VIEWING BEHAVIOR MODEL GRAPHS (VBMG) FOR CHARACTERIZING USER VIEWING BEHAVIOR IN PROGRAM VISUALIZATIONS

by

MANISH AGARWAL

(Under the Direction of Eileen T. Kraemer)

ABSTRACT

We present a methodology for characterizing the gaze behavior of viewers of animated displays. We introduce a transition graph called **V**iewing **B**ehavior **M**odel **G**raph (VBMG) to characterize the behavior of users with similar viewing patterns into separate groups. We apply this methodology to the viewing of program visualizations. In this method the user's eye-fixation sequences are obtained using an Eye-Tracker and per-user viewing behavior models are created. We then cluster these per-user models to build VBMGs for each cluster. The VBMGs are useful because they help us classify users into separate groups, each user within a group having viewing behavior similar to others in the group. One useful application of VBMGs would be to dynamically capture viewing behavior and predict the cluster to which a user belongs, thus permitting on-the-fly adaptation of displays and other teaching materials.

INDEX WORDS: vbmg, viewing behavior model graph, clustering, program visualization

VIEWING BEHAVIOR MODEL GRAPHS (VBMG) FOR CHARACTERIZING USER VIEWING BEHAVIOR IN PROGRAM VISUALIZATIONS

by

MANISH AGARWAL

Bachelor of Technology, Uttar Pradesh Technical University, India, 2005

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2007

VIEWING BEHAVIOR MODELS GRAPHS FOR CHARACTERIZING USER VIEWING

BEHAVIOR IN PROGRAM VISUALIZATIONS

by

MANISH AGARWAL

Major Professor: Eileen T. Kraemer

Committee: Maria Hybinette
Daniel M. Everett

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
May 2007

DEDICATION

To my Parents.

ACKNOWLEDGEMENTS

I wish to thank and express my sincere gratitude to my Major Advisor Dr. Eileen Kraemer for her constant support and encouragement. Dr. Kraemer has been very generous with her time and wisdom and I am greatly inspired by her genuine concern for others and her always smiling attitude.

I would also like to thank my committee members Dr. Maria Hybinette and Dr. Daniel M. Everett for their time and consideration.

I also thank my lab mates Shradha, Shaohua, Phillipa and Kelly who made working in lab so much fun. Special thanks to Shradha for her valuable input. I would also like to thank Vivek, Pawan and Rippy for all those moments of joy, Varun and Sumit for all the emotional support, and my parents who sacrificed a lot to bring me at this point of life.

I am grateful to you all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Visualizations have often been considered an effective way to promote the understanding of both abstract and concrete ideas [Baecker81, Tufte90]. Researchers have taken advantage of new technologies to help them understand difficult and otherwise incomprehensible data. In the context of Computer Science education, Program Visualization, also called Software Visualization or Algorithm Animation, can be used to help illustrate and present computer programs, processes, and algorithms. The idea of using algorithm animations as an aid in teaching and software development began with Baecker's "Sorting out Sorting" video [Baecker81]. Algorithm animations can be used in education, to help students understand data structures and the mechanics of an algorithm. They are also useful in software development to help programmers to debug and understand their code better.

Even though the idea of using program visualizations (PVs) to help users understand concepts has been around for a long time, PVs are yet to be accepted as being widely successful [Gurka96, Mulholland98, Hundhausen02]. Studies into the effectiveness of program visualizations have yielded mixed results. While some have shown improvement in viewer comprehension [Hansen00, Lawrence93] others have argued that there is no significant improvement in viewer comprehension [Gurka96, Mulholland98, Hansen00].

Our research tries to answer the question of why some visualizations are useful while others fail to yield any significant benefit. Based on previous research by members of our group

[Rhodes06a, Rhodes06b, Rhodes06c, Kraemer06], three key approaches were identified: dual-coding, individual differences and levels of engagement. This thesis concentrates on understanding individual differences between viewing behaviors of users in program visualizations through the use of viewing behavior modeling.

In this work we formulate a methodology to characterize viewing behavior of users in program visualizations. For a long time researchers have used stochastic models to model behavioral patterns [Jagerman96, Hlavacs99, Manavoglu03]. We build our methodology using one such stochastic model, The Markov Chain. For our study we use the SSEA software, **S**ystem for **S**tudying the **E**ffectiveness of **A**nimations, which is a testing environment to study visualization design [Kraemer06]. The SSEA interface is divided into four main areas: an animation area, a pseudo-code display, animation controls and a question area. We define three separate areas of interest (AOI) based on the requirements. We selected the Code area, the Animation area and the Animation's Caption area as the three AOI. We selected these AOI with the understanding that these are the main areas that disseminate information about the algorithm and any user would be looking at only one of these areas at a time.

Using an eye-tracker we record the users' eye-gaze pattern while the user is using the SSEA system to study an algorithm. The eye-gaze patterns are the sequences of users' eye-fixations. The eye-gaze patterns are then mapped to the defined areas of interest (AOI), which results in the sequence of area of interests. These sequences of AOI are then used to build Transition Matrices which are essentially counts of the total number of transitions between any two AOI. The transition matrices are then used to calculate Transition Probability Matrices for each user, which model Markov Chains. The transition probability matrices are then clustered based on a distance function. The final cluster matrices thus obtained are used to build the **V**iewing **B**ehavior **M**odel **G**raph (VBMG).

The VBMG help us in addressing one common problem with visualizations, the problem of determining the level of abstraction and the type of information that program visualizations should display. While some visualization place more emphasis on animation and less on text, there are others which stress on textual description. This is an important issue since this tends to make program visualizations suitable only to one kind of user. Our methodology of identifying users into separate groups based on their viewing behavior will help in adapting the algorithm animation based on the individual needs of a user. For example, if a user follows the behavioral pattern of a previously identified group of users, then the algorithm animations can be adapted accordingly.

In Chapter 2, we discuss related research that has been done to model user viewing behavior and to categorize viewing behavior of users in the domain of human-computer interaction. Chapter 3 provides a description of our methodology of characterizing viewing behavior. In Chapter 4, we present the data that was used for experimentation and evaluation. To test our method we created some synthetic data. We present both the manufactured data as well as the experimental data obtained from actual gaze patterns. We conclude our work in Chapter 5 and discuss the results.

CHAPTER 2

BACKGROUND AND RELATED WORK

In the domain of program visualizations very few researchers have tried to assess individual differences in viewing behavior of users. The studies done in [Crosby95] support, to an extent, the theory that individual differences affect learning through algorithm animations but there has been little work done to model the viewing behavior in these animations. Although few studies have been done to characterize viewing behavior in evaluating the effect of individual differences in use of program visualizations, there has been some work done in other areas of HCI. Researchers have used viewing behavior modeling to understand user behavior and help improve the design of websites. The use of viewing behavior modeling has also been investigated in video sessions to aid in audiovisual production techniques.

In this chapter we survey the various approaches that exist in modeling user behavior in various computer interfaces. We briefly describe the Markov Chains (Markov Model) concept and also it's application in user behavior studies. Following this, we focus on past and present research on modeling user behavior. We discuss the work done by some researchers in different areas of Human Computer Interaction (HCI) where behavioral models have been used to characterize user viewing behavior.

2.1 Markov Chains

Models for simulating realistic user behavior have been studied by researchers for many years [Jagerman96, Manavoglu03]. Most of these models are represented by stochastic processes. User behavior models try to build the sequence of user interactions at a higher level of abstraction. One such model, the Markov Model (or Markov Chain), has been used extensively by researchers in modeling user behavior [Menasce99, Mongy, Hlavacs99, Manavoglu03].

A Markov Chain is formed of a set of states S = {s_1, s_2, ……., s_n} where 'n' is the number of states. There is an initial state s_1, from which the process starts and then moves successively from one state to another [Grinstead]. The probability of the chain to move from state s_i to s_j is denoted by p_{ij} [Kemeny74]. The probability p_{ij} does not depend on the previous state, i.e. state s_{i-1}.

The probabilities p_{ij} are called transition probabilities. If the process s_i remains in the same state then its probability is denoted as p_{ii}.

<u>Transition Matrix</u>

The probabilities p_{ij}; where i = {1, 2, …….., n}, j = {1, 2, …….., n} and n is the number of states; can be represented in form of a two dimensional matrix called a Transition Matrix.

For example, consider the following transition matrix:

$$
\begin{array}{c} \\ \\ P = \end{array}
\begin{array}{c} \\ X \\ Y \\ Z \end{array}
\begin{array}{c} \begin{array}{ccc} X & Y & Z \end{array} \\ \begin{pmatrix} 1/4 & 1/4 & 1/2 \\ 1/2 & 1/4 & 1/4 \\ 1/4 & 1/2 & 1/4 \end{pmatrix} \end{array}
$$

The above transition matrix gives the probability of transition between three states X, Y and Z. Each row in the transition matrix gives the probability of going from that state to all other states. For example, if the current state is X then the probability of being in the same state at the next time period is ¼ , the probability of next state being Y is ¼ and the probability of next state to be Z is ½ . Similarly we can get the probability of transition from any one state to any other state from the transition matrix. Predictably, the sum of each row is 1.

<u>Transition Graph</u>

The transition matrix can be represented in pictorial form through a transition graph also called a state transition graph. The figure below shows the transition graph for the example transition matrix discussed above.

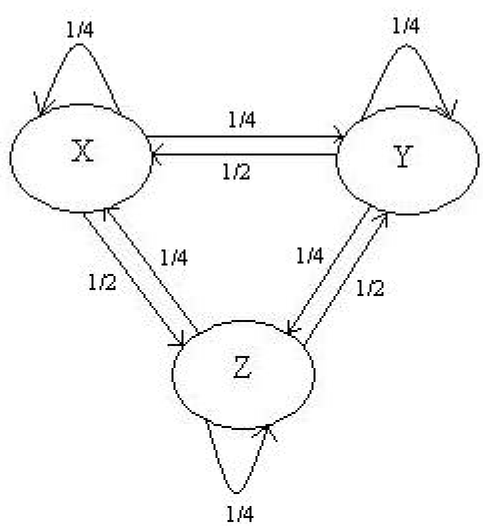

Figure 2.1 State Transition Graph

The states of the probability matrix form the nodes of transition graph and the edges of the transition graph are labeled with the probability between each state.

2.2 Applications of Behavior Modeling

Markov Chains have been used in statistics [Freedman83], physics [Sokal89], genetics [Nix92], and psychology [Miller52] and even in music [McCormack96]. In psychology the Markov model has been used to predict the learning habits of animals as well as humans [Estes50]. As early as 1950 Estes built learning models based on probability matrices [Estes50].

In the field of Human-Computer Interaction, the importance of behavior modeling is more apparent than ever before. As the user base of computers increases and so does its applications, it is desired that the user is presented with only relevant information. Since there is always a difference between individual needs and expectations of each person, computer-human interactions cannot be tied to one general standard. It is imperative that the users interact with the computers at a level that conforms to their ability and their needs. In this context Behavioral Models can prove to be very useful. Researchers are studying applications of behavioral models in all areas of HCI. One such application is computer based learning, where the educator would want to adjust the difficulty of tutorials based on the learning ability of a particular student. Therefore as the student starts the tutorial, the computer system can automatically adjust the difficulty level based on initial responses of the student and matching the responses to the stored behavior models. Another application, which has already been implemented to some extent, is targeted advertising on websites [Johnston06, Chatterjee98, Montgomery03]. The web logs of websites are used to build models and predict user behavior. The advertisers may analyze the sequence in which the user surfs the web-pages or the areas of web-pages that are viewed the maximum amount of time. This helps advertisers in targeting users with relevant advertisements and also to place the advertisements at the right location so that they attract user attention without being annoying.

Another important issue is that of security. The ever increasing usage of the internet also brings with it the problem of security of financial transactions as well as maintaining secrecy of confidential information. The behavior models can be used to detect unusual behavior and subsequent action can be taken. There are a number of other uses of behavior models such as to improve the organization of a Web site to better serve customers, to identify new trends in consumer behavior for improving profit, etc. Our particular interest is the application in program visualization and algorithm animations.

2.3 Approaches to Modeling User Behavior

For a long time, researchers have been trying to build behavior models that can predict user behavior accurately. Researchers have followed different approaches in an effort to build a behavior model that would work best in that particular scenario. In the context of HCI, a survey of approaches to modeling user behavior found that Markov Chains have been the fundamental concept behind most of the approaches taken to model user behavior [Menasce99, Mongy, Manavoglu03]. Each approach differs in the methodology followed to build the behavioral model. Although not much work has been done to model viewing behavior in program visualizations, work has been done in other areas of HCI. We focus on some of the past and present work by other researchers in modeling and characterizing user behavior in the area of HCI.

Menasce *et al.* [Menasce99] use a behavioral model, called the Customer Behavior Model Graph (CBMG) for workload characterization of E-commerce sites. CBMGs are used to "describe the behavior of groups of customers who exhibit similar navigational patterns". These CBMGs are then applied to derive relevant workload metrics for e-commerce sites, which could help in performance evaluation and capacity planning. Metrics such as average session length, average number of items bought per customer and buy to visit ratio can be obtained from the CMBG. It has been discussed that the workload of an e-commerce website can be represented using the sequence of requests of different types made by a customer during his/her visit to the site. A sequence of requests can be browse, search, select, add to cart, and pay. These sequences of requests are represented using the Customer Behavior Model Graph (CBMG).

