

2006

Membership Functions for a Fuzzy Relational Database: A Comparison of the Direct Rating and New Random Proportional Methods

Shweta Sanghi

Virginia Commonwealth University

Follow this and additional works at: <http://scholarscompass.vcu.edu/etd>

 Part of the [Computer Sciences Commons](#)

© The Author

Downloaded from

<http://scholarscompass.vcu.edu/etd/1366>

This Thesis is brought to you for free and open access by the Graduate School at VCU Scholars Compass. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of VCU Scholars Compass. For more information, please contact libcompass@vcu.edu.

**School of Engineering
Virginia Commonwealth University**

This is to certify that the thesis prepared by Shweta Sanghi entitled **Membership Functions for a Fuzzy Relational Database: A Comparison of the Direct Rating and New Random Proportional Methods** has been approved by her committee as satisfactory completion of the thesis requirement for the degree of Master of Science in Computer Science.

Lorraine M. Parker, Ph.D., Director of Thesis, School of Engineering

James E. Ames, IV, Ph.D., Committee Member, School of Engineering

Amita G. Chin, Ph.D., Committee Member, School of Business

David Primeaux, Ph.D., Interim Chairman, Computer Science, School of Engineering

L. Thomas Overby, Ph.D., Assistant Dean of Graduate Affairs, School of Engineering

Robert J. Mattauch, Ph.D., Dean, School of Engineering

F. Douglas Boudinot, Ph.D., Dean, School of Graduate Studies

April 17, 2006

Date

© Shweta Sanghi, 2006

All Rights Reserved

MEMBERSHIP FUNCTIONS FOR A FUZZY RELATIONAL DATABASE: A
COMPARISON OF THE DIRECT RATING AND NEW RANDOM PROPORTIONAL
METHODS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of
Science in Computer Science at Virginia Commonwealth University.

by

SHWETA SANGHI
B. E. Electronics & Communication Engineering
Guru Jambheshwar University, India, 1998

Director: Dr. Lorraine Parker
Associate Professor, Department of Computer Science

Virginia Commonwealth University
Richmond, Virginia
May, 2006

Acknowledgement

I would like to thank Dr Lorraine M. Parker for her constant support and guidance. Her encouragement gave me confidence to complete this thesis. Her questions and comments have helped improve the quality of this thesis.

I would like to thank my committee members, Dr. Ames and Dr. Chin, for their constructive input to this thesis. I would also like to thank all the faculty and staff of the Department of Computer Science at Virginia Commonwealth University for everything I have learned here.

Table of Contents

	Page
Acknowledgements.....	ii
List of Tables	vii
List of Figures.....	xiv
Chapter	
1 CHAPTER 1: INTRODUCTION.....	1
1.1 An Overview of Database Research Group	1
1.2 Overview of Project.....	2
2 CHAPTER 2: FUZZY RELATIONAL DATABASE.....	4
2.1 Need for Fuzzy Relational Database	4
2.2 Fuzzy Sets.....	5
2.3 Fuzzy Membership Function.....	6
2.4 Fuzzy Query Interface	8
3 CHAPTER 3: NATURAL LANGUAGE INTERFACE AND SPATIAL DATABASE	9
3.1 Natural Language Interface	9
3.2 Steps to Transform English Query into SQL	10
3.3 English Language Front-end (ELF)	11
3.4 Transformation of an English Query to SQL in ELF	12

3.5	Synonyms	12
3.6	Spatial Database	13
4	CHAPTER 4: OVERVIEW OF PREVIOUS RESEARCH WORK	15
4.1	Implementing a Fuzzy Relational Database.....	15
4.2	Initialization Methods.....	18
4.3	Compound Queries.....	20
4.4	Determination of Synonyms for Fuzzy Modifiers	23
4.5	Determine Membership Function Value by Direct Rating Method	26
4.6	Best Method to Store Synonyms	28
5	CHAPTER 5: INTERACTIONS, ISSUES AND EXPERIMENT OVERVIEW	33
5.1	Interactions between Different Versions of the Prototype	33
5.2	Issues	36
5.3	Experiment Overview.....	38
6	CHAPTER 6: DIRECT RATING EXPERIMENT	40
6.1	Experiment Setup	40
6.2	Training Session.....	41
6.3	Testing Session.....	44
7	CHAPTER 7: NEW RANDOM PROPORTIONAL EXPERIMENT	46
7.1	Experiment Setup.....	46

7.2 Initializing Membership Values	47
7.3 Querying the Database	47
7.4 Training Session	48
7.5 Analysis of Training Results	51
7.6 New Random Proportional with “Not” modifier Experiment.....	53
7.7 Testing Session.....	55
8 CHAPTER 8: RESULTS AND FUTURE WORK	57
8.1 Analysis of Testing Results for Blue Eye Color Attribute	57
8.2 Analysis of Testing Results for Green Eye Color Attribute.....	58
8.3 Analysis of Testing Result for Brown Eye Color Attribute	60
8.4 Results	61
8.5 Future Work	62
List of References	63
Appendices.....	65
A Initialization Weights for New Random Proportional with/without “Not” Modifier.....	65
B History of User Evaluation for New Random Proportional in the Training Phase	68
C History of User Evaluation for New Random Proportional with “Not” in the Training Phase	95

D	History of User Evaluation for Direct Rating in the Training Phase	109
E	History of User Verifications in the Testing Phase	136
F	Instructions for User Evaluation and User Verification	150
G	Stored Procedure to Initialize New Random Proportional Weights	162
H	Stored Procedure to Retrieve all Images for given Eye Color	168
I	Stored Procedure to Retrieve Images for New Random Proportional Method based on the Query	169
J	Stored Procedure to Retrieve Images for Direct Rating Method based on the Query	170
K	Stored Procedure to Retrieve Images for New Random Proportional Method with “Not” Modifier based on Query	171
L	Stored Procedure to Update Weight for New Random Proportional Method	172
M	Stored Procedure to Update Weights for Direct Rating Method	176
N	Stored Procedure to Update Weights for New Random Proportional Method with “Not” Modifier	178
O	Stored Procedure to Insert Weight for Images to maintain History	181
P	Stored Procedure to Update Weights for Test Data	182
Q	Comparison Charts	183
R	Fuzzy Relational Database Prototype Code	187

List of Tables

	Page
Table 1: Fuzzy Modifier Range Values.....	16
Table 2: Modifier Ranges and Mid Point.	24
Table 3: Interaction Matrix.	34
Table 4: Using Membership Values (Weights) for Each Attribute.	40
Table 5: Modifier Ranges and Midpoints for New Random Proportional.	46
Table 6: Statistical Analysis of Users' Satisfaction for Blue Color.....	58
Table 7: Statistical Analysis of Users' Satisfaction for Green Color.	59
Table 8: Statistical Analysis of Users' Satisfaction for Brown Color.	60
Table A.1: Initialization Values for Blue Eye Color.	65
Table A.2: Initialization Values for Green Eye Color.	66
Table A.3: Initialization Values for Brown Eye Color.....	67
Table B.1: History of New Random Proportional Weights for Image 1.	68
Table B.2: History of New Random Proportional Weights for Image 2.	69
Table B.3: History of New Random Proportional Weights for Image 3.	70
Table B.4: History of New Random Proportional Weights for Image 8.	71
Table B.5: History of New Random Proportional Weights for Image 9.	72
Table B.6: History of New Random Proportional Weights for Image 10.	73
Table B.7: History of New Random Proportional Weights for Image 11.	74

Table B.8: History of New Random Proportional Weights for Image 12.	75
Table B.9: History of New Random Proportional Weights for Image 13.	76
Table B.10: History of New Random Proportional Weights for Image 14.	77
Table B.11: History of New Random Proportional Weights for Image 15.	78
Table B.12: History of New Random Proportional Weights for Image 16.	79
Table B.13: History of New Random Proportional Weights for Image 17.	80
Table B.14: History of New Random Proportional Weights for Image 18.	81
Table B.15: History of New Random Proportional Weights for Image 19.	82
Table B.16: History of New Random Proportional Weights for Image 21.	83
Table B.17: History of New Random Proportional Weights for Image 23.	84
Table B.18: History of New Random Proportional Weights for Image 24.	85
Table B.19: History of New Random Proportional Weights for Image 31.	86
Table B.20: History of New Random Proportional Weights for Image 32.	87
Table B.21: History of New Random Proportional Weights for Image 33.	88
Table B.22: History of New Random Proportional Weights for Image 34.	89
Table B.23: History of New Random Proportional Weights for Image 35.	90
Table B.24: History of New Random Proportional Weights for Image 36.	91
Table B.25: History of New Random Proportional Weights for Image 37.	92
Table B.26: History of New Random Proportional Weights for Image 38.	93
Table B.27: History of New Random Proportional Weights for Image 40.	94

Table C.1: History of New Random Proportional with “Not” Weights for Image 1.....	95
Table C.2: History of New Random Proportional with “Not” Weights for Image 2.....	96
Table C.3: History of New Random Proportional with “Not” Weights for Image 3.....	96
Table C.4: History of New Random Proportional with “Not” Weights for Image 8.....	97
Table C.5: History of New Random Proportional with “Not” Weights for Image 9.....	97
Table C.6: History of New Random Proportional with “Not” Weights for Image 10.....	98
Table C.7: History of New Random Proportional with “Not” Weights for Image 11.....	98
Table C.8: History of New Random Proportional with “Not” Weights for Image 12.....	99
Table C.9: History of New Random Proportional with “Not” Weights for Image 13.....	99
Table C.10: History of New Random Proportional with “Not” Weights for Image 14...100	
Table C.11: History of New Random Proportional with “Not” Weights for Image 15...100	
Table C.12: History of New Random Proportional with “Not” Weights for Image 16...101	
Table C.13: History of New Random Proportional with “Not” Weights for Image 17...101	
Table C.14: History of New Random Proportional with “Not” Weights for Image 18...102	
Table C.15: History of New Random Proportional with “Not” Weights for Image 19...102	
Table C.16: History of New Random Proportional with “Not” Weights for Image 21...103	
Table C.17: History of New Random Proportional with “Not” Weights for Image 23...103	
Table C.18: History of New Random Proportional with “Not” Weights for Image 24...104	
Table C.19: History of New Random Proportional with “Not” Weights for Image 31...104	
Table C.20: History of New Random Proportional with “Not” Weights for Image 32...105	

Table C.21: History of New Random Proportional with “Not” Weights for Image 33...	105
Table C.22: History of New Random Proportional with “Not” Weights for Image 34...	106
Table C.23: History of New Random Proportional with “Not” Weights for Image 35...	106
Table C.24: History of New Random Proportional with “Not” Weights for Image 36...	107
Table C.25: History of New Random Proportional with “Not” Weights for Image 37...	107
Table C.26: History of New Random Proportional with “Not” Weights for Image 38...	108
Table C.27: History of New Random Proportional with “Not” Weights for Image 40...	108
Table D.1: History of Direct Rating Weights for Image 1.	109
Table D.2: History of Direct Rating Weights for Image 2.	110
Table D.3: History of Direct Rating Weights for Image 3.	111
Table D.4: History of Direct Rating Weights for Image 8.	112
Table D.5: History of Direct Rating Weights for Image 9.	113
Table D.6: History of Direct Rating Weights for Image 10.	114
Table D.7: History of Direct Rating Weights for Image 11.	115
Table D.8: History of Direct Rating Weights for Image 12.	116
Table D.9: History of Direct Rating Weights for Image 13.	117
Table D.10: History of Direct Rating Weights for Image 14.	118
Table D.11: History of Direct Rating Weights for Image 15.	119
Table D.12: History of Direct Rating Weights for Image 16.	120
Table D.13: History of Direct Rating Weights for Image 17.	121

Table D.14: History of Direct Rating Weights for Image 18.	122
Table D.15: History of Direct Rating Weights for Image 19.	123
Table D.16: History of Direct Rating Weights for Image 21.	124
Table D.17: History of Direct Rating Weights for Image 23.	125
Table D.18: History of Direct Rating Weights for Image 24.	126
Table D.19: History of Direct Rating Weights for Image 31.	127
Table D.20: History of Direct Rating Weights for Image 32.	128
Table D.21: History of Direct Rating Weights for Image 33.	129
Table D.22: History of Direct Rating Weights for Image 34.	130
Table D.23: History of Direct Rating Weights for Image 35.	131
Table D.24: History of Direct Rating Weights for Image 36.	132
Table D.25: History of Direct Rating Weights for Image 37.	133
Table D.26: History of Direct Rating Weights for Image 38.	134
Table D.27: History of Direct Rating Weights for Image 40.	135
Table E.1: History of User Verifications in the Testing Phase for Image 1.	136
Table E.2: History of User Verifications in the Testing Phase for Image 2.	137
Table E.3: History of User Verifications in the Testing Phase for Image 3.	137
Table E.4: History of User Verifications in the Testing Phase for Image 8.	138
Table E.5: History of User Verifications in the Testing Phase for Image 9.	138
Table E.6: History of User Verifications in the Testing Phase for Image 10.	139

Table E.7: History of User Verifications in the Testing Phase for Image 11.	139
Table E.8: History of User Verifications in the Testing Phase for Image 12.	140
Table E.9: History of User Verifications in the Testing Phase for Image 13.	140
Table E.10: History of User Verifications in the Testing Phase for Image 14.	141
Table E.11: History of User Verifications in the Testing Phase for Image 15.	141
Table E.12: History of User Verifications in the Testing Phase for Image 16.	142
Table E.13: History of User Verifications in the Testing Phase for Image 17.	142
Table E.14: History of User Verifications in the Testing Phase for Image 18.	143
Table E.15: History of User Verifications in the Testing Phase for Image 19.	143
Table E.16: History of User Verifications in the Testing Phase for Image 21.	144
Table E.17: History of User Verifications in the Testing Phase for Image 23.	144
Table E.18: History of User Verifications in the Testing Phase for Image 24.	145
Table E.19: History of User Verifications in the Testing Phase for Image 31.	145
Table E.20: History of User Verifications in the Testing Phase for Image 32.	146
Table E.21: History of User Verifications in the Testing Phase for Image 33.	146
Table E.22: History of User Verifications in the Testing Phase for Image 34.	147
Table E.23: History of User Verifications in the Testing Phase for Image 35.	147
Table E.24: History of User Verifications in the Testing Phase for Image 36.	148
Table E.25: History of User Verifications in the Testing Phase for Image 37.	148
Table E.26: History of User Verifications in the Testing Phase for Image 38.	149

Table E.27: History of User Verifications in the Testing Phase for Image 40.149

List of Figures

	Page
Figure 1: Hierarchy of Research Work.....	3
Figure 2: Membership Function for Crisp Set.....	5
Figure 3: Membership Function for Fuzzy Set.....	6
Figure 4: Fuzzy Membership Function Young Definition	7
Figure 5: Graphical Representation of Membership Function Young.....	7
Figure 6: Query Form	22
Figure 7: Users Feedback for Image 3 for Direct Rating Method	43
Figure 8: Membership Weights for Direct Rating Method.....	44
Figure 9: Percentage of Satisfaction for Direct Rating Method	45
Figure 10: Membership Weights for New Random Proportional Method	50
Figure 11: Direct Rating (No Blue) vs. New Random Proportional.....	52
Figure 12: Direct Rating (Slightly Blue) vs. New Random Proportional.....	53
Figure 13: Direct Rating (Medium Blue) vs. New Random Proportional.....	53
Figure 14: Membership Weights for New Random Proportional Method with Not	55
Figure 15: Percentage of Satisfaction for New Random Proportional	56
Figure 16: Comparison of User Satisfaction for Blue Color	57
Figure 17: Comparison of User Satisfaction for Green Color	59
Figure 18: Comparison of User Satisfaction for Brown Color	60

Figure 19: Membership Weight Comparison	62
Figure F.1: Instructions to Train the Direct Rating Method	150
Figure F.2: Instructions to Train the New Random Proportional Method.....	153
Figure F.3: Instructions to Verify the Direct Rating Method	155
Figure F.4: Instructions to Verify the New Random Proportional Method.....	157
Figure F.5: Instructions to Train the New Random Proportional Method with “Not”	159
Figure Q.1: Direct Rating (No Green) vs. New Random Proportional	183
Figure Q.2: Direct Rating (Slightly Green) vs. New Random Proportional.....	184
Figure Q.3: Direct Rating (Medium Green) vs. New Random Proportional.....	184
Figure Q.4: Direct Rating (No Brown) vs. New Random Proportional	185
Figure Q.5: Direct Rating (Medium Brown) vs. New Random Proportional.....	185
Figure Q.6: Direct Rating (Very Brown) vs. New Random Proportional	186

Abstract

MEMBERSHIP FUNCTIONS FOR A FUZZY RELATIONAL DATABASE: A COMPARISON OF THE DIRECT RATING AND NEW RANDOM PROPORTIONAL METHODS

By Shweta Sanghi, Bachelor in Engineering in Electronics & Communication

A thesis submitted in partial fulfillment of the requirements for the degree of Master of
Science in Computer Science at Virginia Commonwealth University.

Virginia Commonwealth University, 2006

Major Director: Dr. Lorraine M. Parker
Associate Professor of Computer Science

Fuzzy relational databases deal with imprecise data or fuzzy information in a relational database. The purpose of this fuzzy database implementation is to retrieve images by using fuzzy queries whose common-language descriptions are defined by the consensus of a particular user community. The fuzzy set, which is presentation of fuzzy attribute values of the images, is determined through membership function. This paper compares two methods of constructing membership functions, the Direct Rating and New Random Proportional, to determine which method gives maximum users satisfaction with minimum feedback from the community. The statistical analysis of results suggests the use of Direct Rating method. Moreover, the analysis shows that the performance of the New Random Proportional

method can be improved with the inclusion of a “Not” modifier. This paper also identifies and analyzes issues that are raised by different versions of the database system.

CHAPTER 1 Introduction

1.1 An Overview of Database Research Group

The Database Research Group, at Virginia Commonwealth University, aims to use natural language queries to access a spatial database to retrieve facial images that are described using subjective terms. Currently, there are three subgroups.

- 1. Natural Language:** Natural language processing allows database queries using everyday language without directly using standard query languages such as SQL. A natural language interface maintains its own dictionary containing terms associated with the relations and their relationships as well as a standard language dictionary. A natural language interface refers to words in its own dictionary as well as to the words in the standard dictionary, in order to interpret a query. If the interpretation is successful, the interface generates a SQL query corresponding to the natural language request and submits it to the database management system for processing; otherwise, a dialogue is started with the user to clarify the request [7].
- 2. Fuzzy Database:** Fuzzy relational database theory extends the relational model to allow for the representation of imprecise data. Thus, it provides a more accurate representation of the world that it models. In this project, images are

described with subjective attributes. By incorporating user feedback, the descriptions of the attributes are adapted so that the database appropriately represents the consensus of a particular user community. The application takes a SQL query from the Natural Language Interface along with a list of subjective terms and produces a SQL query that uses fuzzy data stored about the images in the database [1].

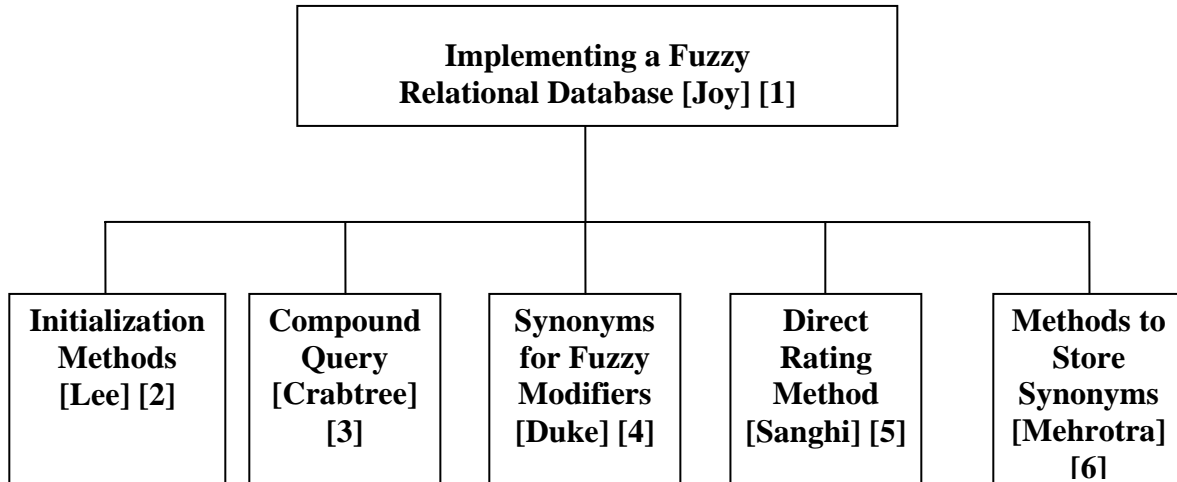
- 3. Spatial Database:** A spatial database stores the same data as an image database, but in a different format. It uses geometric primitives (point, line, polygon, etc.), instead of a collection of pixels, to store the data. A spatial database offers the advantage of being able to enhance the capabilities of database systems to allow for the preservation and retrieval of the various geometric properties of the images [8]. Methods of representing facial images spatially are being investigated [18].

1.2 Overview of Project

Recognizing the variations inherent in different user communities, Smita Dattatri and Karen Joy [1] investigated the use of community defined values in a fuzzy database with a natural language interface. In their work, a prototype was implemented to allow a user community to determine the inclusion weights of the attribute values describing a set of images. Their research was extended further by Lee [2], Crabtree [3], Duke [4], Sanghi [5]

and Mehrotra [6] to find solutions to some of the issues generated by the original prototype as shown in Figure 1. An overview of different versions of the system is given in chapter 4.

Figure 1: Hierarchy of Research Work



In order to integrate these different versions of the system, this paper identifies and analyzes the issues that are raised by each paper. This involves identifying conflicts between the versions. This paper also tries to resolve the issue of whether the New Random Proportional method or the Direct Rating method is the best method of constructing membership functions by achieving maximum users' satisfaction with minimum feedback from the community.

CHAPTER 2 Fuzzy Relational Database

2.1 Need for Fuzzy Relational Database

Most conventional databases are based on the relational model. Values in a relation are taken from a finite set of strictly defined domain values. Each relation in the database represents a proposition and each record in a relation is a statement such that it evaluates to ‘true’ for that proposition ([9], [10]). It could be argued, however, that this required precision actually gives an insufficient representation of the world. As such, the conventional relational database model has limited usefulness.

One area that illustrates this limitation is in everyday, subjective language used to describe people. For instance, a person might be described as being “tall, with a wide face and very dark brown eyes”. This description would be difficult to represent under the conventional relational model both because it uses descriptive words that are inherently imprecise, and also because differing communities (i.e., groups with internal agreement on subjective meanings of these terms) may describe the same person differently. As an approach, fuzzy database systems, which are able to represent and manipulate imprecise information, have been presented [11]. In these systems, imprecise information can be stored using *fuzzy linguistic terms* (e.g. young, big) [12]. Fuzzy relational database theory extends the relational model to allow for the representation of imprecise data and thus, it provides a more accurate representation of the world that it models.

2.2 Fuzzy Sets

The concept of fuzziness as described by Zadeh [12] includes imprecision, uncertainty, and degrees of truthfulness of values. Linguistic variables are used for system input and output, and are represented by words such as “size”, “age” and “temperature”. A fuzzy set is created to describe the linguistic variables in more detail. The linguistic variable “age”, for instance, may have categories (members) of “young”, “very young”, “middle age”, “old”, and “very old”. Once these categories or members are defined, the fuzzy set is obtained, and a membership function is then developed for each member in the set.

Fuzzy sets are derived by generalizing the concept of set theory. Fuzzy sets can be thought of as an extension of classical sets. In a classical set (or crisp set), the objects in the set are called elements or members of the set. An element x belonging to a set A is defined as $x \in A$, an element that is not a member in A is noted as $x \notin A$. A characteristic function or membership function $\mu_A(x)$ is defined as an element in the universe U having a crisp value of 1 or 0. For every $x \in U$,

Figure 2: Membership Function for Crisp Set

$$\mu_A(x) = \begin{cases} 1 & \text{for } x \in A, \\ 0 & \text{for } x \notin A. \end{cases}$$

This can also be expressed as $\mu_A(x) \in \{0,1\}$

For fuzzy sets, the membership function takes values in the interval $[0, 1]$. The range between 0 and 1 is referred to as the membership grade or degree of membership. A fuzzy set A is defined as:

Figure 3: Membership Function for Fuzzy Set

$$A = \{(x, \mu_A(x)) \mid x \in A, \mu_A(x) \in [0, 1]\}$$

Fuzzy set theory has equivalent operations to those of crisp set theory. It includes functions such as equality, union and intersection [12]. Integrity constraints, such as functional dependency, multivalued dependency, join dependency, etc, have been identified and sets of complete inferences rules for such dependencies have been proposed [13] for fuzzy sets.

2.3 Fuzzy Membership Function

The membership function is often conceptualized as an indicator of the degree of possibility that an element has a particular membership value. That is, possible values that an element can assume from a possibility distribution. The membership function is depicted graphically and simple membership functions can be modeled using a triangle or trapezoid.

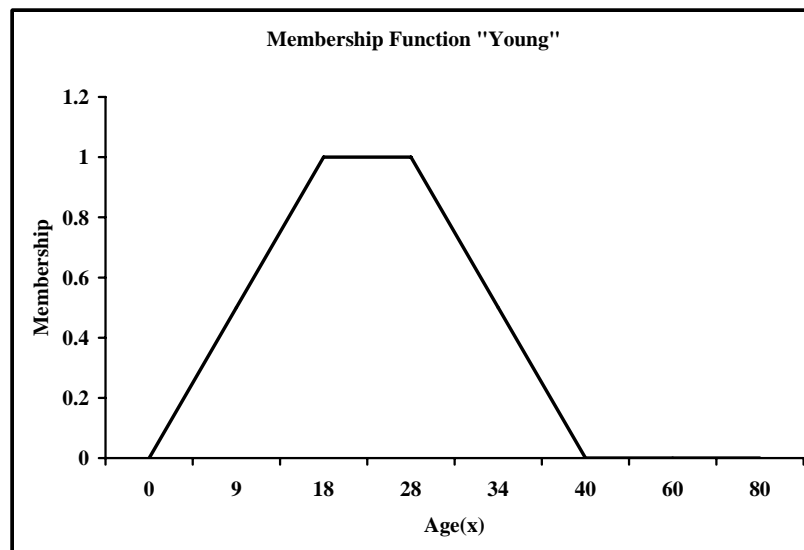
For example, the membership function of linguistic variable “young” is defined as:

Figure 4: Fuzzy Membership Function Young Definition

$$\text{young}(x) = \begin{cases} 0, & \text{if } \text{age}(x) = 0 \text{ or } \text{age}(x) \geq 40 \\ 0.010 - 0.99, & \text{if } 0 < \text{age}(x) < 18 \text{ or } 28 < \text{age}(x) < 40 \\ 1.0, & \text{if } 18 \leq \text{age}(x) \leq 28 \end{cases}$$

young’s membership function is shown graphically in Figure 5.

