed onto a BetaCam tape for digitizing. Using a VLAN controlled SIRIUS board on an SGI Onyx machine, the first ten commercial-free minutes of the episode were digitized at full spatial (640 x 480 x 3) and temporal (30 fps) resolution. The data were smoothed with a 9 to 1 averaging filter and scaled down by a factor of 4 to its final resolution of 320 x 240 x 3. This process produced two 2 GB concatenated image files of raw image data (9000 frames each) which were stored on a local 12 GB disk tower. No compensation for 3:2 pull down redundancy was made. Only the odd video fields were acquired. In order to demonstrate the generality of methods and to avoid compression artifacts in the low-level feature analysis, no signal compression was performed on the data other than the 4:1 size reduction.

The video was displayable using a custom video browser written in C++ and SGI/GL. A screen dump of the browser is show in Figure 5. Shot detection, script and closed caption analysis was performed on the entire 10 minutes of digitized video. Subsequent low-level image processing was performed only on the first 5 minutes (82 shots).

Shot Detection

The video itself was partitioned into shots. The absolute value of the difference between the luminance histogram of each pair of successive frames was computed. Although more sophisticated methods exist e.g.[Zhang, 93], [Astle, 94], a single threshold, which was found experimentally, was able to find 182 shots in the 10 minutes of video. There were four false positives which occurred in two shots on the same set (EXT. NEW YORK STREET - DAY (1) SCENES B and D). Shots 63,64 and Shots 77,78 should have each been a single shot. Both false positives occurred during sequences in which the camera was panning severely. There were no false negatives. In the 18000 frames (first 10 minutes), there were 7 scenes and 182 shots automatically detected. On average this is a shot change every 100 frames or every 3.4 seconds.

Smart Fast Forward: A Simple Browser Indexed by Shot, Scene and Character

With the captions and script correlated and the video partitioned into shots, the video was indexed automatically by character, shot and scene. A browser was constructed which facilitated the fast forwarding of the video based on these simple content-based tags. The illustration below is a screen dump of the browser itself. Users are able to fast forward to the next shot or scene, sampling the video in semantic chunks rather than simply skipping every *N* frames. The combined script and closed caption information also identified the set of frames in which a particular caption was spoken. Therefore, users were also able to index the video by particular characters or groups of characters.

Images Appear Courtesy of Castle Rock Entertainment

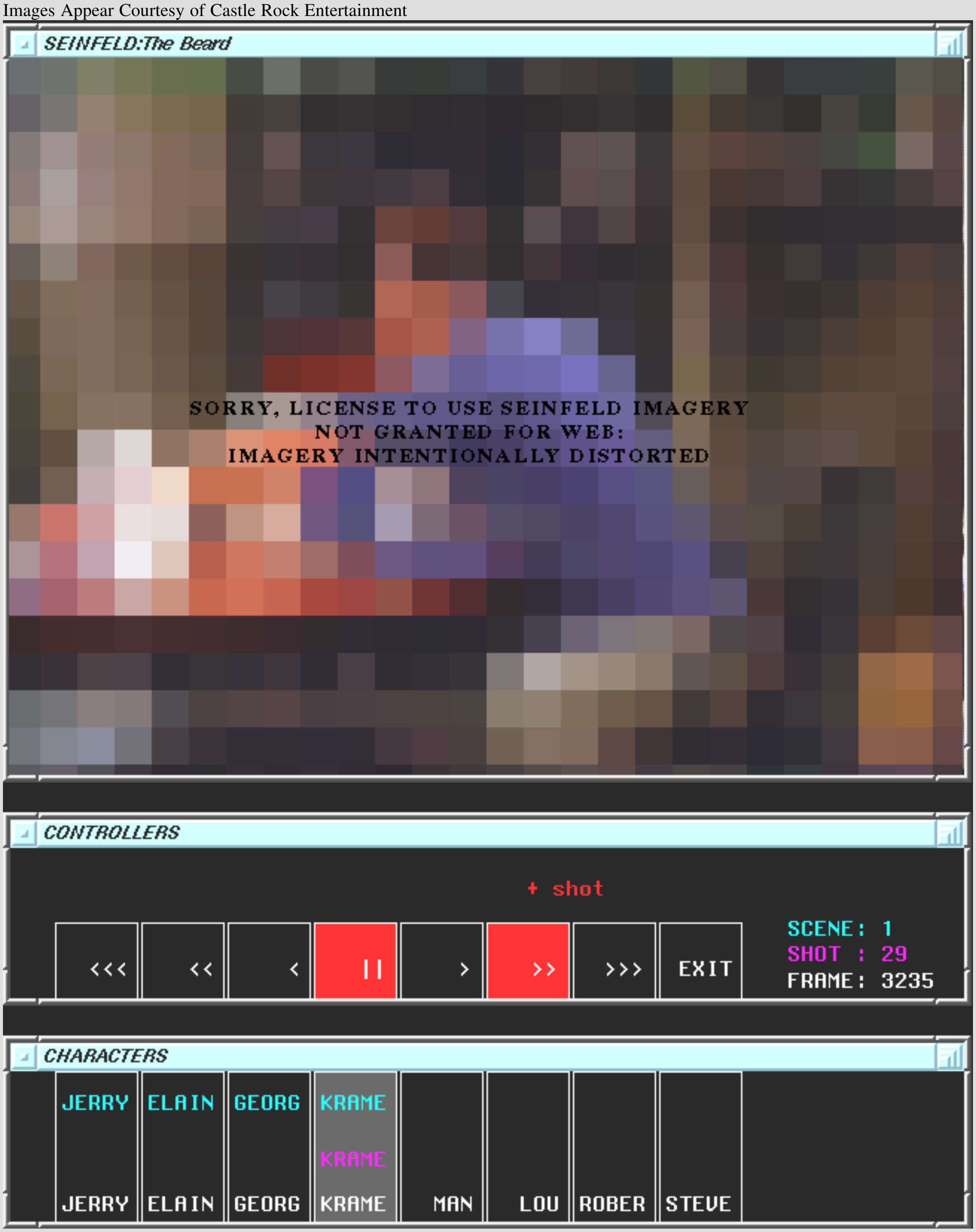

Figure 5 Screen dump of the *Basic Video Browser*. Users can skip ahead to the next shot or scene that contains a given character or group of characters. ‘Find all Shots with Kramer’. Retrieved are key frames for shots/scenes in which the specified character has a caption.

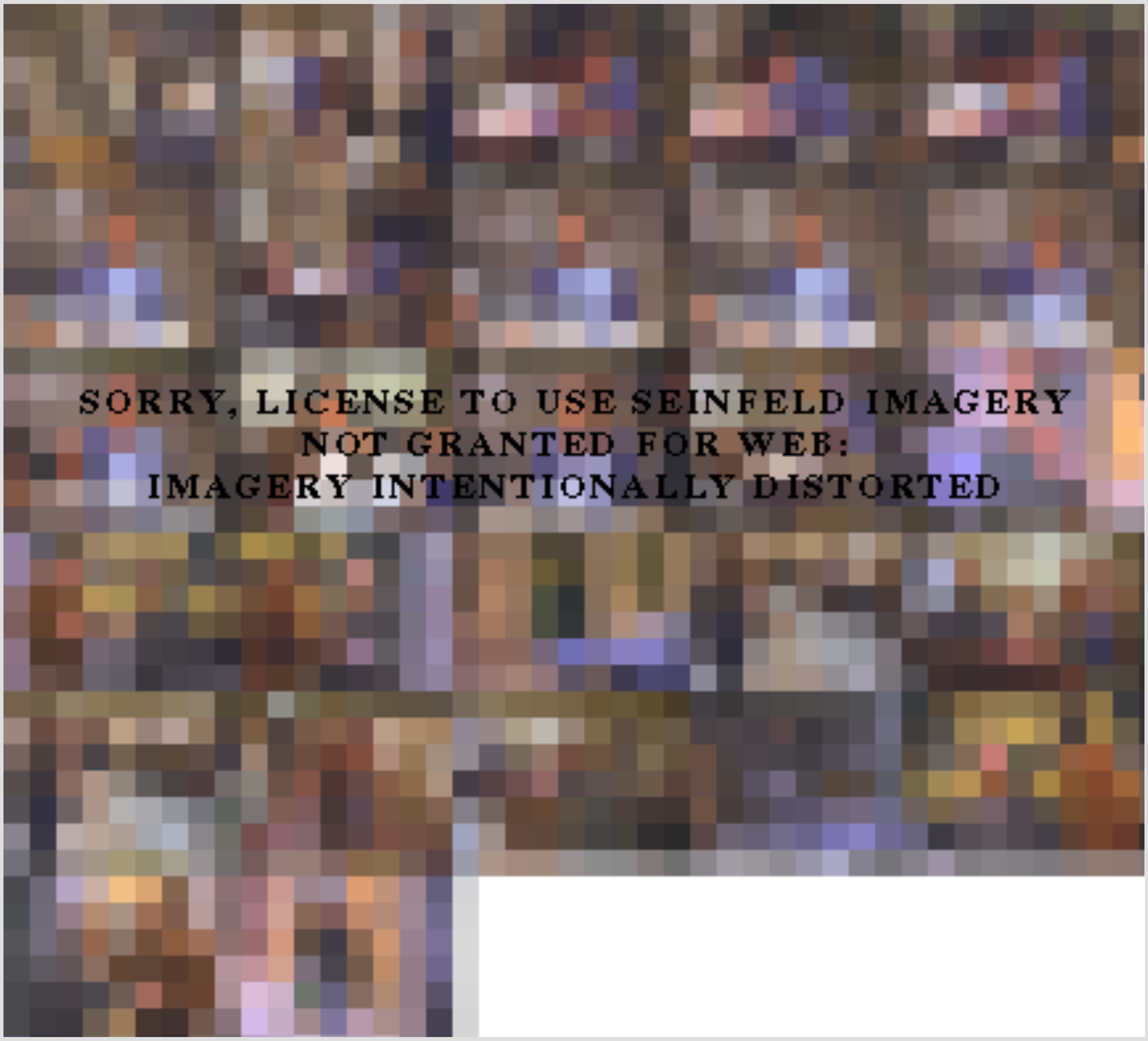

Figure 6 Key frames returned on a query 'Find Shots with Kramer.' Using the *Basic Video Browser* that retrieves shots during which a character had a caption.

Shown above are the first frames in each shot given the query 'Find me shots with Kramer.' For the 82 shots analyzed, there were 27 in which Kramer had at least one caption. The Basic Video Browser was able to find 18 of these shots. There were 9 false negatives (shots skipped that should have been included) and 6 false positives (shots included that should have been skipped). Of the 6 false positives, 4 were in fact shots in which Kramer appeared but did not have a caption. There are three possible types of false errors in this construct:

1) Shots falsely labeled Kramer in which he speaks but the camera is trained on someone else (reaction shot for example)
2) Shots which were improperly labeled due to false script/closed caption correlation. For instance, if there is a single word script snippet like ‘Yes’ but the caption says ‘Yup’, it is virtually impossible to match the two lines without incorporating a vernacular thesaurus.
3) Alignment problems which could be fixed in subsequent implementations.

Shots in which Kramer is not in the first (key) frame, but is in subsequent frames within the sequence do not count as false positives.

Towards a Smarter Browser

The Simple Video Browser described above was constructed using the automatic alignment of captions, script and shot boundaries. It would be possible to apply this technique to many styles of episodic captioned television for which a script exists. However, without the script or closed captions, the method fails. Kramer appears in many other shots in the episode and this approach will not detect him if he does not have a caption during those shots. Under these circumstances what is needed is a means of identifying the pattern in the video which *is* Kramer. To do this, the system has to recognize the subset of the video signal which constitutes Kramer's representation. Ideally, the system should exploit the information regarding who is likely speaking in the shot to seed positive examples of the characters in the learning algorithm discussed below. If it were possible to learn the pattern of Kramer in shots in which Kramer speaks, then when Kramer appeared in shots in which he did not speak, he could be recognized. If the learning and recognition were truly robust, then Kramer could also be found in video in which high-level information was unavailable.

Since the determination of who is speaking is unreliable information, FourEyes cannot use it to make automatic labels. However, the specific shot-by-shot, frame-by-frame labeling information can be imposed if the user is included in the loop. The following section outlines the method by which the computer can learn such patterns bootstrapping off the low-level image features and guided by high-level constraints.

Once the episode is partitioned into individual shots, a sequence-by-sequence analysis is performed. The goal of the following preprocessing steps is to cull regions in the sequence which are likely portions of the character's representation. The premise is that, in a situation comedy, the most salient motion events are people oriented. These events include talking, gesturing, walking and eating. One feature of most situation comedies is a style of cinematography where a set of cameras are allocated to several zones around a theatrical set. Since the cameras are locked off, most of the motion that occurs in situation comedies, and in *Seinfeld*, in particular, is motion of people or parts of people. There is no hand held camera work and few zooms, tilts, trucks or other canonical camera motions. In fact in the first 82 shots of *The Beard* there is one subtle zoom and approximately 15 pans of varying intensity.

An extension to the hypothesis (hypothesis #2 above) that the most salient motion in a shot is human motion, is the assumption that the person moving the most is the one speaking. If this premise were true, then the closed captions would correlate to the on screen activity and the speaker of any given caption could be used to label the regions associated with that sequence. This would provide a means of automatically seeding labels to regions of interest. Of course there are exceptions as described earlier in the this chapter (See section 'Assumptions About Video of People in Situation Comedies')

Camera motion is the most obvious counter example. In the event of camera motion, it is the object which is *not* moving relative to the camera which is more likely of interest. Filtering for this object or set of objects is possible by various means e.g. [Wang, 93]. The premise in this circumstance is that if the camera is moving, it is tracking a character, therefore the character is stationary relative to the frame. Except during the camera acceleration/deceleration or during the character's acceleration/deceleration, the magnitude of the motion should effectively separate the character from the background. Instances of such segmentation can be seen in shots 16 in which Elaine walks across Jerry's apartment. Steam rising (shots 59 and 77) and doors opening (shots 22, 31 and 33) were the only severe non camera, non human motion events which occurred during the first 82 shots of *The Beard*.

Unsupervised Selection of Regions of Interest

A subset of the data preprocessed above was used for the low-level image processing and learning tests reported in the remainder of the thesis. The unsupervised processing stage detailed below generated 15,052 irregularly shaped regions. These regions came from the first 82 shots of the episode which constitutes the first ~8200 frames (approx. 5 minutes of video). The assumption which governs the following processing is that in video the change in information is what is important. After all, change is what distinguishes video from still-imagery. Therefore, all changes in flow above a certain threshold were acquired automatically by the database. These changes included shot boundaries, camera motions and object motions. This step also served a practical purpose in that it reduced the set of data on which the FourEyes processing was performed.

Stage 1) Optical Flow Magnitude

In an effort to extract the regions of activity, the magnitude of the optical flow of the sequences was computed using the method of [Lucas, 81]. The image sequence was converted to luminance, then smoothed with a five tap temporal filter (see appendix C). The optical flow was calculated using a three parameter estimation. The magnitude sequence was scaled and then thresholded. Each resulting magnitude frame was filtered spatially using a 3x3 median filter and then segmented using method detailed below. The net result was the isolation of regions of substantial coherent motion. A bit map for each region was generated and later used to mask the source image for further processing by FourEyes. A bounding box around each region was used to indicate to the viewer the neighborhood of the underlying region.