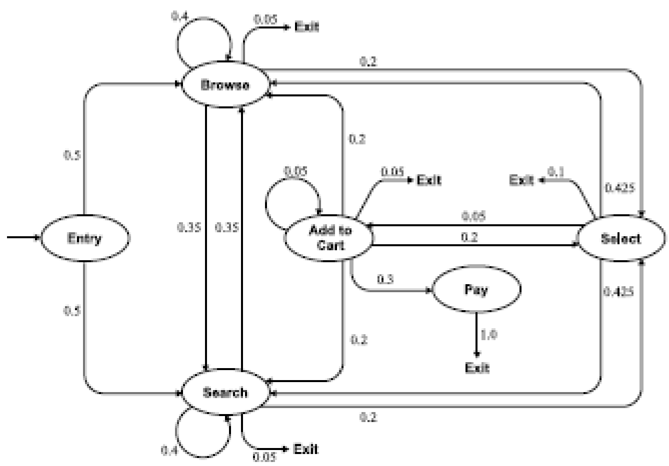

Figure 2.2, [Menasce99], CBMG of an occasional buyer.

Figure 2.2 above shows a CBMG. It has one node for each type of request (or state). Each edge of the above graph is assigned a probability, which is the probability of transition from one state to another state. The idea here is that different types of users can be characterized by different CBMGs. Therefore we can have different graphs for frequent buyers, occasional buyers and window shoppers. These graphs for different categories of users can be built by analyzing the logs of e-commerce websites to identify navigation and buying patterns. The analysis uses a k-means clustering algorithm to identify groups of customers with similar navigational patterns. The Euclidean distance function is used to calculate the distance between two points represented by a transition count matrix for each user. Once the clustering is completed, each cluster centroid represents the navigational pattern of a specific group of buyers. From these cluster centroids, transition probability matrices are calculated for each centroid. Thus a probability matrix is obtained for each buyer group that represents that groups' CBMG. Once the CBMGs have been built, they can be used to provide varying levels of service to customers. When a customer logs on to the website and starts navigating, the system can match the navigation pattern of the user to the profiles already stored and thus provide an appropriate level of service based on their profile. A customer whose navigation pattern matches that of a frequent buyer would thus get priority treatment.

Mongy *et al.* [Mongy], discuss a similar approach to model user behavior in video sessions. In a video an action log of a session is defined by a sequence of play, fast forward, rewind, etc. actions. These logs are analyzed to extract viewing behavior characteristics of users. The first order Markov models are employed to characterize viewing behavior into different categories such as "fast viewing of video", "viewing of a specific video sequence" or "a complete viewing of the video". The sequence of actions and their durations during a video session are used to construct the first order Markov models. It was important to consider time spent between each consecutive action, therefore to account for the time spent in each state it is assumed that the transition is done every one second. Thus if a user spends 'n' number of seconds in a state 'x', it is recorded as 'n' transitions from state 'x' to state 'x'. The nodes of models are the actions such as play, rewind etc. and the edges are the transitions between these actions. Again, as in other applications, the edges are labeled with probabilities of transition between these actions. The characterization is done using a clustering algorithm that clusters the transition matrices. The Kullbach-Leibler distance function [Kullbach93] is used to calculate the distance between two transition matrices.

Table 2.1 and Figure 2.4 below show the transition matrix and the transition graph obtained after the analysis of action logs of video sessions.

Table 2.1, [Mongy], Transition Matrix for a video session

	PLAY	PAUSE	JUMP	FFW	RWD	STOP
PLAY	0.9	0.08	0	0	0	0.02
PAUSE	0.21	0.54	0.07	0.08	0.05	0.05
JUMP	0	1	0	0	0	0
FFW	0	0.88	0	0.12	0	0
RWD	0	0.85	0	0	0.15	0
STOP	0	0	0	0	0	1

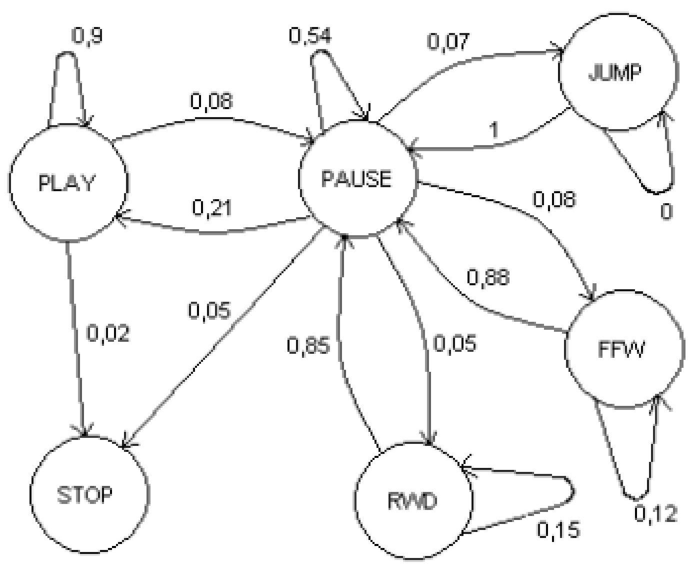

Figure 2.3, [Mongy], Transition Graph of a video session

It has been argued that the categorization of viewing behavior in a video session can help in information publishing through videos. One such application can be analysis of quality of video of a commercial advertisement.

The above discussed approaches help us in formulation of our methodology to characterize viewing behavior in program visualizations. In chapter 3, we explain our approach to this problem of modeling user viewing behavior.

CHAPTER 3

METHODOLOGY

As discussed earlier, viewing behavior modeling is essential to help us understand individual differences in viewing behavior. A viewing behavior model should represent the actual viewing behavior of users of an interface. Our method of building viewing behavior models provides such a characterization of the viewing behavior. In this chapter, first we give a high level description of our approach to build viewing behavior models. Later, we provide a more detailed description of our methodology.

3.1 Metrics

To evaluate the quality of the clustering algorithm that we implement in our methodology we use the metrics described in this section [Ray99]:

1. <u>Average Intra-Cluster Distance</u> (d_{intra}): Average intra cluster distance is the average distance of all points P_i of a cluster from its centroid C. Therefore if there are N points in a cluster then the d_{intra} is defined as follows:

$$d_{intra} = \frac{1}{N}\sum_{i=1}^{N} d(P_i, C)$$

The function $d(x_1,x_2)$ is the Euclidean distance between points x_1, x_2. For optimal clustering the average intra-cluster distance should be minimized.

2. Minimum Inter-Cluster Distance (d_{inter}): Inter-cluster distance is the distance between clusters. It is calculated by computing the distance between centroid of two clusters C_1 and C_2.

$$d_{inter} = d(C_1,C_2)$$

   Again, the distance function used here is the Euclidean Distance. For optimal clustering the clusters should be as far apart as possible and therefore the inter-cluster distance should be maximum. We use only minimum inter-cluster distance in our calculations since we want the smallest of the inter-cluster distance to be maximized, and other larger values will automatically be bigger than this value.

3. Ratio Intra-Cluster Distance to Inter-Cluster Distance (r): Since we know for optimal clustering the intra-cluster distance should be minimum and the inter-cluster distance should be maximum, therefore the ratio r should be minimized to achieve good clustering.

4. Errors: The number of errors in a clustering is equal to the number of points which are present in the wrong cluster.

3.2 Overview of Methodology

Below is a summary of the steps followed to build the viewing behavior model graphs.

1. **User Interface**: For our study we use the SSEA interface, **S**ystem for **S**tudying the **E**ffectiveness of **A**nimations, which is a testing environment to study visualization design [Kraemer06]. The SSEA interface is divided into four main areas: an animation area, a pseudo-code display, animation controls and a question area.

2. **Area of Interest:** We divide the user interface into parts called Areas of Interest (AOI). The areas of interests are chosen such that the user is expected to focus on only one of them at a time. These AOIs serve as the building block of our viewing behavior model.

3. **Eye-gaze sequence recording:** Using an eye-tracker we record the users' eye-gaze pattern while the user is using the SSEA system to study an algorithm. The eye-gaze patterns are converted to sequences of AOI.

4. **Build Transition Count Matrices:** The sequences of AOI are used to build Transition Count Matrices, counts of total number of transitions between any two AOI, for each user.

5. **Calculate Transition Probability Matrices:** The transition matrices are used to calculate Transition Probability Matrices for each user, which are basically Markov Chains.

6. **Clustering:** The transition probability matrices are clustered using the K-means clustering algorithm [McQueen67]. The Euclidian distance is used to calculate the distance between the matrices. Running with multiple values of K, and using multiple runs for each value of K, and then selecting the "best" is part of the methodology.

7. **Viewing Behavior Model Graph:** The final matrices of the centroids of each cluster obtained after clustering are used to build the Viewing Behavior Model Graph (VBMG).

3.3 Detailed Description

3.3.1 User Interface

In this study we use the System for Studying the Effectiveness of Animations (SSEA) [Kraemer06]. SSEA is a testing environment used to study visualization design. The SSEA interface is divided into four main areas: an animation area, a pseudo-code display, animation controls and a question area as shown in the figure 3.1 below.

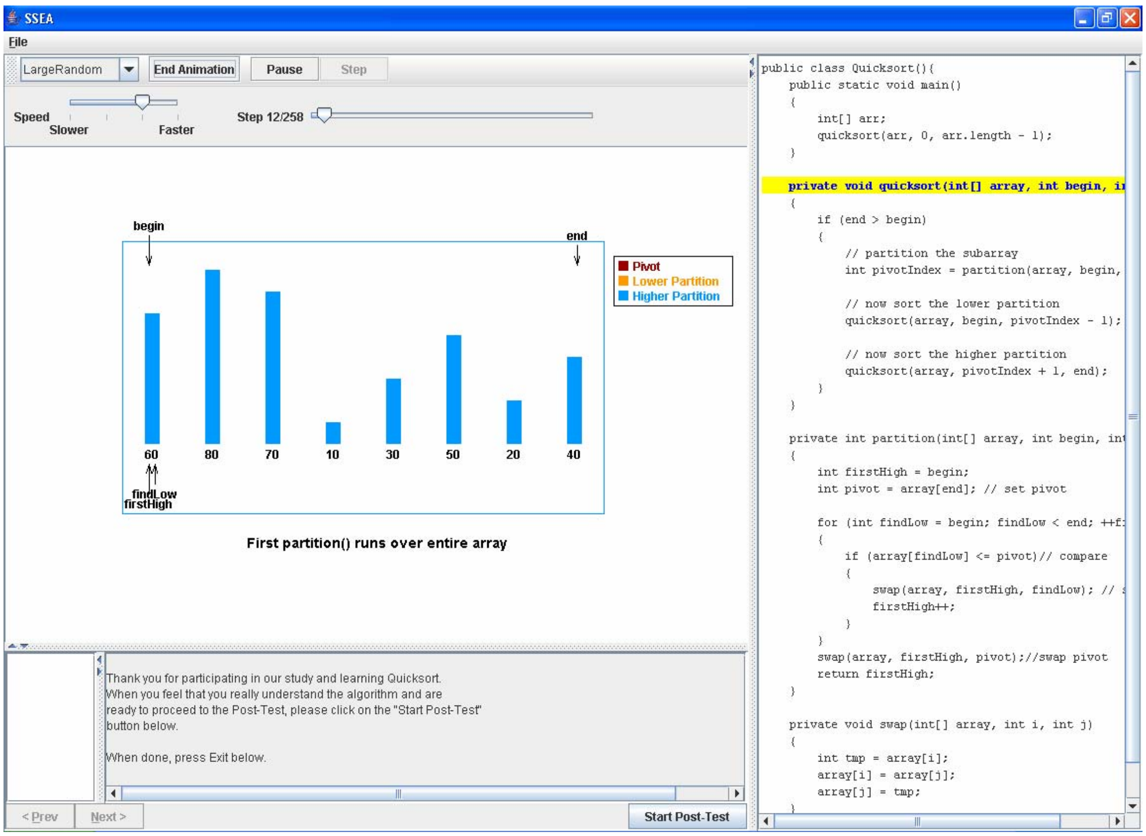

Figure 3.1 SSEA Screenshot

The animation and the pseudo-code in the visualization are synchronized such that the changes in the animation correspond to the highlighted code. The viewer can adjust the speed of animation playback using the controls above the animation area.

3.3.2 Area of Interest

As shown in the figure below, the SSEA interface is divided into units called Areas of Interest (AOI).