Figure 5: Graphic Representation of Membership Function Young



For a person with a high inclusion value, there is a high confidence that they belong to the class of young people. Conversely, people with a low inclusion value in the class young, (say from .5 to 0) shows a low confidence that they are young. For a person that is 34 years old, their corresponding young fuzzy value might be .0.5, signifying that there is some degree of youthfulness as it relates to that specific person.

Semantically, probability theory and fuzzy logic use different notions: Probability and Degree of Membership. Probabilities are the likelihoods that an event does or does not occur. Fuzzy Logic models the extent to which an event occurred or can occur. Even though fuzzy membership values and probabilities values operate over the same functional range $[0, 1]$, the semantic difference between a fuzzy statement and probabilistic statement is significant.

2.4 Fuzzy Query Interface

Bosc and Pivert [14] discussed the architecture of a relational database system required to process fuzzy queries. They suggested that there are two principle ways to accomplish this:

- 1.** Use regular DBMS architecture and develop an additional layer that functions as a user interface;
- 2.** Build a completely new system that includes fuzzy query processing techniques in the kernel.

If the first method is selected then mechanisms such as stored procedures may be used in an additional interface layer and by taking advantage of the implementation mechanisms handled by the DBMS, enabling it to reach acceptable performance levels. The fuzzy relational database developed by the Database Research Group extends a relational database by adding fuzzy logic, fuzzy data and membership functions.

CHAPTER 3 Natural Language Interface and Spatial Database

3.1 Natural Language Interface

Asking questions of a database in a natural language is a very convenient and easy method of data access, especially for casual users who do not understand database query languages such as SQL. Thus the purpose of a Natural Language Interface (NLI) for a database system is to accept requests in English from a user, attempt to “understand” the requests, and then translate them into SQL.

A natural language interface has its own private dictionary [7]. This dictionary contains words related to a database and its relationships. In addition to this, the interface also maintains a standard dictionary (e.g. Webster’s dictionary). The standard approach to database Natural Language Processing (NLP) systems is well established [15]. This approach creates a ‘semantic grammar’ for each database, and uses this to parse the English question. The semantic grammar creates a representation of the semantics of a sentence. If the interpretation is successful, a database query can be generated in SQL after some analysis of the semantic representation; otherwise, a dialogue is started with the user to clarify the request.

The need for a NLI to databases has become increasingly important as more and more people access information through web browsers, PDA’s and cell phones. These people are

casual users and it is necessary to have a way that they can make queries in their own natural language rather than to learn and write SQL queries. But the important point is that NLI's are only usable if they map natural language questions to SQL queries correctly.

3.2 Steps to Transform English Query into SQL

There are some basic steps needed to transform a given English query to an equivalent SQL form. The workings of all natural language to SQL software packages deal with these basic steps in some manner.

First there is a dictionary, where all the words that are expected to be used in any English question are declared. These words consist of all the elements (relations, attributes and values) of the database and their synonyms. Then these words are mapped to the database system. This implies that the meaning of each word needs to be defined. They may be called by different names in different systems but these two (i.e. definition and mapping of the words) form the basis of the conversion.

There are two dictionaries used in transformation. The lexical dictionary lists the root words with their synonyms. It generally defines all the words that may occur in a question. The first step of any system is to find all the words of a question and match them with words in the lexical dictionary.

The semantic dictionary contains mappings. The extracted words from previous step are then mapped to the database using the semantic dictionary. After this process, the English question is transformed to an internal language which is then converted to SQL in the next step.

During the mapping process words are attached to each other or to the entities or to the relations. So the output of this step is a function. For example, consider the question “What is the age of each employee?” Here the attributes, age and employee are attached and so the output is the function has_age (age, employee).

The next step is the conversion of the internal query into an equivalent SQL statement. This is accomplished differently by different systems. The interface, English Language Front-end, uses one approach to get a SQL statement from an English query.

3.3 English Language Front-end (ELF)

ELF is a commercial system that generates Natural Language Processing System for a database [7]. It is developed by ELF Software Co. ELF Access is an interface which works with Microsoft Access.

The lexicon is automatically built in ELF. In other words, ELF takes an existing database and scans through it so that it can understand both the data and the relationships that are set up. Words related to attributes and relationships of the database are stored into the lexicon

dictionary. During the analysis process, ELF Access examines the terms used in defining fields and tables, and uses its built-in dictionary to try to predict what kinds of synonyms might be used in queries. It also stores its type and which table it comes from (builds the Semantic dictionary).

3.4 Transformation of an English Query to SQL in ELF

After the lexicon and semantic dictionary is built, the database system is ready to answer an English query. ELF does this in three steps.

1. The first step is to parse the English question and find the words that are stored in the lexicon.
2. The second step is to find tables for the extracted words. This information is stored in the lexicon.
3. After getting the tables, the last step involves joining these tables on the common attributes.

Once the interpretation is successful, the SQL is generated.

3.5 Synonyms

In Linguistic terms, two expressions are synonymous in a linguistic context L if the substitution of one for the other in L does not alter the truth-value. In a relational database, a select query represents a Boolean function [10]. The result of a select query is the set of tuples with a truth-value of TRUE for the predicate. Hence two queries that return the same relation are synonyms in the context of the base relations. In a fuzzy relational database, an attribute represents a range of membership weights. Hence an attribute that

returns tuples in the same range as another attribute can be considered its synonym. The range of membership weights is subjective in that one modifier's range can be a subset of another's and still be considered a synonym. In the context of a fuzzy relational database with community-defined membership values, the community, logically, must make the designation of synonym.

3.6 Spatial Database

Electronically storing information on a computer reduces the information to a collection of binary bits. An electronic system must be able to reconstruct the information about an item such that a user will understand not only the textural description of an object, but, also the physical description of the object. Spatial databases are an attempt to enhance the capabilities of electronic database systems to allow for the preservation and retrieval of the physical interrelationships of objects in space. Spatial data is comprised of objects in multi-dimensional space. Spatial data is large and voluminous with respect to storage and information.

Spatial Database Management Systems (SDBMS) provide the capabilities of a traditional DBMS while allowing special storage and handling of spatial data. A SDBMS can then be defined as a software model that can work within an underlying DBMS, support multiple spatial data models and spatial abstract data types, a query language to manage these ADT's and support for spatial indexing, algorithms for spatial operations, and domain

specific rules for query optimization. SDBMS work with a spatial application at the front end and a DBMS at the back end.

SDBMS have three layers:

- Interface to spatial application
- Core spatial functionality
- Interface to DBMS

The SDMS is explored to find out the capabilities of the SDBMS to store facial characterization data for the purpose of searching the database with fuzzy queries. Therefore the task is to represent the human face as a collection of geometric shapes using SDBMS that will capture all the information for the fuzzy queries. Currently, an investigation is going on to evaluate the performance of SDBMS over non-SDBMS for the fuzzy application [18].

CHAPTER 4 Overview of Previous Research Work

One of the goals of this project is to create a new system by combining different versions of the existing system. This chapter reviews the database research done by other students at VCU.

4.1 Implementing a Fuzzy Relational Database

The original application Joy [1] was implemented using VB.NET and SQL Server 2000. The fuzzy relational database allowed a user to query and retrieve facial images based on two independent factors: Eye Color, and Shape of the Face. For each factor, there were three fuzzy attributes. For the Eye Color these attributes were: Blue, Green, and Brown. For the Shape of the Face these attributes were: Average, Broad, and Wide. In the prototype implementation, three attribute modifiers which quantify the degree of membership were selected: Very, Medium, and Slightly. The attribute modifiers were given numerical ranges as shown in Table 1.

Every attribute of the images in the database was initially assigned a random membership value between 0 and 1. Each modifier utilized a threshold value that was designed to ‘stabilize’ each attribute’s membership value within the modifier’s range to represent most accurately that community’s consensus.

Table 1: Fuzzy Modifier Range Values

Modifier Name	High Value	Low Value	Threshold Value
Very (V)	1.00	0.75	0.87
Medium (M)	0.74	0.30	0.52
Slightly (S)	0.29	0.1	0.20

The stabilizing property was implemented as follows. When users' viewed images that currently match their criteria, they provided feedback indicating either that the image met their criteria, or that they believed the image could be better defined using a stronger or a weaker modifier. If the user chose the former then the attribute's membership value were moved (0.01) toward the threshold value of that modifier's range. By adjusting the membership weight so that it was more deeply within the modifier's range, the community opinion was strengthened with concurring feedback. However, if the user chose the latter then the attribute's weight was moved slightly (0.01) away from the threshold value in the direction suggested by the user. Thus, it was through use by the community that the correct values were established.

User-entered natural language queries were parsed by a natural query interface into the actual SQL query and into its attributes and its modifiers, and these were stored in data files. Implemented stored procedures processed these data files, i.e., determined the value ranges for the modifiers, run the modified query against the database, and returned the

appropriate images to the user. Below each image were checkboxes asking the user to indicate whether the image satisfies the query, or if instead the image fits the identifier that is one modifier above or below it. If the user had selected a different modifier, then the image's weight was changed accordingly by a stored procedure. Eventually, the single query attributes could result in the community opinion represented in the database. Further, the resulting images were described by the community, not the database designers, in the sense that each image's associated weights for the defined attributes (eyes, face, hair, etc) reflect the feedback of a particular user community. Thus, it was expected that each user community could describe the same images differently.

The ultimate goal of the prototype was to seek an implementation that would provide a basis for further research. Several issues were simplified when developing the prototype allowing expansion of the research in the following areas:

1. Compound Queries of Fuzzy Attributes,
2. Implement "not" Queries,
3. Determine and Include Synonyms of the Attributes in Fuzzy Relational Database,
4. Fully Integrated interface with a Natural Language Querying System,
5. Determine Methods to Calculate Membership Function Values,
6. Initialization Methods of Fuzzy Attribute Weights,
7. Extend the list of attributes, and
8. Prototype Implementation with Spatial Image Data.

4.2 Initialization Methods

As an extension of the original research, Lee [2] compared various initialization methods of the fuzzy membership function values in order to expedite the convergence of the fuzzy values. The original prototype developed was modified for this project. The weighting factors applied to modify fuzzy attribute values were changed from .01 to .02 in [2], since actual community convergence is not used for the experiment portion of this research. The experiment was initially conducted using only the Color table and then results were verified using the Face table.

The first part of this experiment was to initialize the fuzzy membership values for the pictures within the database using four methods:

1. **Independently Random**- Generated randomly between the ranges of 1.0 and 0.0.
2. **Random Proportional** – The membership function value for Brown eye color is generated randomly between the ranges of 1.0 and 0.0. Remaining two values split the difference proportionally.
3. **New Random Proportional** – The membership function value is generated randomly for one color within the valid range [1, 0]. Remaining two values split the difference proportionally.
4. **Fixed**- Fixed to the midpoint of the function value, in this case 0.5.

For the purpose of this experiment, the convergence was based on one user's opinion. This was, in fact, a simulation of the case where the community agrees.

The second part of the experiment was to query the database and provided feedback on the query responses. The range quantifiers were queried in the following order:

1. Medium
2. Slightly
3. Very

The querying continued until all the query responses provided "correct results" or the data converges with respect to the modifier's range. The term "correct results" was an interpretation of the Color and Range variables. In other words, the query ran on "Very Brown", "Medium Brown" and "Slightly Brown" eyes independently until only the query results returned by the system meet the criteria for each range and color in question and so on for each color and range variable. During the querying, the following data points were tracked:

1. Number of times the query was performed for each membership function,
2. The total number of queries for each modifier, and
3. The color responses per query (i.e., 3 Very Brown, 21 Medium Brown, and 7 Slightly Brown) after convergence.

The final part of the experiment was to compare the total number of queries needed for convergence and recommend the fuzzy membership initialization function that could optimize convergence.

Results conclude that the midpoint initialization is the best method. However, new random proportional was the recommended method for the current system because the random proportional method attempted to quantify the small dependency between the attribute values without requiring them to be totally independent. Since the color and facial attributes have some intrinsic relationship, the addition of the proportional restriction on the random initialization better models the attributes depicted within this prototype.

The following issues for the future work arose:

1. Test the New Random Proportional Method by User Community,
2. Methods to Increment the Weight of Membership Values. If the linear function is used then what should be the increment factor?
3. Investigate the Longevity of Convergence based on the Community Stability, and
4. Investigate the Exclusion of an Image within a Fuzzy Function when the Fuzzy Attribute of the Image does not qualify with in Fuzzy Function.

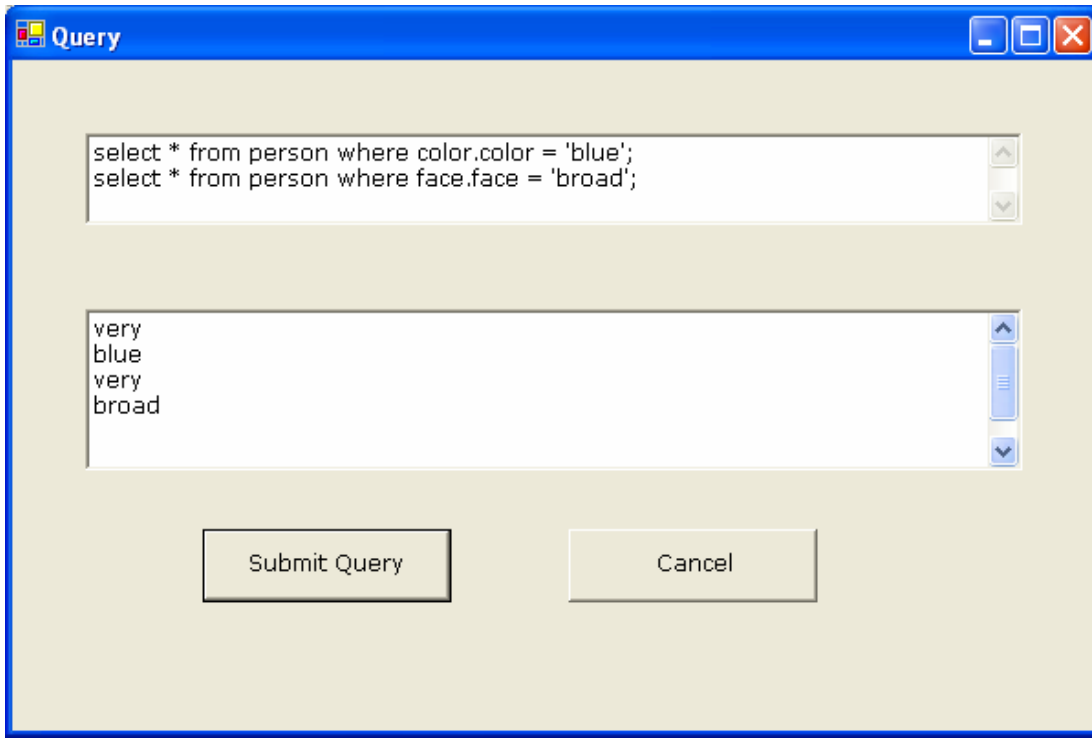
4.3 Compound Queries

Crabtree [3] expanded the original prototype by implementing the capability to search on multiple defined fuzzy characteristics. The objective was to provide users with new querying capabilities based on conditions that involved their preferences for multiple fuzzy

attribute. Users defined flexible queries that yielded results that were ranked according to their preferences. These queries essentially were Boolean queries that had been extended by adding these preferences. The database returned those images that met the subjective criteria.

To allow a query using compound attributes, the system read in SQL queries along with fuzzy attributes and modifiers from data files. The data files were created by parsing the user-entered natural language queries, using a natural query interface into SQL queries and into its fuzzy attributes and modifiers. These data files were displayed to allow the user to confirm or change the criteria of the queries, as shown in Figure 6. Implemented stored procedures processed these data files, i.e., determined the value ranges for the modifiers, ran the modified queries against the database, and returned the appropriate images to the user.

The user interface was modified in order to get feedback on the compound attributes from the user community. The user community provided feedback on whether the result(s) meet their query by agreeing or disagreeing. If they agreed, the weights assigned to each attribute were modified in the database according to the attribute's modifier and the threshold value. If they disagreed, the image was redisplayed allowing the community to use the selections given, to indicate which attribute(s) varied from the query. Selecting the ok checkbox for an attribute could indicate the attribute met the original query.

Figure 6: Query Form

The image shows a window titled "Query" with a blue title bar and standard window controls. It contains two text input fields and two buttons. The top input field contains two SQL queries: "select * from person where color.color = 'blue';" and "select * from person where face.face = 'broad;". The bottom input field displays the results of these queries: "very", "blue", "very", and "broad". At the bottom of the window are two buttons: "Submit Query" and "Cancel".

The weights assigned to each attribute were updated accordingly and stored in the database. The new weights were then used as other queries were made, defining the images based on the particular user community's natural language.

The ultimate goal of this system was to extend the original prototype to handle compound fuzzy queries – more than one fuzzy term in the query. Several issues arose to be explored further:

1. Implement User Friendly User Interface, and
2. Extend the List of Attributes and Modify Application

4.4 Determination of Synonyms for Fuzzy Modifiers

The goal of the Duke's research [4] was to expand the original prototype by developing a method for evaluating potential synonyms of the implemented modifiers in order to deal with the duplicity and ambiguity of natural language. In addition, a proposed implementation was given to address the determination of synonyms and provided a mechanism for their inclusion in a fuzzy database with a natural language interface. In order to determine synonymy for a modifier, a user community was asked to select which of the images stored in the fuzzy database met the criteria of that modifier. Subsequently, a comparison of the membership weights for the modifier in question to the implemented modifiers was made.

The original prototype, which was developed by Joy [1], was modified by the following design changes:

1. User Interface was modified so that the user could evaluate all of the images in the result set rather than the six visible in the original prototype by adding the "More" button to the existing user interface and forced the user to view all the results before weight adjustment to occur.
2. The method for updating weights in the attribute modifier "Slightly" was modified so that if an image currently had a weight in the "Slightly" range and a user's response was that a "Less" modifier could better describe the image, the weight for the image could be decremented by .01 regardless of where in the range the weight was previously.

3. The range of weights for the initial randomization was narrowed to 0.1 – 1.0 from 0.0 – 1.0, so that all images were included in the evaluation.
4. The lower limit of the ‘Slightly’ modifier was adjusted from 0.1 to 0.05. This insured that multiple users were required to confirm that an image was to be excluded from the result set for a given term before that image could no longer be displayed for the color specified in the query.
5. The upper and lower limit of the ranges for each modifier was redistributed as shown in Table 2.

Table 2: Modifier Ranges and Midpoints

Modifier	High	Low	Threshold
Very	1.00	0.69	0.85
Medium	0.69	0.36	0.52
Slightly	0.36	0.05	0.20

6. The weighing factors applied to modify fuzzy attribute values were moved from 0.01 to 0.015 to reflect the user’s community consensus. The increment in weighing factor was done to reduce the number of the queries needed to reach the user community consensus and hence reduce the burden on the user community.

The experiment was conducted in two phases. In the first phase, the user evaluation of the fuzzy relational database to define membership values for the attribute eye color for each

image was completed by the user community which was comprised of students, faculty and staff of the School of Engineering of Virginia Commonwealth University.

The second phase of the project was to have the user community evaluate a set of all combinations of potential synonyms and colors to the fuzzy modifiers: “Very”, “Medium” and “Slightly”. In order to accomplish this evaluation, the user interface was modified. A query was randomly selected for each of the potential synonyms of a fuzzy modifier combined with one of the three colors. For the selected query, all images of that color with a membership weight were displayed and, for each image, the user was asked to indicate whether the image met the criteria or not. The number of times each image was selected by a user as meeting the criteria described by the synonym and color was tallied and recorded. The number of times each image was selected was multiplied by the user defined membership weight for the color aspect of the query. The weighted average of all membership weights for each of the queries was calculated and compared the result to the original modifier ranges.

The results of the evaluation of modifier synonyms showed that some modifiers, such as Normal and Weak, were determined to be synonyms to the implemented modifiers, Medium and Slightly, respectively, for all three eye colors. However, for other modifiers, the user community concluded different implemented modifiers as synonyms depending on the eye color. For example, the modifier Average was determined to be a synonym for

the implemented modifier Medium if the color term of the query was Blue, but for Brown and Green, it was synonymous to Very and Slightly, respectively.

The community evaluation of the database provided a basis for the evaluation of synonyms to the fuzzy modifiers of the initial implementation. Several issues arose to be explored further in the following areas:

1. Implement User Friendly User Interface, and
2. Method to Store Synonyms that are Different for Different Modifiers.

4.5 Determine Membership Function Value by Direct Rating Method

As an extension of the original work, it was proposed to construct membership values using the Direct Rating method, Sanghi [5]. The Direct Rating approach subscribed to the point of view that fuzziness arose from individual subjective vagueness. This method of constructing the membership function was then compared with the original prototype which used the random initialization method [1] to determine which method gave the most user satisfaction with minimum feedback from the community.

The original prototype was modified to alter the method of assigning the membership value to the direct rating method, for the blue eye color attribute of each image. The experiment was conducted for both prototypes in two sessions, which are training and testing. The user interface of both the prototypes was altered to make it more users friendly. The user community which was comprised of students, faculty and staff of the

School of Engineering of Virginia Commonwealth University provided feedback in both sessions.

In the direct rating method, weights were not initially assigned to the blue eye color attribute of the images. In the training session, the question put to the community was “How blue are the eyes? They respond using a simple indicator on a sliding scale. The left most bar on the sliding scale represented that the eyes were not blue whereas the right most bar on the sliding scale represented that the eye were 100% blue. The training session was comprised of 21 people. The training data for each image was collected. It showed the community’s view of the degree to which each image had blue eyes. A frequency distribution table was created that grouped the data into non overlapping intervals called frequency ranges. The frequency ranges were chosen based on range values of the very (75-100), medium (30-74), slightly (0 -29) modifiers. The number of data that fell into each frequency range was calculated. The counts, or frequencies, of data were then listed in their respective frequency ranges. Each image was assigned a weight equal to threshold membership value for the maximum frequency range.

In the original method [1], random weights were assigned initially to the blue eye color attribute of the images. The user evaluation of the fuzzy relational database to define membership values for the blue attribute eye color for each image was completed by the user community in the training session which was comprised of 21 people.

A testing session was conducted to compare both methods of eliciting membership values and to determine which method gives the better community satisfaction. In the testing session, the fuzzy relational database was queried for “slightly”, “medium” and “very” blue eye color of the images based on the membership values assigned by the random method and the direct rating method. The user community was asked whether, they were satisfied with the result or not. The percentage of user satisfaction was calculated.

The research found that the direct rating method of constructing membership value gave higher user community satisfaction compared to the random method. The research arose some questions, which could be explored in future fuzzy relational database research.

1. Improvements in the User Interface,
2. Method of Defining the Range of Modifiers,
3. Method to Handle Unclear Pictures,

4.6 Best Method to Store Synonyms

The purpose of a NLI for a database system was to accept requests in English from a user, attempt to “understand” the requests, and then translate them into SQL.

In this study Mehrotra [6], the experiments were conducted to find the best possible way to store potential synonyms so that a NLI answers the queries correctly. It could expand the use of NLI beyond the words used in the database. This research tested the NLI with the Fuzzy database developed by the group [1]. This research used English Language Front-

End (ELF) which was a commercial system that generates Natural Language Processing System for a database [7].

As discussed in [1], the Natural Language Interface generated a SQL statement and saved it in a file. In addition, they stored the fuzzy words in a separate file. Then the Fuzzy group took the SQL statement and altered it according to the fuzzy words found in the separate file. Now instead of changing the SQL again outside, it was desired to try to do it as a part of Natural Language Interface. Modifying the SQL was successfully incorporated in ELF. In ELF, a lexical dictionary, also called as lexicon, hold all the words that may occur in query. The lexicon, as well as database, could be used to define synonyms.

This research brought up the following perspectives for the synonyms. It was assumed that the user community had a consensus agreement.

Perspective 1: Two words that are definitely synonyms of each other.

Suggestion: Store one of the words as the base word in the database and then declare the other as a synonym of the base word in the lexicon. This worked because when words were declared as synonyms of each other in lexicon, the lexicon changed the word used in the NLI query to the base word stored in database when it generated the SQL.

Perspective 2: Two words that are definitely not synonyms of each other.

Suggestion: Store both words as the base word in the database and no synonyms were defined in the lexicon. This worked because the words were known not to be synonyms of each other and was therefore unrelated.

Perspective 3: When it is unclear whether or not words are synonyms but user community consensus exists.

Suggestion: This perspective occurs when it was up to the user community to decide whether word were synonyms or not and the user community had a consensus. This perspective was treated in same way as fuzzy words were handled. That is, start with a default condition and depending upon the feedback of the user community, the weights were modified. Here the weights were considered to have a different meaning from that in [1]. There the weights were used to determine the extent to which the words were applied but in this research weights determined if a word applied or not. Therefore, if words were not relevant then the weights were decreased.

The first Step was to store both words in the database. The lexical dictionary had separate entries for both the words but neither of them listed one as a synonym of the other. In the second step, duplicate rows were created in the database. For example, consider two words: “broad” and “large”. After creating duplicating rows, every face which was declared as “broad” was now also declared as “large” and

vice versa. The result of query (“broad faces”) either for “broad” or “large” could contain all the people that were declared as broad as well as large faces. If the user community decided that the words were synonyms then weights for “broad” and “large” were not changed. If the user community decided that words were not synonyms then weight of word which was queried (e.g. “broad”), was decreased. Once the weight of one attribute was updated, the interface asked user whether or not that person had “large” face. After running the query each time, it was checked if any row had weight as 0, if so then it was deleted. Since initial weights were declared randomly [1], the same test was repeated with different initial weights and the number of runs required (for the same person) for the weight of “broad” to reach zero were noted. It was found that there was a linear correlation between initial weight value and the number of runs; this was because the weight was decreased by a constant value of 0.1.

Perspective 4: When it is unclear whether or not words are synonyms and there is a divide between users.