Images appear Courtesy of Castle Rock Entertainment

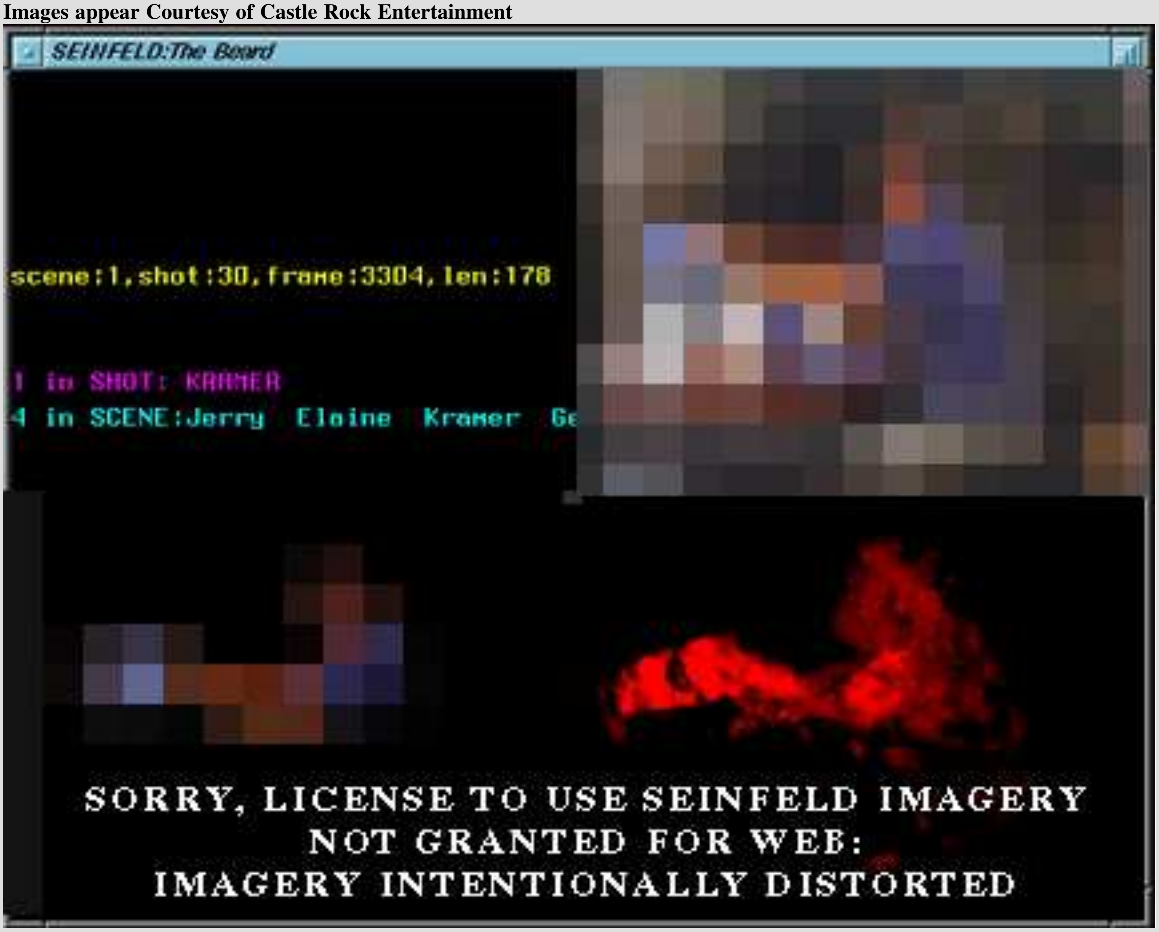

Figure 7 Screen dump of tool which segments optical flow and culls regions of interest based on the segmentation of the flow image. The bottom right is a still of the motion magnitude. The bottom left is the segmentation of the motion magnitude. The upper right is the bounding box of the segmentation superimposed over the source image.

Stage 2) Luminance Segmentation.

Luminance Segmentation Procedure: walk a 3x3 kernel over the image. If the value of the peripheral pixels is close enough to the center pixel by some user prescribed threshold, then include the pixel in the present segment. Else assign that pixel to its own segment. Repeat until walk is done.

Static objects may be of interest to someone doing a database query. The case when an object moves and then stops moving is know as *dwelling*. A dwelling object would be invisible to the present system without this stage, which samples full frames at points of significant change during the video.

Full image frames were extracted under two circumstances:
1) Shot key frames: The 5th frame of each new shot was sampled.

2) Motion key frames: During instances of camera motion, new luminance samples were extracted every 15 frames. These new key frames represented the background imagery revealed during camera pans.

Detecting dwelling regions within the shot in which they are active may be possible using the premise that things that stop moving stay where they were last located. Tracking systems (see [Intil, 94] for survey) are designed on this assumption and might be better at localizing dwelling regions than this method which takes a somewhat random approach in sampling at shot boundaries and during camera motions. However, simple key frame samples may be effective at finding objects that moved in one shot and appear static in other shots. In this scenario, tracking systems would have to be reinitialized. With respect to finding people, motion segmentation can be expected to find regions on the body that move coherently --hands waving, heads shaking, arms swinging, legs walking, etc. It may not be possible to distinguish between individual characters based exclusively on say, hands waving, no more than a human could identify a friend based on the same stimuli.

Both shot and motion key frames were processed as follows:

The color image was converted to luminance and then segmented by means as described above. However, since there is no unique segmentation of an image, segmentations were performed at 4 different thresholds. This pyramid scheme generated multiple segmentations for each image in an attempt to span the space of perceptually relevant segmentations. Regions below a certain minimum size were ignored by the system. This pyramid or multi-threshold scheme was not necessary when segmenting the relatively homogeneous motion magnitude images because a single threshold tended to segment the images into perceptually salient regions.

To summarize, preprocessing culled regions of significant and homogenous change in the episode segment analyzed. These regions of changes included 1) temporal discontinuities in the form of shot boundaries 2) motion discontinuities in the form of segmented optical flow magnitude 3) local luminance discontinuities within key frames and during periods of continuous camera motion. The union of the regions from 2 and 3 were available to FourEyes for subsequent image analysis.

.Images appear Courtesy of Castle Rock Entertainment

Figure 8 Screen dump of tool which segments the luminance image of the key frames. Two of the four levels of the segmentation pyramid are shown in the bottom half.

Results of Preprocessing

The graphs in figure 9 show the shot-by-shot distribution of regions preselected by these means. They are plotted at the same scale to facilitate comparison.

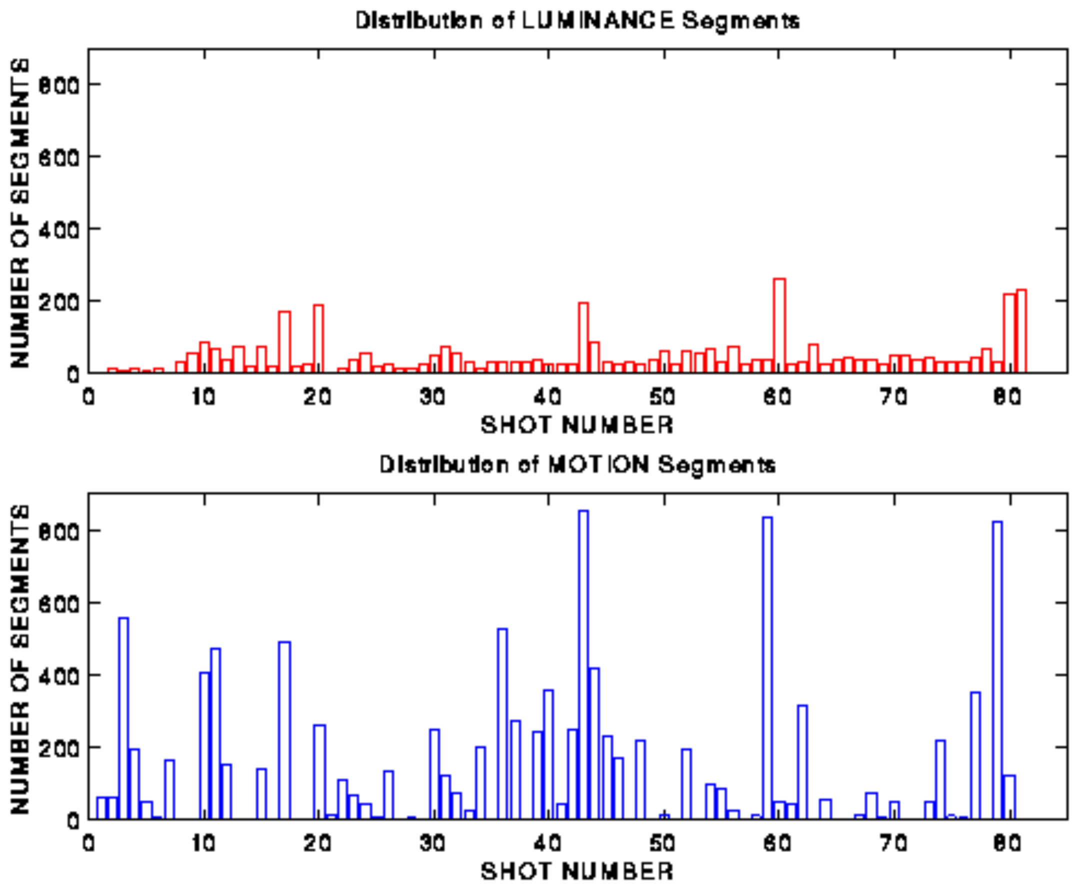

Figure 9 Distribution of segments by source and by shot membership

Notice how the spikes in the motion plot are echoed in the luminance plot. This is as a result of camera motion. Recall that during camera motion, new samples of luminance data were extracted periodically (non shot boundary key frames). Inspection of the video confirms that these spikes correspond to shots with camera motion. Figure 10 presents the same information with all shots normalized to the shot length in frames.

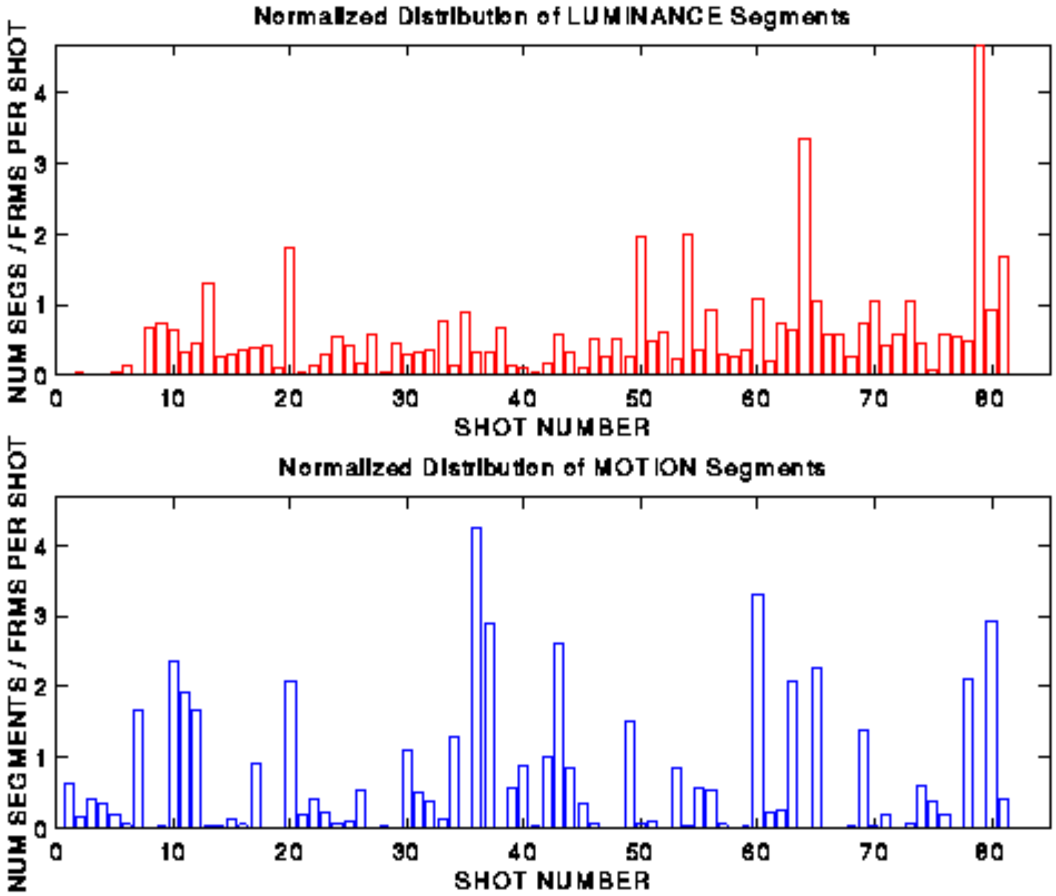

Figure 10 Distribution of segments by source and by shot number normalized by shot length in frames.

Bars on these graphs correspond to the relative 'activity' on a shot-by-shot basis in units of segments per frame per shot. For instance, in the normalized motion plot Shot 60 is a long pan with four characters walking in front of each other and Shot 36 is a pan which follows George as he does a twirl. These activities generate many motion segments despite the shot length normalization. In the luminance plot, Shot 50 is a very high contrast image of George sitting in front of the Chinese food boxes and venetian blinds. The luminance segmentation of this image results in a very large number of regions. Another instance occurs in Shot 79 which contains the very high contrast letters “Tonga Lounge” behind Kramer and Homeless Man. The luminance segmentation of this ‘busy’ image contains the most segments.

With the assumption that all motion is derived from movements of people, any given blob is likely a portion of the representation of whichever character is present in the shot Since the correlation of the script and closed captions have pre labeled each shot, the blob can be labeled with one of the N characters known to be present in the sequence. But these labels will have to be marked as unreliable because precision is not assured.

Output to FourEyes

Each bitmap and rectangular region defined by the bounding box of the bitmap in the source frame was saved for later processing by FourEyes. Each image segment and mask was read in to FourEyes and then mapped to a grid. This permitted analysis of irregularly shaped regions in terms of individual blocks. It also introduced a measure of quantization error as blobs were forced to adopt or abandon more of the background region than was originally extracted by the above mentioned segmentation method. This is an artifact of implementation which should be improved in future systems.

Low-level Feature Extraction: Model Members of Society

Each 320 x 240 x 3 frame image was tessellated into a 16x16 array of 20x15 pixel blocks. Each segment extracted via luminance and motion segmentation in the preprocessing stages mentioned above was requantized to this grid. Following this coarse mapping each block within each segment was processed by the Society of Models discussed in Chapter 2.

Intuition and experience gained from using FourEyes suggested which models might be the most appropriate in distinguishing the many appearance-based attributes of the characters. Given that the objects of interest were people, color and texture metrics seemed to be obvious models to evaluate. By observation, the wardrobes of the characters were quite distinguishable by both color and texture. However, since all of the characters in the episode have virtually identical skin and hair color, it was expected that regions composed of hands and heads, which were abundant in the data set because of their tendency to move, would not be distinguishable by color or texture alone. As mentioned earlier, a face recognition system might also be useful, but this would involve first solving the problem of finding faces and other steps which would reduce the generality of the system.

Figure 11 Histogram intersection between segments derived from luminance and motion segmentations may increase the likelihood that an individual region in the luminance segmented image would be associated with a region derived from a motion segmented image. The luminance segments illustrated above could be considered *children* of the *parent* motion segment of the same object and the segmentation of key frames in shots that had minimal motion could be used to find related objects in other shots in which objects with similar color histograms moved.

Model 1) RGB Color Histogram

Each block was quantized into one of 32 uniformly spaced bins for each of the three color channels. The three 32 level histograms were concatenated into a 96 unit vector which represented each block's distribution of color. A single vector for each segment was computed via block-wise averaging. The histogram intersection [Swain, 91] was used as a similarity measure between regions. Consider, for example, two regions being compared: A to B. A is a full frame image. B is half the size of A by truncation. For real (non uniform) images, Euclidean distance metrics would find these histograms to be dissimilar but the histogram intersection of both finds them to be comparable. This metric is asymmetrical. But it is beneficial when comparing images of different sizes. Below is an example of the benefit of using histogram intersection in this database. A query on Kramer's hands finds other hands, but also finds them in situ. The asymmetry is that given the query of the large frame, one would not expect the hand to be returned.

Another justification for the inclusion of the luminance segmentation imagery during preprocessing is that it may help *bleed* features across shots. Consider the ideal case of a body moving. The motion segmentation may acquire a coherent blob of the torso, head and arm. The luminance segmentation on the same image may segment those regions individually, depending on how the body is clad. Using the histogram intersection method, the individual (luminance segmented) regions should match to the motion segmented collection of body parts. This matching, or *bleeding* across features, may help identify objects across shots where the motions of the individuals may be quite different. Figure 11 illustrates this idea graphically.