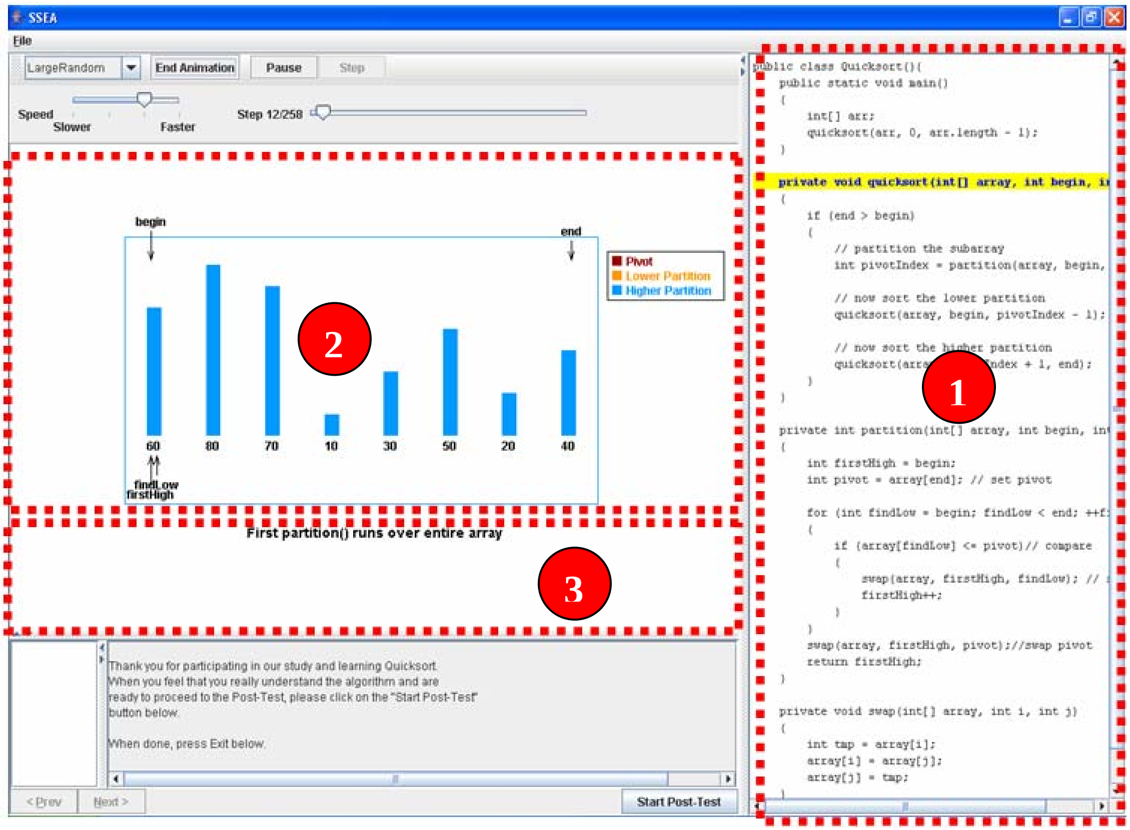

Figure 3.2 SSEA Interface with marked AOI – Area 1:Code, 2:Animation, 3:Caption

Since the SSEA interface is already divided into separate areas: an animation area, a pseudo-code display, animation controls and a question area, we can either assume each of these areas as a separate area of interest (AOI) or we can define our own AOI based on the requirements. In our work we selected three areas of interest. The first AOI is the Code area, the second AOI is the Animation area and the third AOI is the Animation's Caption area. We selected these AOI with the understanding that these are the main areas that disseminate information about the algorithm and any user would be looking at only one of these areas at a time, and since our purpose is to model behavior based on parts of the interface that are focused on by the user, it was a natural choice.

AOIs serve as the basic component of our viewing behavior model. We consider each AOI as a state in the model and thus the user's gaze shifting from one AOI to another AOI are used as transitions in our behavior model.

3.3.3 Eye-Gaze Sequence Recording

For recording the eye-gaze sequence, we recruited 12 subjects from the CSCI 4800 – Human Computer Interaction class at The University of Georgia. All necessary permissions and consents were obtained to conduct human subjects study. The subjects used the SSEA interface to study the quicksort algorithm. While the subjects were studying the algorithm, their eye movements were recorded using the ASL Eye-Trac 6000 System.

Figure 3.3 ASL Eye Trac 6000 System, Courtesy: ASL Labs

The Eye-Trac 6000 system records the eye fixations while the user is looking at the interface. The ASL Eyenal and Fixplot software provided with the Eye-Trac 6000 system were used to define the AOI, (explained in last section), in the user interface. The software is then used to superimpose the eye-fixation sequences on the Areas of interest. The following figure shows an example plot of eye-fixation sequence on the AOI.

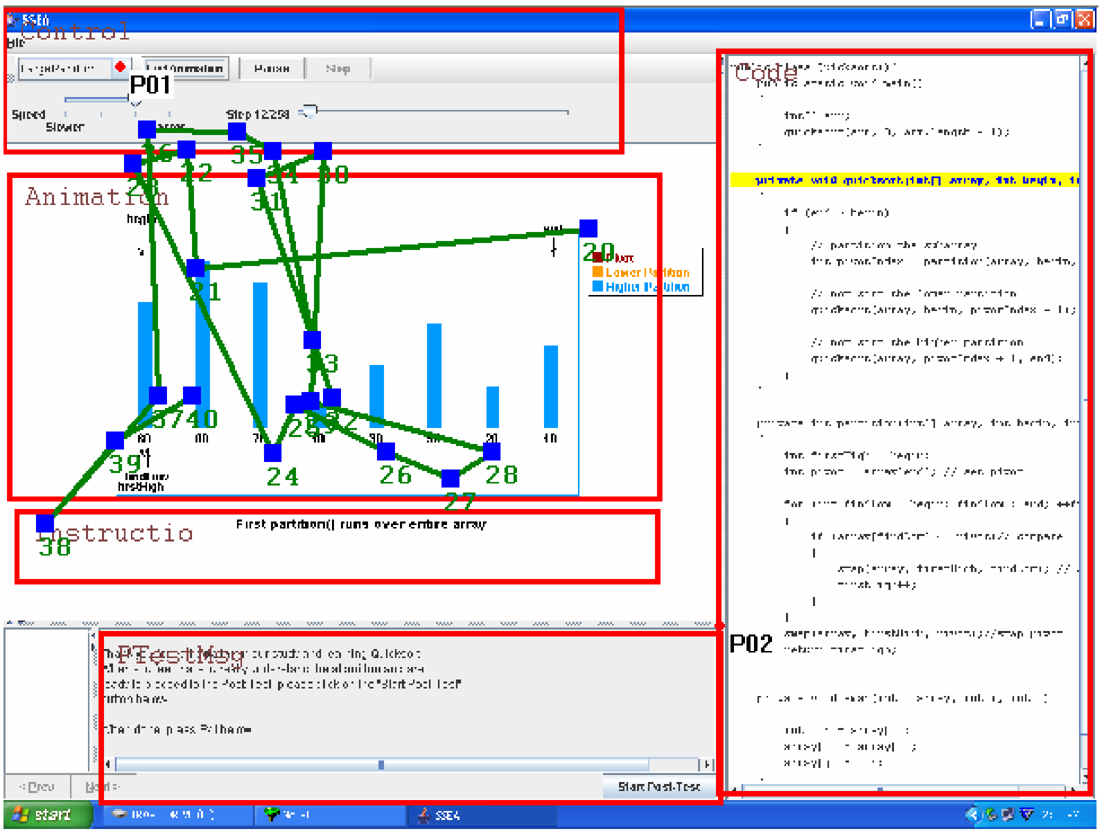

Figure 3.4 Eye-Fixation Sequence superimposed over the defined AOI

By superimposing eye-fixation over the AOI we get the sequence of AOI in which user moves his focus from one AOI to another AOI.

3.3.4 Building Transition Count Matrices

Once we have obtained the sequence of AOIs for each subject, we convert these sequences into transition count matrices. Transition count matrices are two dimensional arrays that store total number of transitions between any pair of AOIs.

Table 3.1 Sample Transition Count Matrix

	Off	*Controls*	*Animation*	*Caption*	*PesudoCode*	*Questions*
Off	13	0	6	6	4	4
Controls	2	8	5	0	0	1
Animation	6	6	265	30	1	16
Caption	7	1	29	43	0	0
PseudoCode	2	0	2	1	0	0
Questions	4	0	17	0	0	57

The table above shows a sample transition matrix obtained after recording the eye-gaze sequence of a human subject. As can be seen in the above transition matrix, there is an additional row for “Off”. This is used to take into account situations in which the user’s eye gaze shifts outside the screen area. Since we are using only Animation, Code, and Caption as our AOIs, we obtain the transition count matrix for each user for only these AOI.

3.3.5 Calculate Transition Probability Matrices

Once we have the transition count matrix for a user, we calculate the transition probability matrix. Transition probability is the probability of transition from one AOI to another AOI. Thus the sum of probabilities in each row should be 1.

Table 3.2 Sample Transition Probability Matrix

	Off	*Controls*	*Animation*	*Caption*	*PesudoCode*	*Questions*
Off	.1	.1	.2	.1	.2	.3
Controls	.3	.1	.1	.1	.2	.2
Animation	.25	.25	.1	.1	.1	.2
Caption	0	.1	.2	.2	.3	.2
PseudoCode	.2	.3	.1	.2	.1	.1
Questions	.1	.2	.3	.2	.1	.1

The Transition Probability Matrices are the two dimensional representations of the first order Markov models of the viewing behavior for each user. Thus we can use these viewing behavior models of all subjects to group behavior models into different clusters.

3.3.6 Clustering

Once we have the transition probability matrices we perform clustering analysis which gives us a group of clusters that represent different viewing behavior patterns of subjects. The centroid of each cluster defines the final VBMG characteristics. There are various approaches available in the literature to cluster matrices [Everitt80, Menasce94]. We use the K-Means clustering algorithm to cluster the transition probability matrices [MCQUEEN67]. The Euclidian distance is used to calculate distance between the matrices.

The K-means clustering algorithm works as follows:

1. Assume K points in the space represented by points to be clusters. These K points are selected randomly from the points in space. The randomly selected K points are the initial group centroids.
2. Choose one point randomly among the points which have not been already added to the cluster and assign the point to the centroid which is closest to that point.
3. Recalculate position of the centroid to which the point has been added.

4. Repeat steps 2 and 3 until all points have been assigned to some cluster.
5. Repeat steps 2 and 3 until no point changes the cluster of which it was part in the last iteration.

The value K, which is the number of clusters, has to be provided to the algorithm. The clustering algorithm needs the definition of distance which is used to calculate the distance of one point (transition probability matrix in our case) from another. To calculate the distance between each probability matrix we used the Euclidian distance function. The Euclidian distance function is defined as follows, where P1, P2 are the two points between which we need to calculate the distance and n is the number of AOIs.

$$d_{p_1, p_2} = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(P_1[i, j] - P_2[i, j]\right)^2}$$

When a point is to be added to a cluster, the new value of the centroid is calculated by taking the weighted mean of the centroid and the new point. For example, if a point P is to be added to the cluster C which already contains k points, the new values of centroid of the cluster are computed as follows:

$$C[i, j] = \frac{k \times C[i, j] + P[i, j]}{k + 1}$$

The clustering algorithm is run multiple times for each value of k ranging from 1 up to k-1. For each value of k we perform 20 iterations and select the "best" clustering for that value of k. The best clustering occurs when the ratio of intra-cluster distance to inter-cluster distance is minimum among the other 19 clustering possibilities. E.g. for 4 clusters we may have 6 different clustering permutations, from these 6 different clustering permutations we select the clustering which has the minimum ratio of intra-cluster distance to inter-cluster distance. This is done because we want the clustering to be as tight as possible.

3.3.7 Viewing Behavior Model Graph

Once the transition probability matrices have been clustered, we obtain groups of subjects who have similar viewing behavior in the program visualization. The centroid of each cluster is representative of the viewing behavior of its members. Finally we build the Viewing behavior model graph using the centroids of cluster matrices that are obtained. The centroid matrices give us the transition probability for each group of users. Thus a VBMG can be built using the AOIs as nodes and labeling the edges of the VBMG with the transition probabilities of the corresponding centroid.