Suggestion: In this experiment, not all the users think that the words were synonyms. To simulate this test, it was assumed that there were three doing testing and the first two people thought that “broad” and “large” were not synonyms and third person thought they were indeed synonyms. It used the same method as defined above. Therefore, after two queries weight of broad was decreased by 0.2. But at the third query, no weight change was done. But now the weights of “broad”

and “large” were not equal for the same face image. Therefore, the weight of “broad” was increased by 0.1 in order to make it close to the weight of “large”. The experiment was carried out again and again in above order until the weight reached zero. Number of queries was recorded. This experiment gave the desired results, but it was found that the implemented system was biased towards the words being synonyms as this was initially assumed.

This paper successfully suggested the best methods to store the synonyms. Several issues arose to be explored further in the following areas:

1. Further Testing Considering Bursts of the Biased Users,
2. Find Methods to Remove Memory Wastage, and
3. Incorporate the Fuzzy Database aspect of Weight Updating.

CHAPTER 5 Interactions, Issues, Experiment Overview

5.1 Interactions between Versions of the Prototype

Some of the original prototype's issues were researched by the members of the Database Research Group. This research created different versions of the prototype and raised some more issues. In order to create one system out of all the versions, it is necessary to analyze the interaction between them and consider issues raised by them. The issues are listed in section 5.2.

The interactions between each version can be shown by means of an **interaction matrix** as shown in Table 3. That matrix is interpreted as follows: Consider research [2]; If [2] interacts with [3] then the Interaction # is given in [2] row and [3] column indicating the interaction between them. An "N" indicates no interaction. Dash (-) represents that interaction/ no interaction is already stated.

Table 3: Interaction Matrix

	Initialization Methods [2]	Compound Query [3]	Modifier's Synonyms[4]	Direct Rating [5]	Store Synonym[6]
Initialization Methods [2]	N	N	Interaction 1	Interaction 2	Interaction 3
Compound Query [3]	-	N	N	N	Interaction 3
Modifier's Synonyms[4]	-	-	N	Interaction 4	Interaction 5
Direct Rating [5]	-	-	-	N	Interaction 3
Store Synonym [6]	-	-	-	-	N

Interaction 1: In the initialization methods [2] research, the weighting factor applied to modify fuzzy attribute weights is 0.02 but in [4] it is 0.015. The upper limit and lower limit of the ranges of each modifier is different for both pieces of research. The range of weights for the initial randomization is changed to 0.1 – 1.0 in [4] from 0.0 – 1.0 in [2]. The method of updating weights in the attribute modifier “Slightly” is different in [4] from [2]. The initialization methods [2] recommends using random proportional method instead of the random method that is used in [4] for initial randomization.

Interaction 2: Initialization methods [2] and Direct Rating Method [5] are the two different methods for constructing the initial membership function value. Initialization methods [2] need to be tested by the user community. In the initialization methods [2], the

weighting factor applied to modify the fuzzy attribute is 0.02 but in Direct Rating Method [5] is 0.01.

Interaction 3: The weights are considered in a different perspective in [2] [3] and [5] from that in [6]. In [2] [3] & [5] the weights are used to determine the extent to which the words are applied but in [6] weights determined if a word applied or not. Therefore, if words are not applied (i.e. not synonyms of each other) then the weights get decreased in [6].

Interaction 4: In the modifier's synonyms [4], the weighting factor applied to modify fuzzy attribute is 0.015 but in direct rating [5] is 0.01. The upper limit and lower limit of the ranges of each modifier is different in both research. The range of weights for the initial randomization is changed to 0.1 – 1.0 in [4] from 0.0 – 1.0 in [5]. The method of updating weights in the attribute modifier “Slightly” is different in [4] from [5]. Research [5] uses direct rating method as a method of constructing membership function but modifier's synonyms [4] uses random method to elicit the membership function.

Interaction 5: The weights are considered in a different perspective in [4] from that in [6]. In [4] the weights are used to determine the extent to which the words are applied but in [6] weights determined if a word applied or not. Therefore, if words are not applied (i.e. not synonyms of each other) then the weights get decreased in [4]. Research [4] develops a method of evaluating the synonyms of the implemented modifiers which are Very”,

“Medium”, and “Slightly”. Research [6] describes the best possible way to store the synonyms in Natural Language Interface.

5.2 Issues

The research that is done under the Database Research Group raises some issues. In the order of priority, the extensive list of issues has been divided and listed in this section.

Priority 1: This category lists the most important issues that need to be addressed.

1. Find out which method of constructing membership function is better : Direct Rating Method or New Random Proportional Method
2. Determine whether we need training phase to learn the system. If yes, then which initialization method should be used and when training phase is complete?

Priority 2:

1. Determine measures for: data set characteristics, user community characteristics, degree to which the system is working correctly, different methodologies comparison, retrieval effectiveness of the query, effort needed to conduct training or testing, user community consensus
2. Determine the method of choosing subset of the user community that will represent whole user community in the training phase
3. Method to define weights. The weights are considered in a different perspective from [1] that in [6]. In [1] the weights are used to determine the extent to which the

words are applied but in [6] weights determined if a word applied or not. Is one is better than other or both can be used together?

4. Investigate the longevity of convergence based on the community stability. Will the community consensus remain same for ever? What will be effect of changing user community?
5. Investigate the exclusion of an image within a fuzzy membership function
6. Determine whether modifier membership function value should be between 0 and 1 or between 0.1 and 1
7. Improvements in ease of use of the user interface because the user interface will affect the process of community learning

Priority 3:

1. Way to handle overlapping modifier ranges. For Example, medium has modifier range between 0.3 and 0.71 but average has modifier range between 0.4 and 0.8
2. Way to handle modifiers/hedges that are different for different fuzzy attribute values. For example, average seems equivalent to very with respect to brown eyes, but equivalent to medium for blue eyes
3. Method of defining the range of modifiers
4. Methods to increment the weight of membership values. If the linear function is used then what should be the increment factor?

Priority 4:

1. Method to handle unclear picture
2. Find out the effect of Burst
3. Determine possibility of using Data Mining to detect possible synonyms

Priority 5:

1. Implement “not” queries
2. Fully integrated interface with a Natural Language Querying System
3. Prototype implementation with Spatial Image Data.
4. Effect of compound queries on the previous research work
5. Extend the list of attributes
6. Incorporate the fuzzy database aspect of weight updating in Natural Language Interface

5.3 Experiment Overview

Fuzzy Set theory has expanded the usefulness of relational databases by allowing a user to retrieve information based on imprecisely defined properties or qualities. One of the most important aspects of Fuzzy Relational Database is to define the membership function. The most important issue for the current prototype is to find better method of constructing membership function as defined in section 5.2. For the purpose this paper, the experiment is conducted by changing the current prototype to determine whether Direct Rating or New Random Proportional is better method to define membership function. One of the most

important issues to conduct the experiments is whether the system needs the training phase to learn the membership value or not.

New Random Proportional Method does not require the training phase to construct membership function as explained in section 4.2. Direct rating method requires the training phase as explained in section 4.5. It is necessary to investigate the method of choosing subset of the user community that will represent whole user community in the training phase. That aspect of choosing the subset of the user community is not addressed within this paper; however it should be addressed in future work.

For the purpose of this paper, a training phase is used for both methods and testing phase is used to test both methods. Hence, in the training phase, the users' community defines membership values for eye color for both methods. The final part of the experiment is to compare satisfaction levels of both methods and determine which method is better method for current project.

CHAPTER 6 Direct Rating Experiment

6.1 Experiment Setup

The original prototype database has a ColorDirect table that holds the fuzzy membership values as shown in table 4. For the color table, attributes are Blue, Green and Brown. Initial random weights were not assigned to the eye color attributes. Users' feedback was taken in two sessions, which were training and testing.

Table 4: Using Membership Values (Weights) for Each Attribute

IMAGE_ID	EYE_COLOR	WEIGHT
1	GREEN	0.6
1	BLUE	0.4
1	BROWN	0.2

The suggestions given by Sanghi [5] to improve the Direct Rating Method were considered while setting up the experiment. The following changes were made:

1. Instead of using a sliding bar to capture the user's response, the user chose from one of the four radio buttons. The radio buttons for blue eye color attributes were

- i. Not Blue
- ii. Slightly blue
- iii. Medium Blue
- iv. Very Blue

In Sanghi [5] was suggested that the community should define the range of modifiers because users can have different perspective for ranges. By choosing the radio button, community was given control over range definition.

2. The “Not” modifier was added to exclude the images from that range/color pair when the image did not satisfy the criteria.
3. The experiment was conducted for all the attributes of the eye color instead of only blue eye color attribute. The users were asked to provide feedback for all three attributes which were generated randomly to prevent bias.

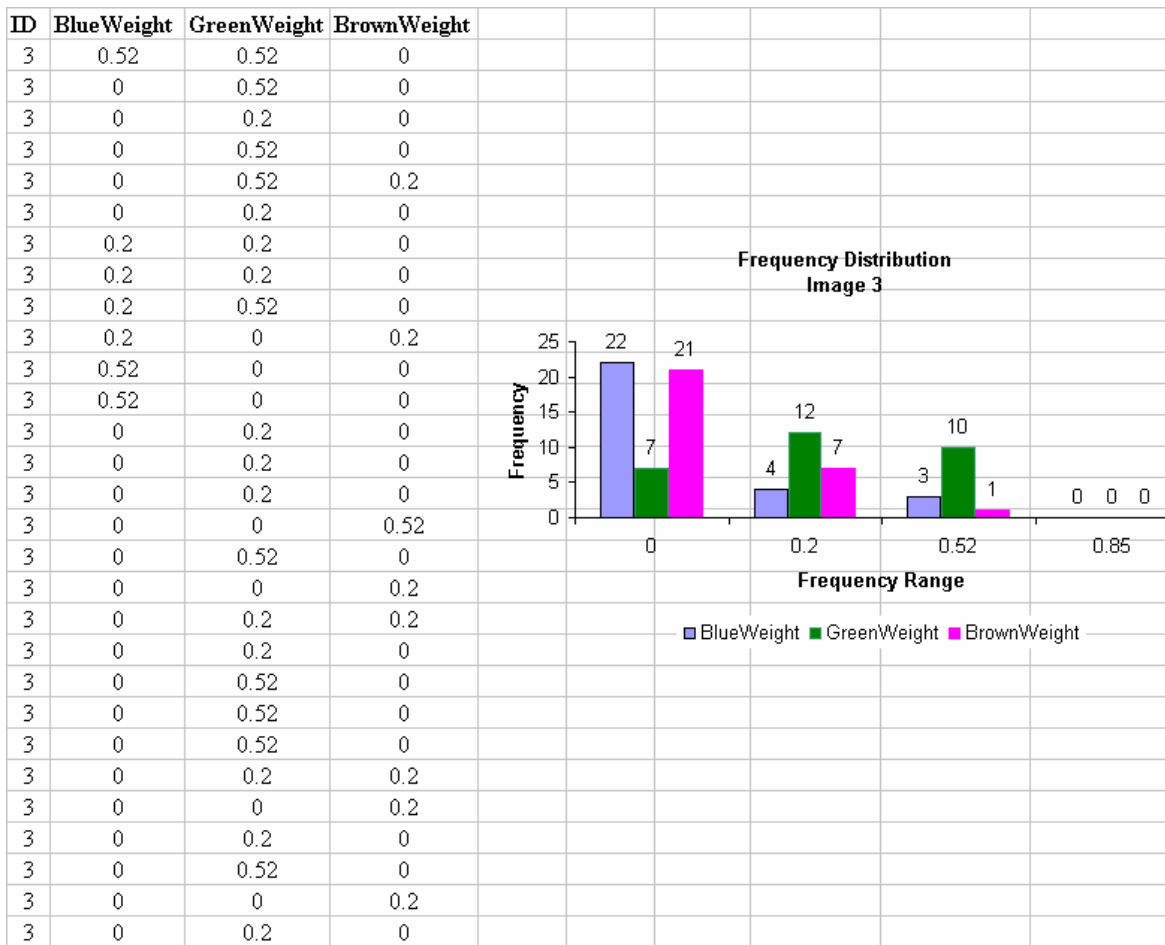
6.2 Training Session

Each user evaluated the fuzzy relational database to define membership values for eye color for direct rating method. The SQL Server Fuzzy database was installed on the database server in the School of Engineering. The user interface to train the system was installed on client computers in the computer classroom and accessed the fuzzy database via the local area network. The user community consisted of students, faculty and staff of the School of Engineering of Virginia Commonwealth University. The instructions which detailed the steps to conduct experiment by the users on the client computers were given to

members of the user community for the determination of membership weights. The instructions can be referenced in Appendix F, Figure F.1.

In the training session, for example, the question put to the community for green eye color was “How green are eyes?” and they responded by choosing one of the four radio buttons. This user feedback process was also completed for all eye colors. The history of user’s feedback for training session was maintained in HistColorDirect table. The database stored “0” value for “Not Color” radio button, “0.2” value for “Slightly Color” radio button, “0.52” value for “Medium Color” radio button and “0.85” value for “Very Color” radio button selection. The users’ feedback for image id 3 is shown in Figure 7.

Blue Weight column in Figure 7 represents the users’ feedback for blue eye color of the image 3. A frequency distribution graph was created that grouped values into non overlapping intervals called frequency ranges. The frequency ranges were chosen based on range values of the “Very” (0.85), “Medium” (0.52), “Slightly” (0.2) and “Not” (0) modifiers. The count of values that fell into each frequency range for each color was calculated. The range with maximum frequency for color represented the modifier for the color/image pair. For example, from Figure 7 using frequency distribution graph, it can be deduced that image 3 is a person with slightly green eyes but no tint of blue or brown. Therefore, the membership value of 0.2 was assigned to the green eye color attribute of the image id 3 with membership value of 0 for blue and brown eye color attribute.

Figure 7: Users Feedback for Image 3 for Direct Rating Method

The frequency distribution method was applied to all images to find out the defined membership weights for each image after training phase as shown in Figure 8.

The history of users' feedback for all images for direct rating method during training session can be referenced in Appendix D, Tables D.1 – D.27.

Figure 8: Membership Weights for Direct Rating Method

Membership Weights Direct Rating			
Image ID	BlueWeight	GreenWeight	BrownWeight
1	0.2 S	0 N	0 N
2	0 N	.85 V	0 N
3	0 N	0.2 S	0 N
8	0 N	0 N	0.85 V
9	0.2 S	0 N	0 N
10	0.2 S	0.2 S	0 N
11	0.2 S	0 N	0 N
12	0 N	0.2 S	0 N
13	0 N	0 N	0.85 V
14	0.2 S	0 N	0 N
15	0 N	0.52 M	0 N
16	0.2 S	0.52 M	0 N
17	0 N	0.2 S	0 N
18	0 N	0 N	0.85 V
19	0.2 S	0.2 S	0 N
21	0 N	0 N	0.85 V
23	0 N	0 N	0.85 V
24	0.85 V	0 N	0 N
31	0.52 M	0 S	0 S
32	0 N	0 N	0.85 V
33	0 N	0 N	0 N
34	0.52 M	0 N	0 N
35	0.52 M	0 N	0 N
36	0.52 M	0 N	0 N
37	0.52 M	0 N	0 N
38	0.52 M	0 S	0 S
40	0 N	0 N	0.52 M

6.3 Testing Session

In the testing session, the database was queried for each range/color pair randomly. Each user was asked to complete the feedback for all range/color pair. The community was asked whether they were satisfied with the query result or not by choosing appropriate radio button. The user interface to test the system for direct rating method was installed on client computers in the computer classroom. The user community consisted of students,

faculty and staff of the School of Engineering of Virginia Commonwealth University. The instructions which detailed the steps to conduct experiment by the users on the client computers were given to members of the user community. The instructions can be referenced in Appendix F, Figure F.3. The users' feedback was maintained in HistTestDirect table. The history of testing session for user's feedback for direct rating can be referenced in Appendix E, Tables E.1 – E.27. The percentage of users' satisfaction for defined membership weights based on direct rating method is shown in Figure 9.

Figure 9: Percentage of Satisfaction for Direct Rating Method

Percentage of Users Satisfaction Direct Rating Method			
ID	% Satisfaction for Blue	% Satisfaction for Green	% Satisfaction for Brown
1	50	100	100
2	100	90	100
3	100	60	100
8	100	100	90
9	80	100	100
10	80	70	100
11	50	100	100
12	100	100	100
13	100	100	90
14	10	100	100
15	100	60	100
16	40	70	100
17	100	30	100
18	100	100	90
19	60	70	100
21	100	100	90
23	100	100	100
24	80	100	100
31	80	100	100
32	100	100	60
33	100	100	100
34	70	100	100
35	100	100	100
36	90	100	100
37	80	100	100
38	70	100	100
40	100	100	70

CHAPTER 7 New Random Proportional Experiment

7.1 Experiment Setup

The original prototype database has a Color table that stores the fuzzy membership value. For the Color table, the possible attribute values are Blue, Green and Brown. The database also contains a table that stores the range values for each fuzzy modifier as shown in Table 5.

Table 5: Modifier Ranges and Midpoints for New Random Proportional

Modifier	High	Low	Threshold
Very	1.00	0.70	0.85
Medium	0.69	0.36	0.52
Slightly	0.35	0.02	0.20

The “Slightly” low range modifier was changed to 0.02 to allow images to fall outside the range of applicable values. After repeated queries of the system, the image could eventually fall out of the range. Therefore the lower limit of the “Slightly” causes images to be excluded from a fuzzy function that does not apply.

7.2 Initializing Membership Values

In this experiment, the membership value was generated randomly for one color within the valid range [1, 0] initially. The initial membership values for other two colors were set proportionally from the first color value using the following formula [2]:

$$V_o = \text{Random} [1, 0];$$

$$V_1, V_2 = (1.0 - (V_o)) / 2.0;$$

The selection of first color (V_o) for initialization is rotated between each color value to prevent bias.

7.3 Querying the Database

Once membership values were initialized, the database was repeatedly queried for each range value of each eye color. The range of quantifiers can be queried by following ways:

1. The range quantifiers can be queried in the order: Medium, Slightly and then Very. Each query is repeated until all the responses delivered by the database met the criteria. This querying process can also be completed for each range/color pairing.

This method of querying the database has the following flaws:

- a. As the target user community contains an unknown number of users each having varied opinions, convergence based on a wider community will be difficult to evaluate. For example, the satisfaction criteria for slightly blue will be different for each user.

- b. The user will need to provide feedback for same range/color pair many times before the pair satisfies the criteria. This could be frustrating for the user and could affect the user's judgment.
 - c. There is a possible bias towards the medium range. In Lee [2], the number of queries for the medium/color pair is higher than any other range.
2. The range of quantifiers can be queried randomly. This method has the advantage of preventing any bias towards any range/color pair. The disadvantage of this method is that it can be possible that queries occur more for one range/color pair than for others. In an experiment using this approach, it was discovered that number of queries for slightly brown was much higher than the number for very brown.
3. The third approach can be to rotate between each range/color pair randomly while forcing the experiment to query all the range/color pairs once for each user. This method has the advantages of the random method. It also prevents any bias for number of queries for any range/color pair.

For this experiment, the third method was chosen. Each user responded to all nine queries for range/color pair but the pair ordering was generated randomly.

7.4 Training Session

Each user evaluated the fuzzy relational database to define membership values for eye color. The SQL Server Fuzzy database was installed on the database server in the School

of Engineering. The user interface to train the system was installed on client computers in the computer classroom. These client machines accessed the fuzzy database via the local area network. The user community consisted of students, faculty and staff of the School of Engineering of Virginia Commonwealth University. The instructions which detailed the steps to conduct experiment by the users on the client computers were given to members of the user community for the determination of membership weights. The instructions can be referenced in Appendix F, Figure F.2.

New Random Proportional Experiment was modified from the suggested method by Lee [2]. Hence simulation was conducted to find out the number of queries needed to reach single user convergence. Five hundred and thirty two queries were needed to update the database to reach single user convergence. Hence to prevent an unreasonable burden on the user community and to prevent fast exclusion of the image from fuzzy function, the factor applied to modify fuzzy attribute weights was changed from .01 to .015.

All the attribute values for the Color table for New Random Proportional Method were initialized using the Initialize_NProp_Random stored procedure in the SQL Server database that can be referenced in Appendix G. The initialized random proportional values can be referenced in Appendix A, Tables A.1 – A.3.

Figure 10: Membership Weights for New Random Proportional Method

Membership Weights for New Random Proportional			
Image ID	BlueWeight	GreenWeight	BrownWeight
1	0.21 S	0.28 S	0.24 S
2	0.24 S	0.42 M	0.48 M
3	0.24 S	0.56 M	0.2 S
8	0.56 M	0.22 S	0.25 S
9	0.2 S	0.22 S	0.5 M
10	0.54 M	0.32 S	0.22 S
11	0.55 M	0.22 S	0.22 S
12	0.36 S	0.52 M	0.22 S
13	0.34 S	0.22 S	0.72 V
14	0.21 S	0.55 M	0.33 S
15	0.44 M	0.54 M	0.22 S
16	0.62 M	0.2 S	0.57 M
17	0.24 S	0.54 M	0.24 S
18	0.28 S	0.22 S	0.5 M
19	0.2 S	0.55 M	0.22 S
21	0.62 M	0.22 S	0.51 M
23	0.24 S	0.22 S	0.52 M
24	0.27 S	0.43 M	0.22 S
31	0.25 S	0.22 S	0.22 S
32	0.22 S	0.22 S	0.85 V
33	0.49 M	0.22 S	0.54 M
34	0.53 M	0.22 S	0.22 S
35	0.26 S	0.22 S	0.41 M
36	0.55 M	0.22 S	0.33 S
37	0.85 V	0.22 S	0.22 S
38	0.3 S	0.22 S	0.4 M
40	0.24 S	0.28 S	0.28 S

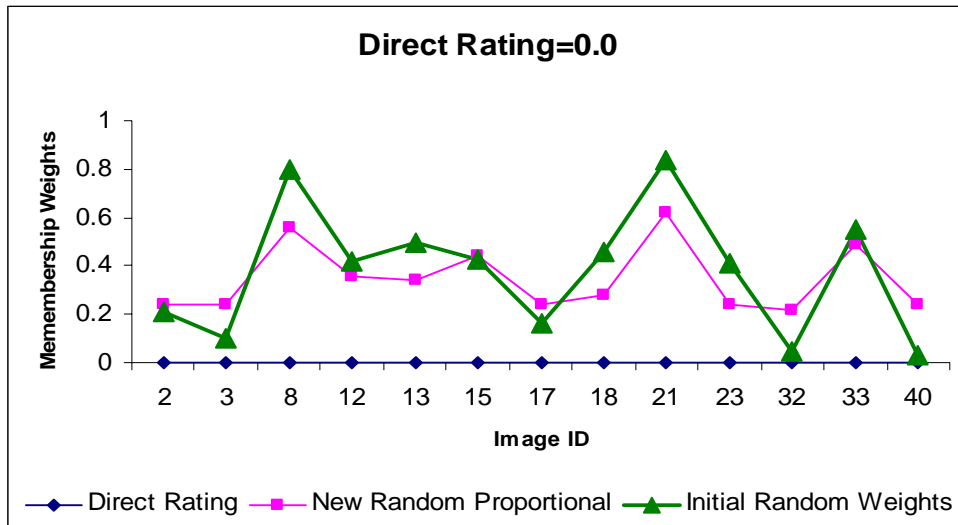
The user feedback was completed for each range/color pair. The history of user's feedback to define membership weights was maintained in the HistColor table. The history was maintained to record the pattern of users' responses. The history of user's feedback in the training phase can be referenced in Appendix B, Tables B.1 – B.27. The final membership weights after are shown in Figure 10.

7.5 Analysis of Training Results

After the completion of training sessions for both methods, analysis was conducted to compare final membership values of Direct Rating method and New Random Proportional method. Unlike the Direct Rating method no absolute zero rating was found in the New Random Proportional method due to the absence of the choice “Not Blue eyes”. For example, image 32 definitely does not have blue eyes according to the Direct Rating method. But the eyes of image 32 are “Slightly” blue based on the New Random Proportional method. The images for which a fuzzy function was not applicable were included in the fuzzy function of New Random Proportional Method. The membership weights approached the threshold value (0.20) of “Slightly” modifier instead of zero, even when the initial membership values were between 0.02 - 0.2 as shown in Figure 11. For example, for image ID 3, the membership value was increased from 0.1 to 0.24 for New Random Proportional method during training session. However the final membership value was 0 according to direct rating method. Therefore, another training session was conducted for New Random Proportional method which included “Not Color” modifier. The experiment for New Random Proportional Method with “Not” is described in section 7.6.

There is a common trend in Figures 11-13 which shows that the New Random Proportional method was approaching the direct rating method except for a few outliers. It appears that New Random Proportional method would need more user feedback to move near to direct rating values.

Figure 11: Direct Rating (No Blue) vs. New Random Proportional



In Figure 12, image id 10 is an outlier which can be explained by the underlying methodology of the direct rating method. 10 users out of 27 said that the eyes were “Slightly blue”, and 9 users out of 27 said that the eyes were “Medium blue” (see Appendix D, Table D.6). Thus the membership weight of 0.2 was assigned to image 10 to satisfy the criteria of the maximum users’ satisfaction.

Similarly the users’ feedback for image id 37 caused another outlier as shown in Figure 13. There were only a few outliers. The similar charts for brown and green eye colors can be referenced in Appendix Q, Figures Q1 – Q6.