Images Appear Courtesy of Castle Rock Entertainment

Figure 12 RGB Histogram intersection gets Kramer's body from a sample of his hand. Screen dump of FourEyes, the upper left is the test image, the others are the ranked responses in raster order.

Model 2) Normalized RGB Color Histogram

The procedure is the same as described above except instead of computing the histogram intersection, the integral of each histogram was normalized to size 1. The Euclidean distance between vectors was used as a measure of similarity between segments.

This normalization step was performed in order to facilitate color comparisons between segments of different sizes. Consider, for example, two regions being compared. One a full frame the other a zoomed in version of that frame. The histograms of these regions are similar, but without the normalization steps, the similarity between them would not be evident. In the database of Seinfeld, one would expect the histogram intersection to out perform the normalized histogram since the distribution of shots at different scales is fairly narrow. There are few zooms and the variance of scales is determined by the character's distance to the camera. This variance is small. This intuition was developed by watching the style of cinematography used in the show and not by any rigorous analytical means.

Model 3) Ohta Color Histogram

The Ohta color space is an expression of the RGB color cube in terms of eigenvectors calculated over real scenes [Ohta, 80]. Histogramming in this space can be expected to produce better segmentations then in the straight RGB space for some data, because it segments real world imagery with respect to its principal components. For purposes of calculation, the Ohta model is effectively a rotation of the data along the eigenvectors of the RGB color cube. As above, each 15 x20 pixel block was quantized into one of 32 uniformly spaced bins for each of the three color channels. The three 32 level histograms were concatenated into a 96 unit vector which represented each block. A single vector was computed for the segment via block-wise averaging. Histogram intersection was used as the measure of similarity.

Model 4) Normalized Ohta Color Histogram

Same process as the Normalized RGB Color Histogram (model 2) only performed in the Ohta color space. The block-wise average histogram was computed over the segment. Euclidean distance was used to measure similarity.

Model 5) DCT of the DFT Magnitudes--a texture metric

Procedure: Tile the image into blocks of size 8x8 (arbitrarily chosen). Take the magnitude of the DFT of each block in order to make comparisons shift invariant. Take the top 10 (arbitrarily chosen) coefficient DCT of the result in order to express the texture discriminant in a compact form. Use the Mahalanobis distance between the vectors as a measure of similarity. Local covariance was used for segments of size greater than 10 blocks. For segments with fewer blocks, an identity matrix was used for the covariance.

Other texture models were considered but were ultimately discarded: The strength of the Mutli Resolution Simultaneous Auto-Regressive (MRSAR) model lies in its ability to characterize complex textures across scales. Its effectiveness on a database of natural scenes which might include foliage, water or bark has been shown [Minka, 96]. In consideration of a database composed of relatively homogenous clothing and man-made interiors, its effectiveness is dubious. The Discrete Cosine Transform (DCT) was considered and preliminary experiments found it to be considerably sensitive to shifts in the placement of the analysis blocks during tiling. That is, the imagery varied enough that as each tile was analyzed its response was considerably different depending on how the tiles were mapped onto the image. Therefore comparisons across image regions were uninformative by this metric. *Eigen*vector analysis was rejected because comparison of irregularly shaped regions of such a variation in size is impractical. The DCT of the DFT magnitude is an approximation to the *eigen*vector decomposition.

Approximately 750,000 15x20x3 pixel tiles were analyzed by FourEyes. This is about 675 MB of image area or 35% of the 8200 frame movie.

Formation of Similarity Trees

With the metrics run on the individual regions a single-link hierarchical clustering was performed on all segments. A similarity tree was constructed for each metric using the FourEyes learning system. This bottom-up approach generated similarity trees like the following in figure 13 where each node collects nodes of similar items.

Other kinds of hierarchical clustering were evaluated but ultimately abandoned. Complete-link, unweighted average, weighted average, and Ward's method [Jain, 88] all resulted in grossly inferior performance in terms of their ability to approximate the underlying feature space.

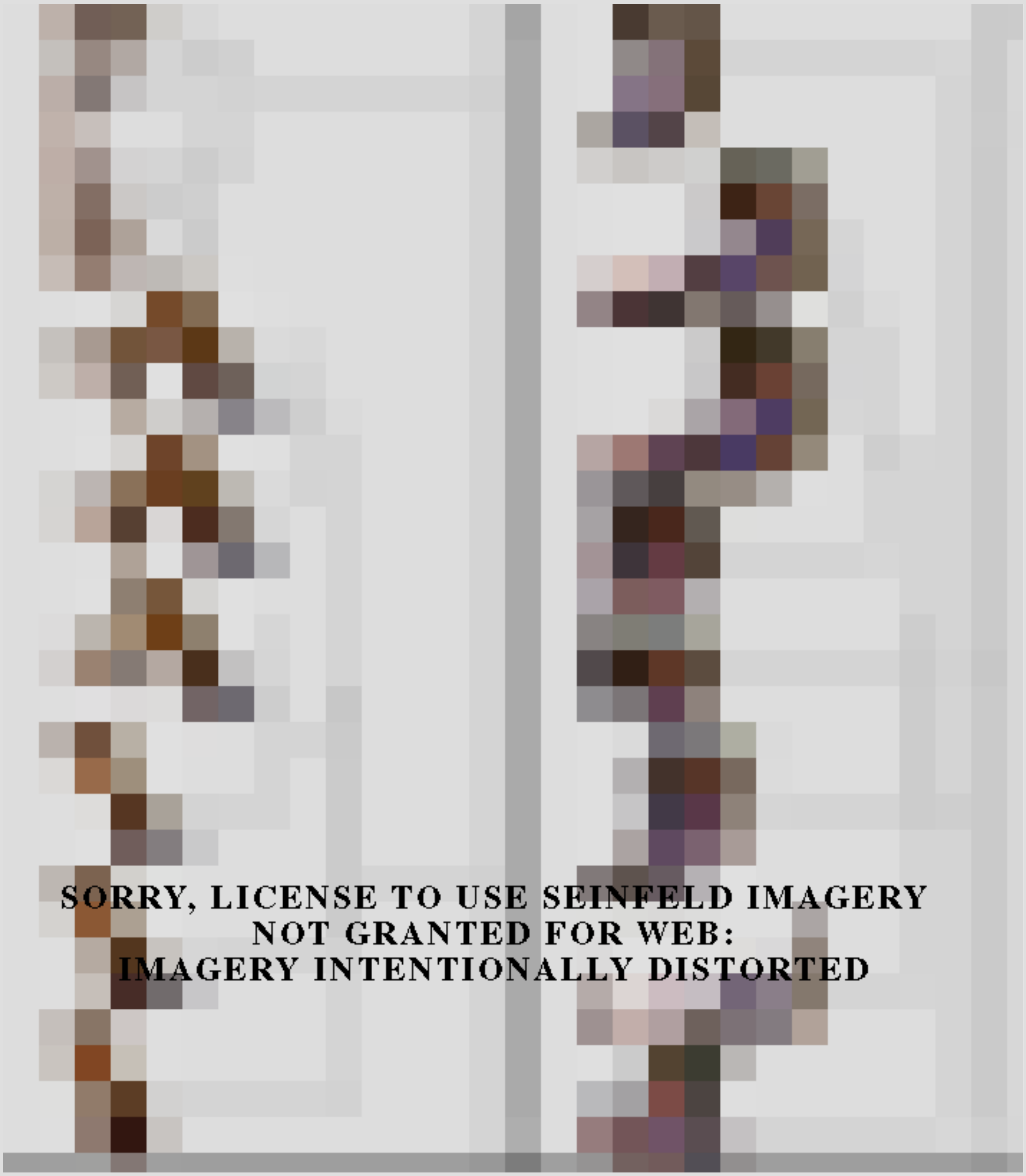

Figure 13 Samples of different branches of the same RGB histogram similarity tree. One the left, George segments, one the right, Kramer.

Chapter 4

LEARNING EXPERIMENTS

Determining Ground Truth Labeling for the Regions

In order to evaluate the success of the preprocessing stage in culling regions related to the characters, the content of each region had to be determined objectively. To generate this classification, the set of 15052 regions was first decimated into 10 equally distributed subsets. Every tenth segment was presented to a human subject for manual labeling. Each subject evaluated a different set of ~1505 segments. The task was to label the region with the name of one of the 6 characters who appears in the first 82 shots or to leave the region blank.

The specific instructions were
1) Ignore full frames
2) Give each region a single label
3) If more than 50% of the box contained a portion of a given character, label the box by that character's name.
4) Leave all other boxes blank.

Potentially there are several sources of error in this approach
1) Since regions are irregularly shaped, bounding boxes were used to rapidly indicate the regions of interest to the viewer. The people determining ground truth evaluated containment based on the bounding box and not the blob itself. Therefore their classification decisions were potentially influenced by superfluous image information.

2) A single label may not be sufficient for a given box. As in the case of box number 11612 (frame 6523) where Homeless Man and Kramer exchange a piece of Tupperware. The motion of both arms forms a single motion blob. These ambiguous regions were left 'blank'. It is indeterminate whether such regions should be labeled 'Kramer', 'Homeless Man', 'blank' or 'Chinese Food'?

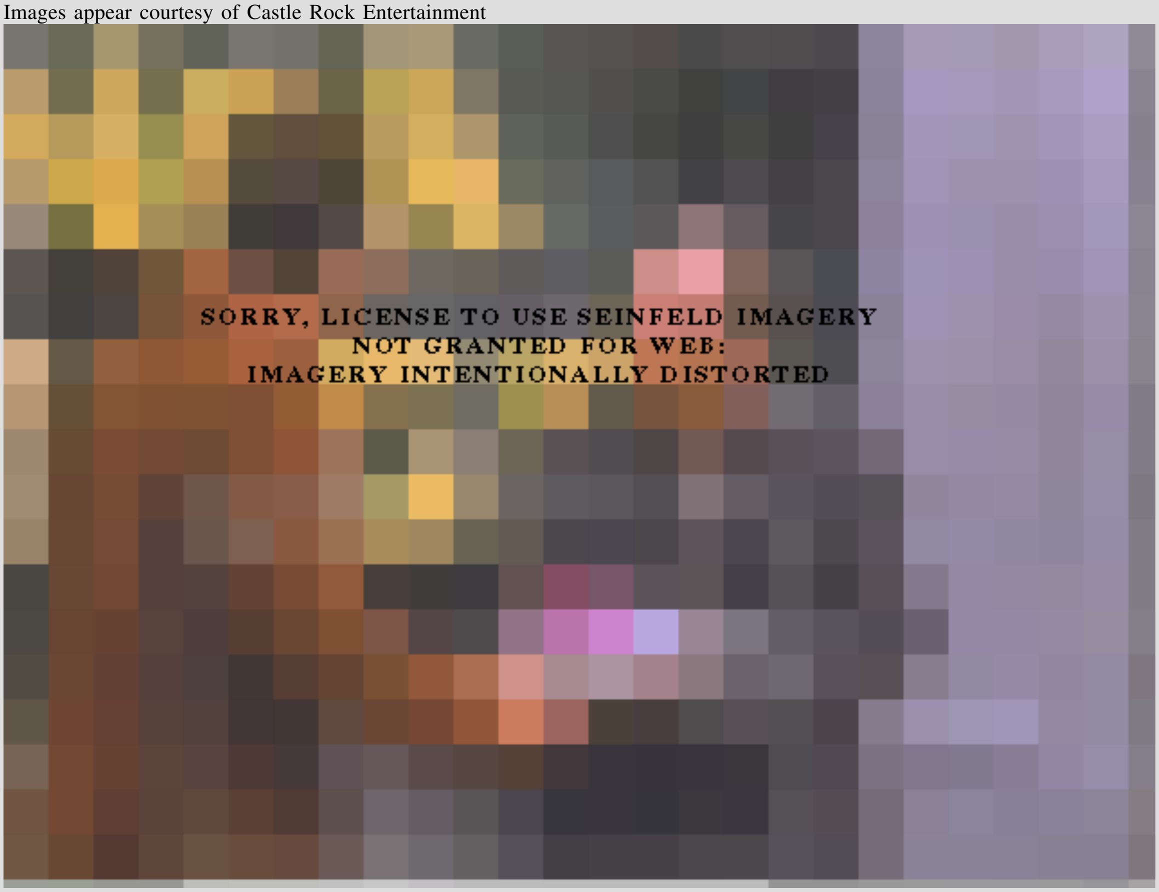

Figure 14 Ambiguous Labeling task. Is it Kramer, Homeless Man or Tupperware?

3) Other sources of error include mistakes or biases from the annotators. It should be noted that since the regions were viewed in the context of the full frame, and not as individually masked regions, the understanding of how people move from frame to frame may have influenced the labeling. A more conservative manual labeling task would have the viewer watch the video and then manually label only those boxes or regions which are recognizable. But perhaps the most precise method would have the user outline the object itself and then the area of overlap with available regions could be computed.

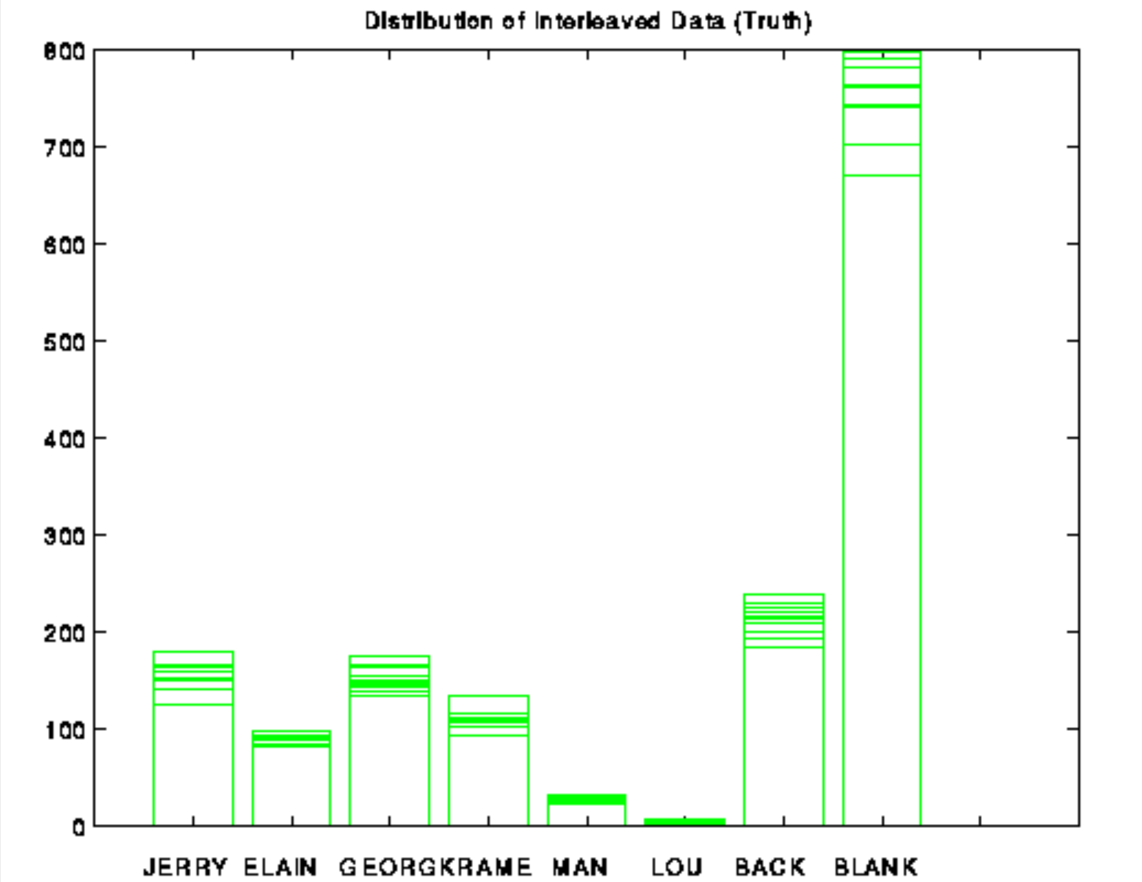

Figure 15 A comparison of the relative distribution of regions in the decimated and manually labeled data set. The variance is narrow.