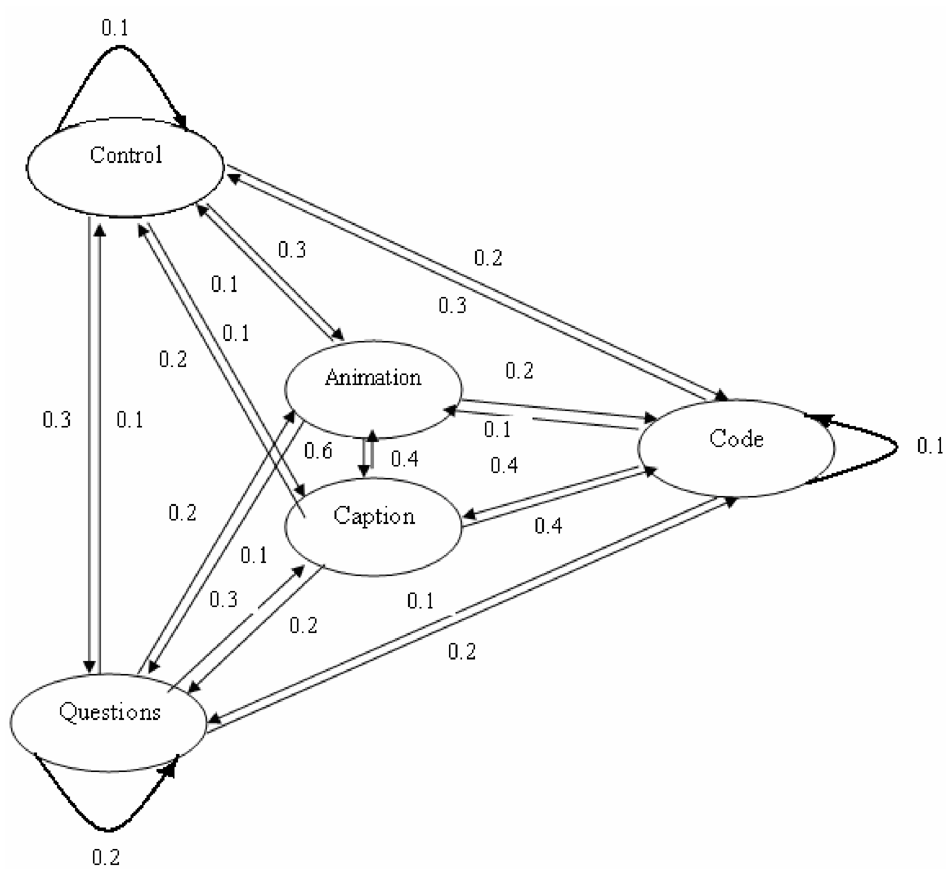

Figure 3.5 Sample Viewing Behavior Model Graph

The VBMG can help us in understanding viewing behavior of users in program visualizations. We can vary the number of clusters to compare and analyze the results. In Chapter 4, we present some experimental data generated in our study. Also, we present the results of testing that was performed using synthetic data to test our methodology.

CHAPTER 4

EXPERIMENT AND EVALUATION

This chapter presents both real experimentatal data and synthetic data that was generated and also the results of our analysis of that data. First we present the synthetic data that we used to evaluate our methodology and the results obtained. Later we present the experimental data that was obtained from the eye-tracker during the experiments with human subjects.

4.1 Manufactured Data

To test our methodology of characterizing viewing behavior through clustering, we generated artificial raw data and then analyzed it using our algorithm. In order to mimic the actual viewing behavior we used pre-determined VBMGs to generate the gaze sequences and then used these gaze sequences to build transition matrices for each hypothetical user. In this section we first explain how we generated the gaze sequences and then present the results of our analysis of that data.

Gaze Sequence Generation

While generating gaze sequences we wanted to generate sequences that would represent a specific type of viewing behavior. For this purpose, our gaze sequence generation method is a function of the VBMG for that specific viewing behavior. E.g. Table 1 shows a transition probability matrix for a VBMG, which represents the viewing behavior of a user whose gaze sequence would have exactly equal number of transitions from any one AOI to any other AOI.

We call this transition probability matrix a Seed Matrix because we use this matrix to generate a large number of gaze sequences, each sequence different from one another but representing the same viewing behavior as its seed matrix.

Table 4.1 Sample Seed Matrix

	AOI 1	AOI 2	AOI 3	AOI 4	AOI 5
AOI 1	0.2	0.2	0.2	0.2	0.2
AOI 2	0.2	0.2	0.2	0.2	0.2
AOI 3	0.2	0.2	0.2	0.2	0.2
AOI 4	0.2	0.2	0.2	0.2	0.2
AOI 5	0.2	0.2	0.2	0.2	0.2

Similar to the VBMG described above, we can have many seed matrices which in turn can be used to generate more gaze sequences. Table 2 shows a transition probability matrix for a VBMG which represents a different viewing behavior than what is represented by Table 1. The **Euclidian distance** between the matrices of Table 1 and Table 2 is **0.40**.

Table 4.2 Sample Seed Matrix

	AOI 1	AOI 2	AOI 3	AOI 4	AOI 5
AOI 1	0.3	0.15	0.15	0.1	0.3
AOI 2	0.3	0.25	0.05	0.3	0.1
AOI 3	0.2	0.25	0.15	0.1	0.3
AOI 4	0.1	0.3	0.2	0.15	0.25
AOI 5	0.3	0.15	0.25	0.15	0.15

Since a gaze sequence is a sequence of AOIs, we need to decide the starting AOI in the sequence. To preserve the randomness we use a Start AOI probability matrix to randomly select the starting AOI. The start AOI probability matrix also allows us to emphasize more on particular AOI if we need an AOI to be starting AOI more often. Table 3 shows an example of a start AOI probability matrix.

Table 4.3 Starting AOI Probability Matrix

AOI	Probability	Cumulative Probability
Control	0.1	0.1
Question	0.1	0.2
Animation	0.4	0.6
Caption	0.2	0.8
Code	0.2	1.0

The seed matrices and the start AOI probability matrices are then used to build a number of eye gaze sequences for each type of viewing behavior represented by its VBMG. This is done as follows:

1. A random number generator generates a random floating point number and matches it to the cumulative probability in starting AOI probability matrix; this gives us the first AOI at which the gaze sequence starts.
2. After we obtain the starting AOI the next AOI is obtained as a function of the VBMG which is represented by the seed matrix. It follows the steps listed below:
    a. A random floating point number is generated.
    b. The random number is looked up in the transition probability matrix of the VBMG. The lookup is performed in the row corresponding to the AOI from last iteration.
    c. The column corresponding to the matched probability cell gives us the next AOI.

In our experiments we built sequences of length 1000, meaning each sequence had 1000 transitions. Once we have obtained gaze sequences, we use these sequences to build Transition Count Matrices, as explained earlier in Chapter 3, for use in our experiments.

4.1.1 Manufactured Data Experiment I

In our first experiment we wanted to check the performance of our algorithm by examining if our approach yields the correct number of clusters or not. To test this we used three VMBGs, whose representative probability matrices are shown in Table 4, 5 and 6.

Table 4.4 Seed Matrix 1

	AOI 1	AOI 2	AOI 3	AOI 4	AOI 5
AOI 1	0.2	0.2	0.2	0.2	0.2
AOI 2	0.2	0.2	0.2	0.2	0.2
AOI 3	0.2	0.2	0.2	0.2	0.2
AOI 4	0.2	0.2	0.2	0.2	0.2
AOI 5	0.2	0.2	0.2	0.2	0.2

Table 4.5 Seed Matrix 2

	AOI 1	AOI 2	AOI 3	AOI 4	AOI 5
AOI 1	0.3	0.15	0.15	0.1	0.3
AOI 2	0.3	0.25	0.05	0.3	0.1
AOI 3	0.2	0.25	0.15	0.1	0.3
AOI 4	0.1	0.3	0.2	0.15	0.25
AOI 5	0.3	0.15	0.25	0.15	0.15

Table 4.6 Seed Matrix 3

	AOI 1	AOI2	AOI 3	AOI 4	AOI 5
AOI 1	0.35	0.15	0.05	0.15	0.3
AOI 2	0.3	0.25	0	0.35	0.1
AOI 3	0.35	0	0.3	0	0.35
AOI 4	0.05	0.3	0.35	0.05	0.25
AOI 5	0.3	0.15	0.25	0	0.3

The table below lists the Euclidean distance between each pair of seed matrix.

Table 4.7 Distances between Seed Matrices

Seed Matrices	Euclidean Distance
[1, 2]	0.4000000000000000
[1, 3]	0.6519202405202640
[2, 3]	0.4690415759823420

We generated 20 gaze sequences from each of the three VBMGs of Table 4, 5 & 6, using the method described in earlier in this section. Therefore in total we get 60 gaze sequences. For each of these 60 gaze sequences we build transition count matrices for further analysis using the clustering algorithm.

Once we have obtained 60 transition count matrices, we cluster these matrices using our algorithm. Our goal is to cluster the points such that the intra cluster distance is minimum and the inter cluster distance is maximum. It appears that this case would occur when the number of clusters is equal to the number of points but since we want fewer clusters therefore we use the ratio of intra-cluster distance to inter-cluster distance to help us decide on the number of clusters [Ray99]. Table 8 presents the ratio of intra-cluster distance to inter-cluster distance, for varying number of clusters *k*, obtained after clustering the 60 transition count matrices that were generated.

For each value of k the clustering algorithm was run for 20 iterations and the clustering with best ratio of intra-cluster to inter-cluster distance was chosen. Since we want the clusters to be as tight as possible, the ideal clustering would be the one which has minimum intra-cluster distance within clusters and maximum inter-cluster distance between clusters. Therefore we selected the clusters with minimum ratio of intra-cluster to inter-cluster distance.

Table 4.8 Ratio intra-cluster/inter-cluster distances

No. of Clusters	**Ratio** intra-cluster/inter-cluster distance
1	Infinity
2	0.438654
3	0.329148
4	0.896797
5	0.881402
6	0.919299
7	0.880399
8	0.826609
9	0.966546
10	0.939831
11	0.910447
12	0.88944
13	0.948472
14	0.932206
15	0.909169
16	0.868453
17	0.8458
18	0.818492
19	0.788443
20	0.733161
21	0.721524
22	0.68925
23	0.676002
24	0.649048
25	0.631596

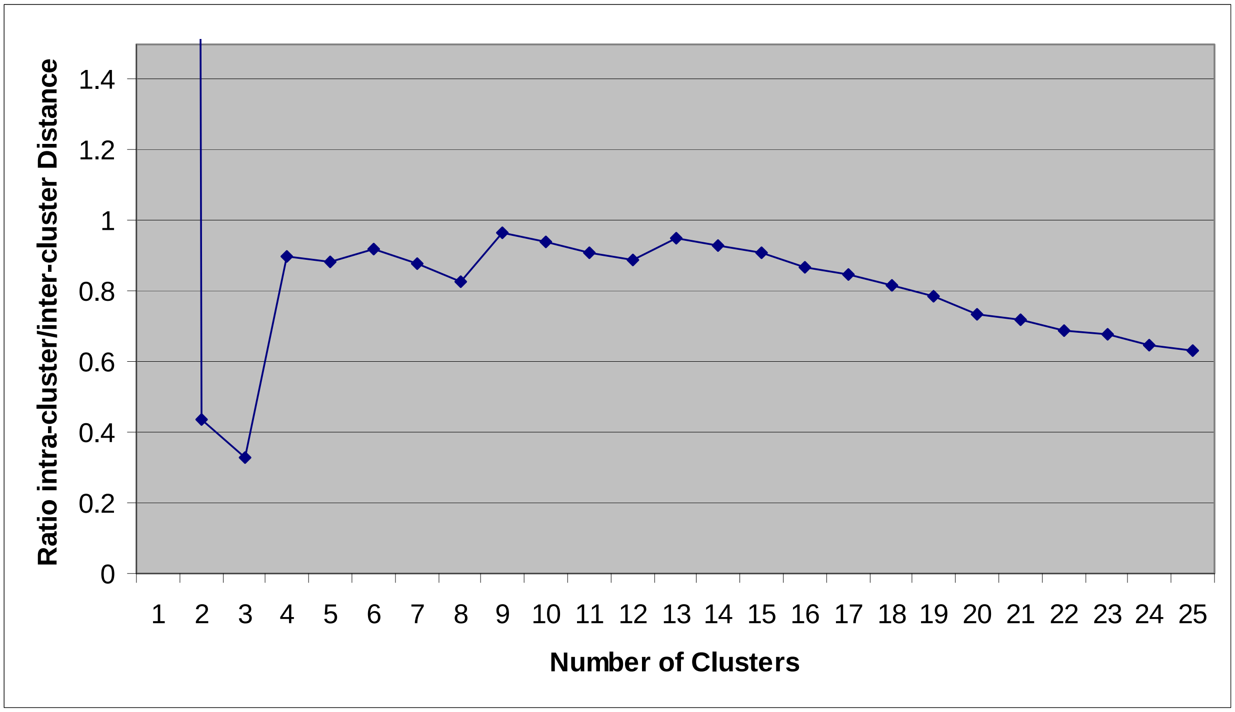

Figure 4.1 Ratio of intra-cluster to inter-cluster distance Vs. Number of Clusters

The graph in Figure 4.1 summarizes the result presented in table 8. It can be observed from Figure 4.1 that the local minimum of the value of ratio occurs when the number of clusters is 3. Also the value of ratio drops consistently if the number of clusters is increased to more than 13. When the number of cluster is 1 then the value of ratio is Infinity. The local minimum of the ratio gives us the optimal value of number of clusters, which in this case are 3. This results confirms that our methodology successfully identifies the optimal number of clusters since we used 3 VBMGs to generate the transition matrices.

4.1.2 Manufactured Data Experiment II

In the second experiment we evaluated how our algorithm worked with varying inter cluster distance between clusters. In this experiment we used 16 different VBMG and formed 15 pairs of VBMGs out of these 16 VBMGs. For each pair of VBMG we generated 100 transition count matrices, i.e. 50 transition count matrices for each VBMG in a pair. The VBMG pairs were built such that the distance between the two members of a pair of VBMG is in increasing/decreasing order. Table 9 presents the distance between pairs of VBMGs e.g. the distance between VBMG 1 and VBMG 2 is 0.1. Appendix A lists the transition probability matrices that represent these VBMGs.