Figure 12: Direct Rating (Slightly Blue) vs. New Random Proportional

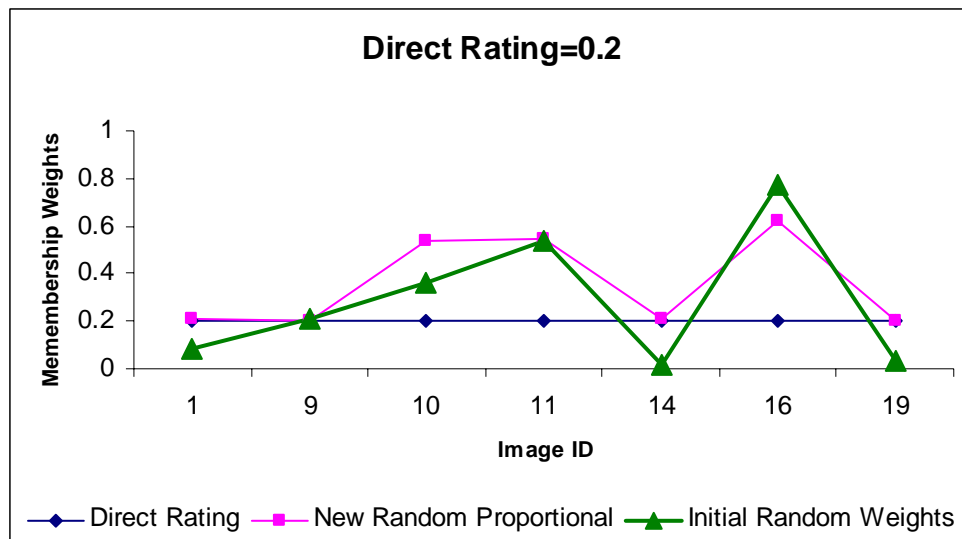
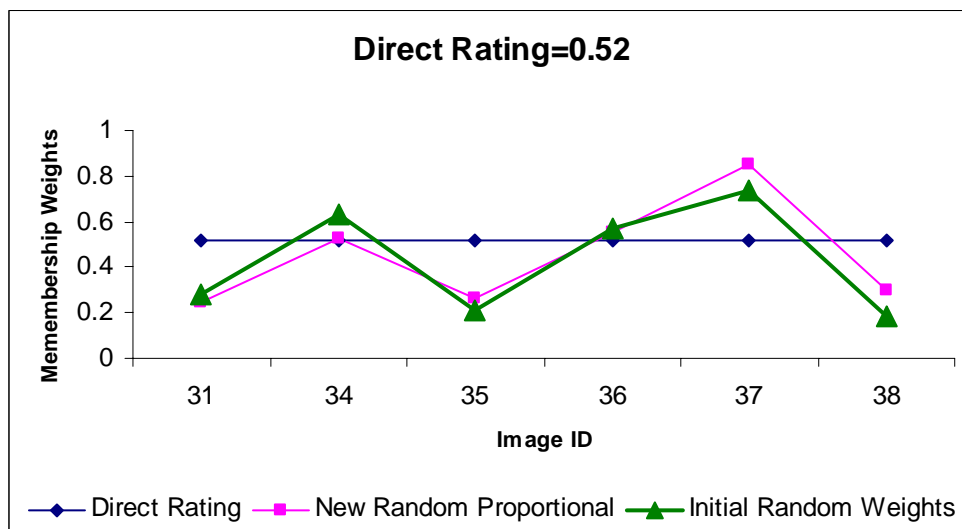


Figure 13: Direct Rating (Medium Blue) vs. New Random Proportional



7.6 New Random Proportional with “Not” modifier Experiment

Another training session was conducted for New Random Proportional Method which included the “Not” modifier. The experiment set up was same as described in sections 7.1-

7.4. “Not Color” quantifier range was added to existing quantifier ranges. Users chose the “Not Color” radio button whenever the range/color pair was not applicable to that image. The weight was decreased by 0.25 when the users’ response was “Not Color”. Appendix N contains the SQL Server stored procedure used to update the weights in the database to reflect the feedback given by the user after the query evaluation. The users’ evaluation was conducted. The instructions can be referenced in Appendix F, Figure F.5.

All the attribute values for the Color table were initialized with same initial value as was done for New Random Proportional method and can be referenced in Appendix A, Tables A.1 – A.3.

The user feedback was completed for each range/color pair as explained in section 7.4. The history was maintained to record the pattern of users’ response. The history of user’s feedback for new random proportional method in training phase can be referenced in Appendix C, Tables C.1 – C.27. The final membership weights are shown in Figure 14.

Figure 14: Membership Weights for New Random Proportional Method with Not

Membership Weights for New Random Proportional Method with No			
Image ID	BlueWeight	GreenWeight	BrownWeight
1	0.2 S	0.32 S	0.32 S
2	0.11 S	0.31 S	0.38 M
3	0 N	0.51 M	0.12 S
8	0.63 M	0.08 S	0.21 S
9	0.2 S	0.2 S	0.41 M
10	0.41 M	0.2 S	0.17 S
11	0.6 S	0.12 S	0.05 S
12	0.27 S	0.52 M	0.1 S
13	0.28 S	0.07 S	0.62 M
14	0 N	0.43 M	0.32 S
15	0.27 S	0.44 M	0 N
16	0.67 M	0.2 S	0.6 M
17	0.2 S	0.51 M	0.23 S
18	0.32 S	0.23 S	0.45 M
19	0 N	0.42 M	0 N
21	0.64 M	0 N	0.21 S
23	0.24 S	0.23 S	0.27 S
24	0.2 S	0.3 S	0 N
31	0.41 M	0.22 S	0.17 S
32	0 N	0 N	0.83 V
33	0.45 M	0.12 S	0.52 M
34	0.52 M	0.2 S	0.08 S
35	0.31 S	0.12 S	0.38 M
36	0.54 M	0.12 S	0.36 M
37	0.82 V	0.00	0.00
38	0.21 S	0.03 S	0.43 M
40	0 N	0.31 S	0.2 S

7.7 Testing Session

In the testing session, the database was queried for each range/color pair randomly. Each user was asked to complete the feedback for all range/color pair. The community was asked whether they were satisfied with the query result or not by choosing appropriate radio button. The user interface to test the system for new random proportional method was installed on client computers in the computer classroom. The user community consisted of students, faculty and staff of the School of Engineering of Virginia Commonwealth University. The instructions which detailed the steps to conduct

experiment by the users on the client computers were given to members of the user community. The instructions can be referenced in Appendix F, Figure F.4.

The users' feedback was maintained in HistTestColor table. The history of testing session for user's feedback for direct rating can be referenced in Appendix E, Tables E.1 – E.27. The percentage of users' satisfaction for defined membership weights based on new random proportional method is shown in Figure 15.

Figure 15: Percentage of Satisfaction for New Random Proportional

Percentage of Users Satisfaction New Random Proportional Method			
ID	% Satisfaction for Blue	% Satisfaction for Green	% Satisfaction for Brown
1	40	50	0
2	0	0	100
3	10	50	50
8	10	0	60
9	90	70	10
10	60	80	10
11	50	40	20
12	30	60	30
13	10	0	90
14	20	40	40
15	20	40	10
16	70	90	0
17	30	60	20
18	10	20	60
19	80	40	20
21	0	0	50
23	10	0	60
24	100	0	10
31	70	40	10
32	20	10	60
33	50	30	40
34	90	70	10
35	70	30	0
36	100	40	10
37	80	10	20
38	100	0	0
40	10	10	70

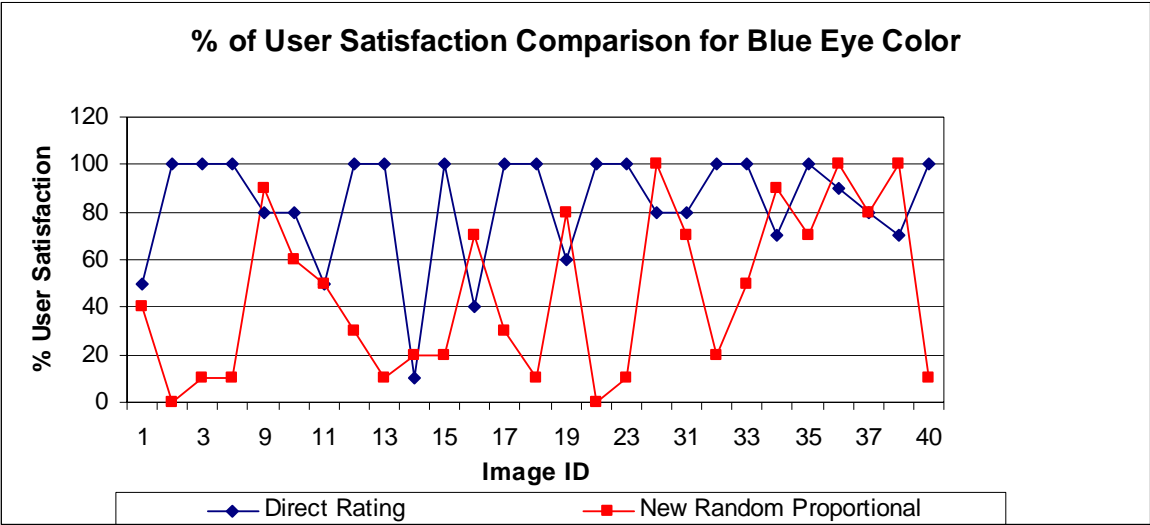
CHAPTER 8 Results and Future Work

The two methods, Direct Rating and New Random Proportional, for constructing membership function for eye color attribute of images have been analyzed by using descriptive statistics – mean, median, and standard deviation.

8.1 Analysis of Testing Results for Blue Eye Color Attribute

Figure 16 compares the percentage of users' satisfaction for the two methods for the blue eye color attribute. The graph shows that the Direct Rating method has a much higher average user satisfaction compared to the New Random Proportional method.

Figure 16: Comparison of User Satisfaction for Blue Color



75% of the images have a user satisfaction rate of 80% and above with the Direct Rating method, compared to only 26% for the New Random Proportional method. Moreover, the higher average satisfaction and the lower variability (Table 6) around the mean for the Direct Rating method make it a more powerful tool for constructing membership functions. Another point of observation is the very high median value in the Direct Rating method.

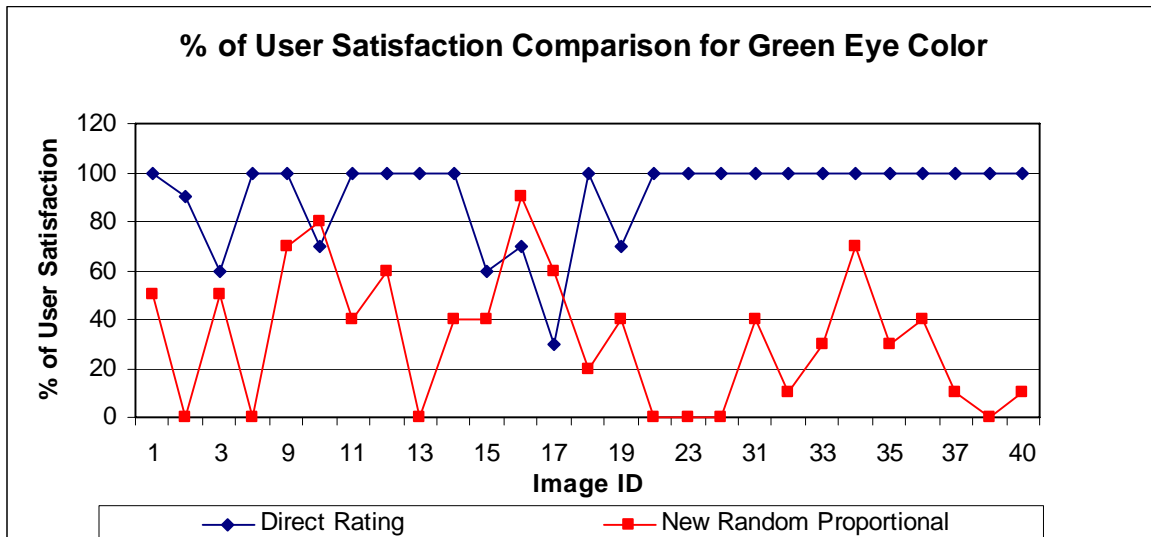
Table 6: Statistical Analysis of Users' Satisfaction for Blue Color

	Average	Standard Deviation	Median
Direct Rating	82.96	23.50	100
New Random Proportional	45.55	34.79	40

8.2 Analysis of Testing Results for Green Eye Color Attribute

Figure 17 compares users' satisfaction for Direct Rating and New Random Proportional for the green eye color attribute. The results for green color are quite similar to that for the blue color attribute. Under Direct Rating method, 21 out of 27 images have user satisfaction rate higher than 80%, whereas the New Random Proportional method has only 2 out of 27 images showing user satisfaction rate higher than 80%.

Figure 17: Comparison of User Satisfaction for Green Color



The statistical analysis of users' satisfaction for green color shows much higher average and lower standard deviation for the Direct Rating method than for the New Random Proportional method (Table 7).

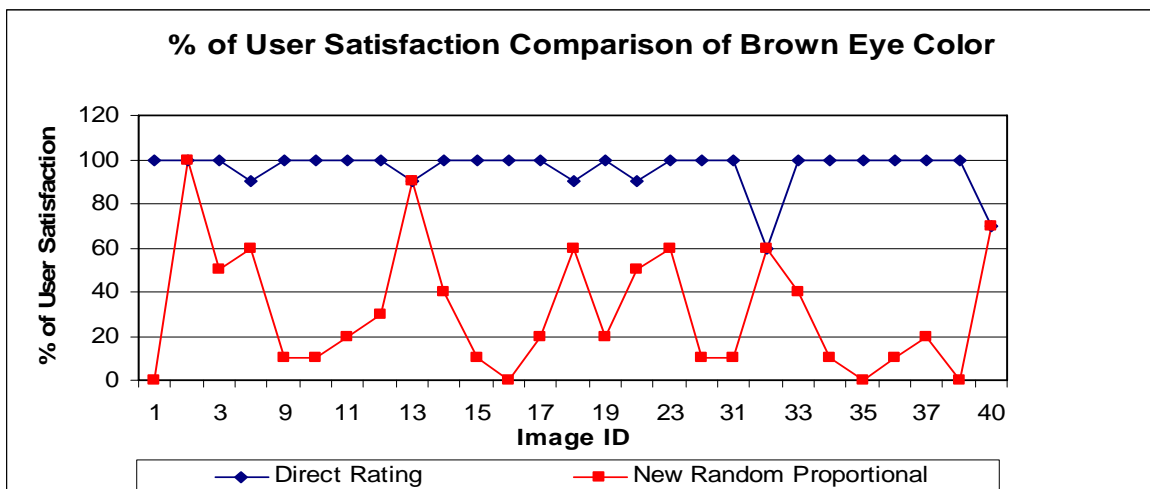
Table 7: Statistical Analysis of Users' Satisfaction for Green Color

	Average	Standard Deviation	Median
Direct Rating	90.74	18.17	100
New Random Proportional	32.59	27.68	40

8.3 Analysis of Testing Result for Brown Eye Color Attribute

The comparison graph (Figure 18) for brown color shows that 25 out of 27 images have a users' satisfaction higher than 80% for the Direct Rating method, while only 2 out of 27 images show 80% and above users' satisfaction for the New Random Proportional method.

Figure 18: Comparison of User Satisfaction for Brown Color



Statistical analysis of the results of brown color further confirms that Direct Rating is a better method of determining hedges relative to New Random Proportional method.

Table 8: Statistical Analysis of Users' Satisfaction for Brown Color

	Average	Standard Deviation	Median
Direct Rating	95.92	9.71	100
New Random Proportional	31.85	28.69	20

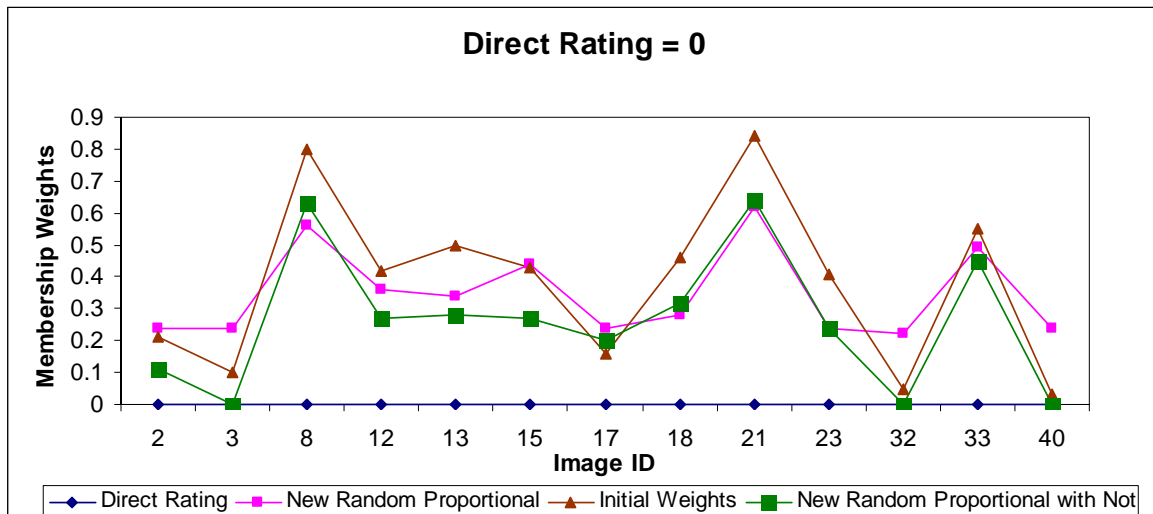
8.4 Results

The statistical analysis of the results proves that the Direct Rating method is better than the New Random Proportional method. Based on Figures 16-18, the Direct Rating achieved higher user satisfaction compared to the New Random Proportional method. It is to be noted that Direct Rating method does not give 100% users' satisfaction for all the images because of the subjective nature of the queries and the underlying principle of choosing maximum frequency value.

The membership weights assigned after the training session for the New Random Proportional method with "Not" were compared with the weights assigned by the New Random Proportional method. For the images where the fuzzy function did not apply, the membership weights approached zero faster with the inclusion of "Not" in the New Random Proportional method, as shown in Figure 19. For example, for image 32, the final membership weight was zero for the New Random Proportional method with "Not", compared to 0.22 for the New Random Proportional method.

New Random Proportional with No method causes exclusion of the image from the evaluating attribute. Hence New Random Proportional with Not is better method than New Random Proportional. For facial features, such as Nose, the modifiers could be small, average or big but the "Not" modifier does not make sense. Therefore, New Random Proportional with Not is applicable to only those facial features which can have a "Not" modifier range.

Figure 19: Membership Weight Comparison



8.5 Future Work

There are many issues that should be resolved for the current prototype as discussed in section 5.2. This paper tries to resolve one of the most important issues and suggests that the Direct Rating is a better method of constructing membership functions.

One suggested area of the future work is to find a method for choosing a subset of the users' community that will represent the whole community during the training period.

List of References

- [1] Joy, K., Dattatri, S., “*Implementing a Fuzzy Relational Database and Querying System with Community Defined Membership Values*”, VCU Directed Research Report, <http://www.people.vcu.edu/%7Elparker/DBGGroup.html>, December 2004
- [2] Lee, S.S., “*Fuzzy Membership Function Initial Values: Comparing Initialization Methods that Expedite Convergence*”, VCU Master Thesis Report, <http://www.people.vcu.edu/%7Elparker/DBGGroup.html>, June 2005
- [3] Crabtree, P., “*Fuzzy Relation Database and Querying System with Compound Query Capabilities*”, VCU Directed Research Report, <http://www.people.vcu.edu/%7Elparker/DBGGroup.html>, May 2005
- [4] Duke, D.M., “*The determination of Synonyms for Fuzzy Modifiers in a Fuzzy Database based on User Community Consensus*”, VCU Directed Research Report, <http://www.people.vcu.edu/%7Elparker/DBGGroup.html>, June 2005
- [5] Sanghi, S., “*Determining Membership Function Values to Optimize Retrieval in a Fuzzy Relational Database*”, VCU Directed Research Report, <http://www.people.vcu.edu/%7Elparker/DBGGroup.html>, May 2005
- [6] Mehrotra, A., “*Natural Language Interface: Finding the best possible way to handle potential synonyms*”, VCU Master Thesis Report, <http://www.people.vcu.edu/%7Elparker/DBGGroup.html>, May 2005
- [7] Bhootra, R. and Mehrotra, A., “*Overview of Natural Language Interfaces*”, VCU Directed Research Report, <http://www.people.vcu.edu/%7Elparker/DBGGroup.html>, December 2004
- [8] Crouse, B. and Pandrinath, A., “*Spatial Databases*”, VCU Directed Research Report, <http://www.people.vcu.edu/%7Elparker/DBGGroup.html>, December 2004
- [9] Codd, E. F., “*A relational model for large shared data banks*”, Communications of the ACM, 13, 6 (1970), 377-387.
- [10] Date, C. J. (2004). “*A Note on Relation-Valued Attributes*”, in An Introduction to Database Systems (8th ed.), pp. 373-375.

- [11] Date, C. J. (2003). “*On Fuzzy Databases*”, from Database Debunkings, [http://www.dbdebunk.com,\(01/15/2005\)](http://www.dbdebunk.com,(01/15/2005))
- [12] Zadeh, L. A. (1965). “*Fuzzy Sets*”, Information and Control, 8, 338-353.
- [13] Raju, K.V.S.V.N., Majumdar, Arun K. (1988), “*Fuzzy Functional Dependencies and Lossless Join Decomposition of Fuzzy Relational Database Systems*”, ACM Transactions on Database Systems, 13, 129-166
- [14] Bosc, P., Pivert, O. (2000), “*SQLf Query Functionality on Top of a Regular Relational Database Management System.*” Studies in Fuzziness and Soft Computing, 39, 171-191
- [15] Bhootra, R., “*Natural Language Interfaces: Comparing English front End and English Query*”, VCU Master Thesis Report, <http://www.people.vcu.edu/%7Elparker/DBGroup.html>, December 2004
- [16] List of Latin phrases From Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/List_of_Latin_phrases#S
- [17] Miller, G. A., and Fellbaum, C. (1991), “*Semantic Networks of English.*” Cognition 1991 Dec 41 (1-3) 197-229.
- [18] Mastros, J., “*Spatial Database Feasibility for Facial Characterization using Fuzzy Logic Queries*”, VCU Directed Research Report, <http://www.people.vcu.edu/%7Elparker/DBGroup.html>, September 2005.

APPENDIX A

Initialization Weights for New Random Proportional original or with “Not”

Appendix A contains the initialized weights that are created by running the stored procedure called Initialize_NProp_Random for the color table. These initialized weights are used both for New Random Proportional and New Random Proportional with “Not” methods. This appendix contains one table per each eye color.

Table A.1: Initialization Values for Blue Eye Color

ID	Color	Weight	ID	Color	Weight
1	Blue	0.08	19	Blue	0.03
2	Blue	0.21	21	Blue	0.84
3	Blue	0.1	23	Blue	0.41
8	Blue	0.8	24	Blue	0.03
9	Blue	0.21	31	Blue	0.28
10	Blue	0.36	32	Blue	0.05
11	Blue	0.54	33	Blue	0.55
12	Blue	0.42	34	Blue	0.63
13	Blue	0.5	35	Blue	0.21
14	Blue	0.02	36	Blue	0.57
15	Blue	0.43	37	Blue	0.74
16	Blue	0.77	38	Blue	0.18
17	Blue	0.16	40	Blue	0.03
18	Blue	0.46			

Table A.2: Initialization Values for Green Eye Color

ID	Color	Weight	ID	Color	Weight
1	Green	0.46	19	Green	0.49
2	Green	0.21	21	Green	0.08
3	Green	0.45	23	Green	0.41
8	Green	0.1	24	Green	0.48
9	Green	0.21	31	Green	0.36
10	Green	0.32	32	Green	0.05
11	Green	0.23	33	Green	0.22
12	Green	0.42	34	Green	0.18
13	Green	0.25	35	Green	0.21
14	Green	0.49	36	Green	0.22
15	Green	0.43	37	Green	0.13
16	Green	0.12	38	Green	0.18
17	Green	0.42	40	Green	0.48
18	Green	0.46			

Table A.3: Initialization Values for Brown Eye Color

ID	Color	Weight	ID	Color	Weight
1	Brown	0.46	19	Brown	0.03
2	Brown	0.58	21	Brown	0.08
3	Brown	0.1	23	Brown	0.17
8	Brown	0.1	24	Brown	0.03
9	Brown	0.58	31	Brown	0.36
10	Brown	0.36	32	Brown	0.9
11	Brown	0.23	33	Brown	0.55
12	Brown	0.17	34	Brown	0.18
13	Brown	0.5	35	Brown	0.59
14	Brown	0.49	36	Brown	0.57
15	Brown	0.13	37	Brown	0.13
16	Brown	0.77	38	Brown	0.64
17	Brown	0.42	40	Brown	0.03
18	Brown	0.08			

APPENDIX B

History of User Evaluation for New Random Proportional in the Training Phase

Appendix B contains the history of the user evaluations to set the membership values for the New Random Proportional method during the training phase with one table per image. The last row of each table contains the final convergence membership value for the image after training phase.