Figure 15 is a graph of the ground truth of each of the decimated data sets superimposed. It illustrates the rather narrow variance in the method used of decimating data to collect ground truth labels.

Regions of area greater than 95% (Background Frames) of the image frame were discarded from the labeling task.

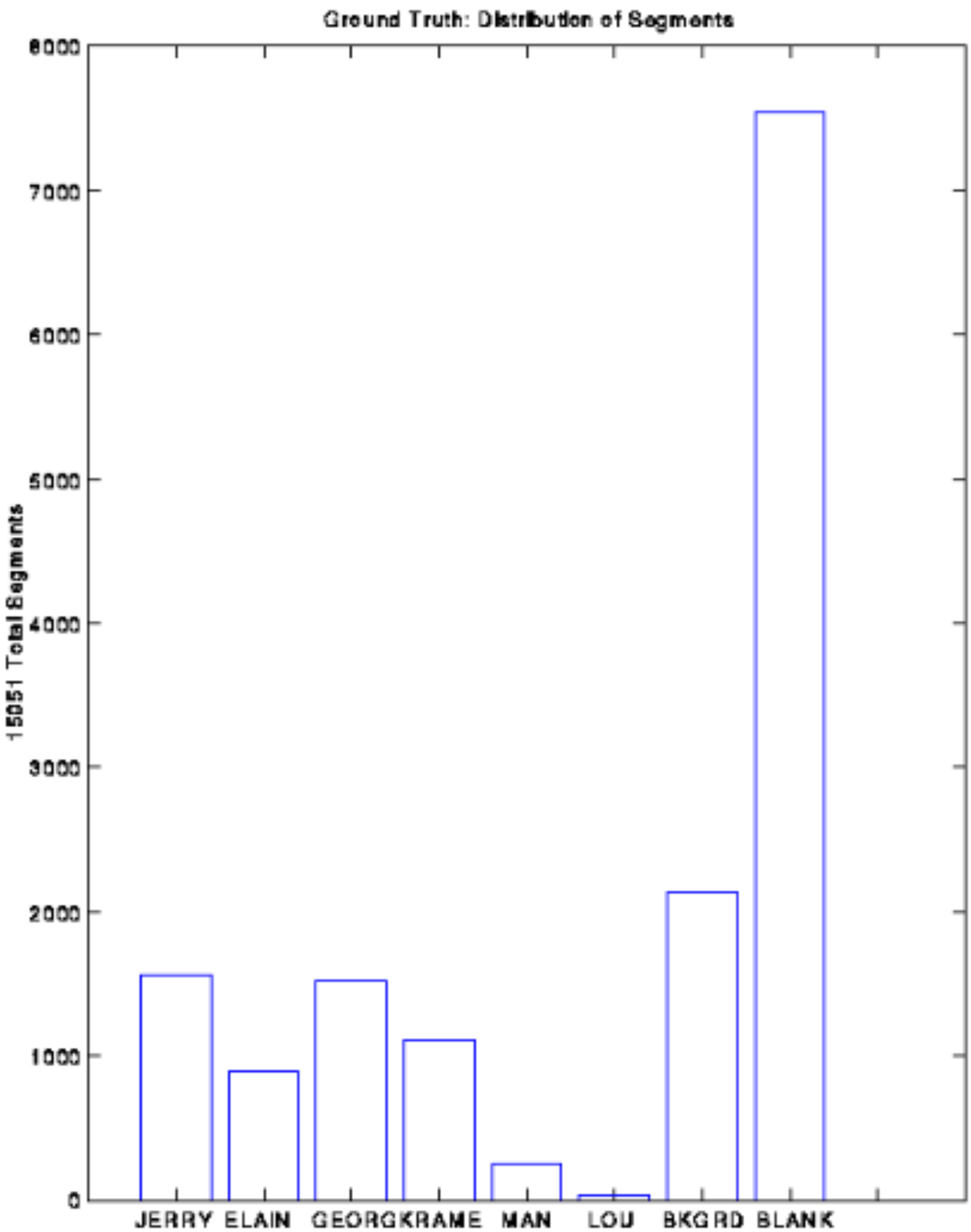

Figure 16 Distribution of 7 alternative forced choice ground truth manual labeling of all automatically extracted segments in the first 82 shots of *The Beard*.

The results of the 7 alternative forced choice labeling is graphed in figure 16. These distributions constitute the priors and were used to measure learning performance. Ideally these data would be gathered over a library of episodes and would be normalized on an episode-by-episode basis given the tendency for some episodes to feature one character more than another. Notice the large number of blanks in the right-most 'blank' column. A shot-by-shot analysis of these unlabeled regions is shown in figure 17.

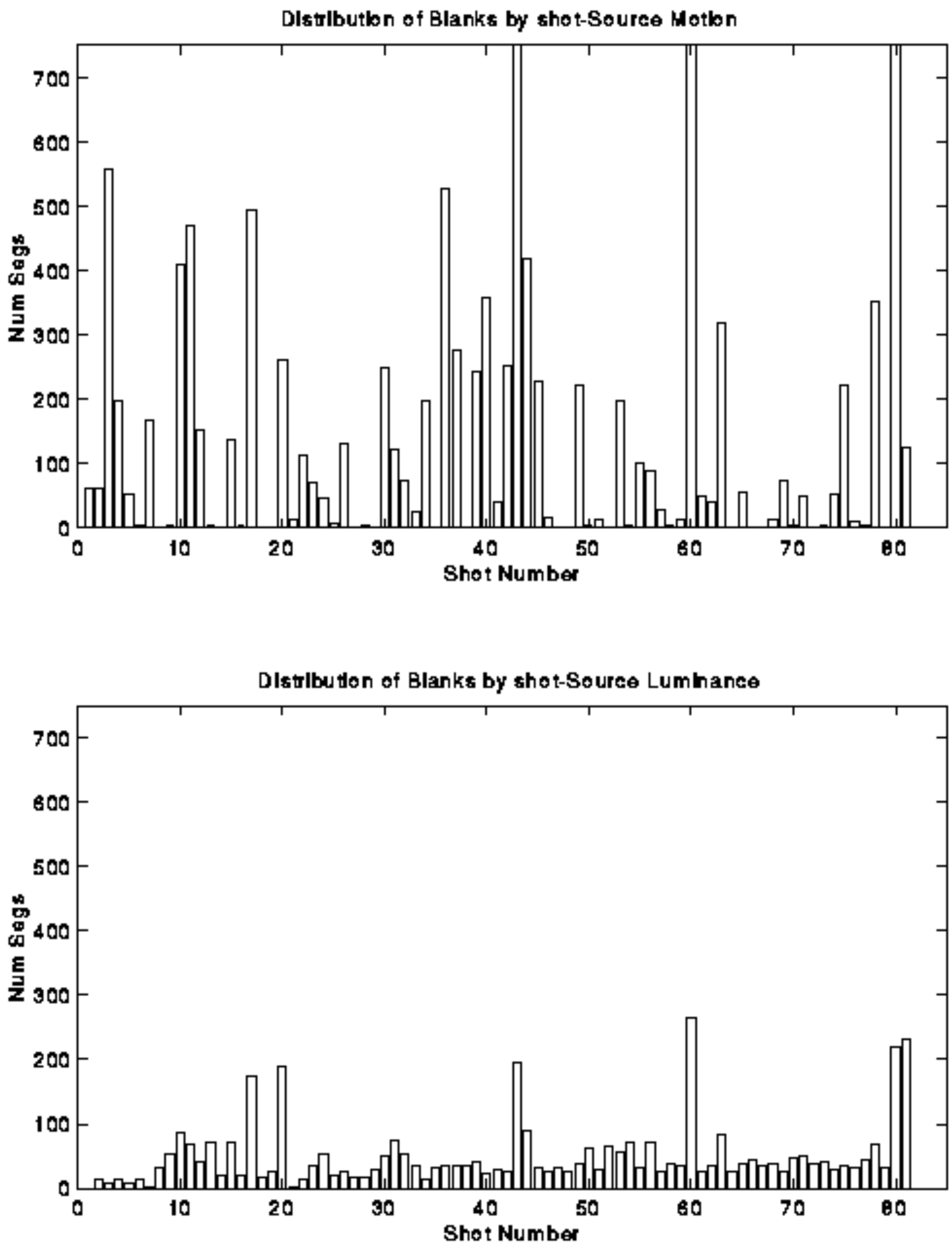

Figure 17 Analysis of manually labeled blank segments. A shot-by-shot comparison between those derived from segmentation of the optical flow images and those from luminance key frames.

The spikes correspond to shots in which the camera moved. The top graph illustrates the blank regions that were derived from motion segmentation, the bottom graph illustrates those derived from luminance segmentation. These spikes indicate that the largest number of blobs, not associated with characters, occurred during camera motions. Also notice how the spikes in the upper graph echo those in the lower graph. Recall from the description of luminance segmentation above, that during camera motions, new samples were acquired periodically from the luminance image in order to cover all of the background during these instances of significant change. Inspection of the video confirms that these spikes correspond to these new samples.

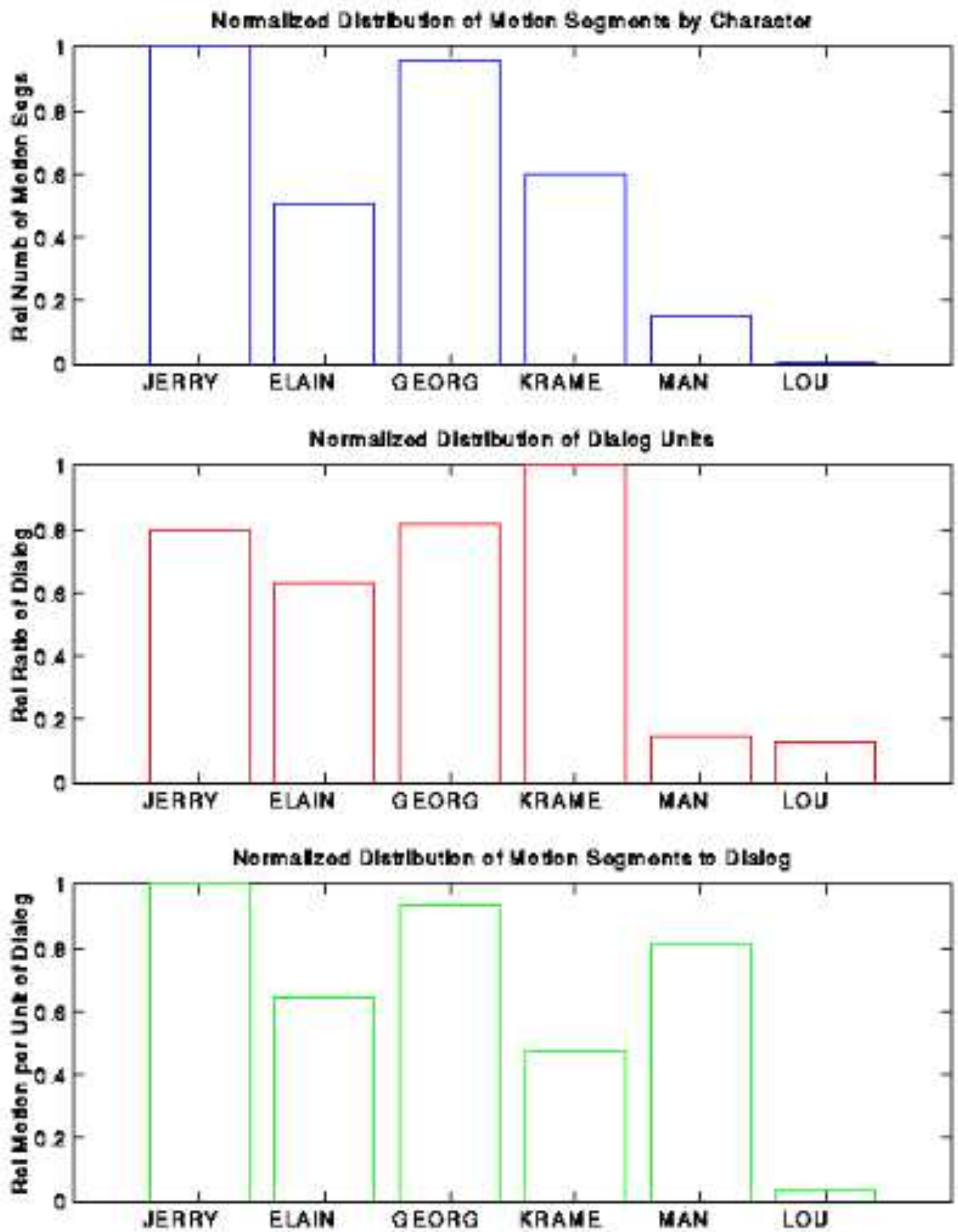

Figure 18 A measure of a character's relative 'presence' or 'activity' can be determined by the Normalized Distribution of Segments to Dialog unit.

The histogram in figure 16 is one measure of the presence of a character in the episode. However a more telling measure of the relative *presence* or *activity* of a may be calculated by normalizing the number of motion regions associated with the character to the frequency with which the character speaks. This relationship is illustrated in the bottom graph of figure 18 entitled ‘Normalized Distribution of Motion Segments to Dialog.’ The measure 'dialog unit' was calculated as follows:

$$\frac{(\text{number of captions attributed to character}) + (\text{number of snippets attributed to character})}{2}$$

Analysis of the graphs in figure 18 suggests the relationships between the frequency of speech and the extent of motion. During the 82 shots analyzed, Jerry is the most outspoken and active character in the episode. The disproportional distribution of his activity as compared to that of other characters in the episode is confirmed by the observation that in Show Open, he is gesturing and speaking by himself during 6 shots.

Eliminating the 'blank' Class

The 6 distinct character classes corresponded to regions which were hand labeled as positive examples of individual characters. The blank classes, which include the background frames could also be members of the 6 character classes. They represented ambiguously segmented data. Segments in the 'blank' class could in fact be composed of portions of individual or groups of individual classes. Therefore, they were disregarded from the benchmark tests because of their ambiguity. In future work, the manual labeling could explicitly tag each blank class as a negative example of one of the other 6 classes, designating it as some other class (e.g., Chinese food, window, door, or some multiple groupings of existing classes such as the segment mentioned above which includes both a Kramer and Homeless Man). This finer grained labeling would make the gathering of ground truth a rather complicated task. Even though a default tagging scheme could be set up where each positive example of a character is also negative example of all other classes, the fact that only about a third of the segments were actually characters would make the explicit labeling of the non character classes quite difficult. Compound this complexity with hierarchical class groupings such as 'Jerry's hand' or 'Kramer's head' and the task of hand labeling becomes unmanageable.

The Symmetric Set Cover Algorithm

FourEyes is equipped with a reconfigurable bank of learning algorithms. Of these algorithms, set covering is the method which was used for the analysis performed in this thesis. Given a set of data, the set cover method forms the smallest union of data which includes all of the positive examples and none of the negative examples. Everything else in the data set remains unclassified. Minka implemented a *symmetric* set cover method for this work which is more optimistic than the standard set cover in FourEyes. The *symmetric* set cover (see figure 19 below) finds the set of all of positive examples similar to the instance, then finds all of the negative examples which are not examples of the instance. The assumption in the symmetric set cover is that false positives are popular in these data and their effect should be ignored. False positives refer to regions included in the set which should not be members. Given that the data set is composed of real world imagery and that each model does not well constrain its members by perceptual metrics, it is fair to expect spurious members. Specifically it is likely that a given node in the similarity tree will group regions with a similar histogram together, but there is no guarantee that in culling through those nodes, that the leaves will be pointers to perceptually similar items. The symmetric set cover will include the false positives without penalty. In a browsing venue this should be acceptable.