Table 4.9 Euclidean Distance between pairs of VBMGs

VBMG Pair	Distance B/W Members
[1, 2]	0.1000000000000000
[1, 3]	0.1174734012447070
[1, 4]	0.1224744871391580
[1, 5]	0.1349073756323200
[1, 6]	0.1462873883832770
[1, 7]	0.1568438714135810
[1, 8]	0.1732050807568870
[1, 9]	0.1907878400283380
[1, 10]	0.2024845673131650
[1, 11]	0.2144761058952720
[1, 12]	0.2236067977499780
[1, 13]	0.2366431913239840
[1, 14]	0.2469817807045690
[1, 15]	0.2565151067676130
[1, 16]	0.2645751311064590

The clustering algorithm was run to cluster the 100 transition probability matrices obtained from one pair of VBMG. In a similar manner the clustering algorithm was run for each of the 15 pairs of VBMGs. After all clustering is completed we wanted to examine the number of errors in clustering. As explained earlier, the number of errors in a clustering is equal to the number of points which are present in the wrong cluster.

Table 10 presents the results obtained after clustering. The values are obtained after averaging over 20 runs. It can be noted that the average intra-cluster distance and the average inter-cluster distance obtained after clustering reflect the increasing distance between the seed matrices.

Table 4.10 Results obtained after clustering

VBMG Pair	Distance B/W Members (Seeds)	Avg Intra-Cluster Distance	Avg Inter-Cluster Distance	Avg Errors
[1, 2]	0.1000000000000000	0.1173417853864970	0.1306165431059130	27.4
[1, 3]	0.1174734012447070	0.1152712322108770	0.1470241440340000	24.6
[1, 4]	0.1224744871391580	0.1222311243955490	0.1444823644935920	18.55
[1, 5]	0.1349073756323200	0.1291719977787420	0.1462268291804200	14.9
[1, 6]	0.1462873883832770	0.1333777582229840	0.1523380612606260	10.4
[1, 7]	0.1568438714135810	0.1284557914216670	0.1659769005746650	11.95
[1, 8]	0.1732050807568870	0.1330555942345680	0.1812232573037140	7.2
[1, 9]	0.1907878400283380	0.1349446493513590	0.1962068171524650	3.5
[1, 10]	0.2024845673131650	0.1383658380791800	0.2050781634818430	0.35
[1, 11]	0.2144761058952720	0.1360055618666270	0.2182623709135990	0.45
[1, 12]	0.2236067977499780	0.1370377641333890	0.2271584924650070	0.1
[1, 13]	0.2366431913239840	0.1367644676545480	0.2352209545138670	0.05
[1, 14]	0.2469817807045690	0.1366657378938260	0.2503640052591860	0
[1, 15]	0.2565151067676130	0.1370522243571290	0.2574533750190250	0
[1, 16]	0.2645751311064590	0.1367462122661020	0.2701286571244540	0

As expected, it can be seen from Table 10 that as the inter-cluster distance between the clusters increases, the number of errors decreases. Figure 4.2 presents the graph for the data in Table 10. This information can be useful for researchers when they need to make decisions such as whether there is any benefit of building different algorithm animations for groups whose viewing behavior is not very different. For example if the distance between two clusters is very small then there may be no significant benefit in building different algorithm animations for those two groups.

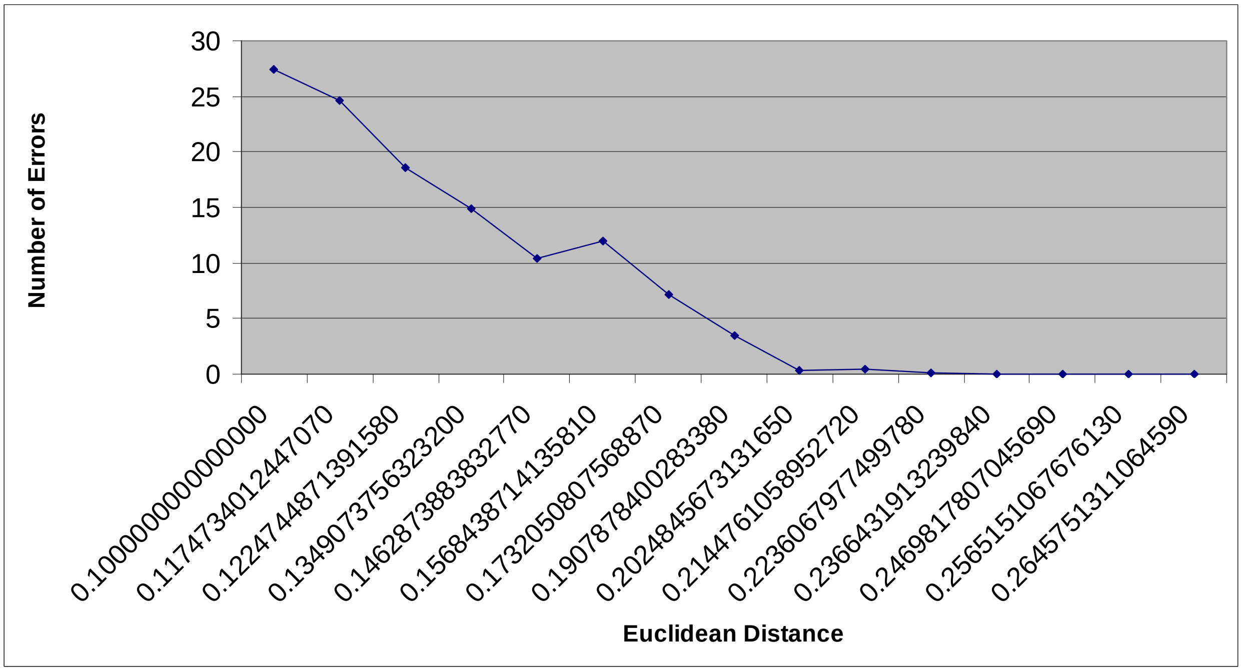

Figure 4.2 Number of Errors Vs. Distance between Seed Matrices

4.2 Experimental Data

As discussed in Chapter 3, we recruited 12 Human subjects from the CSCI 4800 – Human Computer Interaction class at The University of Georgia to record their eye-gaze sequences. All necessary permissions and consents were obtained. The subjects used the SSEA interface to study the Quick-sort algorithm. While the subjects were studying the algorithm, their eye movements were recorded using the ASL Eye-Trac 6000 System. The eye-gaze sequences obtained from the ASL system was used to construct transition count matrices and later transition probability matrices. For details refer to Chapter 3. We focused on only three areas of interest i.e. Animation area, Caption area, and Code area.

Out of the 12 subjects the data from 3 was discarded because of problems such as color blindness, failed calibration and head movement. More details of the experimentation can be found in [Kaldate07]. Appendix B lists the transition count matrices obtained from the ASL Eye-Trac for each subject.

Once we obtained the transition count matrices for the remaining 9 human subjects we used the varying number of clusters in our algorithm to analyze this data. This approach is similar to the one discussed in Section 4.2.1 used for analyzing manufactured data in Experiment-I. Table 11 presents the result of clustering and Figure 4.3 presents the graph.

Table 4.11 Results obtained after clustering

No. of Clusters	**Ratio** intra-cluster/inter-cluster distance
1	Infinity
2	0.5881871835998510
3	0.4957003798264610
4	0.5210362744481650
5	0.4147027386366350
6	0.3354295145703010
7	0.1865308564869910
8	0.1240756359227120

It can be observed from Figure 4.3 that the local minimum of ratio of intra-cluster to inter-cluster distance occurs at 3 clusters. Therefore we hypothesize that the viewing behavior of users can be categorized into three clusters. We aim to compare the results obtained here with the empirical experiments done in [Kaldate07].

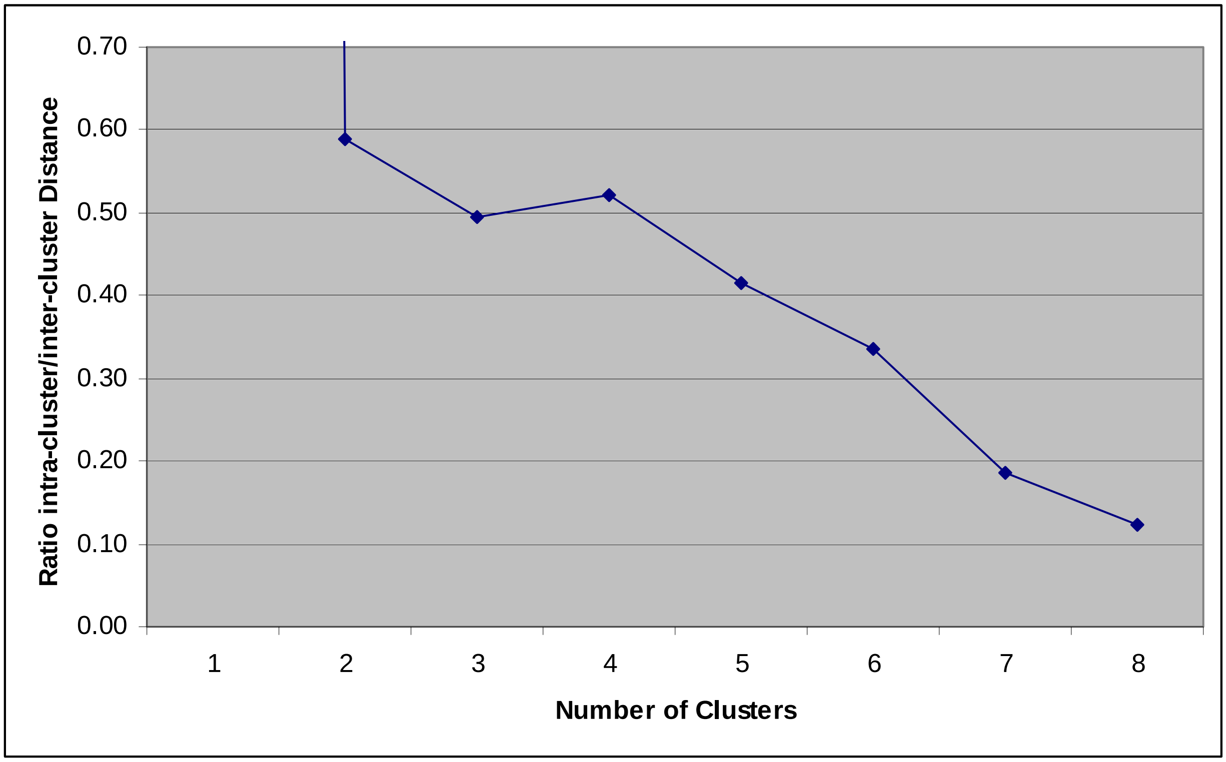

Figure 4.3 Ratio of intra-cluster to inter-cluster distance Vs. Number of Clusters

Once we know that we can group the viewing behavior in three groups, we build a VBMG for each of these three groups. The cluster centroid represents the VBMG for each cluster. The edges of VBMGs are labeled with the transition probabilities in the centroid's matrix. Figure 4.4, Figure 4.5, and Figure 4.6 represent VBMG for each of the cluster obtained from our experimental data. Appendix B lists the transition probability matrices for these VBMGs.

It can be seen from Figure 4.4 that subjects in Cluster 1 have high probability of remaining in the Animation and Code area but lower probability of remaining in Caption area. Cluster 1 was formed by subjects 2, 3, 4, 7, and 10.