Table B.1: History of New Random Proportional Weights for Image 1

ID	BlueWeight	GreenWeight	BrownWeight
1	0.12	0.45	0.45
1	0.12	0.44	0.44
1	0.14	0.43	0.43
1	0.16	0.42	0.42
1	0.18	0.44	0.44
1	0.20	0.43	0.43
1	0.20	0.41	0.41
1	0.20	0.41	0.41
1	0.21	0.39	0.39
1	0.21	0.38	0.38
1	0.20	0.37	0.37
1	0.20	0.36	0.36
1	0.20	0.33	0.33
1	0.20	0.32	0.32
1	0.22	0.31	0.31
1	0.22	0.3	0.30
1	0.21	0.32	0.28
1	0.22	0.31	0.27
1	0.22	0.3	0.26
1	0.21	0.27	0.24
1	0.21	0.27	0.24
1	0.22	0.26	0.22
1	0.21	0.28	0.24

Table B.2: History of New Random Proportional Weights for Image 2

ID	BlueWeight	GreenWeight	BrownWeight
2	0.2	0.21	0.56
2	0.2	0.21	0.55
2	0.2	0.2	0.54
2	0.2	0.22	0.53
2	0.2	0.21	0.52
2	0.20	0.22	0.51
2	0.2	0.22	0.52
2	0.2	0.22	0.52
2	0.2	0.22	0.51
2	0.2	0.22	0.5
2	0.2	0.24	0.49
2	0.2	0.26	0.48
2	0.2	0.25	0.45
2	0.2	0.27	0.45
2	0.2	0.29	0.43
2	0.2	0.31	0.43
2	0.2	0.3	0.42
2	0.2	0.32	0.44
2	0.2	0.34	0.42
2	0.2	0.4	0.42
2	0.2	0.4	0.44
2	0.22	0.4	0.46
2	0.24	0.42	0.48

Table B.3: History of New Random Proportional Weights for Image 3

ID	BlueWeight	GreenWeight	BrownWeight
3	0.14	0.47	0.14
3	0.14	0.49	0.14
3	0.16	0.51	0.16
3	0.18	0.53	0.18
3	0.2	0.52	0.2
3	0.20	0.52	0.2
3	0.2	0.52	0.2
3	0.2	0.52	0.2
3	0.2	0.52	0.2
3	0.2	0.52	0.2
3	0.2	0.52	0.2
3	0.2	0.51	0.2
3	0.2	0.52	0.2
3	0.2	0.52	0.2
3	0.2	0.53	0.2
3	0.2	0.53	0.20
3	0.2	0.52	0.2
3	0.2	0.51	0.22
3	0.2	0.52	0.21
3	0.2	0.54	0.2
3	0.2	0.54	0.2
3	0.22	0.54	0.2
3	0.24	0.56	0.2

Table B.4: History of New Random Proportional Weights for Image 8

ID	BlueWeight	GreenWeight	BrownWeight
8	0.78	0.14	0.14
8	0.78	0.14	0.14
8	0.77	0.16	0.16
8	0.76	0.18	0.18
8	0.75	0.2	0.2
8	0.74	0.2	0.2
8	0.72	0.2	0.22
8	0.72	0.2	0.21
8	0.7	0.2	0.22
8	0.7	0.2	0.21
8	0.68	0.2	0.2
8	0.67	0.2	0.2
8	0.65	0.2	0.2
8	0.65	0.2	0.22
8	0.63	0.2	0.24
8	0.63	0.22	0.26
8	0.62	0.21	0.25
8	0.6	0.2	0.27
8	0.57	0.2	0.26
8	0.54	0.2	0.24
8	0.54	0.2	0.24
8	0.54	0.2	0.26
8	0.56	0.22	0.25

Table B.5: History of New Random Proportional Weights for Image 9

ID	BlueWeight	GreenWeight	BrownWeight
9	0.2	0.2	0.56
9	0.2	0.2	0.55
9	0.2	0.2	0.54
9	0.22	0.2	0.53
9	0.21	0.2	0.52
9	0.20	0.2	0.51
9	0.2	0.2	0.52
9	0.2	0.2	0.52
9	0.21	0.2	0.51
9	0.21	0.2	0.53
9	0.2	0.2	0.52
9	0.2	0.2	0.51
9	0.2	0.2	0.49
9	0.2	0.2	0.49
9	0.22	0.2	0.47
9	0.22	0.2	0.47
9	0.21	0.22	0.45
9	0.22	0.21	0.47
9	0.2	0.2	0.44
9	0.2	0.2	0.44
9	0.2	0.2	0.46
9	0.2	0.2	0.48
9	0.2	0.22	0.5

Table B.6: History of New Random Proportional Weights for Image 10

ID	BlueWeight	GreenWeight	BrownWeight
10	0.4	0.33	0.35
10	0.4	0.33	0.35
10	0.39	0.32	0.34
10	0.41	0.31	0.33
10	0.39	0.33	0.32
10	0.41	0.32	0.31
10	0.38	0.36	0.3
10	0.38	0.36	0.28
10	0.4	0.38	0.27
10	0.42	0.37	0.26
10	0.44	0.36	0.25
10	0.43	0.34	0.24
10	0.47	0.33	0.22
10	0.47	0.32	0.21
10	0.47	0.31	0.2
10	0.47	0.3	0.20
10	0.49	0.32	0.2
10	0.53	0.34	0.2
10	0.54	0.33	0.2
10	0.52	0.34	0.2
10	0.52	0.34	0.2
10	0.52	0.33	0.2
10	0.54	0.32	0.22

Table B.7: History of New Random Proportional Weights for Image 11

ID	BlueWeight	GreenWeight	BrownWeight
11	0.55	0.24	0.21
11	0.55	0.24	0.21
11	0.54	0.22	0.2
11	0.56	0.24	0.2
11	0.55	0.22	0.2
11	0.54	0.21	0.2
11	0.52	0.22	0.2
11	0.52	0.22	0.22
11	0.54	0.2	0.21
11	0.53	0.2	0.2
11	0.52	0.2	0.2
11	0.52	0.2	0.2
11	0.53	0.2	0.2
11	0.53	0.2	0.2
11	0.54	0.2	0.2
11	0.54	0.2	0.20
11	0.53	0.22	0.2
11	0.57	0.21	0.22
11	0.57	0.2	0.21
11	0.57	0.2	0.2
11	0.57	0.2	0.2
11	0.57	0.2	0.2
11	0.55	0.22	0.22

Table B.8: History of New Random Proportional Weights for Image 12

ID	BlueWeight	GreenWeight	BrownWeight
12	0.39	0.44	0.21
12	0.39	0.46	0.21
12	0.38	0.48	0.2
12	0.37	0.5	0.2
12	0.39	0.49	0.2
12	0.38	0.51	0.2
12	0.36	0.52	0.22
12	0.36	0.52	0.21
12	0.38	0.52	0.2
12	0.4	0.52	0.2
12	0.39	0.52	0.2
12	0.38	0.51	0.2
12	0.36	0.52	0.2
12	0.36	0.52	0.2
12	0.34	0.52	0.2
12	0.34	0.52	0.20
12	0.36	0.54	0.2
12	0.36	0.53	0.22
12	0.33	0.52	0.21
12	0.32	0.52	0.2
12	0.32	0.52	0.2
12	0.34	0.52	0.2
12	0.36	0.52	0.22

Table B.9: History of New Random Proportional Weights for Image 13

ID	BlueWeight	GreenWeight	BrownWeight
13	0.48	0.22	0.52
13	0.48	0.22	0.54
13	0.47	0.21	0.56
13	0.45	0.2	0.58
13	0.47	0.2	0.6
13	0.45	0.2	0.62
13	0.43	0.2	0.63
13	0.43	0.2	0.63
13	0.42	0.2	0.65
13	0.41	0.2	0.64
13	0.39	0.2	0.66
13	0.38	0.2	0.65
13	0.39	0.2	0.66
13	0.39	0.2	0.66
13	0.37	0.2	0.67
13	0.37	0.2	0.67
13	0.36	0.2	0.66
13	0.33	0.2	0.65
13	0.31	0.2	0.69
13	0.3	0.2	0.69
13	0.3	0.2	0.71
13	0.32	0.2	0.7
13	0.34	0.22	0.72

Table B.10: History of New Random Proportional Weights for Image 14

ID	BlueWeight	GreenWeight	BrownWeight
14	0.06	0.48	0.48
14	0.06	0.47	0.47
14	0.08	0.49	0.45
14	0.1	0.48	0.44
14	0.12	0.5	0.46
14	0.14	0.49	0.45
14	0.18	0.5	0.43
14	0.18	0.5	0.43
14	0.2	0.52	0.42
14	0.2	0.51	0.41
14	0.2	0.5	0.39
14	0.2	0.49	0.38
14	0.2	0.5	0.36
14	0.2	0.5	0.36
14	0.22	0.51	0.35
14	0.22	0.51	0.34
14	0.21	0.5	0.33
14	0.2	0.49	0.35
14	0.21	0.52	0.34
14	0.2	0.53	0.32
14	0.2	0.53	0.32
14	0.22	0.53	0.31
14	0.21	0.55	0.33

Table B.11: History of New Random Proportional Weights for Image 15

ID	BlueWeight	GreenWeight	BrownWeight
15	0.41	0.45	0.16
15	0.41	0.44	0.16
15	0.43	0.46	0.18
15	0.42	0.48	0.2
15	0.41	0.47	0.2
15	0.39	0.49	0.2
15	0.37	0.5	0.2
15	0.37	0.5	0.2
15	0.36	0.52	0.2
15	0.38	0.51	0.2
15	0.37	0.53	0.2
15	0.39	0.52	0.2
15	0.39	0.53	0.2
15	0.39	0.53	0.2
15	0.37	0.52	0.2
15	0.37	0.52	0.2
15	0.39	0.54	0.2
15	0.4	0.56	0.2
15	0.41	0.58	0.2
15	0.42	0.55	0.2
15	0.42	0.55	0.2
15	0.42	0.55	0.2
15	0.44	0.54	0.22

Table B.12: History of New Random Proportional Weights for Image 16

ID	BlueWeight	GreenWeight	BrownWeight
16	0.75	0.16	0.75
16	0.75	0.16	0.75
16	0.74	0.18	0.74
16	0.73	0.2	0.73
16	0.72	0.2	0.72
16	0.71	0.2	0.71
16	0.69	0.2	0.72
16	0.69	0.2	0.72
16	0.68	0.21	0.7
16	0.67	0.21	0.7
16	0.66	0.2	0.68
16	0.65	0.2	0.67
16	0.63	0.2	0.65
16	0.63	0.2	0.65
16	0.61	0.2	0.63
16	0.61	0.22	0.63
16	0.6	0.21	0.62
16	0.61	0.22	0.61
16	0.62	0.24	0.59
16	0.6	0.21	0.59
16	0.6	0.21	0.57
16	0.6	0.2	0.55
16	0.62	0.2	0.57

Table B.13: History of New Random Proportional Weights for Image 17

ID	BlueWeight	GreenWeight	BrownWeight
17	0.2	0.44	0.41
17	0.2	0.46	0.39
17	0.2	0.48	0.38
17	0.2	0.5	0.37
17	0.2	0.49	0.39
17	0.20	0.51	0.38
17	0.2	0.52	0.36
17	0.2	0.52	0.36
17	0.2	0.52	0.34
17	0.2	0.51	0.33
17	0.2	0.5	0.32
17	0.2	0.52	0.31
17	0.2	0.53	0.3
17	0.2	0.53	0.28
17	0.2	0.54	0.27
17	0.2	0.54	0.26
17	0.2	0.53	0.25
17	0.22	0.52	0.27
17	0.22	0.53	0.26
17	0.21	0.52	0.24
17	0.21	0.52	0.24
17	0.22	0.52	0.22
17	0.24	0.54	0.24

Table B.14: History of New Random Proportional Weights for Image 18

ID	BlueWeight	GreenWeight	BrownWeight
18	0.44	0.45	0.12
18	0.44	0.44	0.12
18	0.43	0.43	0.14
18	0.42	0.42	0.16
18	0.44	0.41	0.18
18	0.43	0.39	0.2
18	0.41	0.37	0.22
18	0.41	0.37	0.24
18	0.39	0.36	0.26
18	0.38	0.35	0.28
18	0.37	0.34	0.3
18	0.36	0.33	0.28
18	0.32	0.32	0.3
18	0.32	0.31	0.32
18	0.3	0.3	0.34
18	0.3	0.28	0.36
18	0.28	0.27	0.38
18	0.27	0.26	0.4
18	0.25	0.25	0.44
18	0.24	0.22	0.44
18	0.24	0.22	0.46
18	0.26	0.21	0.48
18	0.28	0.22	0.5

Table B.15: History of New Random Proportional Weights for Image 19

ID	BlueWeight	GreenWeight	BrownWeight
19	0.07	0.48	0.07
19	0.07	0.5	0.07
19	0.09	0.49	0.09
19	0.11	0.51	0.11
19	0.13	0.53	0.13
19	0.15	0.52	0.15
19	0.18	0.5	0.16
19	0.18	0.5	0.18
19	0.2	0.52	0.2
19	0.2	0.51	0.2
19	0.2	0.53	0.2
19	0.2	0.52	0.2
19	0.2	0.5	0.2
19	0.2	0.5	0.2
19	0.2	0.51	0.2
19	0.2	0.51	0.2
19	0.22	0.53	0.2
19	0.24	0.52	0.2
19	0.21	0.52	0.2
19	0.2	0.53	0.2
19	0.2	0.53	0.2
19	0.2	0.53	0.2
19	0.20	0.55	0.22

Table B.16: History of New Random Proportional Weights for Image 21

ID	BlueWeight	GreenWeight	BrownWeight
21	0.82	0.12	0.12
21	0.82	0.12	0.12
21	0.81	0.14	0.14
21	0.8	0.16	0.16
21	0.79	0.18	0.18
21	0.78	0.2	0.2
21	0.76	0.2	0.22
21	0.76	0.2	0.24
21	0.74	0.2	0.26
21	0.74	0.2	0.28
21	0.73	0.2	0.3
21	0.72	0.2	0.32
21	0.7	0.2	0.34
21	0.7	0.2	0.36
21	0.67	0.2	0.4
21	0.67	0.2	0.4
21	0.66	0.2	0.42
21	0.64	0.2	0.41
21	0.62	0.2	0.45
21	0.6	0.2	0.45
21	0.6	0.2	0.47
21	0.6	0.2	0.49
21	0.62	0.22	0.51

Table B.17: History of New Random Proportional Weights for Image 23

ID	BlueWeight	GreenWeight	BrownWeight
23	0.38	0.39	0.21
23	0.38	0.38	0.21
23	0.37	0.37	0.22
23	0.36	0.36	0.24
23	0.38	0.35	0.26
23	0.37	0.34	0.25
23	0.34	0.32	0.27
23	0.34	0.32	0.29
23	0.32	0.3	0.31
23	0.32	0.3	0.33
23	0.31	0.28	0.35
23	0.3	0.27	0.37
23	0.27	0.26	0.41
23	0.27	0.25	0.41
23	0.25	0.24	0.45
23	0.25	0.22	0.45
23	0.24	0.21	0.47
23	0.22	0.2	0.45
23	0.2	0.2	0.49
23	0.2	0.2	0.49
23	0.2	0.2	0.51
23	0.22	0.2	0.53
23	0.24	0.22	0.52

Table B.18: History of New Random Proportional Weights for Image 24

ID	BlueWeight	GreenWeight	BrownWeight
24	0.07	0.47	0.07
24	0.07	0.45	0.07
24	0.09	0.44	0.09
24	0.11	0.43	0.11
24	0.13	0.45	0.13
24	0.15	0.44	0.15
24	0.18	0.45	0.16
24	0.18	0.45	0.18
24	0.22	0.44	0.2
24	0.22	0.43	0.2
24	0.24	0.42	0.2
24	0.22	0.41	0.2
24	0.22	0.38	0.2
24	0.22	0.38	0.2
24	0.2	0.36	0.2
24	0.2	0.36	0.2
24	0.22	0.38	0.2
24	0.24	0.37	0.2
24	0.24	0.37	0.2
24	0.26	0.41	0.2
24	0.26	0.41	0.2
24	0.28	0.41	0.2
24	0.27	0.43	0.22

Table B.19: History of New Random Proportional Weights for Image 31

ID	BlueWeight	GreenWeight	BrownWeight
31	0.28	0.35	0.35
31	0.28	0.35	0.35
31	0.27	0.34	0.34
31	0.29	0.33	0.33
31	0.31	0.32	0.32
31	0.30	0.31	0.31
31	0.31	0.28	0.3
31	0.31	0.28	0.28
31	0.28	0.26	0.27
31	0.28	0.26	0.26
31	0.27	0.25	0.25
31	0.29	0.24	0.24
31	0.26	0.22	0.22
31	0.26	0.21	0.21
31	0.27	0.2	0.2
31	0.27	0.22	0.2
31	0.26	0.21	0.2
31	0.28	0.2	0.22
31	0.28	0.2	0.21
31	0.27	0.2	0.2
31	0.27	0.2	0.2
31	0.26	0.2	0.2
31	0.25	0.22	0.22

Table B.20: History of New Random Proportional Weights for Image 32

ID	BlueWeight	GreenWeight	BrownWeight
32	0.09	0.09	0.88
32	0.09	0.09	0.88
32	0.11	0.11	0.87
32	0.13	0.13	0.86
32	0.15	0.15	0.85
32	0.16	0.16	0.85
32	0.2	0.2	0.85
32	0.2	0.2	0.85
32	0.2	0.2	0.84
32	0.2	0.2	0.84
32	0.2	0.2	0.86
32	0.2	0.2	0.85
32	0.2	0.2	0.85
32	0.2	0.2	0.85
32	0.2	0.2	0.86
32	0.2	0.2	0.86
32	0.2	0.2	0.85
32	0.2	0.2	0.84
32	0.2	0.2	0.85
32	0.2	0.2	0.85
32	0.2	0.2	0.85
32	0.2	0.2	0.85
32	0.2	0.2	0.85
32	0.22	0.22	0.85

Table B.21: History of New Random Proportional Weights for Image 33

ID	BlueWeight	GreenWeight	BrownWeight
33	0.53	0.22	0.54
33	0.53	0.22	0.53
33	0.52	0.21	0.52
33	0.51	0.2	0.52
33	0.53	0.2	0.52
33	0.52	0.2	0.52
33	0.54	0.2	0.5
33	0.54	0.2	0.5
33	0.53	0.2	0.52
33	0.52	0.2	0.54
33	0.51	0.2	0.53
33	0.5	0.2	0.52
33	0.48	0.2	0.54
33	0.48	0.2	0.54
33	0.45	0.2	0.52
33	0.45	0.2	0.52
33	0.44	0.2	0.52
33	0.45	0.2	0.51
33	0.43	0.2	0.52
33	0.47	0.2	0.52
33	0.47	0.2	0.52
33	0.47	0.2	0.52
33	0.49	0.22	0.54

Table B.22: History of New Random Proportional Weights for Image 34

ID	BlueWeight	GreenWeight	BrownWeight
34	0.61	0.2	0.2
34	0.61	0.2	0.2
34	0.6	0.2	0.2
34	0.59	0.2	0.2
34	0.57	0.2	0.2
34	0.55	0.22	0.2
34	0.56	0.2	0.2
34	0.56	0.2	0.2
34	0.55	0.2	0.2
34	0.54	0.2	0.2
34	0.53	0.2	0.2
34	0.52	0.2	0.2
34	0.5	0.2	0.2
34	0.5	0.2	0.2
34	0.52	0.2	0.2
34	0.52	0.2	0.2
34	0.51	0.2	0.2
34	0.55	0.2	0.2
34	0.56	0.2	0.2
34	0.54	0.2	0.2
34	0.54	0.2	0.2
34	0.54	0.2	0.2
34	0.53	0.22	0.22

Table B.23: History of New Random Proportional Weights for Image 35

ID	BlueWeight	GreenWeight	BrownWeight
35	0.21	0.2	0.57
35	0.21	0.2	0.55
35	0.2	0.2	0.54
35	0.22	0.2	0.53
35	0.21	0.2	0.52
35	0.22	0.2	0.51
35	0.22	0.2	0.49
35	0.22	0.2	0.49
35	0.22	0.2	0.48
35	0.22	0.2	0.47
35	0.21	0.2	0.45
35	0.2	0.2	0.44
35	0.2	0.2	0.42
35	0.2	0.2	0.42
35	0.22	0.2	0.39
35	0.22	0.2	0.39
35	0.21	0.2	0.38
35	0.22	0.2	0.4
35	0.26	0.2	0.38
35	0.28	0.2	0.38
35	0.28	0.2	0.37
35	0.27	0.2	0.39
35	0.26	0.22	0.41

Table B.24: History of New Random Proportional Weights for Image 36

ID	BlueWeight	GreenWeight	BrownWeight
36	0.54	0.2	0.55
36	0.54	0.2	0.54
36	0.53	0.2	0.53
36	0.52	0.2	0.52
36	0.51	0.2	0.52
36	0.53	0.2	0.51
36	0.54	0.2	0.49
36	0.54	0.2	0.49
36	0.53	0.2	0.48
36	0.52	0.2	0.47
36	0.52	0.2	0.45
36	0.52	0.2	0.44
36	0.53	0.2	0.42
36	0.53	0.2	0.42
36	0.52	0.2	0.39
36	0.52	0.2	0.39
36	0.51	0.2	0.38
36	0.55	0.2	0.37
36	0.56	0.2	0.34
36	0.56	0.2	0.32
36	0.56	0.2	0.32
36	0.56	0.2	0.31
36	0.55	0.22	0.33

Table B.25: History of New Random Proportional Weights for Image 37

ID	BlueWeight	GreenWeight	BrownWeight
37	0.78	0.16	0.16
37	0.78	0.16	0.16
37	0.77	0.18	0.18
37	0.79	0.20	0.20
37	0.81	0.22	0.20
37	0.83	0.21	0.20
37	0.85	0.20	0.20
37	0.85	0.20	0.20
37	0.85	0.20	0.20
37	0.85	0.20	0.20
37	0.85	0.20	0.20
37	0.85	0.20	0.20
37	0.85	0.20	0.20
37	0.85	0.20	0.20
37	0.85	0.20	0.20
37	0.85	0.20	0.20
37	0.85	0.22	0.20
37	0.85	0.21	0.20
37	0.85	0.20	0.20
37	0.83	0.20	0.20
37	0.83	0.20	0.20
37	0.83	0.20	0.20
37	0.85	0.22	0.22

Table B.26: History of New Random Proportional Weights for Image 38

ID	BlueWeight	GreenWeight	BrownWeight
38	0.22	0.20	0.63
38	0.22	0.20	0.62
38	0.21	0.20	0.61
38	0.22	0.20	0.60
38	0.21	0.20	0.59
38	0.22	0.20	0.57
38	0.22	0.20	0.54
38	0.22	0.20	0.54
38	0.22	0.20	0.53
38	0.22	0.20	0.52
38	0.24	0.20	0.51
38	0.26	0.20	0.50
38	0.27	0.20	0.48
38	0.27	0.20	0.48
38	0.25	0.20	0.45
38	0.25	0.20	0.45
38	0.24	0.22	0.44
38	0.26	0.21	0.43
38	0.3	0.20	0.41
38	0.32	0.20	0.41
38	0.32	0.20	0.39
38	0.31	0.20	0.38
38	0.30	0.22	0.40

Table B.27: History of New Random Proportional Weights for Image 40

ID	BlueWeight	GreenWeight	BrownWeight
40	0.07	0.50	0.07
40	0.07	0.49	0.07
40	0.09	0.48	0.09
40	0.11	0.47	0.11
40	0.13	0.45	0.13
40	0.15	0.44	0.15
40	0.18	0.42	0.16
40	0.18	0.42	0.18
40	0.20	0.41	0.20
40	0.20	0.39	0.22
40	0.20	0.38	0.21
40	0.20	0.37	0.22
40	0.20	0.34	0.24
40	0.20	0.33	0.26
40	0.20	0.32	0.25
40	0.20	0.31	0.27
40	0.20	0.30	0.29
40	0.20	0.28	0.31
40	0.20	0.30	0.30
40	0.20	0.27	0.31
40	0.20	0.27	0.31
40	0.22	0.26	0.30
40	0.24	0.28	0.28

APPENDIX C

History of User Evaluation for New Random Proportional with “Not” in the Training Phase

Appendix C contains the history of the user evaluations to set the membership values for the New Random Proportional with “Not” method during the training phase. This appendix contains one table per image. The last row of each table contains the final convergence membership value for the image after training phase.

Table C.1: History of New Random Proportional with “Not” Weights for Image 1

ID	BlueWeight	GreenWeight	BrownWeight
1	0.16	0.48	0.48
1	0.18	0.5	0.5
1	0.2	0.52	0.52
1	0.2	0.52	0.52
1	0.10	0.45	0.48
1	0.14	0.43	0.45
1	0.14	0.22	0.43
1	0.16	0.43	0.41
1	0.14	0.41	0.43
1	0.16	0.38	0.41
1	0.18	0.37	0.38
1	0.20	0.36	0.36
1	0.20	0.32	0.32

Table C.2: History of New Random Proportional with “Not” Weights for Image 2

ID	BlueWeight	GreenWeight	BrownWeight
2	0.19	0.2	0.56
2	0.21	0.2	0.55
2	0.2	0.2	0.54
2	0.2	0.2	0.53
2	0.19	0.22	0.56
2	0.15	0.26	0.54
2	0.15	0.26	0.52
2	0.13	0.28	0.5
2	0.11	0.3	0.48
2	0.09	0.32	0.45
2	0.07	0.3	0.43
2	0.09	0.32	0.41
2	0.11	0.31	0.38

Table C.3: History of New Random Proportional with “Not” Weights for Image 3

ID	BlueWeight	GreenWeight	BrownWeight
3	0.2	0.47	0.12
3	0.2	0.49	0.14
3	0.2	0.51	0.16
3	0.2	0.53	0.18
3	0.08	0.47	0.12
3	0.03	0.44	0.08
3	0.03	0.46	0.08
3	0.05	0.45	0.1
3	0.03	0.47	0.12
3	0	0.45	0.14
3	0	0.47	0.16
3	0	0.49	0.14
3	0	0.51	0.12

Table C.4: History of New Random Proportional with “Not” Weights for Image 8

ID	BlueWeight	GreenWeight	BrownWeight
8	0.82	0.12	0.12
8	0.84	0.14	0.14
8	0.86	0.16	0.16
8	0.84	0.18	0.18
8	0.78	0.12	0.12
8	0.74	0.12	0.16
8	0.74	0.12	0.16
8	0.72	0.14	0.18
8	0.7	0.12	0.2
8	0.68	0.1	0.22
8	0.67	0.08	0.24
8	0.65	0.1	0.22
8	0.63	0.08	0.21

Table C.5: History of New Random Proportional with “Not” Weights for Image 9

ID	BlueWeight	GreenWeight	BrownWeight
9	0.2	0.2	0.56
9	0.2	0.2	0.55
9	0.2	0.2	0.54
9	0.2	0.2	0.53
9	0.2	0.22	0.56
9	0.24	0.22	0.54
9	0.24	0.22	0.52
9	0.22	0.21	0.51
9	0.2	0.19	0.49
9	0.22	0.21	0.47
9	0.21	0.19	0.45
9	0.2	0.21	0.43
9	0.2	0.2	0.41

Table C.6: History of New Random Proportional with “Not” Weights for Image 10

ID	BlueWeight	GreenWeight	BrownWeight
10	0.38	0.31	0.38
10	0.4	0.3	0.4
10	0.38	0.28	0.42
10	0.36	0.27	0.44
10	0.38	0.34	0.34
10	0.35	0.31	0.3
10	0.35	0.31	0.3
10	0.39	0.28	0.27
10	0.41	0.27	0.25
10	0.43	0.26	0.23
10	0.41	0.24	0.21
10	0.39	0.22	0.19
10	0.41	0.2	0.17

Table C.7: History of New Random Proportional with “Not” Weights for Image 11

ID	BlueWeight	GreenWeight	BrownWeight
11	0.53	0.21	0.22
11	0.52	0.2	0.21
11	0.52	0.2	0.2
11	0.52	0.2	0.2
11	0.53	0.25	0.21
11	0.54	0.21	0.17
11	0.54	0.21	0.17
11	0.56	0.19	0.15
11	0.58	0.17	0.13
11	0.6	0.15	0.11
11	0.59	0.16	0.09
11	0.61	0.14	0.07
11	0.6	0.12	0.05

Table C.8: History of New Random Proportional with “Not” Weights for Image 12

ID	BlueWeight	GreenWeight	BrownWeight
12	0.44	0.44	0.19
12	0.46	0.46	0.21
12	0.48	0.48	0.2
12	0.5	0.5	0.2
12	0.41	0.44	0.19
12	0.36	0.46	0.15
12	0.36	0.48	0.15
12	0.38	0.47	0.16
12	0.36	0.49	0.14
12	0.34	0.48	0.16
12	0.32	0.5	0.14
12	0.3	0.52	0.12
12	0.27	0.52	0.1