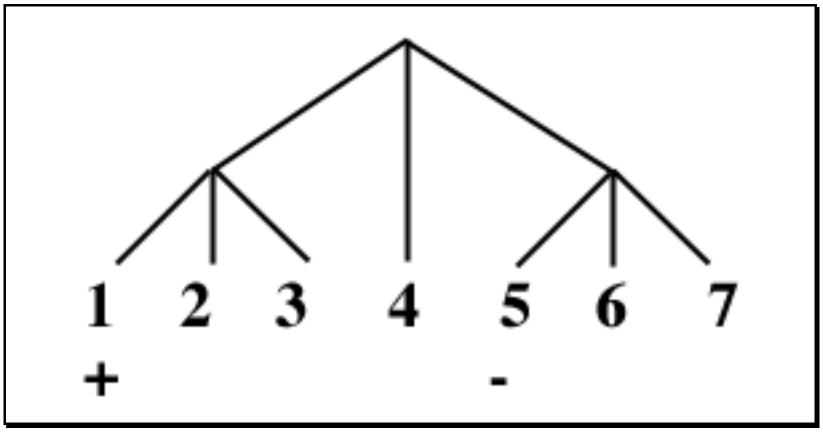

Figure 19 Example of Set Covering. Given that the user labeled segment 1 *positive* and segment 5 *negative*, the following illustrate the behavior of various set cover methods.

Non-Symmetric Optimistic Set Cover
Positive 1,2,3
Don't Know 4,6,7
Negative 5

Non-Symmetric Pessimistic Set Cover
Positive 1
Don't Know 2,3,4
Negative 5,6,7

Symmetric Set Cover
Positive 1,2,3
Negative 5,6,7
Don't Know 4

Benchmarking the Data

Given the ground truth labeling, several benchmark experiments were performed. The effectiveness of the tree clustering can be evaluated in terms of a learning curve. The faster the curve drops, the better the performance. The procedure for generating the learning curve is prescribed by [Minka, 96 (p26)] as follows:

> At each step, an instance which was unclassified by the learner was scored as one error; an instance which was misclassified by the learner was scored as two errors, to make blind guessing disadvantageous. Thus the formula for error count is (False negatives) + 2 * (False Positives). The progression of error counts forms a learning curve by which algorithms can be judged.

> The learner remembered all examples it had received and reconstructed the candidate concept from scratch after every new example. The learner also assumed that the classes were disjoint, i.e, a positive example for one class was a negative example for all other classes. These two facts imply that the learner will always converge to zero error in at most *N* steps, where *N* is the number of instances. The minimum number of examples required is equal to the number of classes, since a learner must see at lease one member of a class in order to speculate what is in it."

Most learning curves have three characteristic phases. In the initial phase the results of trials may vary wildly until some convergence is achieved. In the middle phase, where the slope is steepest, learning occurs most rapidly. During the final phase the system learns less rapidly with more experience. This is referred to as the breakdown phase. During breakdown, the curve often approaches the x-axis asymptotically. An intersection with the x-axis can be projected by drawing a tangent line at the maximum rate of curvature in the breakdown range. In most circumstances the curve actually intersects the x-axis during breakdown. The ratio of the point of this intersection to the total number of segments is the performance ratio. Although various learning algorithms and data sets will affect the shape of the learning curve, most will have these characteristic phases. Characterizing the learning curve in terms of these phases helps evaluate the performance of a set of data with a particular set of similarity metrics.

Learning the Segments

In figure 21, the learning performance of the RGB, Ohta and DCT of the DFT magnitude models were plotted against a baseline classifier that learns nothing. The y-axis of the learning graph plots the number of errors during learning. The extent of the y-axis is bounded by the number of possible segments (5382) times the number of wrong labels it can inherit times the penalty for being wrong (2). For instance, for a given patch labeled ‘Kramer’ by one tree and ‘Elaine’ by another tree, but for which the ground truth label is ‘Jerry’, would be scored with four errors. In figure 21, the graph was cropped and scaled to increase readability so the limit of the y-axis is around 5500. The x-axis is simply the number of training examples (maximally 5382). The fact that the learning curve starts to be significant in the range of the number of segments means that the initial rate of learning is rapid.

Notice that of the individual models, color outperformed texture and Ohta performed best. The normalized Ohta and RGB were also evaluated. They each performed slightly worse then their unnormalized counterparts and were left off the plot to increase readability. The DCT of the DFT Magnitudes appears to be a poor model of this particular data-set. Its initial performance was worse than baseline. This can be explained by the fact that there is a double penalty for false positives. The combined models (RGB and Ohta) were also plotted and performed noticeably better than the individual models. Notice also, that the combined RGB and Ohta curve converges to zero after about 2000 training examples. This means that there is a grouping of 2000 training examples from which FourEyes can successfully extrapolate the labels to the remaining 5382 samples. Since the learning performance graph is generated by picking segments at random the performance will vary on each run depending on which segments are used for training. Nevertheless, these data appear representative of performance and it can be said that the combined RGB and Ohta models learned at a rate of 2.7:1. This measure was calculated by dividing the 5382 segments by 2000 training examples.

The Effect of High-level Information on Learning Performance.

The incorporation of high-level contextual information constrains the classes in which a given segment could be a member. The following details about high-level information are presented before their effect on learning performance is considered.

Perceptual and Semantic issues

People's appearances change. The characters in the sit-com may change their clothes as the fictional day or occasion in the story changes. On the first fictional day in *The Beard* Jerry wears a blue shirt, on the second day he wears a red shirt. In Scene 1, George walks into Jerry's apartment holding his coat. The coat is part of George, so when he puts it down, should it still be considered part of his representation? In Scene 2, when George is wearing his coat, is it him?

Although each character's wardrobe may change radically across scenes, observations made over a library of episodes would bear out that their wardrobes converge to a specific style. For instance, generally, George wears plaid shirts, Jerry wears solid colored shirts and Kramer wears Hawaiian or Golf shirts. In *The Beard* specifically, when the characters go outside, they put on winter coats which obscure much of their clothing. Sit-com viewers, and people more generally, are not often confused by such radical superficial changes because they monitor and recognize other features of people, including height, voice, behavior, gait and of course face. In the episode *The Beard* in fact, George sports a toupee. This constitutes such a radical change in his appearance in the fiction of the episode, it becomes a major plot point. Whereas when George puts on his jacket to walk down the street, it is an acceptable appearance change. In the space of faces (*not* a distinct class within this configuration of FourEyes) the distance between George with a toupee and without a toupee is likely smaller than the distance between George's face and anyone else's face. However, from the computer's standpoint patches of George with a jacket and patches of George without a jacket are entirely dissimilar. This is a problem. What is needed is a way of adjusting the set of features selected based on the high-level information available in the script.

1) Location and Setting Information:

The script may offer assistance in classifying with respect to wardrobe changes. In each scene header, the script indicates the day number, the time of day, and the location of the scene.

For instance:

> "SHOW OPEN STAND-UP #1 INT. COMEDY CLUB - NIGHT"
> "ACT ONE SCENE A INT. JERRY'S APARTMENT - DAY (1)"
> "ACT ONE SCENE B EXT. NEW YORK STREET - DAY (1)"
> "ACT ONE SCENE C INT. POLICE STATION - DAY (1)"
> "ACT ONE SCENE D EXT. NEW YORK STREET - DAY (1)"

Since the regions of interest have membership in a particular setting or location, the expectation of a particular labeled region can be modified according to this knowledge.

2) Scene Membership Information:

Using the reliable knowledge of who is in each scene, the system can exclude those actors who are known NOT to be in the scene from the classification task. Figure 20, the distribution of ground truth segments by shot dramatically confirms that the characters are only present in certain scenes. This information can be automatically extracted on a scene by scene basis from the script.

3) Entrance/Exit Information:

The script also indicates when a particular actor enters and exits the scene. This information can be used to further limit the classifications on a given region. For example, in Scene 1, although Jerry, Elaine, Kramer and George are eventually all present, Kramer and George make individual and separate entrances. Prior to their respective entrances, it is possible to exclude them from the classification task. Although George speaks on the

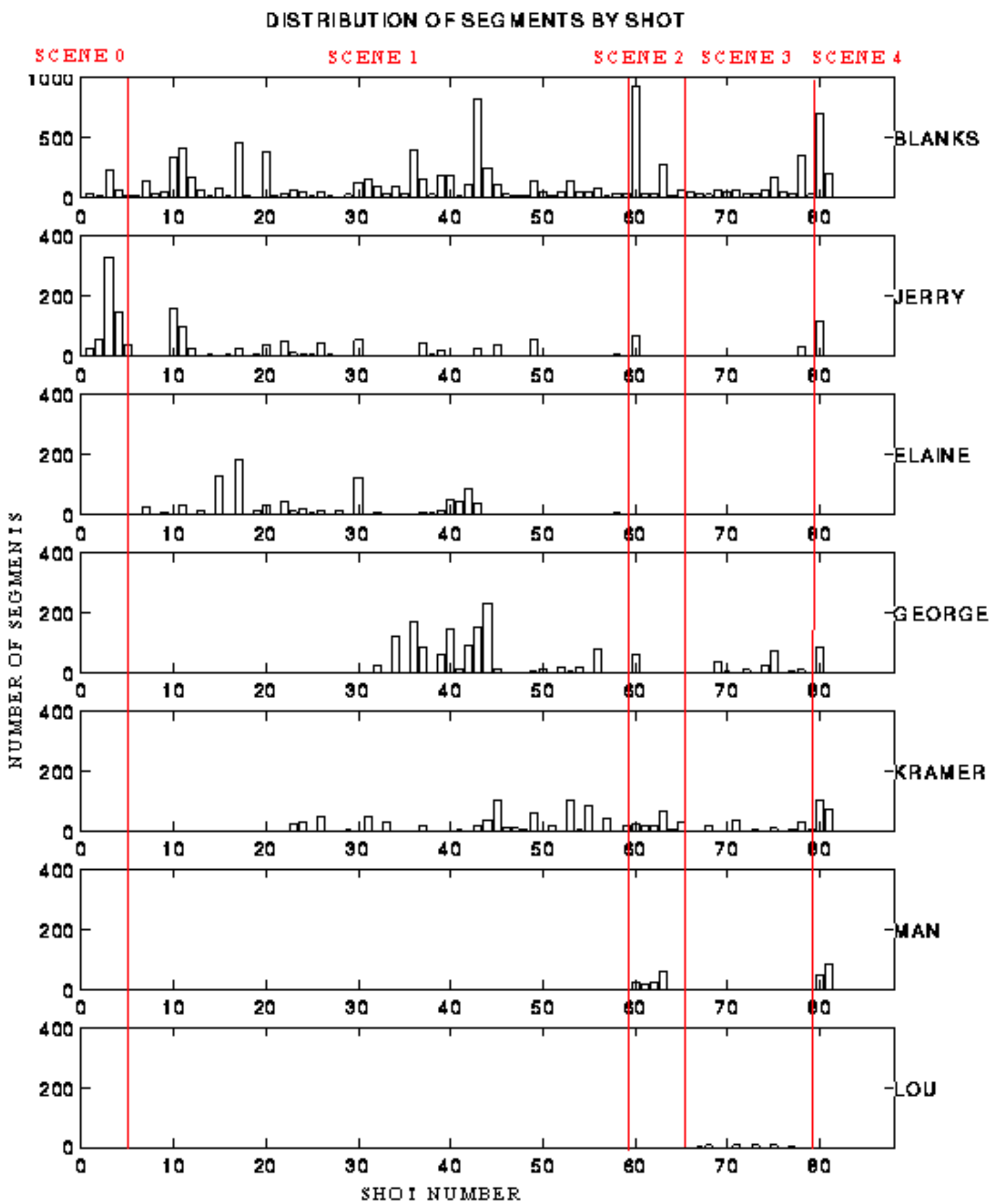

Figure 20 Analysis of the distribution of manually labeled segments indicates in which shots the characters are present. Vertical lines indicate scene membership.

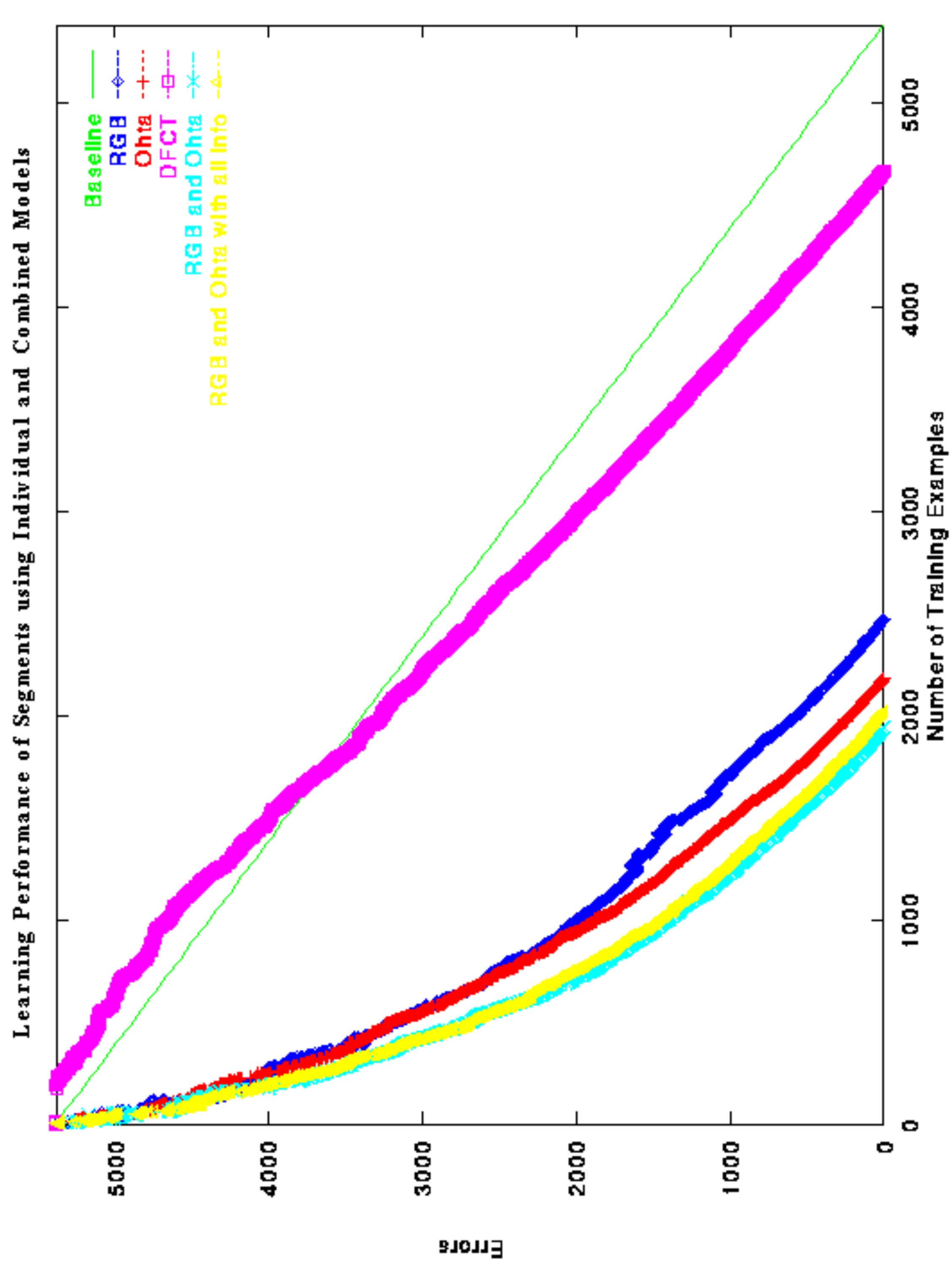

Figure 21 The faster the curve drops the better the performance. The graph illustrates the performance of individual and combined models against baseline. The graph also illustrates the negligible effect of high-level information on performance of the combined RGB and Otha model.

intercom in Shot 21, he is not physically in the apartment until Shot 32. So none of the extracted regions until Shot 32 could possibly be classified as George.