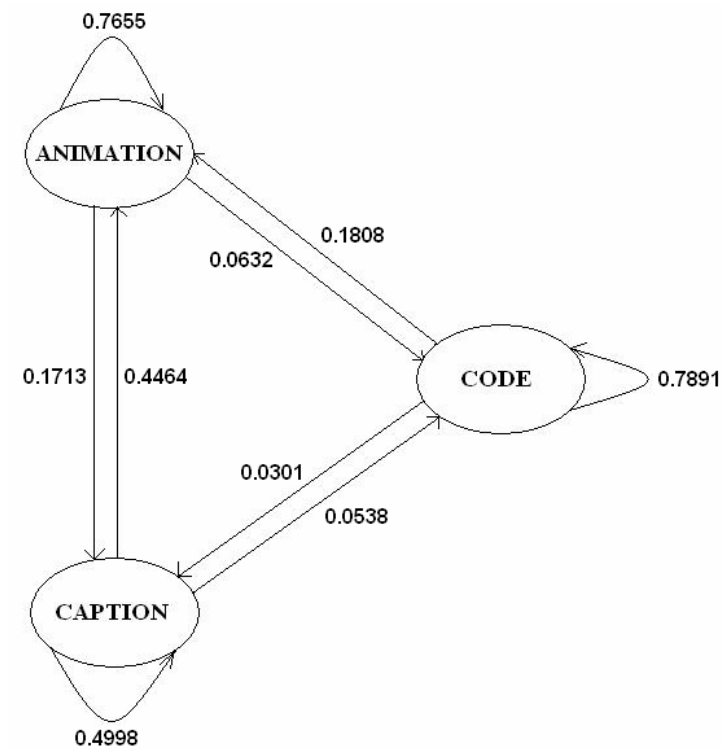

Figure 4.4 VBMG for Cluster 1

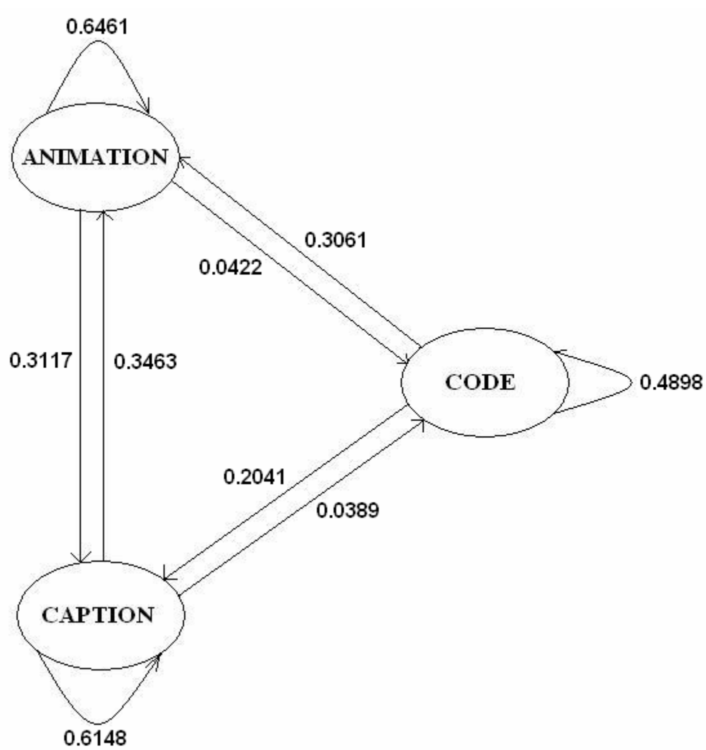

Figure 4.5 VBMG for Cluster 2

In Figure 4.5 it can be seen that subjects in Cluster 2 have a high probability of remaining in the Animation and Caption area but lower probability of remaining in Code area. Cluster 2 had only one member i.e. subject 9. Therefore, it appears that subjects in Cluster 1 focus less on Caption area while subjects in Cluster 2 read Code for lesser amount of time as compared to subjects in Cluster 1.

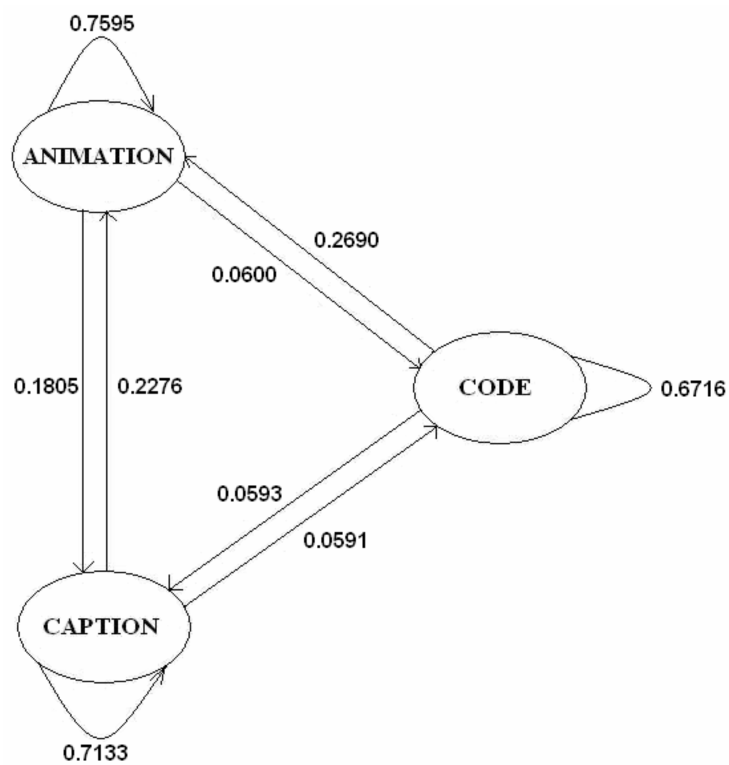

Figure 4.6 VBMG for Cluster 3

The chart in Figure 4.7 shows the percentage time spent by each user in the three AOIs. In the clustering algorithm the subjects 2, 3, 4, 7, and 10 were members on first cluster, subject 9 forms the second cluster and subjects 5, 6 and 11 form the third cluster.

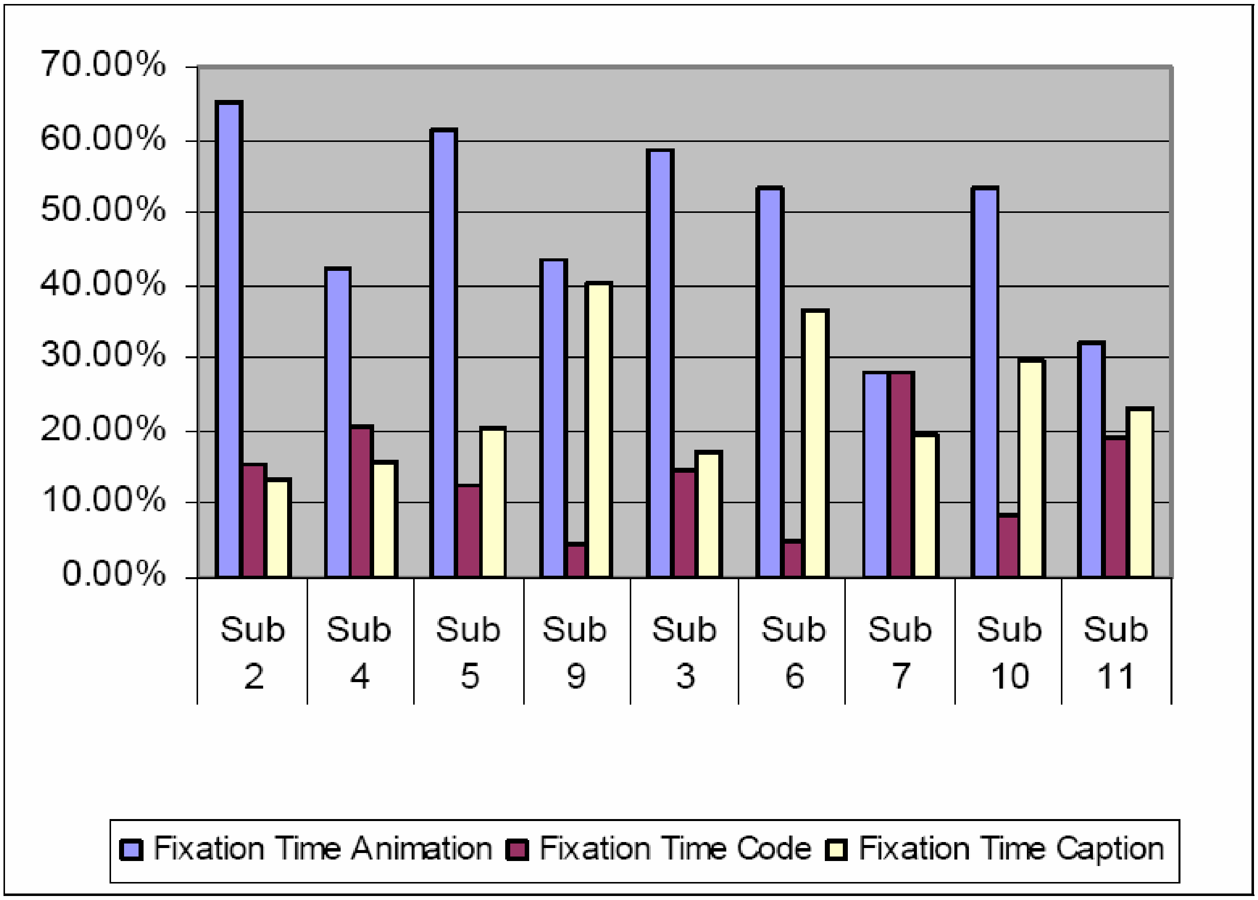

Figure 4.7 Percentage time in each AOI by each subject

4.3 Correlation of VBMGs with Cognitive study

We calculated the correlation of the distance of subjects from their cluster centroid with various assessment scores obtained from study of individual differences based on the preference of learning style, perceptual, attention and cognitive capabilities. For details on cognitive study refer to [Kaldate07]. Because cluster 2 had only 1 subject, we were not able to calculate the correlation of cluster 2 with the cognitive studies.

4.3.1 Correlation of VBMG with Viewing Behavior

We calculate a distance score for each point (subject) in a cluster using the formula (1 – distance-from-centroid). Thus the point closest to the centroid has a high score and a point further from the centroid has lower score.

The correlation of the distance score as defined above and fixation duration at various areas of interest is shown in Table 4.12. From Table 4.12 we can see that participants belonging to cluster 1 has moderate positive correlation with fixation duration at code (0.422), whereas participants belonging to cluster 3 had moderate negative correlation with fixation duration at code (-0.671). Cluster 1 has high negative correlation with fixation duration at caption (-0.73) and cluster 2 has high positive correlation with fixation duration at caption (0.982).

The participants in different clusters shows distinct viewing pattern in terms of code and captions. Participants in both the clusters have very low negative correlation with fixation at animation. Therefore we can say that there is not much of a difference in viewing pattern of animation for participants in both clusters. However, participants closely belonging to cluster 1 (closer to centriod of cluster 1) are likely to seek more information from code rather than from caption and participants closely belonging to cluster 2 are likely to seek more information from caption rather than from code.

Table 4.12: Correlation of distances of subjects from centroid with fixation duration at areas of interest

	Animation	Code	Caption
Cluster 1	-0.10742	0.422209	-0.73609
Cluster 3	-0.08874	-0.67166	0.982997

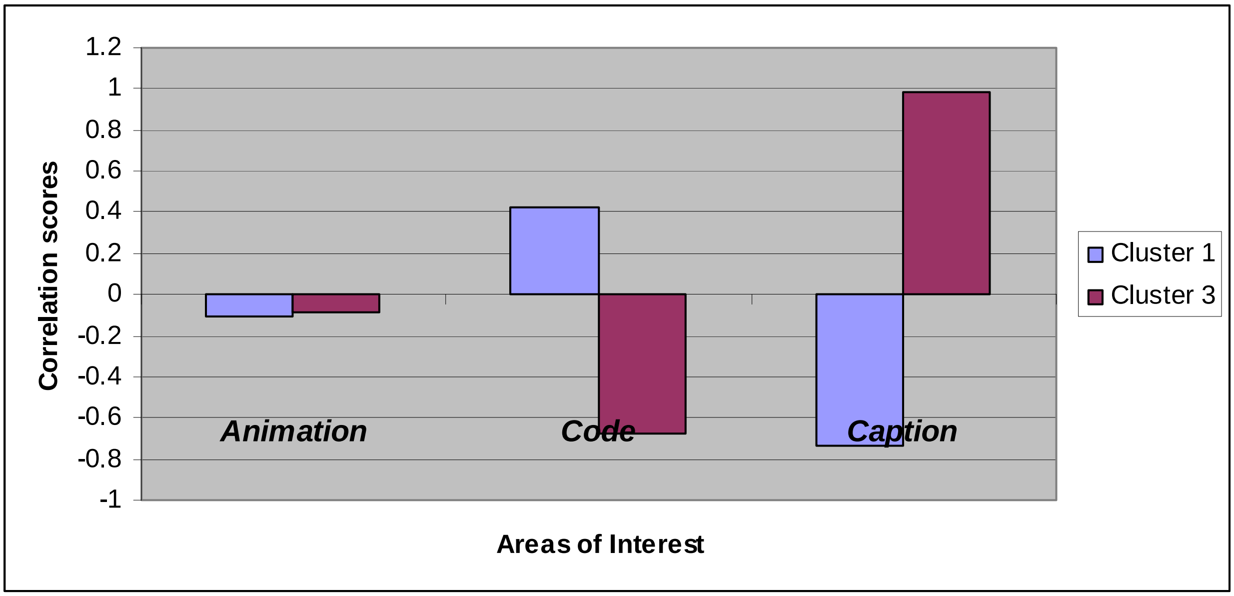

Figure 4.8: Correlation of distance of subjects from centroid with fixation duration

4.3.2 Correlation of VBMG with paper based assessments

The correlation of distance score with paper based assessments will give us insight on similarity of cognitive, attentional and perceptual capabilities of participants belonging to same cluster. Table 4.13 shows the correlation of distance from cluster centroid with scores on various paper based assessments. We see that cluster 1 has very low positive correlation for inference test (0.035) however cluster 3 shows very high positive correlation with this test (0.971). On figure classification test participants in cluster 1 shows low positive correlation (0.143) and participants in cluster 3 shows moderate negative correlation (-0.554). Cluster 1 has moderate positive correlation with surface development test (0.445) and cluster 3 has high negative correlation (-0.80). Size span test shows very low correlation for cluster 1 however, cluster 3 shows moderate negative correlation (-0.592).