Table C.9: History of New Random Proportional with “Not” Weights for Image 13

ID	BlueWeight	GreenWeight	BrownWeight
13	0.52	0.24	0.52
13	0.52	0.22	0.52
13	0.52	0.21	0.52
13	0.52	0.2	0.52
13	0.48	0.23	0.52
13	0.43	0.19	0.54
13	0.43	0.19	0.56
13	0.41	0.17	0.58
13	0.38	0.15	0.6
13	0.36	0.13	0.62
13	0.33	0.11	0.64
13	0.31	0.09	0.63
13	0.28	0.07	0.62

Table C.10: History of New Random Proportional with “Not” Weights for Image 14

ID	BlueWeight	GreenWeight	BrownWeight
14	0.18	0.51	0.51
14	0.2	0.53	0.53
14	0.2	0.52	0.52
14	0.2	0.52	0.52
14	0	0.48	0.51
14	0	0.5	0.49
14	0	0.52	0.47
14	0	0.52	0.45
14	0	0.5	0.43
14	0	0.48	0.41
14	0	0.47	0.38
14	0	0.45	0.36
14	0	0.43	0.32

Table C.11: History of New Random Proportional with “Not” Weights for Image 15

ID	BlueWeight	GreenWeight	BrownWeight
15	0.42	0.45	0.11
15	0.44	0.47	0.13
15	0.46	0.49	0.15
15	0.48	0.51	0.16
15	0.42	0.45	0.15
15	0.42	0.43	0.11
15	0.42	0.42	0.11
15	0.41	0.41	0.09
15	0.38	0.39	0.07
15	0.37	0.38	0.05
15	0.32	0.4	0.03
15	0.3	0.42	0
15	0.27	0.44	0

Table C.12: History of New Random Proportional with “Not” Weights for Image 16

ID	BlueWeight	GreenWeight	BrownWeight
16	0.79	0.14	0.79
16	0.81	0.16	0.81
16	0.83	0.18	0.83
16	0.85	0.2	0.85
16	0.76	0.14	0.75
16	0.76	0.18	0.71
16	0.76	0.18	0.71
16	0.75	0.2	0.7
16	0.73	0.2	0.68
16	0.72	0.2	0.66
16	0.7	0.2	0.64
16	0.68	0.2	0.62
16	0.67	0.2	0.6

Table C.13: History of New Random Proportional with “Not” Weights for Image 17

ID	BlueWeight	GreenWeight	BrownWeight
17	0.2	0.44	0.44
17	0.2	0.46	0.46
17	0.2	0.48	0.48
17	0.2	0.5	0.5
17	0.14	0.44	0.41
17	0.18	0.46	0.38
17	0.18	0.48	0.36
17	0.2	0.5	0.35
17	0.18	0.52	0.32
17	0.2	0.52	0.3
17	0.18	0.51	0.27
17	0.2	0.53	0.25
17	0.2	0.51	0.23

Table C.14: History of New Random Proportional with “Not” Weights for Image 18

ID	BlueWeight	GreenWeight	BrownWeight
18	0.48	0.48	0.27
18	0.5	0.5	0.26
18	0.52	0.52	0.25
18	0.52	0.52	0.24
18	0.44	0.44	0.3
18	0.44	0.42	0.34
18	0.44	0.39	0.34
18	0.42	0.37	0.38
18	0.39	0.32	0.4
18	0.37	0.3	0.42
18	0.36	0.27	0.41
18	0.34	0.25	0.43
18	0.32	0.23	0.45

Table C.15: History of New Random Proportional with “Not” Weights for Image 19

ID	BlueWeight	GreenWeight	BrownWeight
19	0.2	0.51	0.05
19	0.2	0.53	0.07
19	0.2	0.52	0.09
19	0.2	0.52	0.11
19	0.05	0.48	0.05
19	0.05	0.45	0
19	0.05	0.43	0
19	0.07	0.42	0
19	0.05	0.44	0
19	0.03	0.46	0
19	0	0.45	0
19	0	0.44	0
19	0	0.42	0

Table C.16: History of New Random Proportional with “Not” Weights for Image 21

ID	BlueWeight	GreenWeight	BrownWeight
21	0.86	0.1	0.1
21	0.85	0.12	0.12
21	0.85	0.14	0.14
21	0.85	0.16	0.16
21	0.82	0.06	0.1
21	0.78	0	0.14
21	0.78	0	0.14
21	0.76	0	0.16
21	0.74	0	0.18
21	0.72	0	0.2
21	0.7	0	0.2
21	0.66	0	0.22
21	0.64	0	0.21

Table C.17: History of New Random Proportional with “Not” Weights for Image 23

ID	BlueWeight	GreenWeight	BrownWeight
23	0.43	0.43	0.19
23	0.45	0.45	0.21
23	0.47	0.47	0.2
23	0.49	0.49	0.2
23	0.43	0.43	0.19
23	0.38	0.41	0.22
23	0.38	0.38	0.22
23	0.36	0.36	0.24
23	0.32	0.32	0.26
23	0.3	0.3	0.28
23	0.28	0.27	0.27
23	0.26	0.25	0.29
23	0.24	0.23	0.27

Table C.18: History of New Random Proportional with “Not” Weights for Image 24

ID	BlueWeight	GreenWeight	BrownWeight
24	0.21	0.5	0.05
24	0.2	0.52	0.07
24	0.2	0.52	0.09
24	0.2	0.52	0.11
24	0.05	0.47	0
24	0.09	0.44	0
24	0.09	0.42	0
24	0.11	0.39	0
24	0.13	0.37	0
24	0.15	0.35	0
24	0.16	0.34	0
24	0.18	0.32	0
24	0.2	0.3	0

Table C.19: History of New Random Proportional with “Not” Weights for Image 31

ID	BlueWeight	GreenWeight	BrownWeight
31	0.2	0.38	0.38
31	0.2	0.4	0.4
31	0.2	0.42	0.42
31	0.2	0.44	0.44
31	0.3	0.32	0.34
31	0.34	0.32	0.3
31	0.34	0.32	0.3
31	0.38	0.31	0.27
31	0.36	0.28	0.25
31	0.38	0.27	0.23
31	0.4	0.25	0.21
31	0.39	0.24	0.19
31	0.41	0.22	0.17

Table C.20: History of New Random Proportional with “Not” Weights for Image 32

ID	BlueWeight	GreenWeight	BrownWeight
32	0.2	0.07	0.89
32	0.2	0.09	0.88
32	0.2	0.11	0.87
32	0.2	0.13	0.86
32	0.03	0.03	0.89
32	0.02	0	0.86
32	0.02	0	0.86
32	0	0	0.85
32	0	0	0.85
32	0	0	0.85
32	0	0	0.85
32	0	0	0.85
32	0	0	0.85
32	0	0	0.83

Table C.21: History of New Random Proportional with “Not” Weights for Image 33

ID	BlueWeight	GreenWeight	BrownWeight
33	0.54	0.21	0.54
33	0.53	0.2	0.53
33	0.52	0.2	0.52
33	0.52	0.2	0.52
33	0.53	0.2	0.54
33	0.54	0.2	0.52
33	0.54	0.2	0.5
33	0.53	0.18	0.52
33	0.51	0.16	0.54
33	0.49	0.14	0.53
33	0.48	0.12	0.55
33	0.47	0.14	0.54
33	0.45	0.12	0.52

Table C.22: History of New Random Proportional with “Not” Weights for Image 34

ID	BlueWeight	GreenWeight	BrownWeight
34	0.62	0.2	0.2
34	0.61	0.2	0.2
34	0.6	0.2	0.2
34	0.59	0.2	0.2
34	0.62	0.2	0.2
34	0.60	0.2	0.16
34	0.6	0.2	0.16
34	0.59	0.18	0.18
34	0.57	0.16	0.16
34	0.55	0.18	0.14
34	0.53	0.16	0.12
34	0.52	0.18	0.1
34	0.52	0.2	0.08

Table C.23: History of New Random Proportional with “Not” Weights for Image 35

ID	BlueWeight	GreenWeight	BrownWeight
35	0.2	0.2	0.57
35	0.2	0.2	0.55
35	0.2	0.2	0.54
35	0.2	0.2	0.53
35	0.22	0.2	0.56
35	0.22	0.16	0.54
35	0.22	0.16	0.52
35	0.24	0.18	0.5
35	0.26	0.16	0.48
35	0.28	0.18	0.45
35	0.3	0.16	0.43
35	0.32	0.14	0.41
35	0.31	0.12	0.38

Table C.24: History of New Random Proportional with “Not” Weights for Image 36

ID	BlueWeight	GreenWeight	BrownWeight
36	0.55	0.21	0.55
36	0.54	0.2	0.54
36	0.53	0.2	0.53
36	0.52	0.2	0.52
36	0.55	0.21	0.54
36	0.55	0.2	0.52
36	0.55	0.2	0.5
36	0.54	0.18	0.48
36	0.53	0.16	0.45
36	0.52	0.18	0.43
36	0.52	0.16	0.41
36	0.52	0.14	0.38
36	0.54	0.12	0.36

Table C.25: History of New Random Proportional with “Not” Weights for Image 37

ID	BlueWeight	GreenWeight	BrownWeight
37	0.76	0.15	0.15
37	0.78	0.16	0.16
37	0.8	0.18	0.18
37	0.82	0.20	0.20
37	0.76	0.15	0.11
37	0.76	0.11	0.07
37	0.76	0.11	0.07
37	0.78	0.09	0.05
37	0.77	0.07	0.03
37	0.79	0.05	0.00
37	0.78	0.03	0.00
37	0.8	0.00	0.00
37	0.82	0.00	0.00

Table C.26: History of New Random Proportional with “Not” Weights for Image 38

ID	BlueWeight	GreenWeight	BrownWeight
38	0.24	0.20	0.63
38	0.22	0.20	0.62
38	0.21	0.20	0.61
38	0.2	0.20	0.60
38	0.2	0.20	0.62
38	0.20	0.16	0.60
38	0.2	0.16	0.57
38	0.22	0.14	0.54
38	0.21	0.12	0.52
38	0.2	0.10	0.50
38	0.2	0.08	0.48
38	0.22	0.06	0.45
38	0.21	0.03	0.43

Table C.27: History of New Random Proportional with “Not” Weights for Image 40

ID	BlueWeight	GreenWeight	BrownWeight
40	0.20	0.50	0.05
40	0.20	0.52	0.07
40	0.20	0.52	0.09
40	0.20	0.52	0.11
40	0.00	0.45	0.05
40	0.00	0.47	0.09
40	0.00	0.44	0.09
40	0.00	0.42	0.11
40	0.00	0.39	0.13
40	0.00	0.37	0.15
40	0.00	0.34	0.16
40	0.00	0.33	0.18
40	0.00	0.31	0.20

APPENDIX D

History of User Evaluation for Direct Rating in the Training Phase

Appendix D contains the history of the user evaluations to set the membership values for the Direct Rating method during the training phase. This appendix contains one table per image. The final convergence value after training phase for direct rating is obtained by highest frequency per color.

Table D.1: History of Direct Rating Weights for Image 1

ID	BlueWeight	GreenWeight	BrownWeight
1	0.52	0.52	0
1	0.2	0.2	0.2
1	0.52	0	0
1	0	0.2	0.2
1	0.2	0	0
1	0.2	0	0
1	0.2	0	0
1	0.2	0	0
1	0.2	0	0
1	0.2	0	0
1	0	0	0
1	0	0	0
1	0.85	0	0
1	0.2	0	0
1	0.85	0	0
1	0.85	0	0
1	0.2	0.2	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0.52	0.52	0
1	0.52	0.52	0
1	0.2	0.52	0
1	0.2	0	0
1	0	0.2	0
1	0	0	0
1	0.52	0	0
1	0.52	0	0
1	0.52	0	0

Frequency Range	BlueWeight	GreenWeight	BrownWeight
0	8	21	27
0.2	11	4	2
0.52	7	4	0
0.85	3	0	0

Table D.2: History of Direct Rating Weights for Image 2

ID	BlueWeight	GreenWeight	BrownWeight
2	0.2	0.52	0
2	0	0.85	0
2	0	0.85	0
2	0	0.85	0
2	0.52	0	0
2	0.2	0.2	0
2	0	0.2	0
2	0	0.2	0
2	0	0.85	0
2	0	0.85	0.2
2	0	0.85	0
2	0	0.2	0
2	0	0.85	0
2	0	0.85	0
2	0	0.85	0
2	0	0.85	0
2	0	0.85	0
2	0	0.85	0
2	0	0.85	0
2	0	0.85	0
2	0	0.85	0
2	0	0.85	0
2	0	0.85	0
2	0.2	0.52	0
2	0	0.52	0
2	0	0	0.2
2	0	0.85	0
2	0	0.52	0
2	0	0.2	0
2	0	0.85	0

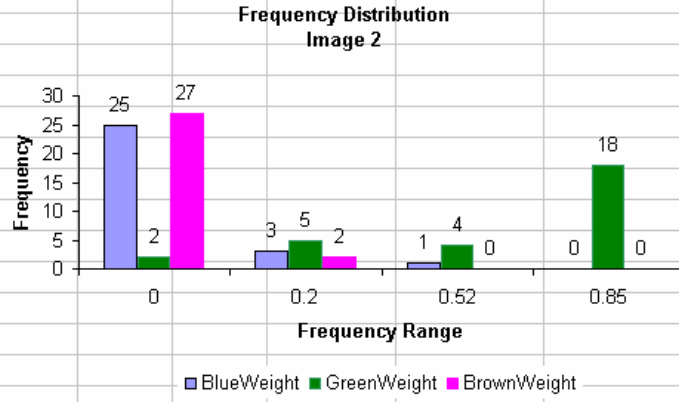


Table D.3: History of Direct Rating Weights for Image 3

ID	BlueWeight	GreenWeight	BrownWeight
3	0.52	0.52	0
3	0	0.52	0
3	0	0.2	0
3	0	0.52	0
3	0	0.52	0.2
3	0	0.2	0
3	0.2	0.2	0
3	0.2	0.2	0
3	0.2	0.52	0
3	0.2	0	0.2
3	0.52	0	0
3	0.52	0	0
3	0	0.2	0
3	0	0.2	0
3	0	0.2	0
3	0	0	0.52
3	0	0.52	0
3	0	0	0.2
3	0	0.2	0.2
3	0	0.2	0
3	0	0.52	0
3	0	0.52	0
3	0	0.52	0
3	0	0.2	0.2
3	0	0	0.2
3	0	0.2	0
3	0	0.52	0
3	0	0	0.2
3	0	0.2	0

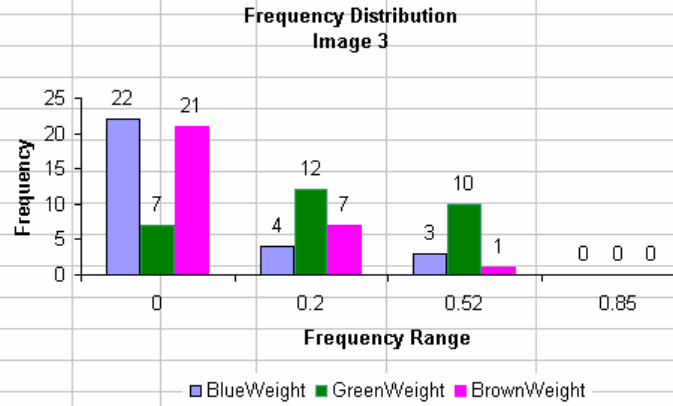


Table D.4: History of Direct Rating Weights for Image 8

ID	BlueWeight	GreenWeight	BrownWeight
8	0	0	0
8	0	0	0.52
8	0	0	0.52
8	0	0	0.85
8	0	0.85	0.85
8	0	0.2	0.52
8	0	0	0.85
8	0	0	0.52
8	0	0	0.52
8	0	0	0.85
8	0	0	0.85
8	0	0	0.52
8	0	0	0.85
8	0	0	0.85
8	0	0	0.85
8	0	0	0.85
8	0	0	0.85
8	0	0.52	0
8	0	0	0.52
8	0	0	0
8	0	0.2	0.85
8	0	0.2	0.52
8	0	0	0.2
8	0	0	0.52
8	0	0	0.52
8	0	0	0.85
8	0	0.2	0.2
8	0	0	0.85
8	0	0	0.52

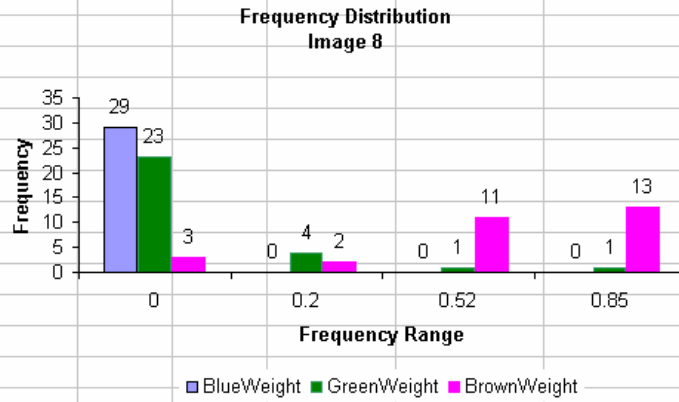


Table D.5: History of Direct Rating Weights for Image 9

ID	BlueWeight	GreenWeight	BrownWeight
9	0.52	0.52	0
9	0	0.2	0
9	0.52	0	0
9	0	0.2	0
9	0.2	0	0.52
9	0.2	0	0
9	0.2	0	0.2
9	0.2	0	0
9	0.2	0	0
9	0.2	0	0
9	0.52	0	0
9	0.52	0	0
9	0.52	0	0
9	0	0	0
9	0.2	0	0
9	0.2	0.2	0
9	0.2	0	0
9	0	0	0
9	0	0	0.2
9	0	0	0
9	0.2	0	0
9	0.2	0	0
9	0	0.2	0
9	0.2	0	0
9	0.2	0.2	0
9	0	0	0
9	0.2	0	0
9	0.2	0.2	0

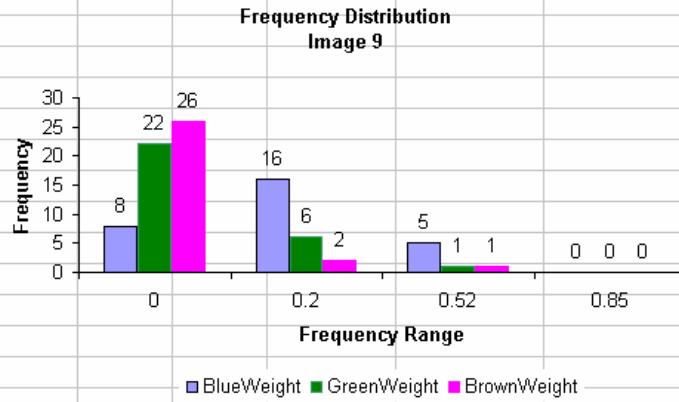


Table D.6: History of Direct Rating Weights for Image 10

ID	BlueWeight	GreenWeight	BrownWeight
10	0.85	0	0
10	0.2	0.2	0
10	0.2	0	0
10	0.52	0.2	0
10	0.52	0	0
10	0.52	0	0
10	0.2	0	0
10	0.2	0	0
10	0.2	0.2	0
10	0.2	0.2	0
10	0.52	0.2	0
10	0.52	0	0
10	0.85	0	0
10	0	0	0
10	0.52	0	0
10	0.52	0.2	0
10	0.85	0	0
10	0	0	0
10	0.2	0	0
10	0.2	0	0
10	0.2	0.2	0
10	0.2	0.2	0
10	0.85	0.52	0
10	0.52	0.2	0
10	0.85	0.52	0
10	0.85	0	0
10	0	0.85	0
10	0.52	0	0
10	0	0.2	0

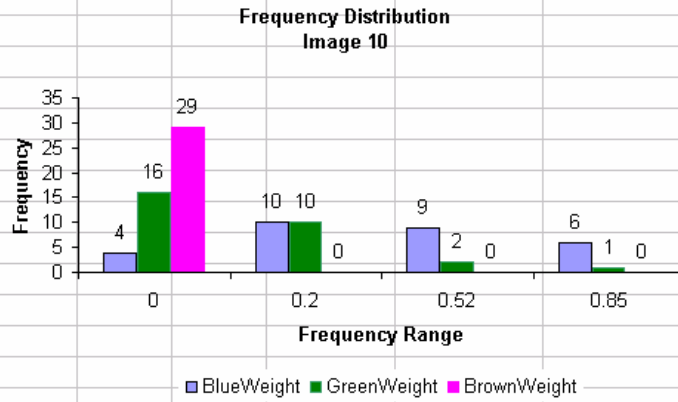


Table D.7: History of Direct Rating Weights for Image 11

ID	BlueWeight	GreenWeight	BrownWeight
11	0.85	0	0
11	0.52	0.2	0.2
11	0.85	0	0
11	0	0.85	0.2
11	0	0.2	0
11	0.2	0	0
11	0.52	0	0
11	0.52	0	0
11	0.52	0.2	0
11	0.52	0	0
11	0	0	0
11	0	0	0
11	0	0.85	0
11	0.52	0	0
11	0.52	0	0
11	0.52	0	0
11	0.2	0	0
11	0.2	0	0
11	0	0	0
11	0	0	0
11	0.85	0	0
11	0.85	0	0
11	0.2	0	0
11	0.52	0.2	0
11	0.85	0.2	0
11	0.85	0	0
11	0.85	0	0
11	0.52	0	0
11	0.52	0	0

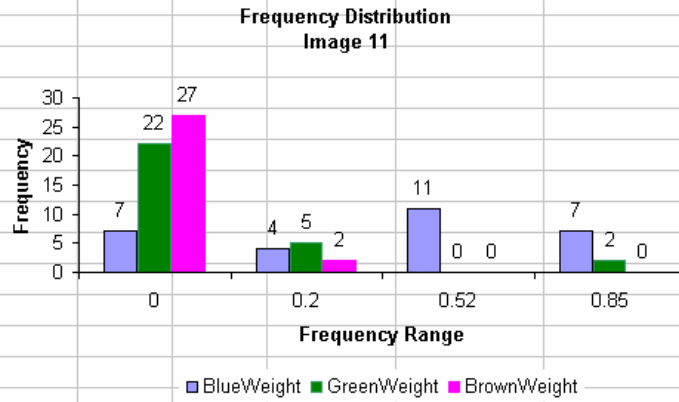


Table D.8: History of Direct Rating Weights for Image 12

ID	BlueWeight	GreenWeight	BrownWeight
12	0.52	0.85	0
12	0	0.52	0
12	0	0.52	0
12	0	0.85	0
12	0	0.2	0.2
12	0	0.52	0
12	0	0.2	0
12	0	0.2	0
12	0	0.2	0
12	0	0.2	0.2
12	0	0.2	0
12	0	0.2	0
12	0	0.2	0
12	0	0.52	0
12	0	0.52	0
12	0	0.52	0
12	0	0.52	0
12	0	0.2	0
12	0	0.85	0.2
12	0	0.85	0
12	0.2	0.52	0
12	0.2	0.52	0
12	0	0.2	0
12	0	0.2	0
12	0	0	0.2
12	0	0.52	0
12	0	0.2	0
12	0	0.2	0
12	0	0.2	0

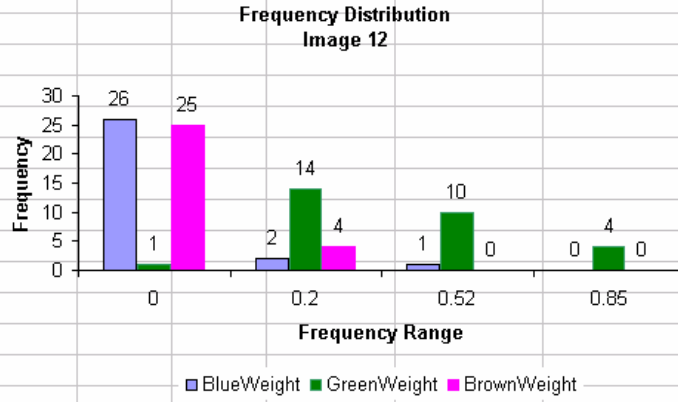


Table D.10: History of Direct Rating Weights for Image 14

ID	BlueWeight	GreenWeight	BrownWeight
14	0.85	0	0
14	0.2	0.2	0
14	0	0.52	0
14	0	0	0
14	0.2	0.2	0.2
14	0.2	0.2	0
14	0.52	0	0
14	0.52	0	0
14	0.52	0.2	0
14	0.52	0	0
14	0.52	0	0
14	0.52	0	0
14	0.52	0	0
14	0	0	0.52
14	0.85	0	0
14	0.85	0	0
14	0	0	0
14	0.2	0	0
14	0.52	0	0.2
14	0.52	0	0
14	0.52	0.2	0
14	0.52	0.2	0
14	0.2	0.2	0.2
14	0.2	0	0
14	0.52	0.2	0
14	0.52	0	0
14	0	0.85	0
14	0.52	0	0.52
14	0.2	0	0

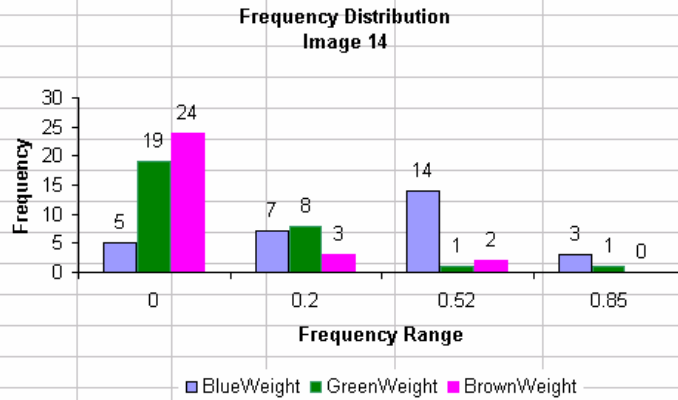


Table D.11: History of Direct Rating Weights for Image 15

ID	BlueWeight	GreenWeight	BrownWeight
15	0.2	0.85	0
15	0.2	0.52	0
15	0	0.2	0
15	0	0.85	0
15	0.2	0.52	0
15	0.52	0.52	0
15	0	0.2	0
15	0	0.2	0
15	0	0.52	0
15	0	0.52	0.2
15	0	0.52	0
15	0	0.52	0
15	0	0.2	0
15	0.2	0	0
15	0	0	0
15	0	0.85	0
15	0	0.52	0
15	0	0	0
15	0	0.2	0
15	0	0.2	0
15	0	0.52	0
15	0	0.52	0
15	0.52	0.85	0
15	0	0.52	0
15	0	0	0.2
15	0	0.52	0
15	0	0.52	0
15	0	0	0
15	0	0.2	0