Figure 21 also illustrates that the incorporation of high-level information did not increase learning performance on the combined RGB and Ohta model. The difference between learning performance with and without high-level information is so small it can be considered insignificant on these data for the segment classification task. It was observed by inspection of the similarity trees, that segments tended to be clustered within shots. Since the high-level information available was relevant at the scene classification level, the performance on segment classification could not be expected to be enhanced.

Preliminary Experiments on Learning Shots with Particular Actors

In a browsing venue, users often want to find shots in which a particular character is present. In a preliminary set of experiments, each shot was classified by learning whether its segments were regions of each actor. If a shot contained one or more segments with a character's name then the shot was classified as containing that character. Since a shot can contain segments from multiple characters, these were not mutually exclusive tests. In other words, an example of a shot with Kramer could not be used as a negative example of a shot for any other characters. The rate of learning these shot classifications was slower than for learning the classification of individual segments. This is because the rather optimistic symmetric set cover algorithm was penalized for having to learn the mutual exclusivity of segments in each shot.

On the other hand, these preliminary experiments suggested that learning performance was marginally enhanced when high-level information was incorporated. Further they indicated that the more high-level information present, the better the learning performance. Incorporating all three types of available information described above (Location and Setting, Scene Membership, Entrance/Exit) improved learning performance the most.

When the browser responds to a user's query, the shots often need to be ranked in relevance order for retrieval. Although this is a subjective criteria, in the classification of shots by character, the relative presence of the character may be used and can be evaluated by several means. A simple measure would determine the popularity of a given actor in a shot by normalizing the number of segments recognized as the character to the total number of segments in the shot, or to the total number of other segments of actors known to be in the shot. Another simple measure would determine whether the character had a caption during the shot and rank the retrieved shots according to the premise that characters that have captions are significant to the shot. A more complex measure might gather statistics on the temporal and spatial co-occurrence of segments. A hypothesis is that an actor is more significant to a shot if the segments associated with him clump together in time or space, than if the same number of segments were evenly distributed throughout the shot. For instance, if a Kramer segment shows up in an isolated frame his 1/30th sec appearance may not be significant to a viewer. Further research is necessary to validate this hypothesis and to determine what role context might play in evaluating the significance of shots retrieved.

Discussion of the Classification Results

In a random database of images, the ability to label a single image region and have the system learn relationships with other images in the 2.7:1 ratio achieved in this data-set would be a great starting point. This grouping across images means the similarity metrics are robust enough to find salient relationships among several images. However, in a video database, across image groupings are less significant unless the related images occur in shots, scenes, or episodes other than in the example shot. Since a shot is a coherent sequence composed of slowly changing images, it is important for the similarity metrics to be able to form groupings that span shots. For much of the data, across shot groupings of segments was poor. This means that 1) the trees splintered along shot boundaries and the models did not characterize the data well, 2) that ground truth was gathered imprecisely or inconsistently by the subjects. There are three ways to improve the performance: 1) identify a better set of similarity measures for the data set 2) integrate more high-level attributes which coerce the low-level attributes to be grouped together 3) collect ground truth more precisely. All are equally valid directions for further development.

From a human perceptual standpoint characters do not vary their appearance radically between scenes, let alone between shots. George is still George even if he changes his pose or clothes. This is because humans have constructed robust cognitive models of

Figure 22 Screen dump of *Advanced Video Browser* demonstrating the ability to find shots in which Kramer did *not* have a caption, but in which he appears. This browser is more powerful than the *Basic Video Browser* because regions have been learned by FourEyes.

Images Appear Courtesy of Castle Rock Entertainment

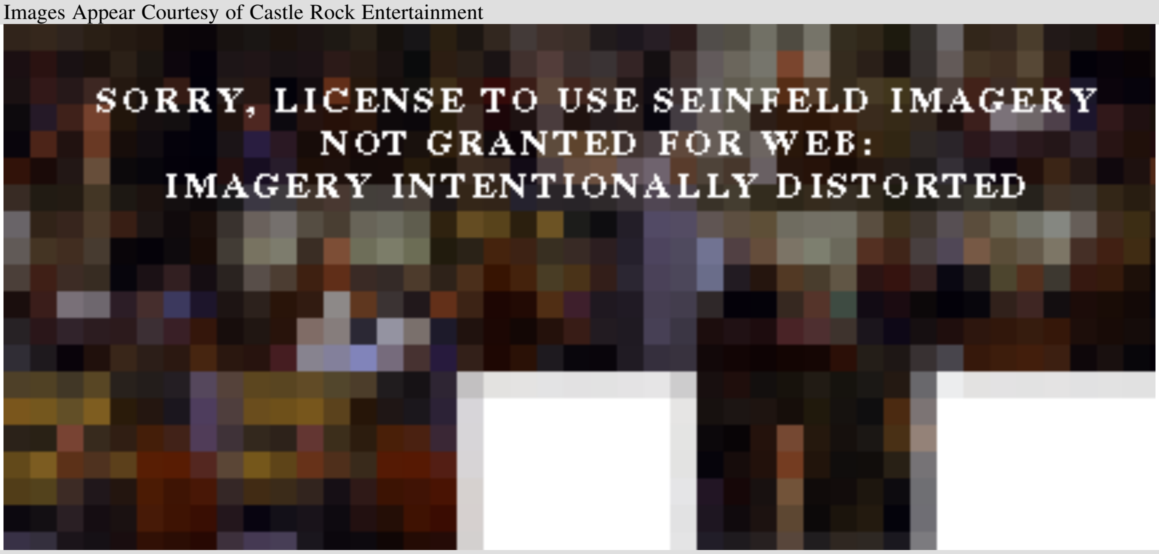

Figure 23 Key frames returned on a query ‘Find Shots with Kramer’. Using the *Advanced Video Browser* shots are retrieved in which Kramer appears but does not necessarily have a caption. The last image, set off by itself, is a shot in which Kramer appears (if you look very carefully you can see Kramer’s back in the bottom right side of the frame), but could not be retrieved by the system. Since during the preprocessing stage, no luminance or motion segments isolated Kramer in this shot he is effectively invisible to the system.

what it means to appear like a given person. The computer's representation is much more brittle. A more explicit model might also improve performance, but would sacrifice generality.

A Smarter Fast Forward: Advanced Video Browser

Since individual regions can be learned through human-computer interaction, a video browser can be constructed that indexes shots by actors even when the actors do not have a caption. Figure 22 is a screen dump of the Advanced Video Browser which illustrates what a user would see when teaching the computer by example. Figure 23 shows key frames the computer can retrieve on shots, given that it learned the classification of all character segments. The key frames returned represent an improvement over the *Basic Video Browser* that could only retrieve shots in which characters had a caption.

Chapter 5

SUMMARY AND FUTURE WORK

Summary

Presented was a general purpose video browser which learns about its content by integrating high-level narrative elements with low-level signal features. The system demonstrated the ability to browse video based on a set of content specific attributes relating to a television situation comedy. In preliminary experiments its performance was enhanced, albeit slightly, by the integrated knowledge provided by the high-level script and closed caption information. A strength of the work is its ability to configure itself in an unsupervised manner by preselecting regions of significant change. A further contribution of the work is the human-computer interaction and feedback which makes the system responsive to perceptual similarities even when they do not correspond to statistical similarities.

It is possible that video without closed caption, script or shot boundary information, could be processed in the manner described. Regions of coherent motion and intelligently selected and segmented frames could be used to construct similarity trees and these trees traversed using one of the traditional set cover methods.

At the outset of this research effort it was unclear, what if any influence high-level script and closed caption information could have on learning performance. The performance of the Basic Video Browser demonstrated that a browser could be constructed which relied almost exclusively on the availability of high-level information. Although preliminary experiments suggest that the incorporation of low-level information could increase accuracy and functionality of the Advanced Video Browser, more experiments are needed to state conclusively how much performance increases by integrating the high-level and low-level information. Another contribution of this work therefore, is the definition of a role which high-level information could play in video annotation.

Future Work

In future work, research into more tuned human sensitive filters could be integrated into the present system to make the preprocessing stage more robust [Pentla, 96]. For instance, of the ~15000 preselected regions approximately 250 were faces. These regions could seed a face tracking or expression detection system which might be used to corral regions in some spatio-temporally neighborhood together.

In a generic video database venues, it may not be fair to expect that closed caption or script information will always be available. In future work, the objective would be to affect training over several episodes in order to bootstrap the system and then to disregard the ancillary textual representations while still achieving successful content-based retrieval on the learned patterns.

Additional Approaches

The Motion Energy Image

In situation comedies it is common for the action to take place on several staged sets during the course of an episode. Although these physical locations may vary from episode to episode, some sets remain constants and become hallmarks of the show. In the Seinfeld catalog for instance, common sets include Jerry's Apartment, The Restaurant, The Comedy Club, A New York Street. How the actors move around these sets is constrained by the physical layout of the set and by the action prescribed by the script. These factors govern the position of the cameras.

As characters move about the space of a given set, they leave behind a motion 'footprint'. These footprints may be used to classify the sequences. A course block-wise motion energy image was generated for each sequence (shot) and collection of shots (scenes) in the episode. The motion energy image (MEI) is effectively a frame relative mapping of the location of the motion in a given sequence to a single image. In the MEI intensity at each location is proportional to the integral of motion at that location. This may be considered as a compression of the motion in an entire sequence into a single image representation. Since the MEIs are a frame relative recording of the motion within a sequence, they encode camera motion as well as individual object motion. For instance, when Jerry is on stage giving his monologue he uses the stage in a consistent way. His motions tend to be restricted to hand and head gestures. The motion energy for these shots are self similar in that there is a blob of energy in the center and the surround tends to be empty. This means that he stayed camera center through much of his comedic monologue. Notice how distinct these are as compared to MEI's of Kramer and Homeless Man on the Street.

Figure 24 Motion Energy Images for two different shots in two different Scenes.

Shots and scenes could be classified by these MEI's. In combination with constraints of who is in each shot and scene the MEIs could be used to determine whether a given shot is a close up, medium or wide shot. Although these data were collected, they were not incorporated into the learning. Discrimination of shots based on these collapsed motion 'snapshots' might be most useful in a large browsing database where the user might want to find scenes that have a characteristic pattern of motion.

The Multiple Labels Approach

An obvious next step would be to repeat the manually labeled ground truth test allowing each blob to be given more than one label. For instance, these multiple labels could be hierarchical,--Jerry's hand, Elaine's face, George' jacket, Kramer and Homeless man's

Chinese food. Could finer grained labels improve the collection of relevant data during a query? Can the similarity measures corral all of the 'hands' together or all of the faces together? The multiple labels approach would also permit many of the regions previously labeled blank to get incorporated into the analysis. Regions which have membership in multiple characters could also be included. It is instructive to think of a person as a collection of dissimilar stuff. Hands, faces, sweaters, jeans, may individually be identifiable as relatively homogeneous patterns, but collectively the distribution of these dissimilar patterns is a meta pattern which may be recognizable as an individual person. Consider the representation of the similarity tree where each branch is a set of similar stuff. One could imagine constructing a *forest* of similarity trees where a meta pattern unique to an individual person is formed. The ultimate goal of that research thread would be to compare a Jerry *forest* to an Elaine *forest*. This method was originally considered in the present work however inspection of the data set suggested that it was too small to support this approach. In future work, if the system could support two or three episodes worth of data, this idea might be pursued more thoroughly. It might also increase the performance and utility of the present system if users could explicitly outline regions of the image which depicted the actor or object of interest. Manual specification of a region of interest within, say, an ambiguously labeled region, might help salvage portions of blank regions currently excluded from analysis.

Further Questions

In an attempt to understand situation more thoroughly, perhaps the system could deduce concurrence or proximity information. How often and how close do Jerry and Elaine appear together? [Pinha, 95] explored the use of scripted video to identify and understand actions in video. The present work could be extended in the same spirit in an effort to begin to understand stage directions like "KRAMER BENDS DOWN AND GIVES HIM (Homeless Man) THE CONTAINER OF CHINESE FOOD. Act I, Scene B. *The Beard*. If each of the objects 'Kramer', 'container' could be identified automatically. Perhaps 'bends' as a verb could be deduced from Kramer's behavior, 'gives' could be inferred by the translation of the 'container' toward the Homeless Man ("him").

Together the captions and script provide support which can be used to train the learning algorithm in an automatic, almost unsupervised pipeline. The idea of using the correlation of the script and closed captions is that it can be used to train the system on the pattern of video which corresponds to a specific individual. If the broadcast reliably included the names and locations of characters on screen, the shot boundaries or even the script, then much of the preprocessing stages performed here would not have been necessary. Broadcast of the Edit Decision List might be able to provide some of this information. Then the goal of constructing an intelligent browser that learns is relegated to deciding how to use the information and how to determine the correspondence between the low-level information and this high-level input. As before, once the characters representation is learned, this high-level information can be discarded. In the analysis of video which does not have this high-level information, the burden is on the user to provide context.

A valuable direction for future work in this area would include audio analysis. In the domain of TV sit-coms specifically, character voice recognition and analysis could play a major role. If the present system were combined with speaker identification software, then a more robust representation of the character could be constructed. Cross-modal research has already proven useful in similar domains and it may ultimately be the most powerful means to detect and distinguish people in video databases [Casey, 96]. The audio could be correlated with the caption and script information to isolate a period of time during which the character is likely present. The video features could be used to determine a spatio-temporal window during which the character is likely on screen. In combination, these features may help resolve instances when people speak off camera or when captions are displayed synchronously to their vocalization. It is even conceivable that in the future, the technology of learning the representation of a character in a video based on the script, audio and video, could assist broadcasters in semi-automatic captioning of their content.

It is not difficult to envision future computer systems watching television and automatically constructing associations between visual, textual and auditory patterns. In this scenario, patterns of video information could be automatically associated with textual concepts, and computers would be empowered to learn about the world as represented by Television. Since vast archives of data already exist in the scripted and closed caption format and new episodes are generated weekly, closed captioned television may offer an untapped resource for automatic computer learning.

BIBLIOGRAPHY

[Astle, 94] Astle, B., "Video Database Indexing and Method of Presenting Video Database Index to a User", US Patent Number 5,485,611 Jan 16, 1996. Assigned to Intel Corporation, Santa Clara CA, Dec. 30, 1994.

[Bouch, 96] Boucher, Roland “Skip”, Broadcast Group at AVID Technology, personal communication.

[Bove, 83] Bove, V. M, Jr., "Personalcasting: Interactive Local Augmentation of Television Programming," MIT M.S. Thesis, 1983.

[Brook, 95] Brooks, T. and Marsh, E., "The Complete Directory to Prime Time Network and Cable TV Shows 1946-Present" Ballantine Books, a Division of Random House, NY. 1995. p 1215.

[Casey, 96] Casey, M. and Wachman, J., "Unsupervised Cross-Modal Characterization of Expressive Gesture in Professional Monologue Discourse" to appear at the *Workshop on the Integration of Gesture in Language and Speech WILGS* Satellite workshop of the Fourth International Conference on Spoken Language Processing. ICSLP '96. October 7-8, 1996.

[Chakr, 94] Chakravarthy, A. S., "Toward Semantic Retrieval of Pictures and Video," *AAAI-94 Workshop on Indexing and Reuse in Multimedia Systems*. Seattle, Washington, August 1994.

[Chang, 95] Chang, S., Smith, J. and Wang, H., "Automatic Feature Extraction and Indexing for Content-Based Visual Query," Columbia University/CTR Technical Report # 408-95-14. 1995.

[Davis, 95] Davis, M. E., "Media Streams: Representing Video for Retrieval and Repurposing," MIT Ph.D. Thesis, 1995. p133.

[Hadam, 23] Hadamard, J. "Lectures on the Cauchy Problem in Linear Partial Differential Equations," New Haven, CT: Yale University Press. 1923.