Table 4.13: Correlation of paper-based test and cluster distance

	Inference Test - RL 3	Fig ClassI-3	Surface Deve Vz-3	Size Span Test
Cluster 1	0.035458822	0.143751145	0.44527259	0.008915988
Cluster 3	0.971901132	-0.554819482	-0.802034802	-0.59206198

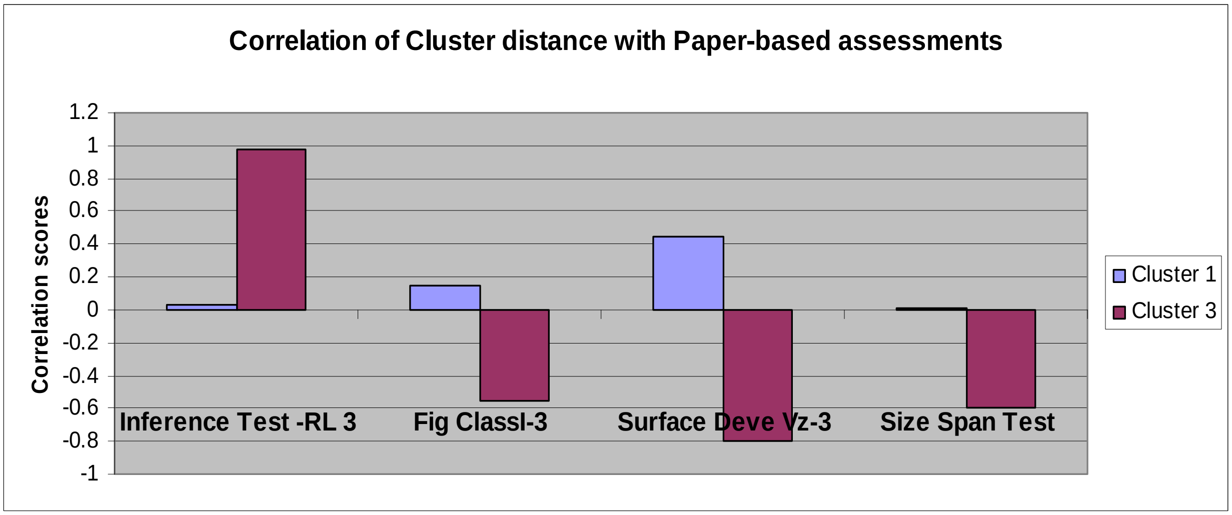

Figure 4.9: Correlation of cluster distance with scores on various paper based assessments

From the Figure 4.9 we see the trend that participants closer to cluster 3 are good at fluid intelligence but they show negative trend for figure classification, surface development and size span test. These users tended to seek information from the captions rather than code. Participants in cluster 1 did not have significant correlation for inference test and size span test, but they show a positive trend figure classification and surface development abilities. These users tended to seek information from the code rather than from captions.

4.3.3 Correlation of VBMG with computer based assessments

Correlation of distance scores and scores on computer based assessments are provided in Table 4.14. Cluster 1 shows low negative correlation (-0.20) and cluster 2 shows high positive correlation (0.91) with the reading span. Scores for symmetry span has low correlation for cluster 1 (0.30) and moderate correlation for cluster 3 (0.516). Cluster 1 (-0.664) and cluster 2 (-0.487) showed moderate negative correlation with operating span. Both group had low positive correlation on the Color Stroop test, cluster 3 (0.345) with slightly higher correlation than cluster 1(0.162).

Table 4.14: Correlation of cluster distance with computer based assessments

	R-Span	S- Span	O-Span	Color Stroop
Cluster 1	-0.20646	0.301437	-0.66419	0.1624415
Cluster 3	0.919095	0.516455	-0.48709	0.3453204

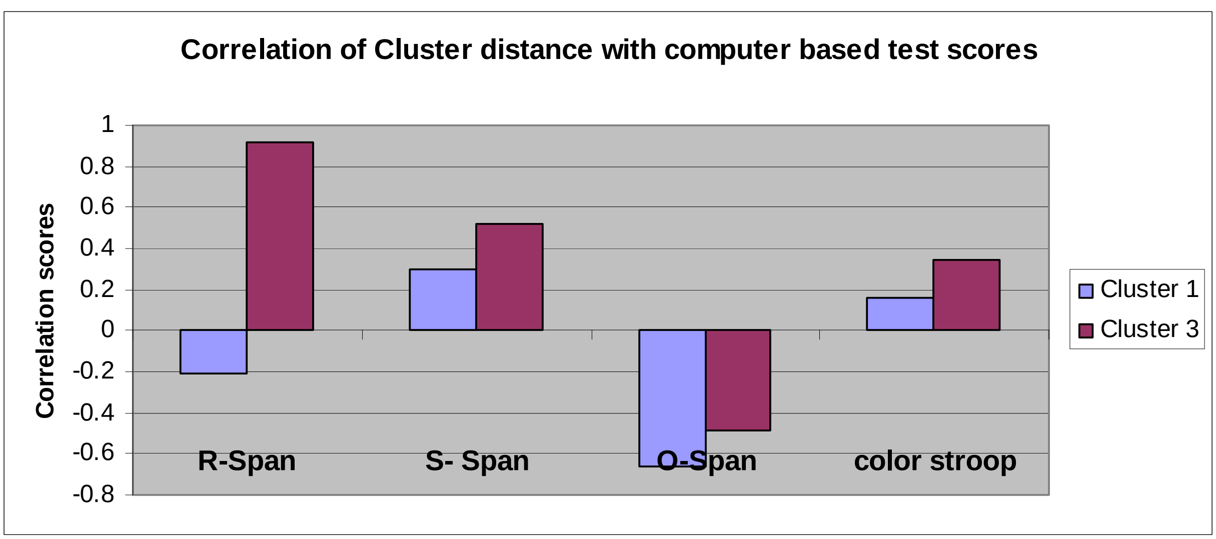

Figure 4.10: Correlation of cluster distance with computer based assessments

Cluster 1 and cluster 3 exhibit different patterns on reading span. Participants closer to the centroid of cluster 3 are likely to have better reading span. Both clusters shows a similar pattern on the symmetry span, the operating span and the Color Stroop test. Participants closer to centroid are likely to have better symmetry span and better able to handle interference due to color, whereas they are likely to have low score on the operating span test.

The clustering was based on the transition of fixation from one area of interest to another. The correlation of cluster distance with the various assessments shows whether participants with similar viewing pattern have other similarities. We observe that participants in cluster 3 have better fluid intelligence than participants in cluster 1, where as participants in cluster 1 have better visualization capability. The clusters did not show any difference in working memory capacity except for reading span. Participants in cluster 3 are likely to have better reading span than participants in cluster 1 and they also spent more time reading captions.

4.3.4 Correlation of VBMG with post-test scores

The correlation of distance scores and post-test scores are provided in Table 4.15. Cluster 1 shows low negative correlation on post-test (-0.128) and cluster 2 shows moderate positive correlation (0.483) with the post-test. On subset of post-test with questions similar to pre-test participants in cluster 1 showed a low negative correlation (-0.250) and participants in cluster 3 showed a high positive correlation. Participants in both the clusters have high positive correlation, cluster 3 showing higher correlation of (0.938) than cluster 1 (0.667). Figure 4.11 shows a bar chart for these correlations.

Table 4.15: Correlation of cluster distance with post-test scores and improvement from pre-test to post-test

	Post-test	Post-test Subset	Improvement
Cluster 1	-0.128965276	-0.250365993	0.667140135
Cluster 3	0.483362594	0.773752196	0.938484852

From Figure 4.11 we see that participants who are closer to cluster 3 tend to comprehend better from program visualization. However, the average score of participants in both clusters shows that participants in cluster 1 had better performance in post-test and had better improvement from post-test. Table 4.16 shows average scores of participants in these cluster for the pre-test, the post-test, the subset of post-test similar to pre-test and the improvement from pre-test to post-test. We see that participants belonging to cluster 1 tend to show better improvement than participants in cluster 3.

Table 4.16: Average scores of pre-test, post-test and improvement for both clusters

	Pre-test	Post-test	Post-test Subset	Improvement
Cluster 1	67.50%	72.50%	92.50%	25.00%
Cluster 3	83.33%	70.83%	66.67%	-16.67%

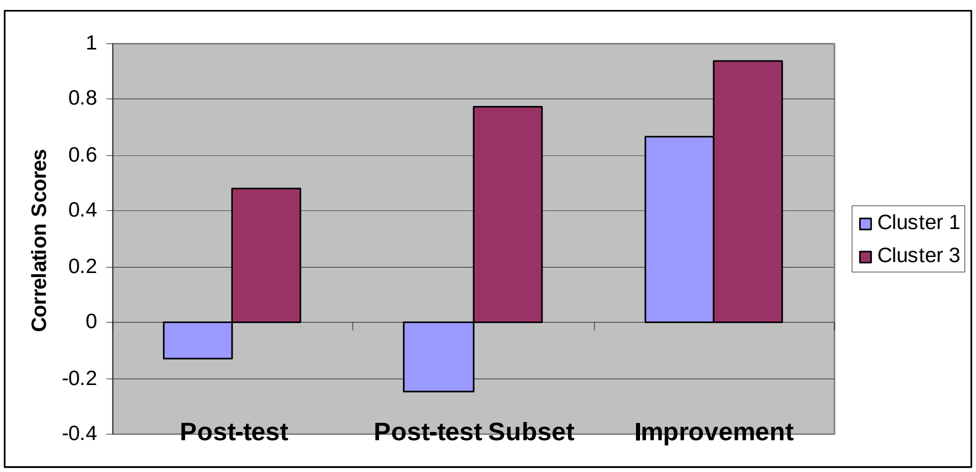

Figure 4.11: Correlation of cluster distance with post-test scores

CHAPTER 5

CONCLUSION

In this thesis we focused on understanding individual differences between viewing behaviors of users in program visualizations. We introduced a new approach to characterize viewing behavior of users of program visualizations. Our methodology of building Viewing Behavior Model Graphs (VBMG) to group viewing behavior of users of program visualizations into separate groups enables us to understand how different users view program visualization. A VBMG describes the gaze behavior, which is a sequence of areas of interests (AOI), for a user. Physically, a VBMG is a Transition Probability Matrix that stores the probabilities of transitions between different AOIs. Using the information available from VBMGs we can study how users within a group and in separate groups relate to each other in terms of their understanding of algorithms that are taught using program visualizations.

In our experiments the user's eye-fixation sequences are obtained using an Eye-Tracker. We then define AOIs for the program visualization interface and map the eye-fixation sequences to the AOIs. The mapping results in a sequence of AOIs for each user. We then use these AOI sequences to build VBMGs, physically represented by transition probability matrices, for each user. The per-user VBMGs are then clustered using a clustering algorithm to obtain VBMGs that represent a group of per-user VBMGs.

We presented both synthetic data and the experimental data collected from 12 Human subjects. We also showed how to select the optimal number of clusters using different metrics.

The results obtained from the experiments were convincing. Our experiments showed that the users can be grouped into seperate groups based on their viewing behavior. More convincing results can be obtained if the number of human subjects is large.

We believe that better understanding of individual differences in viewing behavior will enable us to create better program visualization systems. The level of abstraction problem is rightly addressed by this approach and the potential exists for the advanced program visualization systems will greatly benefit from this work. Also, this work will help us understand the relative importance of each type of information, namely animation, code, and captions, in program visualizations.

If a user follows the behavioral pattern of a previously identified group of users, then the algorithm animations can be adapted accordingly. One useful application of VBMG would be to dynamically capture viewing behavior and predict the cluster to which a user belongs, thus permitting on-the-fly adaptation of displays and other teaching materials.

VBMGs can find application in many areas. One of the applications is computer-based learning where the approach of tutorials can be adjusted based on the abilities and preferences of a particular student. As a student studies using the tutorial, the visualization system can automatically adjust based on the viewing behavior of the student and matching the behavior to the stored VBMGs. For example, if a user is focusing more on animation, then the visualization system can make the animation window bigger. Another application is the targeted advertising on websites [Johnston06, Chatterjee98, Montgomery03]. The VBMG can be built to analyze user viewing behavior while the user surfs the web-pages or the areas of web-pages. This helps advertisers to place the advertisements at the right location so that they attract user attention without being annoying. A method similar to VBMG, that of Customer Behavior Model Graphs has been applied for workload characterization of e-commerce websites.

In the future we will integrate this work with our other approaches in the study of program visualization. We also plan to recruit a larger number of Human Subjects and run the experiments again in hopes of obtaining a statistically significant result. We will compare the results we obtained in our experiments with the results obtained in the empirical study of algorithm animation systems [Kaldate07].

REFERENCES

[Baecker81] Baecker, R. and D. Sherman. “Sorting Out Sorting. 30 minute color sound film”. Dynamic Graphic Project, University of Toronto, 1981.