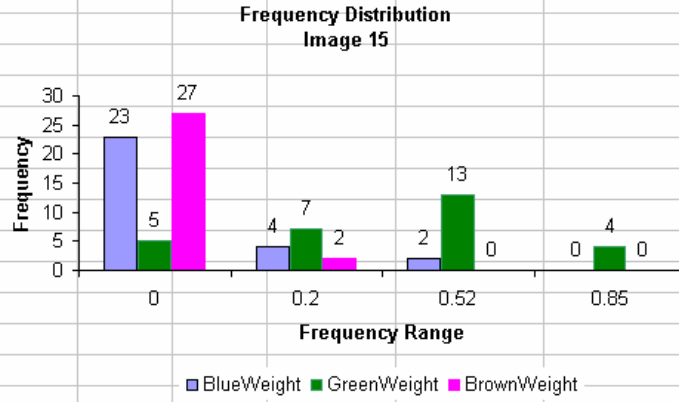


Table D.12: History of Direct Rating Weights for Image 16

ID	BlueWeight	GreenWeight	BrownWeight
16	0.52	0	0
16	0.2	0.52	0.2
16	0	0	0
16	0	0.52	0
16	0.2	0.52	0
16	0.2	0	0
16	0.2	0.2	0
16	0.2	0.2	0
16	0.2	0.52	0
16	0.2	0.2	0
16	0.52	0.2	0
16	0.52	0.52	0
16	0	0.52	0
16	0	0	0
16	0.52	0	0
16	0.52	0.52	0
16	0	0	0
16	0	0.52	0
16	0	0.52	0
16	0	0.52	0
16	0.52	0.52	0
16	0.52	0.52	0
16	0.52	0.52	0
16	0.2	0.2	0
16	0.2	0	0
16	0.2	0.2	0
16	0	0.2	0
16	0.2	0	0

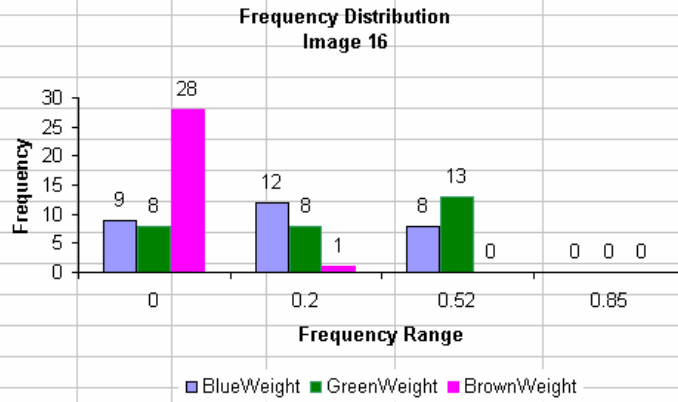


Table D.13: History of Direct Rating Weights for Image 17

ID	BlueWeight	GreenWeight	BrownWeight
17	0.2	0.2	0
17	0.2	0.85	0
17	0	0	0
17	0	0.52	0
17	0.2	0.2	0
17	0	0.2	0.2
17	0.2	0	0.52
17	0.2	0	0
17	0.2	0.2	0
17	0.2	0.2	0.2
17	0.2	0.2	0
17	0.2	0.52	0
17	0	0.85	0
17	0	0	0
17	0	0	0
17	0	0	0
17	0	0	0
17	0	0.85	0
17	0	0.2	0.2
17	0	0.2	0
17	0.52	0.52	0
17	0.52	0.52	0
17	0	0	0.2
17	0	0.52	0
17	0	0	0.2
17	0	0.52	0
17	0	0.2	0
17	0	0.2	0
17	0	0.2	0
17	0	0.52	0

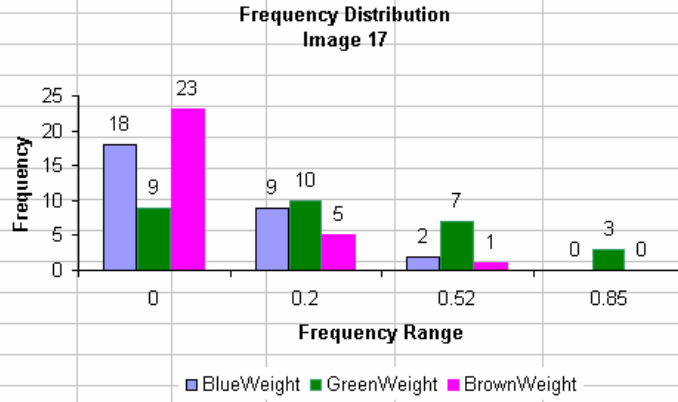


Table D.15: History of Direct Rating Weights for Image 19

ID	BlueWeight	GreenWeight	BrownWeight
19	0.2	0.2	0
19	0.85	0.2	0
19	0.2	0.2	0
19	0.52	0.52	0
19	0.52	0.52	0
19	0.52	0.2	0
19	0.52	0.2	0
19	0.52	0.2	0
19	0.52	0.2	0
19	0.52	0	0
19	0.52	0	0
19	0.52	0	0
19	0.2	0.2	0
19	0.52	0	0
19	0.2	0	0
19	0.2	0.85	0
19	0.2	0.2	0
19	0	0	0
19	0	0.2	0
19	0	0.2	0
19	0	0.52	0
19	0	0.52	0
19	0.2	0.2	0
19	0.2	0.2	0
19	0.2	0.2	0
19	0.2	0	0
19	0.2	0	0
19	0.2	0	0
19	0	0	0

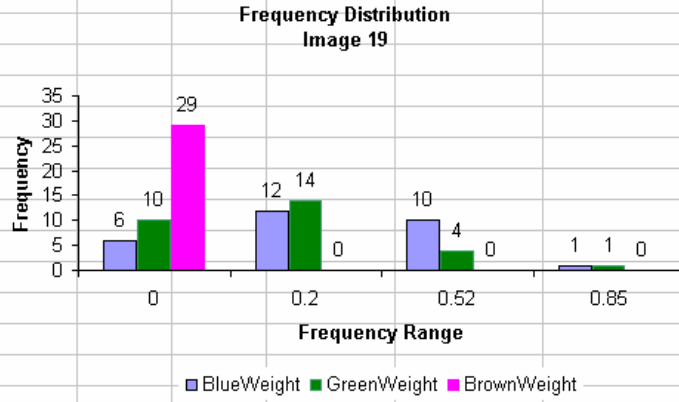


Table D.17: History of Direct Rating Weights for Image 23

ID	BlueWeight	GreenWeight	BrownWeight
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0
23	0	0	0.85
23	0	0	0.52
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.52
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0	0	0.85
23	0.2	0	0.85
23	0.2	0	0.85
23	0	0	0
23	0	0	0.85
23	0	0	0.85

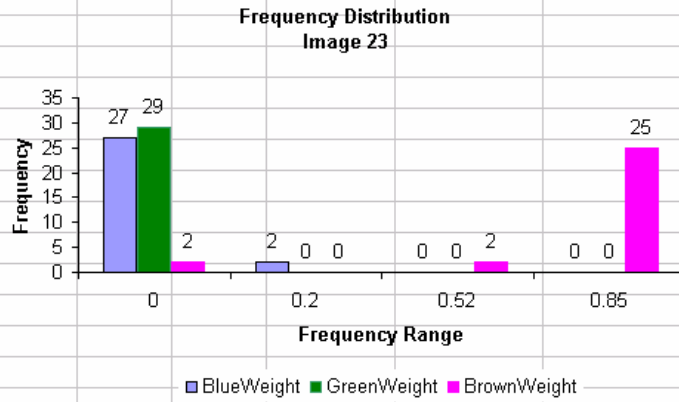


Table D.18: History of Direct Rating Weights for Image 24

ID	BlueWeight	GreenWeight	BrownWeight
24	0.85	0	0
24	0.52	0	0
24	0.52	0	0
24	0.85	0.2	0
24	0.85	0.85	0
24	0.85	0	0
24	0.85	0	0
24	0.85	0	0
24	0.85	0.2	0
24	0.85	0	0
24	0.2	0	0
24	0.2	0.52	0
24	0.85	0	0
24	0.52	0	0
24	0.2	0	0
24	0.2	0.2	0
24	0.85	0	0
24	0.2	0	0
24	0	0	0
24	0	0	0
24	0.2	0	0
24	0.2	0	0
24	0.85	0.2	0
24	0.52	0	0
24	0.52	0.52	0
24	0.52	0	0
24	0.52	0	0
24	0.2	0	0
24	0.2	0	0

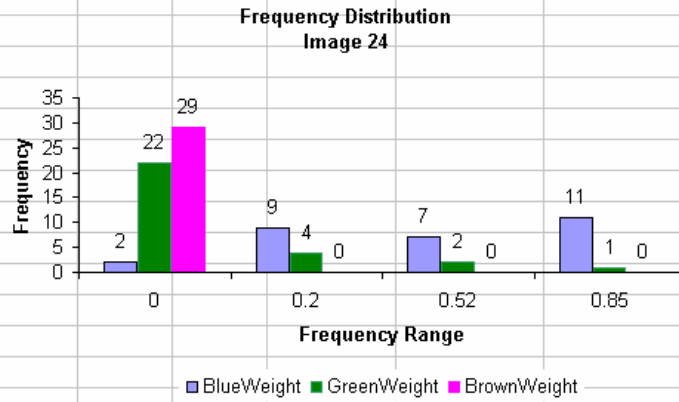


Table D.19: History of Direct Rating Weights for Image 31

ID	BlueWeight	GreenWeight	BrownWeight
31	0.2	0	0
31	0.2	0	0
31	0.52	0	0
31	0.85	0.2	0
31	0.52	0.52	0
31	0.2	0	0
31	0.85	0	0
31	0.85	0	0
31	0.85	0	0
31	0.85	0	0
31	0.52	0	0
31	0.52	0	0
31	0.85	0	0
31	0	0	0
31	0	0	0
31	0	0	0
31	0.2	0	0
31	0.52	0	0
31	0	0	0
31	0	0	0
31	0.52	0.2	0
31	0.52	0.2	0
31	0.2	0	0
31	0	0	0
31	0.52	0.2	0
31	0.52	0	0
31	0	0	0
31	0.2	0	0
31	0.2	0	0

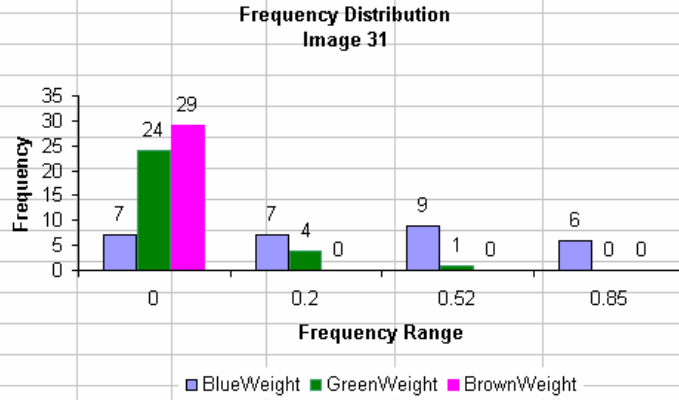


Table D.20: History of Direct Rating Weights for Image 32

ID	BlueWeight	GreenWeight	BrownWeight
32	0	0	0.85
32	0	0	0.85
32	0	0	0.85
32	0	0	0.85
32	0.2	0.2	0
32	0	0	0.85
32	0	0	0.85
32	0	0	0
32	0	0	0
32	0	0	0.85
32	0	0	0.52
32	0	0	0
32	0	0	0.85
32	0	0	0.85
32	0	0	0.85
32	0	0	0.85
32	0	0	0.85
32	0	0	0.85
32	0	0	0.52
32	0	0	0.52
32	0	0	0.85
32	0	0	0.85
32	0	0	0.2
32	0	0	0.85
32	0	0	0.52
32	0	0	0.2
32	0	0	0.85
32	0	0	0.52
32	0	0	0.2
32	0	0	0.85
32	0	0	0.85
32	0	0	0.85

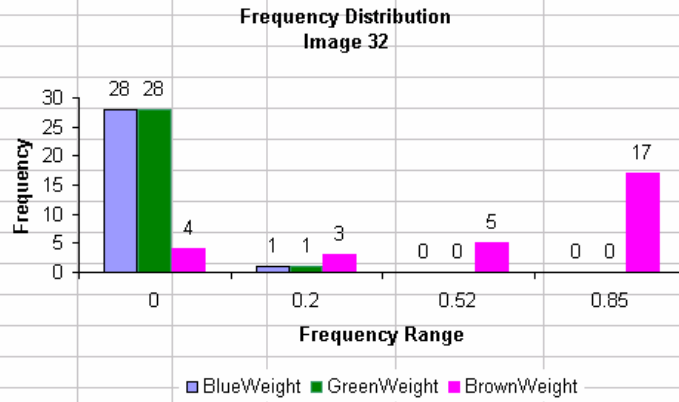


Table D.21: History of Direct Rating Weights for Image 33

ID	BlueWeight	GreenWeight	BrownWeight
33	0	0	0.2
33	0	0.52	0.2
33	0	0	0
33	0	0.2	0.2
33	0	0	0.52
33	0	0	0.85
33	0.2	0	0.52
33	0.2	0	0
33	0.2	0.2	0
33	0.2	0	0
33	0.52	0	0
33	0.52	0	0
33	0.52	0	0
33	0	0	0.85
33	0	0	0.52
33	0	0	0
33	0	0	0
33	0	0	0
33	0	0	0.85
33	0	0	0.2
33	0	0.85	0
33	0	0.85	0.2
33	0	0	0.52
33	0	0	0
33	0	0.2	0
33	0	0.2	0
33	0	0	0
33	0	0	0.52
33	0	0.2	0

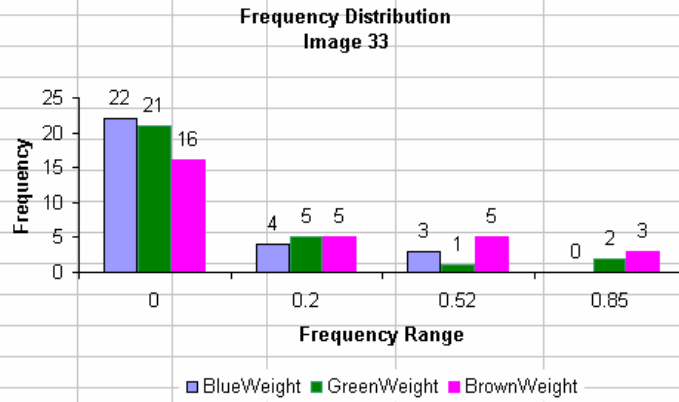


Table D.22: History of Direct Rating Weights for Image 34

ID	BlueWeight	GreenWeight	BrownWeight
34	0.2	0	0
34	0.2	0	0
34	0.52	0	0
34	0.85	0	0
34	0	0	0.2
34	0.2	0.2	0
34	0.85	0	0
34	0.85	0	0
34	0.85	0	0
34	0.85	0	0.52
34	0.52	0	0
34	0.52	0	0
34	0.52	0	0
34	0.2	0	0
34	0	0	0
34	0	0	0
34	0.52	0	0
34	0	0	0
34	0	0	0.2
34	0	0	0
34	0.52	0.2	0
34	0.52	0.2	0
34	0	0.2	0
34	0.2	0.2	0
34	0.2	0.2	0
34	0.2	0	0
34	0.2	0.2	0
34	0.52	0	0
34	0	0	0

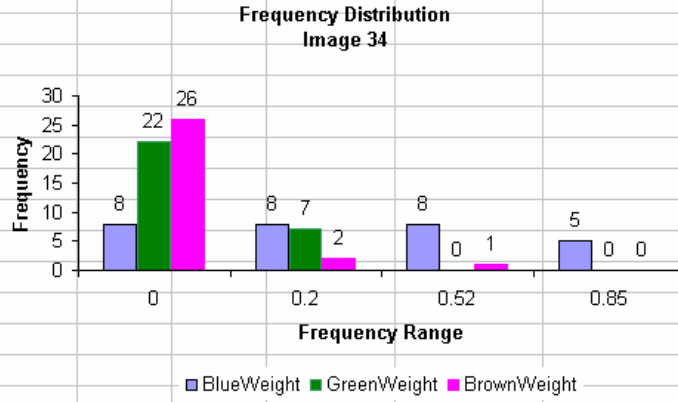


Table D.23: History of Direct Rating Weights for Image 35

ID	BlueWeight	GreenWeight	BrownWeight
35	0.2	0	0
35	0.52	0	0
35	0.85	0	0
35	0.85	0	0
35	0	0.52	0
35	0.52	0	0
35	0.85	0	0
35	0.85	0	0
35	0.85	0	0
35	0.85	0	0
35	0.85	0	0
35	0.52	0	0
35	0.52	0	0
35	0.52	0	0
35	0.52	0	0
35	0.85	0	0
35	0.52	0	0
35	0.52	0.2	0
35	0.52	0	0
35	0.52	0	0
35	0.2	0	0.2
35	0.2	0	0
35	0.52	0	0
35	0.52	0	0
35	0.2	0.52	0
35	0.52	0	0
35	0.52	0.52	0
35	0.52	0	0
35	0.85	0	0
35	0.52	0	0
35	0.52	0	0

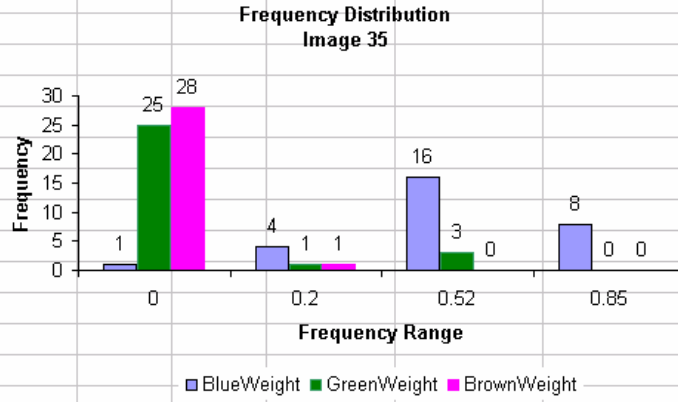


Table D.24: History of Direct Rating Weights for Image 36

ID	BlueWeight	GreenWeight	BrownWeight
36	0.2	0	0
36	0.52	0.2	0
36	0.52	0	0
36	0.52	0	0
36	0.52	0.52	0
36	0.2	0	0
36	0.85	0	0
36	0.85	0	0
36	0.85	0	0
36	0.85	0	0
36	0.52	0	0
36	0.52	0	0
36	0.52	0	0
36	0.52	0	0
36	0.52	0	0
36	0.52	0	0
36	0.52	0	0
36	0.85	0	0
36	0.2	0	0
36	0.2	0	0
36	0.2	0	0
36	0.52	0.2	0
36	0.52	0.2	0
36	0.2	0.85	0
36	0.2	0	0
36	0.2	0.2	0
36	0.2	0.2	0
36	0.2	0	0
36	0.52	0	0
36	0.52	0	0

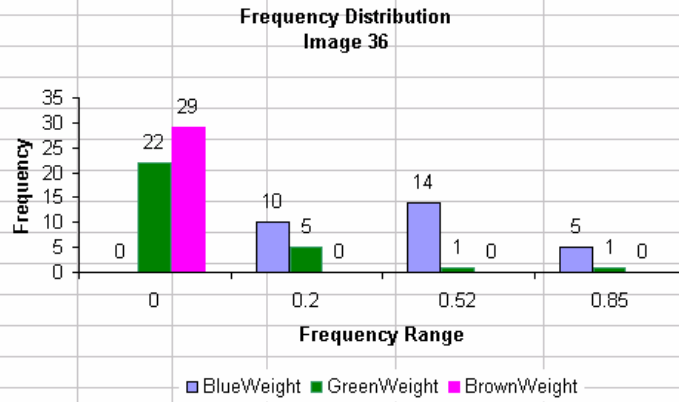


Table D.25: History of Direct Rating Weights for Image 37

ID	BlueWeight	GreenWeight	BrownWeight
37	0.2	0	0
37	0.85	0	0
37	0.52	0	0
37	0.85	0	0
37	0	0.52	0
37	0.85	0	0
37	0.85	0	0
37	0.85	0	0
37	0.85	0	0
37	0.85	0	0
37	0.85	0	0
37	0.52	0	0
37	0.52	0	0
37	0.85	0	0
37	0.52	0	0
37	0.85	0	0
37	0.85	0	0
37	0.52	0	0
37	0.2	0	0
37	0.52	0	0
37	0.52	0	0
37	0.52	0	0
37	0.52	0	0
37	0.52	0	0
37	0.52	0	0
37	0.85	0.2	0
37	0.85	0	0
37	0.52	0.2	0
37	0.52	0	0
37	0.52	0	0
37	0.52	0	0
37	0.2	0	0

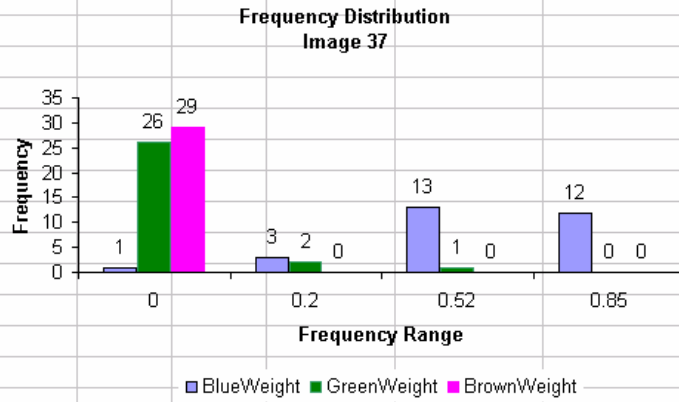


Table D.26: History of Direct Rating Weights for Image 38

ID	BlueWeight	GreenWeight	BrownWeight
38	0.2	0	0
38	0.85	0	0
38	0.52	0	0
38	0.85	0	0
38	0.52	0.52	0
38	0.85	0	0
38	0.52	0	0
38	0.52	0	0
38	0.52	0	0
38	0.52	0	0
38	0.2	0	0
38	0.2	0	0
38	0.85	0	0
38	0.52	0	0
38	0.85	0	0
38	0.85	0	0
38	0.85	0	0
38	0.85	0	0
38	0.52	0	0
38	0.2	0	0
38	0.2	0	0
38	0.52	0	0
38	0.52	0	0
38	0.52	0	0
38	0.85	0.2	0
38	0.85	0	0
38	0.85	0.2	0
38	0.85	0	0
38	0.52	0	0
38	0.2	0	0
38	0.52	0	0

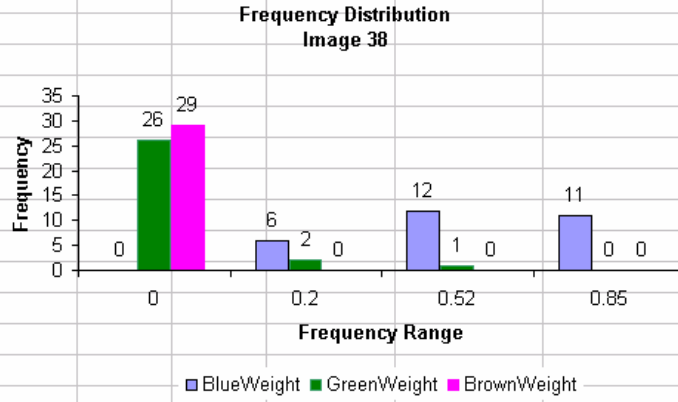
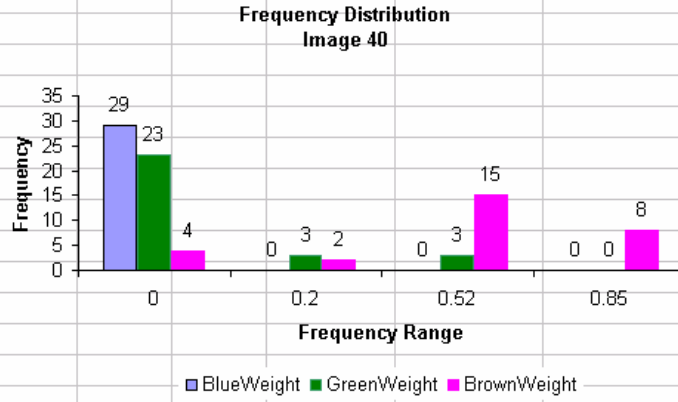


Table D.27: History of Direct Rating Weights for Image 40

ID	BlueWeight	GreenWeight	BrownWeight
40	0	0	0.52
40	0	0	0.52
40	0	0	0.52
40	0	0.2	0.85
40	0	0	0.52
40	0	0	0.52
40	0	0.2	0.85
40	0	0.2	0
40	0	0	0
40	0	0.52	0.85
40	0	0.52	0.52
40	0	0	0.85
40	0	0	0.52
40	0	0	0
40	0	0	0.52
40	0	0	0.85
40	0	0	0.85
40	0	0	0.85
40	0	0	0.52
40	0	0	0.85
40	0	0	0.52
40	0	0	0.52
40	0	0.52	0
40	0	0	0.85
40	0	0	0.52
40	0	0	0.2
40	0	0	0.52
40	0	0	0.52
40	0	0	0.52



APPENDIX E

History of User Verifications in the Testing Phase

Appendix E contains the history of the user verifications of both New Random Proportional and Direct Rating method during the testing phase. This appendix contains one table per image. Value 0 means that user is satisfied with the result and 1 means that user is not satisfied with the result.