[Haupt, 95] Hauptmann, A.G., and Smith, M.A., "Text, Speech, and Vision for Video Segmentation: The Informedia Project" *AAA-I Fall Symposium Series Computational Models for Integrating Language and Vision*, MIT, Proceedings. Nov. 10-12, 1995. p90-95.

[Intil, 94] Intille, S.S and Bobick, A.F., "Visual Tracking Using Closed-Worlds," MIT Media Laboratory Technical Report # 294. 1994.

[Jain, 88] Jain, A.K., and Dubes, R.C., *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ. 1988.

[Karah, 95] Karahalios, K., "Salient Movies," MIT M.S. Thesis. 1995.

[Lucas, 81] Lucas, B. and Kanade, T. "An Iterative Image Registration Technique with an Application to Stereo Vision," *Image Understanding Workshop* April l981. p121-130.

[Marro, 87] Marroquin, J, Mitter, S. and Poggio, T., "Probabilistic Solution of Ill-Posed Problems in Computational Vision," *American Statistical Association Journal of the American Statistical Association*, Vol 82. No 397, Theory and Methods. March 1987. p76-89.

[Meng, 96] Meng J. and Chang, S.F. "Tools for Compressed-Domain Video Indexing and Editing," *IS&T/SPIE Symposium on Electronic Imaging: Science and Technology -- Storage & Retrieval for Image and Video Databases* IV, VOL. 2670, San Jose, CA. February, 1996.

[Minka, 95] Minka , T.P. and Picard, R.W., "Interactive Learning using a 'Society of Models'", *Special Issue of Pattern Recognition on Image Databases: Classification and Retrieval*, 1995. (MIT Media Lab Perceptual Computing Section Technical Report # 349)

[Minka, 96] Minka, T.P. "An Image Database Browser that Learns From User Interaction" MIT Master of Engineering thesis Department of Electrical Engineering Feb. 1996 (MIT Media Lab Perceptual Computing Section Technical Report #365)

[Nibla, 93] Niblack, et al. "The QBIC Project: Querying images by Content Using Color, Texture and Shape", IBM Research Report, RJ 9203 (81511) Feb 1, 1993. http://wwwqbic.almaden.ibm.com/~qbic/qbic.html.

[Ohta, 80] Ohta, Y. I, Kanade., T. and Sakai, T., "Color Information for Region Segmentation," *Computer Graphics and Image Processing*, 13:222-241, 1980.

[Pentl, 96] Pentland, A.P., “Smart Rooms,” *Scientific American*, April 1996. p68-76.

[Picar, 94] Picard, R.W. and Minka, T.P., "Vision Texture for Annotation,” ACM/Springer-Verlag Journal of Multimedia Systems, 1995, Vol. 3, pp 3-15. (MIT Media Laboratory Perceptual Computing Section Technical Report #302).

[Picar, 95a] Picard, R.W., "Digital Libraries: Meeting Place for High-Level and Low-Level Vision," *Asian Conference on Computer Vision,* Singapore, Dec. 1995. vol I. p 1-5. (MIT Media Lab Perceptual Computing Section Technical Report #354).

[Picar, 96a] Picard. R.W., "A Society of Models for Video and Image Libraries" IBM Systems Journal 1996. (MIT Media Lab Perceptual Computing Technical Report #360).

[Pinha, 95] Pinhanez, C. and Bobick, A., "Scripts in Machine Understanding of Image Sequences" p 96-101. *Computational Models for Integration of Language and Vision*, AAAI-95 Workshop Fall Symposium, MIT Nov 10-12, 1995.

[Salt, 83] Salt, B., "Film Style and Technology : History and Analysis," London, Starwood, 1983.

[Schan, 95] Schank, R. and Abelson, R., "Knowledge and Memory: The Real Story Advance in Social Cognition," Vol. VIII Edited by Wyer. Lawrence Erlbaum Associates, Publishers, Hillsdale, N.J. 1995.

[Shake, 36] Shakespeare, William. *Hamlet* "The Complete Works of Shakespeare," The Cambridge Edition Text. Edited by W.A.Wright, Garden City Books, Garden City New York, 1936.

[Sriha, 94] Srihari, R.K., et al., "Use of Collateral Text in Image Interpretation" CEDAR *Proceedings of The Image Understanding Workshop*, Vol II ARPA Software and Intelligent Systems Technology Office Nov 13-16, Monterey, CA. 1994. p897-907.

[Sriha, 95] Srihari, R.K., "Linguistic Context in Vision," *AAA-I Fall Symposium Series Computational Models for Integrating Language and Vision*. MIT, Proceedings Nov 10-12, 1995. p78-88.

[Swain, 91] Swain, M. and Ballard, D.,"Color Indexing" *International Journal of Computer Vision* 7:1, 1991 p 11-32.

[Virage] Virage, http://www.virage.com/

[Wang, 93] Wang, J.Y.A. and Adelson, E.A., "Layered Representation for Motion Analysis" *Proceedings of the Computer Vision and Pattern Recognition Conference*, June 1993. (MIT Media Lab Perceptual Computing Section Technical Report #221).

[Yeung, 95] Yeung, M. M. and Liu, B., "Efficient Matching and Clustering of Video Shots," Princeton University *Proceedings IEEE International Conference on Image Processing Vol 1*. Washington, D.C., October 23-26 1995. p338-341.

[Zhang, 95] Zhang H., Low C.Y., and Smoliar, S., "Video Parsing and Browsing Using Compressed Data" *Journal of Multimedia Tools and Applications*, Vol. 1, No. 1, Kluwer Academic Publishers. March 1995. p 89-111.

[Zhang, 93] Zhang, H., Kankanhalli, A. Smoliar, S., "Automatic partitioning of full-motion video," *Multimedia Systems* 1:10-28 (1993) Springer-Verlag.

Appendix A. **Episode Summary Seinfeld *The Beard***

Excerpt from SEINFELD *The Beard*:

As broadcast Feb 9, 1995
Written by Carol Leifer, Directed by Andy Ackerman
#04-0615

Rebroadcast in syndication on (excerpt from Seinfeld home page:
http://www.spe.sony.com/Pictures/tv/seinfeld/seinfeld.html)

July 20th - *The Beard*
9:00 p.m. (Eastern & Pacific) on NBC

"While posing as a *beard* for a gay acquaintance needing an escort, Elaine becomes infatuated with the handsome man and sets out to convert him to heterosexuality. Meanwhile, George--now sporting a toupee--lands a date with a beautiful friend of Kramer only to find that she's covering up a cosmetic problem too. Also, Jerry frets when a female police officer he likes wants to give him a lie-detector test that could reveal his secret TV viewing habits.

Appendix B. **Manual Annotations of *The Beard***

The Beard
SHOW OPEN (Jerry)

Shot	Characters	Time	
1,1	Jerry	0:00	START
2,2	Jerry	0:06	
3,3	Jerry	0:10	
4,4	Jerry	0:28	
5,5	Jerry	0:35	
6,6	Jerry	0:38	
7,7	Jerry and crowd	0:40	FADE OUT

ACT ONE
SCENE A: (Jerry, Elain, Kramer, George) Jerry's Apartment

Shot	Characters	Time	
8,1	Elaine	2:46	FADE IN
9,2	Elaine, Jerry	2:49	
10,3	Elaine	2:51	
11,4	Jerry	2:52	
12,5	Jerry, Elaine	2:56	Jerry exits frame
13,6	Jerry	3:02	
14,7	Elaine	3:04	
15,8	Jerry	3:06	
16,9	Elaine	3:08	
17,10	Jerry	3:14	
18,11	Elaine	3:16	
19,12	Elaine...Jerry	3:27	pan over to include Jerry
20,13	Elaine, Jerry	3:28	
21,14	Elaine, Jerry	3:30	
22,15	Elaine, Jerry	3:32	
23,16	Jerry	3:35	
24,17	Elaine, Jerry	3:37	
25,18	Elaine, Jerry..Kramer	3:39	Kramer Enters
26,19	Elaine, Jerry, Kramer	3:43	
27,20	Elaine, Jerry, Kramer	3:46	
28,21	Elaine, Jerry	3:49	
29,22	Kramer	3:52	
30,23	Jerry, Elaine	3:54	
31,24	Kramer	3:57	
32,25	Elaine, George	4:03	George Enters (Jerry)
33,26	Kramer	4:08	
34,27	George	4:08	
35,28	Elaine, Jerry	4:11	
36,29	George	4:12	
37,30	George	4:14	(Kramer, Jerry, Elaine)
38,31	Elaine, Jerry	4:16	
39,32	George, Elaine, Jerry	4:17	
40,33	George, Elaine	4:27	
41,34	George, Elaine	4:33	(Jerry, Kramer)
42,35	George, Elaine	4:40	(Jerry)
43,36	George, Elaine, Jerry..	4:44	pan over to include Kramer
44,37	George	4:52	(Kramer)
45,38	Kramer	5:00	(George) (Jerry walks behind)
46,39	Kramer, George, Jerry	5:08	
47,40	Kramer	5:10	(George)
48,41	Kramer, George, Jerry	5:13	

49,42	Kramer	5:14	(Jerry walks back)
50,43	George	5:17	
51,44	Kramer	5:17	
52,45	George	5:19	
53,46	Kramer	5:21	(George)
54,47	George	5:27	(Kramer's hand)
55,48	Kramer	5:27	
56,49	George	5:29	
57,50	Kramer	5:30	
58,51	Jerry, Elaine	5:32	
59,52	George, Kramer	5:35	

SCENE B: (Jerry, George, Kramer, Homeless Man)

Shot	Characters	Time	
60,1	All	5:39	(Additional random)
61,2	Kramer, H.P.	5:47	
62,3	Kramer, H.P.	5:49	
63,4	Kramer, H.P.	5:50	(Additional random) pans outright

SCENE C: (Jerry, George, Kramer, Lou, (Cathy))

Shot	Characters	Time	
64,1	Police Station	5:55	
65,2	Jerry, George, Kramer	5:57	Random (Lou)
66,3	Jerry, Lou, Kramer	5:59	
67,4	George	6:02	(Random)
68,5	George, Kramer, Lou, Jerry	6:03	(Random)
69,6	Kramer, Lou	6:05	(Random)
70,7	George	6:08	(Random)
71,8	Jerry, Lou, Kramer	6:09	
72,9	George, Cathy	6:10	(Random)
73,10	Jerry, Lou, Kramer,Geor	6:12	
74,11	Cathy	6:24	(Random)
75,12	Jerry, Lou, Kramer,Geor	6:26	

CUT TO: Street Random

SCENE: D: (Kramer, H.P. George, Jerry)

Shot	Characters	Time	
76,1	Kramer,George,Jerry,HP	6:28	(Random)
77,2	Kramer, H.P.	6:39	
78,3	Kramer, H.P.	6:42	
79,4	Kramer, H.P.	6:44	
80,5	Kramer, H.P.	6:46	
81,6	Kramer, H.P.	6:47	
82,7	Kramer, H.P.	6:49	
83,8	Kramer, H.P.	6:52	
84,9	Kramer, H.P.	6:55	(Random)

EXTENT OF LEARNING ANALYSIS PERFORMED IN THESIS

CUT TO: City, Night

SCENE: E: (Mr and Mrs Stevenson, Elaine, Robert)

Shot	Characters	Time	
85,1	City, Night	7:01	
86,2	ALL	7:04	
87,3	Elaine, Robert	7:08	
88,4	Elaine, Robert	7:10	
89,5	ALL	7:13	(Robert Exits)
90,6	Elaine, Mrs Stevenson	7:19	

91,7	Elaine, Mr & Mrs.	7:22	
92,8	Elaine, Mrs.	7:24	
93,9	Elaine, Mr & Mrs.	7:26	
94,10	Elaine, Mrs.	7:27	
95,11	Elaine, Mr & Mrs.	7:30	
96,12	Elaine	7:31	
97,13	Elaine, Mr & Mrs.	7:34	
98,14	Elaine	7:37	
99,15	Elaine, Mr & Mrs.	7:43	
100,16	Elaine	7:49	
101,17	Elaine, Mr & Mrs.	7:52	
102,18	Elaine	7:57	
103,19	ALL	8:03	(Robert Enters)
104,20	ALL	8:06	(Robert Enters..)
105,21	ALL	8:10	

CUT TO: Coffee Shop
SCENE F: (G in Script) (Jerry, Elaine)

Shot	Characters	Time	
106,1	Coffee Shop	8:15	(Random)
107,2	Jerry, Elaine	8:18	(Random)
108,3	Elaine	8:19	(Random) (Jerry)
109,4	Jerry	8:26	(Random) (Elaine)
110,5	Elaine	8:33	(Random) (Elaine)
111,6	Jerry	8:36	(Random) (Elaine)
112,7	Elaine	8:38	(Random) (Elaine)
113,8	Jerry	8:39	(Random) (Elaine)
114,9	Elaine	8:40	(Jerry)
115,10	Jerry	8:44	(Random) (Elaine)
116,11	Elaine	8:48	(Jerry)
117,12	Jerry	8:49	(Random) (Elaine)
118,13	Elaine, Jerry	8:51	
119,14	Jerry	8:55	(Random) (Elaine)
120,15	Elaine	8:55	(Jerry)
121,16	Jerry	8:58	(Random) (Elaine)
122,17	Elaine	8:59	(Jerry)
123,18	Jerry	9:01	(Random) (Elaine)
124,19	Elaine	9:06	(Jerry)
125,20	Jerry	9:09	(Random) (Elaine)
126,21	Elaine	9:14	(Jerry)
127,22	Jerry	9:16	(Random) (Elaine)
128,23	Elaine	9:24	(Jerry)
129,24	Jerry	9:27	(Random) (Elaine)
130,25	Elaine	9:29	(Jerry)
131,26	Jerry	9:32	(Random) (Elaine)
132,27	Elaine	9:39	(Jerry)
133,28	Jerry	9:44	(Random) (Elaine)
134,29	Elaine	9:47	(Jerry)
135,30	Jerry	9:53	(Random) (Elaine)
136,31	Elaine	10:00	(Jerry) (Random)
137,32	Jerry	10:02	(Random) (Elaine)
138,33	Elaine	10:05	(Jerry)

CUT TO: Jerry's Apartment
SCENE G: (Jerry, Kramer, George)(SCENE H: in Script)

Shot	Characters	Time
139,1	Jerry's Apartment	10:08
140,2	Jerry	10:10

141,3	Jerry, Kramer	10:31	(Kramer Enters) pan over to Kramer
142,4	Jerry	10:35	
143,5	Kramer	10:37	
144,6	Jerry	10:39	
145,7	Jerry, Kramer	10:42	
146,8	Jerry	10:48	(Kramer)
147,9	Jerry, Kramer	10:49	
148,10	Jerry	10:51	
149,11	Jerry, Kramer	10:53	
150,12	George	10:57	(Jerry, Kramer) George Enters
151,13	Jerry, Kramer	10:58	
152,14	George	10:59	(Jerry, Kramer) George Enters...
153,15	Kramer	11:06	(Jerry)
154,16	George	11:07	
155,17	Kramer	11:10	(Jerry, George)
156,18	George	11:11	
157,19	Kramer	11:14	(Jerry, George)
158,20	George	11:17	(Kramer walks behind)
159,21	Jerry, George	11:19	
160,22	Jerry, George	11:21	
161,23	Jerry	11:22	
162,24	George	11:23	(Jerry) opens Fridge
163,25	Jerry	11:25	
164,26	George	11:26	
165,27	Jerry	11:29	
166,28	George	11:32	
167,29	Jerry, Kramer	11:35	
168,30	George	11:38	
169,31	Jerry	11:41	(Kramer)
170,32	George	11:44	
171,33	Jerry	11:46	(Kramer)
172,34	Jerry, George, Kramer	11:48	
173,35	Jerry	11:52	(Kramer)
174,36	George	11:56	
175,37	Jerry	11:59	(Kramer)
176,38	George	12:01	
177,39	Jerry	12:05	(Kramer)
178,40	Jerry, Kramer	12:07	
179,41	George	12:11	
180,42	Jerry, Kramer	12:13	