[Chatterjee98] Chatterjee, P., Hoffman, D. L. & Novak, T. P. “Modeling the Clickstream: Implications for Web-Based Advertising Efforts”. Working Paper. Vanderbilt University, 1998.

[Crosby95] Crosby, M. and Stelovsky, J. “From multimedia instruction to multimedia evaluation”. Journal of Educational Multimedia and Hypermedia, 4:147--162, 1995.

[Estes50] Estes, W.K. “Toward a statistical theory of learning”. Psychological Review, 57, 94-107, 1950.

[Everitt80] B. Everitt, “Cluster Analysis”, Halsted Press, New York, 1980.

[Freedman83] Freedman,D. “Markov Chains”, New York: Springer-Verlag, 1983.

[Grinstead] Grinstead, C. M. and Snell, J. L. “Introduction to Probability”, 2nd edition', American Mathematical Society, 2003.

[Gurka96] Gurka, J. and Citrin W., "Testing effectiveness of algorithm animation", Proceedings of the IEEE Symposium on Visual Languages, Boulder, CO, 1996.

[Hansen00] Hansen, S., Narayanan N. H. and Schrimpsher D. "Helping Learners Visualize and Comprehend Algorithms." Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 2000, 2(1).

[Hlavacs99] Hlavacs, H.; Kotsis, G , "Modeling user behavior: a layered approach". "MASCOTS'99", IEEE Computer Society, Los Alamitos, California 1999, pp. 218---225.

[Hundhausen02] Hundhausen, D. C., Douglas A.S. and Stasko T.J. "A Meta-Study of Algorithm Visualization Effectiveness". J. Visual Languages and Computing, 13: 259-290, 2002.

[Jagerman96] D. Jagerman, Melamed B., and Willinger W. "Stochastic modeling of traffic processes". In J. Dshalalow, editor, Frontiers in Queuing:Models, Methods and Problems. CRC Press, 1996.

[Johnston06] Johnston, P. Kristoff, N. Mcginness, H. Phuong Vu Wong, N. Wright, J. Scherer, W.T. Burkett, M.W., "Strategic Online Advertising: Modeling Internet User Behavior with Advertising.com", Systems and Information Engineering Design Symposium , IEEE 2006.

[Kaldate07] Kaldate, S. , "Program Visualization and Interaction with Individual Differences", MS Thesis, University of Georgia, 2007.

[Kemeny74] Kemeny J. G., Snell J. L., Thompson G. L., "Introduction to Finite Mathematics", 3rd ed., Englewood Cliffs, NJ: Prentice-Hall, 1974.

[Kraemer06] Kraemer E. T., Reed B., Rhodes P., and Taylor A., "SSEA: A system for studying the effectiveness of animations". In Proceedings of Fourth Program Visualization Workshop (PVW2006), University of Florence, Italy, June 29-30, 2006.

[Kullbach93] Kullbach S., Leibler R.A., “On Information Theory and Sufficiency” Annals of Mathematical Statistics, 1951.

[Lawrence93] Lawrence, A. "Empirical Studies of the Value of Algorithm Animation in Algorithm Understanding." PhD, Georgia Institute of Technology, 1993.

[Manavoglu03] Manavoglu E., Pavlov D., Giles C. L., "Probabilistic user behavior models," Third IEEE International Conference on Data Mining (ICDM 2003), 203-210, 2003.

[McCormack96] McCormack J., “Grammar based music composition”, Complex Systems, 1996.

[McQueen67] McQueen, J., “Some methods for classification and analysis of multivariate observations”. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 281–297, 1967.

[Menasce94] Menasce, D. A., Almeida V. A. F., Dowdy L. W., “Capacity Planning and Performance Modeling: from mainframes to client-server systems”, Prentice Hall, Upper Saddle River, 1994.

[Menasce99] Menasc´e D. A., Almeida V. A. F., Fonseca R., and Mendes M. A.. “A Methodology for Workload Characterization of E-commerce Sites”, Proc. 1999 ACM Conference on Electronic Commerce, Denver, CO, November, 1999.

[Miller52] Miller G.A., “Finite markov processes in psychology”, Psychometrika, Springer, 1952.

[Mongy] Mongy S., Djeraba C. and Simovici D., “On Clustering Users’ Behaviors in Video Sessions”, submitted.

[Montgomery03] Montgomery A L., Li S., Srinivasan K., and Liechty J.C. , "Modeling Online Browsing and Path Analysis Using Clickstream Data", Marketing Science, forthcoming. GSIA Working Paper #2003-E26., 2004.

[Mulholland98] Mulholland P. “A principled approach to the evaluation of SV: a case study in Prolog”. In: Software visualization: Programming as a multimedia experience (M. Brown, J. Domingue, B. Price, and J. Stasko, eds.) The MIT Press, Cambridge, MA, pp. 439-452, 1998.

[Nix92] Nix A. E. , Vose M. D., “Modeling genetic algorithms with Markov chains, Annals of Mathematics and Artificial Intelligence”, Springer, 1992.

[Ray99] Ray S. and Turi R. H. “Determination of number of clusters in K-means clustering and application in colour image segmentation”, invited paper, in N. R. Pal, A. K. De and J. Das (eds), Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT’99), Calcutta, India, 27-29 December, 1999, Narosa Publishing House, New Delhi, India, pp 137-143.

[Rhodes06a] Rhodes P., and Kraemer E., “Software visualization: Using perceptual and cognitive concepts to quantify quality, support instruction and improve interactions”. In Proceedings of Grace Hopper Celebration (GHC) of Women in Computing 2006, San Diego, CA, October 4-6, 2006.

[Rhodes06b] Rhodes P., Kraemer E., Taylor A. H. , Thomas S., Ross M., Davis E., Hailston K., and Main K., 2006. “VizEval: An experimental system for the study of program visualization quality”. In Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) 2006, Brighton, UK, September 4-8, 2006.

[Rhodes06c] Rhodes P., Kraemer E., and Reed B., “The importance of interactive questioning techniques in the comprehension of software visualizations”. In Proceedings of ACM Symposium on Software Visualization 2006 (SOFTVIS'06), Brighton, UK, September 4-5, 2006.

[Sokal89] Sokal A. D. “Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms”, Cours de Troisieme Cycle de la Physique en Suisse Romande,1989.

[Tufte90] Tufte E. R. , “Envisioning Information”, Graphic Press, 1990.

APPENDIX A

MANUFACTURED DATA

Below are the transition probability matrices that were used to evaluate our algorithm as explained in Section 4.2.2 Manufactured Data Experiment II. Each of these matrices represent a VBMG and their pairs were used as seed matrices to simulate gaze behavior as explained in Chapter 4.

0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2

Table A.1 VBMG Transition Probability Matrix 1

0.2	0.2	0.2	0.2	0.2
0.25	0.25	0.2	0.15	0.15
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.1000000000000000	

Table A.2 VBMG Transition Probability Matrix 2

0.2	0.2	0.2	0.2	0.2
0.3	0.2	0.18	0.17	0.15
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.1174734012447070	

Table A.3 VBMG Transition Probability Matrix 3

0.2	0.2	0.2	0.2	0.2
0.3	0.15	0.15	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.1224744871391580	

Table A.4 VBMG Transition Probability Matrix 4

0.3	0.15	0.15	0.2	0.2
0.15	0.21	0.21	0.21	0.22
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.1349073756323200	

Table A.5 VBMG Transition Probability Matrix 5

0.3	0.15	0.15	0.2	0.2
0.15	0.21	0.21	0.21	0.22
0.15	0.21	0.21	0.22	0.21
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.1462873883832770	

Table A.6 VBMG Transition Probability Matrix 6

0.3	0.15	0.15	0.2	0.2
0.15	0.21	0.21	0.21	0.22
0.13	0.23	0.21	0.22	0.21
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.1568438714135810	

Table A.7 VBMG Transition Probability Matrix 7

0.3	0.15	0.15	0.2	0.2
0.3	0.15	0.15	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.1732050807568870	

Table A.8 VBMG Transition Probability Matrix 8

0.3	0.15	0.15	0.2	0.2
0.15	0.21	0.21	0.22	0.21
0.3	0.15	0.15	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.16	0.24	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.1907878400283380	

Table A.9 VBMG Transition Probability Matrix 9

0.3	0.15	0.15	0.2	0.2
0.14	0.22	0.21	0.22	0.21
0.3	0.15	0.15	0.2	0.2
0.16	0.24	0.2	0.2	0.2
0.16	0.24	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.2024845673131650	

Table A.10 VBMG Transition Probability Matrix 10

0.3	0.15	0.15	0.2	0.2
0.14	0.22	0.21	0.22	0.21
0.3	0.15	0.15	0.2	0.2
0.16	0.24	0.2	0.16	0.24
0.15	0.25	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.2144761058952720	

Table A.11 VBMG Transition Probability Matrix 11

0.3	0.15	0.15	0.2	0.2
0.3	0.25	0.05	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.2236067977499780	

Table A.12 VBMG Transition Probability Matrix 12

0.3	0.15	0.15	0.15	0.25
0.14	0.22	0.21	0.22	0.21
0.3	0.15	0.15	0.15	0.25
0.16	0.24	0.2	0.16	0.24
0.15	0.25	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.2366431913239840	

Table A.13 VBMG Transition Probability Matrix 13

0.3	0.15	0.15	0.15	0.25
0.14	0.22	0.21	0.22	0.21
0.3	0.15	0.15	0.15	0.25
0.16	0.24	0.2	0.16	0.24
0.15	0.25	0.25	0.15	0.2
Euclidean Distance from VBMG 1 =			0.2469817807045690	

Table A.14 VBMG Transition Probability Matrix 14

0.3	0.15	0.15	0.15	0.25
0.14	0.22	0.21	0.22	0.21
0.3	0.15	0.15	0.15	0.25
0.16	0.24	0.2	0.16	0.24
0.15	0.25	0.25	0.12	0.23
Euclidean Distance from VBMG 1 =			0.2565151067676130	

Table A.15 VBMG Transition Probability Matrix 15

0.3	0.15	0.15	0.2	0.2
0.3	0.25	0.05	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.1	0.3	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
Euclidean Distance from VBMG 1 =			0.2645751311064590	

Table A.16 VBMG Transition Probability Matrix 16

APPENDIX B

EXPERIMENTAL DATA

The data obtained from human subjects is presented below. For subject 1, 8 and 12 the data was corrupted and therefore was not used.

	Animation	Instruction	Code
Animation	249	33	16
Instruction	31	49	0
Code	16	0	54

Table B.1 Transition Count Matrix for Subject 2

	Animation	Instruction	Code
Animation	265	47	29
Instruction	51	51	8
Code	32	6	58

Table B.2 Transition Count Matrix for Subject 3

	Animation	Instruction	Code
Animation	414	79	22
Instruction	83	90	12
Code	23	5	153

Table B.3 Transition Count Matrix for Subject 4

	Animation	Instruction	Code
Animation	584	60	49
Instruction	63	184	13
Code	45	12	82

Table B.4 Transition Count Matrix for Subject 5

	Animation	Instruction	Code
Animation	307	94	11
Instruction	93	297	5
Code	14	0	34

Table B.5 Transition Count Matrix for Subject 6

	Animation	Instruction	Code
Animation	180	58	28
Instruction	57	119	18
Code	36	13	247

Table B.6 Transition Count Matrix for Subject 7

	Animation	Instruction	Code
Animation	199	96	13
Instruction	98	174	11
Code	15	10	24

Table B.7 Transition Count Matrix for Subject 9

	Animation	Instruction	Code
Animation	276	89	11
Instruction	83	42	5
Code	17	3	162

Table B.8 Transition Count Matrix for Subject 10

	Animation	Instruction	Code
Animation	134	44	16
Instruction	25	83	14
Code	23	11	86

Table B.9 Transition Count Matrix for Subject 11

Following are the transition probability matrices of three clusters that were obtained after clustering data of the human subjects as discussed in Chapter 4. Each of the cluster matrices represents a VBMG for that group.

	Animation	Caption	Code
Animation	0.7654628694331660	0.1713426931501940	0.0631944374166389
Caption	0.4464121967472480	0.4998203670110880	0.0537674362416630
Code	0.1808009600274790	0.0301053489589401	0.7890936910135800
Represents Subject Number 2, 3, 4, 7, 10			

Table B.10 Transition Probability Matrix for Cluster 1

	Animation	Caption	Code
Animation	0.6461038961038960	0.3116883116883110	0.0422077922077922
Caption	0.3462897526501760	0.6148409893992930	0.0388692579505300
Code	0.3061224489795910	0.2040816326530610	0.4897959183673460
Represents Subject Number 9			

Table B.11 Transition Probability Matrix for Cluster 2

	Animation	Caption	Code
Animation	0.7595267077551130	0.1805131833657500	0.0599601088791360
Caption	0.2275562543564200	0.7133063035739940	0.0591374420695856
Code	0.2690247801758590	0.0593325339728217	0.6716426858513190
Represents Subject Number 5, 6, 11			

Table B.12 Transition Probability Matrix for Cluster 3