CUT TO:
SCENE H: (Jerry, Kramer, Homeless Man) (SCENE J: in Script)

Shot	Characters	Time
181,1	H.P., Jerry, Kramer	12:19
182,2	H.P., Jerry, Kramer	12:22
183,3	Jerry, Kramer	12:26

CUT TO:
SCENE I: (Jerry, Kramer, Cathy, Lou) (SCENE K: in Script)

Shot	Characters	Time	
184,1	Police Station	12:30	
185,2	Cathy, Lou, Kramer, Jerry	12:31	(Random)
186,3	Cathy, Jerry	12:33	(Random, Kramer)
187,4	Kramer, Jerry, Cathy	12:40	(Random)
188,5	Jerry, Cathy	12:43	(Random, Kramer)
189,6	Jerry, Cathy, Kramer	12:47	(Random)

CUT TO:
SCENE I: (Kramer, Officer #1 (V.O) Lineups) (SCENE L: in Script)

Shot	Characters	Time	
190,1	Kramer, Lineups	12:52	(Random)
191,2	Kramer, Lineups	12:58	
192,3	Kramer	13:01	(Lineups)
193,4	Kramer, Lineups	13:06	

CUT TO:
SCENE J: (George, Denise) (SCENE M: in Script)

Shot	Characters	Time	
194,1	Coffee Shop	13:25	
195,2	Denise	13:27	(Random) (George) Denise Enters
196,3	George	13:31	(Random)
197,4	Denise	13:32	(Random)
198,5	George...Denise	13:34	(Random) They meet and shake hands
199,6	George	13:43	(Denise) (Random)

CUT TO:
SCENE K: (Jerry, Cathy, Lou) (SCENE N: in Script)

Shot	Characters	Time	
200,1	Police Station	13:53	
201,2	Jerry, Cathy	13:55	(Randoms) (Polygraph...)
202,3	Jerry, Cathy	13:56	
203,4	Jerry, Cathy	13:59	(Random)
204,5	Jerry, Cathy	14:05	(Random)
205,6	Jerry, Cathy	14:08	(Random)
206,7	Jerry, Cathy	14:10	(Random)
207,8	Jerry, Cathy	14:14	(Random)
208,9	Jerry, Cathy	14:17	(Random)
209,10	Jerry, Cathy	14:20	(Random)
210,11	Jerry, Cathy, Lou	14:24	(Random)
211,12	Jerry, Cathy	14:27	(Random)
212,13	Jerry, Cathy	14:31	(Random)
213,14	Jerry, Cathy	14:33	FADE OUT

END OF ACT ONE

ACT TWO
SCENE L: (Jerry, Elaine, George,Kramer) (SCENE P: in Script)

Shot	Characters	Time	
213,1	Jerry Apt.	17:32	
214,2	Jerry, Elaine	17:35	
215,3	Jerry	17:40	(Elaine)
216,4	Jerry, Elaine	17:43	
217,5	Jerry, Elaine	17:50	(Elaine)
218,6	Jerry, Elaine	17:53	
219,7	Jerry, Elaine	17:58	
220,8	Jerry, Elaine	18:03	
221,9	Jerry, Elaine	18:05	
222,10	Elaine	18:09	(Jerry)
223,11	Jerry	18:15	(Elaine)
224,12	Elaine	18:19	(Jerry)
225,13	Jerry, Elaine	18:20	(Jerry exits frame, Elaine opens fridge)
226,14	Jerry	18:27	(Elaine enters frame)
227,15	Jerry, Elaine	18:37	(Elaine grimace)
228,16	Jerry, Elaine	18:40	

229,17	Jerry, Elaine	18:42	
230,18	Jerry, Elaine	18:46	(Jerry shakes)
231,19	Jerry, Elaine	18:54	
232,20	Elaine...Jerry	18:58	
233,21	Jerry	19:06	
234,22	Elaine	19:07	(Jerry)
235,23	George	19:09	Enters thru door
236,24	George,Jerry, Elaine	19:12	
237,25	George,Jerry	19:15	
238,26	George	19:17	(Jerry)
239,27	Jerry	19:18	(George)
240,28	George	19:19	(Jerry)
241,29	Jerry	19:26	(George)
242,30	Elaine	19:27	
243,31	George	19:29	(Elaine, Jerry)
244,32	Jerry, Elaine	19:33	(George)
245,33	George	19:35	
246,34	Jerry, Elaine	19:38	(George)
247,35	Jerry, Elaine,George	19:41	
248,36	Jerry, Elaine,George	19:50	
249,37	Jerry, Elaine,George	19:54	
250,38	Jerry, George	20:56	
251,39	Elaine	20:05	(George) Elaine eating "Reeses Peanut Butter Puffs"
252,40	Kramer...George	20:08	Door entrance
253,41	George	20:12	(Kramer)
254,42	Kramer	20:14	(George) Kramer surprised
255,43	George	20:17	
256,44	Kramer, Elaine,Geor	20:18	
257,45	Jerry, George	20:26	(Elaine)
258,46	George, Elaine	20:29	
259,47	George, Elaine	20:32	Elaine and George Wrestle for Toupee (Kramer, Jerry)
260,48	George, Elaine	20:47	Elaine tosses it out the window

CUT TO: street day
SCENE M: (Homeless Person) (SCENE R: in Script)

Shot	Characters	Time
261,1	H.P.	20:56
262,2	H.P.	20:59

CROSS FADE: street night
SCENE N: (Robert, Elaine) (SCENE S: in Script)

Shot	Characters	Time	
263,1	Street Scene	21:07	(Robert, Elaine, Random, Car)
264,2	Elaine	21:10	(Robert)
265,3	Robert	21:14	(Elaine)
266,4	Elaine	21:15	(Robert)
267,5	Robert	21:18	(Elaine)
268,6	Robert,Elaine	21:20	
269,7	Elaine	21:23	(Robert)
270,8	Robert	21:27	(Elaine)
271,9	Elaine	21:29	(Robert)
272,10	Robert	21:33	(Elaine)
273,11	Elaine	21:35	(Robert)
274,12	Robert	21:41	(Elaine)
275,13	Elaine	21:44	(Robert)
276,14	Robert	21:48	(Elaine)

277,15	Elaine	21:50	

CUT TO: Coffee Shop
SCENE O: (Jerry, George) (SCENE T: in Script)

Shot	Characters	Time	
278,1	Restaurant	21:57	(Randoms)
279,2	George, Jerry	22:01	(Randoms)
280,3	George	22:05	(Jerry) (Randoms)
281,4	Jerry	22:10	(George)
282,5	George	22:12	(Jerry) (Randoms)
283,6	Jerry	22:15	(George)
284,7	George	22:17	(Jerry) (Randoms)
285,8	Jerry	22:26	(George)
286,9	George, Jerry	22:30	(Randoms)
287,10	Jerry	22:34	(George)
288,11	George	22:35	(Jerry) (Randoms)
289,12	Jerry	22:38	(George)
290,13	George	22:41	(Jerry) (Randoms)
291,14	Jerry	22:47	(George)
292,15	George, Jerry	22:53	(Randoms)
293,16	George	22:56	(Randoms)
294,17	Jerry	23:01	door opens behind him

CUT TO: Street
SCENE P: (Jerry, George) (SCENE V: in Script)

Shot	Characters	Time	
295,1	Elaine	23:04	(Randoms)
296,2	Jerry	23:06	(Randoms)
297,3	Elaine	23:08	(Jerry)
298,4	Jerry	23:12	(Elaine) Pushes Elaine
299,5	Jerry, Elaine	23:13	Push continues
300,6	Jerry, Elaine	23:21	
301,7	Elaine	23:27	(Jerry)
302,8	Jerry	23:31	(Elaine)
303,9	Elaine	23:32	(Jerry)
304,10	Jerry, Elaine	23:37	

CUT TO: Police Station
SCENE Q: (Kramer, (Lineups)) (SCENE W: in Script)

Shot	Characters	Time	
305,1		23:37	(Random Cop)
306,2	Lineups, Kramer	23:43	
307,3	Lineups, Kramer	23:46	
308,4	Lineups, Kramer	23:57	
309,5	Lineups, Kramer	24:04	

CUT TO: Police Station
SCENE R: (Jerry, Gus, Cathy) (SCENE Y: in Script)

Shot	Characters	Time	
310,1	Jerry, Gus, Cathy	24:09	(Randoms)
311,2	Jerry	24:18	(Randoms)
312,3		24:26	Polygraph
313,4	Gus, Cathy	24:27	(Randoms)
314,5	Gus, Cathy, Jerry	24:30	(Randoms)
315,6		24:32	Polygraph
316,7	Gus, Cathy, Jerry	24:33	(Randoms)
317,8	Jerry	24:42	(Randoms)
318,9	Gus, Cathy	24:47	(Randoms)

319,10	Jerry	24:50	(Randoms)
320,11	Jerry, Gus	24:54	(Randoms) Polygraph
321,12	Jerry, Gus, Cathy	24:59	(Randoms) Polygraph
322,13	Jerry	25:02	(Randoms)
323,14	Jerry	25:06	(Randoms)
324,15	Gus, Cathy	25:08	(Randoms)

CUT TO: Jerry's Apartment
SCENE S: (Jerry, Elaine, George, Kramer) (SCENE Z: in Script)

Shot	Characters	Time	
325,1	Apartment	25:11	
326,2	Jerry	25:13	
327,3	Jerry	25:19	(Elaine)
328,4	Jerry, Elaine	25:21	
329,5	Elaine	25:27	(Jerry)
330,6	Jerry	25:53	(Elaine)
331,7	Elaine	25:59	(Jerry)
332,8	Jerry, Elaine	26:02	
333,9	Elaine, Jerry	26:05	
334,10	Elaine, Jerry, George	26:10	George Enters
335,11	Jerry...George...Elaine	26:17	George removes coat pan
336,12	Elaine	26:36	
337,13	George	26:37	
338,14	Elaine	26:41	(George)
339,15	George	26:42	
340,16	Kramer, Jerry	26:46	Kramer Enters
341,17	George	26:48	
342,18	Jerry, Kramer...(Elaine)	26:54	
343,19	Kramer	27:01	
344,20	George	27:05	
345,21	Kramer	27:06	
346,22	Jerry, Elaine	27:10	
347,23	All	27:13	
348,24	Jerry, Elaine, Kramer	27:17	
349,25	All	27:21	
FADE OUT:		27:36	

CUT TO: Lineups
SCENE T: (Kramer, Homeless Man,
Officer #2, Officer #3, (Lineups)))
(SCENE AA: in Script)

Shot	Characters	Time	
350,1	Kramer, Lineups	29:42	
351,2	Kramer, Lineups	29:48	
352,3	Office #2,#3, Kramer,H.P.	29:55	(Lineups)
353,4	H.P.	30:05	
354,5	Officer #3	30:14	
355,6	Kramer	30:15	
FADE OUT:		30:22	

Appendix C. **Image Processing Scripts**

This code uses the DYNAMO software package developed by J.Y.Wang of the Brain and Cognitive Sciences Department at MIT. This code is not publically available. It was developed for work on [Wang, 93].

```
#!/bin/csh -f

# script to generate segmentation of color key frames and color.cut frames in /feedfour/sift

set colorfile = $1
set currentframe = $2
set width = $3
set height = $4
set destfile = $5
set pyramid = $6

@ spatial_bleed = $pyramid + 5
@ size = ($pyramid + 1) * 50

/bin/cat $colorfile | \
rawtorle -w $width -h $height -n 3 | \
rletoraw -N -r | \
select-image -xdim $width -ydim $height -n 3 -intype ib |\
filter-image -xdim $width -ydim $height -median uniform |\
region-image2 -xdim $width -ydim $height -n 3 -s_terr $spatial_bleed -t_terr 1 -method
image -size $size -end 0 |\
region-image2 -xdim $width -ydim $height -n 1 -s_terr 1 -t_terr 1 -method image -size
100 -end 0 -o $destfile
```

```
#!/bin/csh -f

# script to generate motion magnitudes in /feedfour/sift

set rawfile = $1
set start = $2
set end = $3
set imagesizex = $4
set imagesizey = $5
set destfile = $6

/bin/cat $rawfile | \
convert-interleave -xdim $imagesizex -ydim $imagesizey -n 3 -start $start -end $end |\
select-image -xdim $imagesizex -ydim $imagesizey -n 3 -intype ib |\
image-transform -xdim $imagesizex -ydim $imagesizey -n 3 -convert rgby -intype ib -
outtype ib |\
```

```
temporal-filter  -xdim $imagesizex -ydim $imagesizey -intype ib -xfdim 5 -filt ".15 .35 .5
.35 .15" \
          -yfdim 5 -filt ".15 .35 .5 .35 .15" -op tfilt -outtype ib | \
optic-flow -analysis 2 -threshold 0 -xdim $imagesizex -ydim $imagesizey -intype ib | \
uniops -intype if -xdim $imagesizex -ydim $imagesizey -n 2 -op mag | \
convert-sh -xdim $imagesizex -ydim $imagesizey -pedestal 1 -scale 30 -outtype ib -o
$destfile
```

```
#!/bin/csh -f

# script to generate segmentation of motion magnitudes in /feedfour/sift

set magfile = $1
set currentframe = $2
set imagesizex = $3
set imagesizey = $4
set destfile = $5

/bin/cat $magfile | \
select-image -xdim $imagesizex -ydim $imagesizey -n 1 -intype ib -start $currentframe -
end $currentframe |\
filter-image -xdim $imagesizex -ydim $imagesizey -median uniform |\
region-image2 -xdim $imagesizex -ydim $imagesizey -n 1 -s_terr 1 -t_terr 1 -method
image -size 40 -end 0 | \
region-image2 -xdim $imagesizex -ydim $imagesizey -n 1 -s_terr 1 -t_terr 1 -method
image -size 100 -end 0 -o $destfile

echo $destfile done!
```

ACKNOWLEDGMENTS

No piece of work of this scope can be accomplished in a vacuum. I would like to recognize a few individuals who helped bring this project to fruition.

• *Thanks go to Ken Lowe at Castle Rock Entertainment who took the time to recognize that our use of the Seinfeld material was legitamate and worthwhile; and to MIT Counsel Connie Mitchell for guiding Roz and me through the legalese of the release process.*

• *I also appreciated the assistance of Professors Henry Jenkins, Glorianna Davenport and Whitman Richards who were all instrumental in helping me track down references.*

• *I am thankful to John Y. Wang for his assistance with his DYNAMO software package and for his willingness and efforts to make his tools more responsive to my research needs.*

• *Thanks go to a thorough and attentive research assistant, Holly Grabowski, who wrote the serial port listening code which parsed the closed captions.*

• *To Laurie Ward for helping me deal with vendors, order equipment or track down Roz. To Linda Peterson for her good humor and guidance with administrative requirements and formatting issues.*

• *I am grateful to the following Seinfeld fans and Media Lab residents who assisted me in gathering the ground truth labels for the 15,000 segments: Michelle McDonald, Wasi Wassid, Karrie Karahalios, Alex Sherstinsky, Chloe Chao and Anne Bui.*

• *My appreciation goes to my readers, Dr. Andy Lippman and Dr. HongJong Zhang for their patience and tolerance when reviewing early drafts of the thesis ,and for their insightful and directed comments.*

• *On a personal note, it is with great respect and admiration that I would like to recognize Tom Minka. I am indebted to him for his scholarship, sound advice and interest in this project which made my time working on it a worth while learning experience. Whether it was debugging code, debating the direction of the project or demonstrating some new concept, it was a genuine pleasure for me to have collaborated with him.*

• *For her strong positive example, commanding intellect, supportive nature and level headed perspective on research and life in general, I am particularly grateful and honored to have had Prof. Roz Picard as my advisor. Thanks for making graduate school the rewarding and satisfying experience it has been.*

• *Thanks also goes to my Father who read early drafts of the thesis and who provided sanctuary in his office in the form of luncheon escapes from the pressures of graduate school and who, together with my Mother, provided the support to make my retreival from LALA land the right decision.*