Table E.1: History of User Verifications in the Testing Phase for Image 1

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
1	1	1	1	1	1	0	0
1	1	0	1	1	1	0	0
1	0	1	1	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	1	1	1	0	0
1	1	1	1	1	1	0	0
1	1	0	1	1	1	0	0
1	1	0	1	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	1	1	0	0	0

Table E.2: History of User Verifications in the Testing Phase for Image 2

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
2	1	0	1	2	0	0	0
2	1	0	1	2	0	0	0
2	1	0	1	2	0	0	0
2	1	0	1	2	0	0	0
2	1	0	1	2	0	0	0
2	1	0	1	2	0	0	0
2	1	0	1	2	0	0	0
2	1	0	1	2	0	0	0
2	1	0	1	2	0	1	0
2	1	0	1	2	0	0	0
2	1	0	1	2	0	0	0

Table E.3: History of User Verifications in the Testing Phase for Image 3

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
3	1	0	0	3	0	1	0
3	0	0	1	3	0	1	0
3	1	1	1	3	0	0	0
3	1	1	1	3	0	0	0
3	1	1	0	3	0	0	0
3	1	0	0	3	0	0	0
3	1	0	0	3	0	0	0
3	1	1	0	3	0	0	0
3	1	0	1	3	0	0	0
3	1	1	1	3	0	1	0

Table E.4: History of User Verifications in the Testing Phase for Image 8

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
8	1	1	1	8	0	0	0
8	1	1	0	8	0	0	0
8	1	1	1	8	0	0	0
8	1	1	0	8	0	0	0
8	1	1	0	8	0	0	0
8	1	1	1	8	0	0	1
8	1	1	1	8	0	0	0
8	0	1	0	8	0	0	0
8	1	1	0	8	0	0	0
8	1	1	0	8	0	0	0

Table E.5: History of User Verifications in the Testing Phase for Image 9

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
9	0	0	1	9	0	0	0
9	0	1	1	9	1	0	0
9	0	0	1	9	0	0	0
9	0	0	1	9	0	0	0
9	0	0	0	9	0	0	0
9	1	1	1	9	0	0	0
9	0	0	1	9	0	0	0
9	0	1	1	9	1	0	0
9	0	0	1	9	0	0	0
9	0	0	1	9	0	0	0

Table E.6: History of User Verifications in the Testing Phase for Image 10

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
10	0	0	1	10	0	0	0
10	0	0	1	10	1	0	0
10	1	1	1	10	0	1	0
10	1	0	1	10	0	1	0
10	0	0	1	10	1	0	0
10	1	0	0	10	0	0	0
10	0	0	1	10	0	0	0
10	1	0	1	10	0	0	0
10	0	0	1	10	0	0	0
10	0	1	1	10	0	1	0

Table E.7: History of User Verifications in the Testing Phase for Image 11

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
11	0	0	1	11	1	0	0
11	0	1	1	11	0	0	0
11	1	1	1	11	0	0	0
11	1	0	1	11	0	0	0
11	0	0	1	11	1	0	0
11	1	1	0	11	0	0	0
11	1	1	1	11	1	0	0
11	0	0	0	11	1	0	0
11	0	1	1	11	1	0	0
11	1	1	1	11	0	0	0

Table E.8: History of User Verifications in the Testing Phase for Image 12

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
12	1	0	1	12	0	0	0
12	1	0	0	12	0	0	0
12	0	1	1	12	0	0	0
12	0	1	1	12	0	0	0
12	1	0	1	12	0	0	0
12	1	1	0	12	0	0	0
12	0	0	0	12	0	0	0
12	1	0	1	12	0	0	0
12	1	0	1	12	0	0	0
12	1	1	1	12	0	0	0

Table E.9: History of User Verifications in the Testing Phase for Image 13

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
13	1	1	1	13	0	0	0
13	1	1	0	13	0	0	0
13	1	1	0	13	0	0	0
13	1	1	0	13	0	0	0
13	1	1	0	13	0	0	0
13	1	1	0	13	0	0	1
13	1	1	0	13	0	0	0
13	0	1	0	13	0	0	0
13	1	1	0	13	0	0	0
13	1	1	0	13	0	0	0

Table E.10: History of User Verifications in the Testing Phase for Image 14

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
14	1	1	0	14	1	0	0
14	1	0	0	14	1	0	0
14	1	0	1	14	1	0	0
14	1	0	1	14	1	0	0
14	1	0	0	14	1	0	0
14	1	1	1	14	1	0	0
14	1	1	1	14	1	0	0
14	1	1	1	14	0	0	0
14	0	1	1	14	1	0	0
14	0	1	0	14	1	0	0

Table E.11: History of User Verifications in the Testing Phase for Image 15

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
15	0	1	1	15	0	1	0
15	0	1	1	15	0	0	0
15	1	1	1	15	0	0	0
15	1	1	1	15	0	0	0
15	1	0	1	15	0	0	0
15	1	1	0	15	0	0	0
15	1	1	1	15	0	1	0
15	1	0	1	15	0	1	0
15	1	0	1	15	0	0	0
15	1	0	1	15	0	0	0

Table E.12: History of User Verifications in the Testing Phase for Image 16

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
16	0	1	1	16	0	0	0
16	0	0	1	16	1	0	0
16	0	0	1	16	1	0	0
16	0	0	1	16	1	0	0
16	0	0	1	16	0	0	0
16	1	0	1	16	1	0	0
16	1	0	1	16	0	0	0
16	1	0	1	16	1	1	0
16	0	0	1	16	0	1	0
16	0	0	1	16	1	0	0

Table E.13: History of User Verifications in the Testing Phase for Image 17

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
17	1	1	1	17	0	1	0
17	0	0	1	17	0	1	0
17	1	1	1	17	0	1	0
17	1	1	1	17	0	1	0
17	1	0	0	17	0	0	0
17	1	1	0	17	0	0	0
17	1	0	1	17	0	1	0
17	1	0	1	17	0	0	0
17	0	0	1	17	0	0	0
17	0	0	1	17	0	1	0

Table E.14: History of User Verifications in the Testing Phase for Image 18

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
18	1	1	1	18	0	0	0
18	1	1	1	18	0	0	0
18	1	1	1	18	0	0	0
18	1	0	1	18	0	0	0
18	1	1	0	18	0	0	0
18	1	1	0	18	0	0	1
18	1	1	0	18	0	0	0
18	0	1	0	18	0	0	0
18	1	1	0	18	0	0	0
18	1	0	0	18	0	0	0

Table E.15: History of User Verifications in the Testing Phase for Image 19

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
19	0	1	1	19	0	0	0
19	0	1	1	19	0	0	0
19	0	1	1	19	1	1	0
19	0	1	1	19	1	1	0
19	0	0	1	19	0	0	0
19	0	0	0	19	0	0	0
19	1	0	0	19	1	0	0
19	1	0	1	19	0	0	0
19	0	1	1	19	0	0	0
19	0	1	1	19	1	1	0

Table E.16: History of User Verifications in the Testing Phase for Image 21

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
21	1	1	1	21	0	0	0
21	1	1	1	21	0	0	0
21	1	1	1	21	0	0	0
21	1	1	1	21	0	0	0
21	1	1	0	21	0	0	0
21	1	1	1	21	0	0	1
21	1	1	0	21	0	0	0
21	1	1	0	21	0	0	0
21	1	1	0	21	0	0	0
21	1	1	0	21	0	0	0

Table E.17: History of User Verifications in the Testing Phase for Image 23

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
23	1	1	0	23	0	0	0
23	1	1	1	23	0	0	0
23	1	1	1	23	0	0	0
23	1	1	1	23	0	0	0
23	1	1	0	23	0	0	0
23	1	1	1	23	0	0	0
23	1	1	0	23	0	0	0
23	0	1	0	23	0	0	0
23	1	1	0	23	0	0	0
23	1	1	0	23	0	0	0

Table E.18: History of User Verifications in the Testing Phase for Image 24

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
24	0	1	1	24	1	0	0
24	0	1	1	24	0	0	0
24	0	1	1	24	0	0	0
24	0	1	1	24	0	0	0
24	0	1	1	24	0	0	0
24	0	1	0	24	0	0	0
24	0	1	1	24	1	0	0
24	0	1	1	24	0	0	0
24	0	1	1	24	0	0	0
24	0	1	1	24	0	0	0

Table E.19: History of User Verifications in the Testing Phase for Image 31

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
31	1	1	1	31	1	0	0
31	0	0	1	31	0	0	0
31	0	1	1	31	0	0	0
31	0	1	1	31	0	0	0
31	0	0	1	31	0	0	0
31	1	1	1	31	1	0	0
31	1	0	1	31	0	0	0
31	0	1	1	31	0	0	0
31	0	0	1	31	0	0	0
31	0	1	1	31	0	0	0

Table E.20: History of User Verifications in the Testing Phase for Image 32

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
32	1	1	0	32	0	0	0
32	1	1	0	32	0	0	0
32	1	1	1	32	0	0	1
32	1	1	1	32	0	0	0
32	1	0	0	32	0	0	1
32	1	1	0	32	0	0	1
32	1	1	0	32	0	0	0
32	0	1	1	32	0	0	0
32	1	1	0	32	0	0	0
32	0	1	1	32	0	0	1

Table E.21: History of User Verifications in the Testing Phase for Image 33

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
33	1	1	0	33	0	0	0
33	1	1	0	33	0	0	0
33	0	1	1	33	0	0	0
33	0	0	1	33	0	0	0
33	1	0	0	33	0	0	0
33	1	1	1	33	0	0	0
33	1	1	1	33	0	0	0
33	0	1	0	33	0	0	0
33	0	0	1	33	0	0	0
33	0	1	1	33	0	0	0

Table E.22: History of User Verifications in the Testing Phase for Image 34

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
34	0	1	1	34	1	0	0
34	0	0	1	34	1	0	0
34	0	1	1	34	0	0	0
34	0	0	1	34	0	0	0
34	0	0	1	34	0	0	0
34	0	0	1	34	0	0	0
34	0	0	1	34	0	0	0
34	0	0	0	34	0	0	0
34	0	0	1	34	0	0	0
34	1	1	1	34	1	0	0

Table E.23: History of User Verifications in the Testing Phase for Image 35

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
35	0	1	1	35	0	0	0
35	0	0	1	35	0	0	0
35	1	1	1	35	0	0	0
35	1	1	1	35	0	0	0
35	0	0	1	35	0	0	0
35	0	1	1	35	0	0	0
35	1	0	1	35	0	0	0
35	0	1	1	35	0	0	0
35	0	1	1	35	0	0	0
35	0	1	1	35	0	0	0

Table E.24: History of User Verifications in the Testing Phase for Image 36

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
36	0	1	1	36	0	0	0
36	0	0	1	36	0	0	0
36	0	1	1	36	0	0	0
36	0	0	1	36	0	0	0
36	0	0	1	36	0	0	0
36	0	1	1	36	0	0	0
36	0	1	1	36	0	0	0
36	0	0	0	36	1	0	0
36	0	1	1	36	0	0	0
36	0	1	1	36	0	0	0

Table E.25: History of User Verifications in the Testing Phase for Image 37

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
37	1	1	1	37	1	0	0
37	0	1	1	37	0	0	0
37	0	1	1	37	0	0	0
37	0	0	1	37	0	0	0
37	0	1	1	37	0	0	0
37	1	1	0	37	1	0	0
37	0	1	1	37	0	0	0
37	0	1	0	37	0	0	0
37	0	1	1	37	0	0	0
37	0	1	1	37	0	0	0

Table E.26: History of User Verifications in the Testing Phase for Image 38

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
38	0	1	1	38	1	0	0
38	0	1	1	38	0	0	0
38	0	1	1	38	0	0	0
38	0	1	1	38	0	0	0
38	0	1	1	38	0	0	0
38	0	1	1	38	1	0	0
38	0	1	1	38	0	0	0
38	0	1	1	38	1	0	0
38	0	1	1	38	0	0	0
38	0	1	1	38	0	0	0

Table E.27: History of User Verifications in the Testing Phase for Image 40

New Random Proportional Verification Data				Direct Rating User Verification Data			
ID	Blue	Green	Brown	ID	Blue	Green	Brown
40	1	1	0	40	0	0	1
40	1	1	0	40	0	0	0
40	1	1	1	40	0	0	1
40	1	1	0	40	0	0	0
40	1	1	0	40	0	0	0
40	1	1	1	40	0	0	1
40	1	1	1	40	0	0	0
40	1	1	0	40	0	0	0
40	1	0	0	40	0	0	0
40	0	1	0	40	0	0	0

APPENDIX F

Instructions for User Evaluation and User Verification

Appendix F contains the instructions which were given to the user community for the training phase, where user determined the membership weights and for the testing phase, where user evaluated the membership weights. In the training phase, instruction sheet per method was given to the user. In the testing phase, instruction sheet contained instructions for both methods.

Figure F.1: Instructions to Train the Direct Rating Method

Thanks a lot for helping in the research study. Your time and effort is appreciated.

Please follow the following instructions

Note: Whenever you want to abort the experiment, click “Cancel” button.

Step 1: Select the desktop icon labeled “**DirectT**”



DirectT.exe

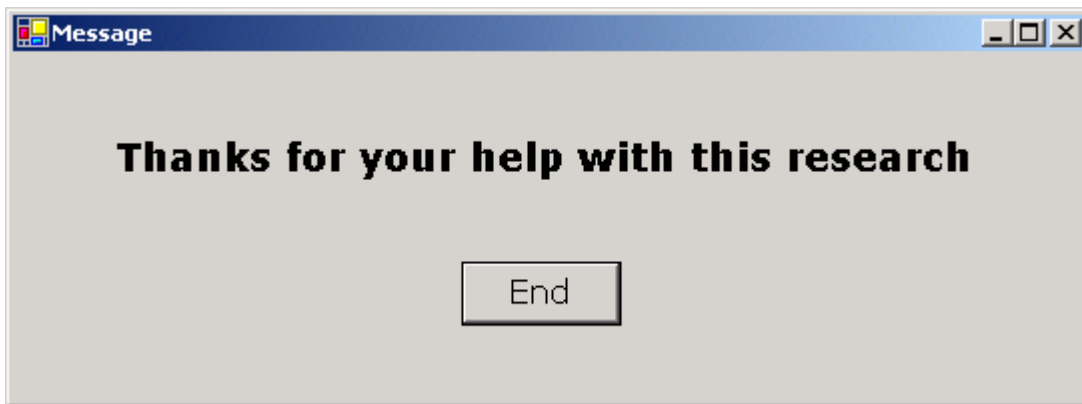
The following window will open

For each image displayed, select one of the radio buttons indicating whether the image has no, slightly, medium or very particular eye color by answering the question related to the eye color which is displayed at the top of the page.

If the “More” button is enabled, there are more images to be evaluated and clicking on it will open those images.

If “End Color” button is enabled, there are no more images for this particular eye color. Click it.

Step 3: Repeat step 2 until the following window opens which indicates that experiment is over.



Click “End” button.

Thanks for your participation.

Figure F.2: Instructions to Train the New Random Proportional Method

Thanks a lot for helping in the research study. Your time and effort is appreciated.

Please follow the following instructions

Note: Whenever you want to abort the experiment, click “Cancel” button.

Step 1: Select the desktop icon labeled “**RandomT**”



The following window will open


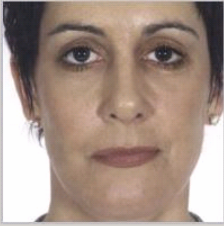
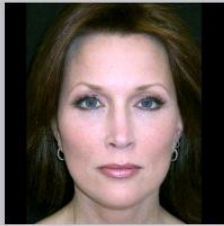
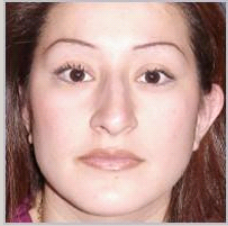

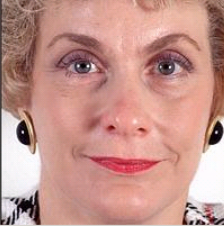


Click “Ok” button

Step 2: A new window will open displaying a series of images

Query Results

To what extent do you agree these people have very Blue Eyes?

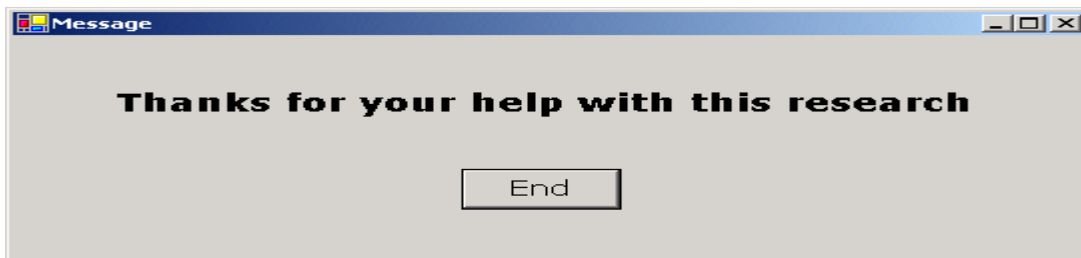
 <p> <input checked="" type="radio"/> Meets The Criteria <input type="radio"/> More Blue <input type="radio"/> Less Blue </p>	 <p> <input checked="" type="radio"/> Meets The Criteria <input type="radio"/> More Blue <input type="radio"/> Less Blue </p>	 <p> <input checked="" type="radio"/> Meets The Criteria <input type="radio"/> More Blue <input type="radio"/> Less Blue </p>
 <p> <input checked="" type="radio"/> Meets The Criteria <input type="radio"/> More Blue <input type="radio"/> Less Blue </p>	 <p> <input checked="" type="radio"/> Meets The Criteria <input type="radio"/> More Blue <input type="radio"/> Less Blue </p>	 <p> <input checked="" type="radio"/> Meets The Criteria <input type="radio"/> More Blue <input type="radio"/> Less Blue </p>

For each image displayed, select one of the radio buttons indicating whether the image meets the criteria displayed at the top of the page, or if the image would be better described by a more or less intense description.

If the “More” button is enabled, there are more images to be evaluated and clicking on it will open those images.

If “End Criteria (example: very blue) Color” button is enabled, there are no more images for this query. Click it.

Step 3: Repeat Step 2 until the following window opens which indicates that the experiment is over.



Click “End” button.

Thanks for your participation.

Figure F.3: Instructions to Verify the Direct Rating Method

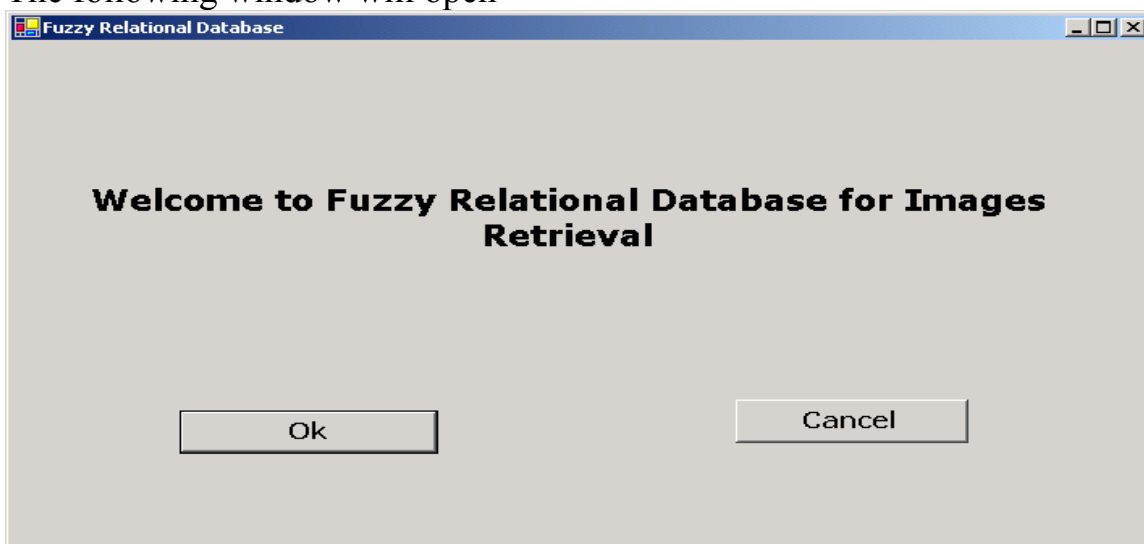
Thanks a lot for helping in the research study. Your time and effort is appreciated.

Please follow the following instructions

Note: Whenever you want to abort the experiment, click “Cancel” button.

Step 1: Select the desktop icon labeled “Experiment 1”

The following window will open



Click “Ok” button

Step 2: A new window will open displaying a series of images

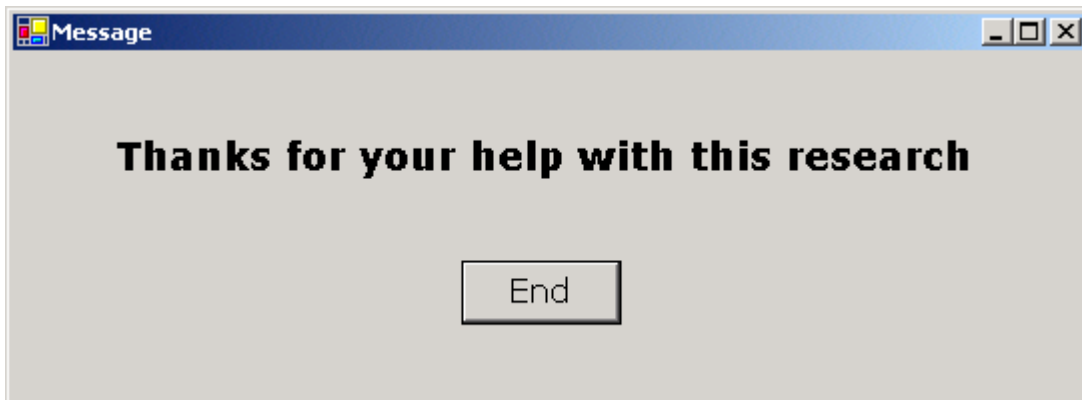
The screenshot shows a window titled "Testing Direct Method" with a subtitle "People with Very Brown Eyes". It contains six portrait images of people, arranged in two rows of three. Each image has two radio buttons below it: "Satisfied with result" (selected) and "Not satisfied with result". At the bottom of the window are three buttons: "More", "End Very Brown", and "Cancel".

For each image displayed, please click one of the two radio button based on where you are satisfied with image’s eye color the criteria displayed in the title (i.e. very brown eyes in case shown above) or not

If the “More” button is enabled, there are more images to be evaluated and clicking on it will open those images.

If “End Color” button is enabled, there are no more images for this particular eye color. Click it.

Step 3: Repeat step 2 until the following window opens which indicates that experiment is over.



Click “End” button.

Thanks for your participation.

Figure F.4: Instructions to Verify the New Random Proportional Method

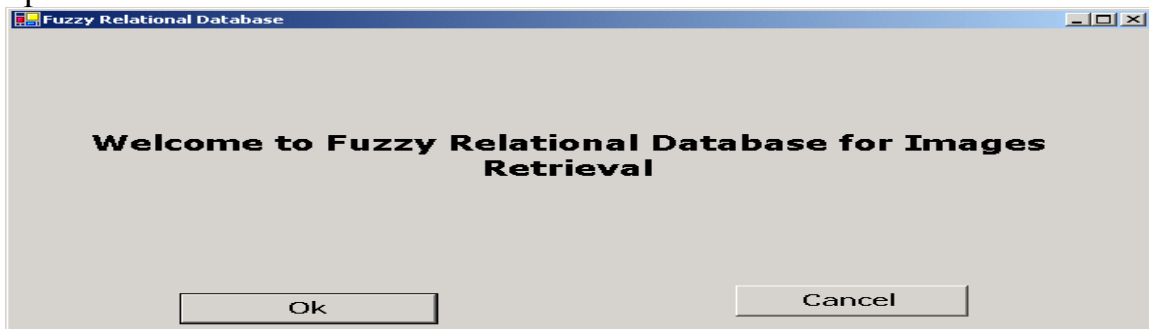
Thanks a lot for helping in the research study. Your time and effort is appreciated.

Please follow the following instructions

Note: Whenever you want to abort the experiment, click “Cancel” button.

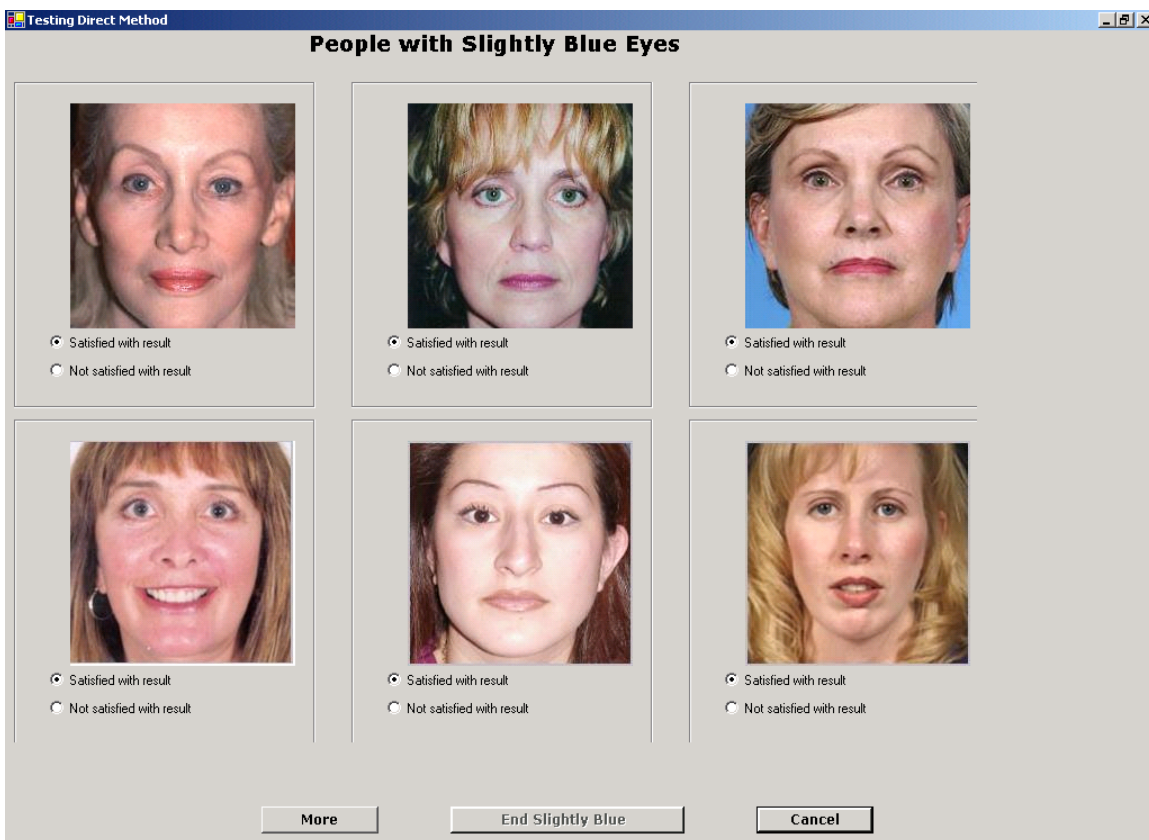
Step 1: Select the desktop icon labeled “Experiment 2”

The following window will open



Click "Ok" button

Step 2: A new window will open displaying a series of images

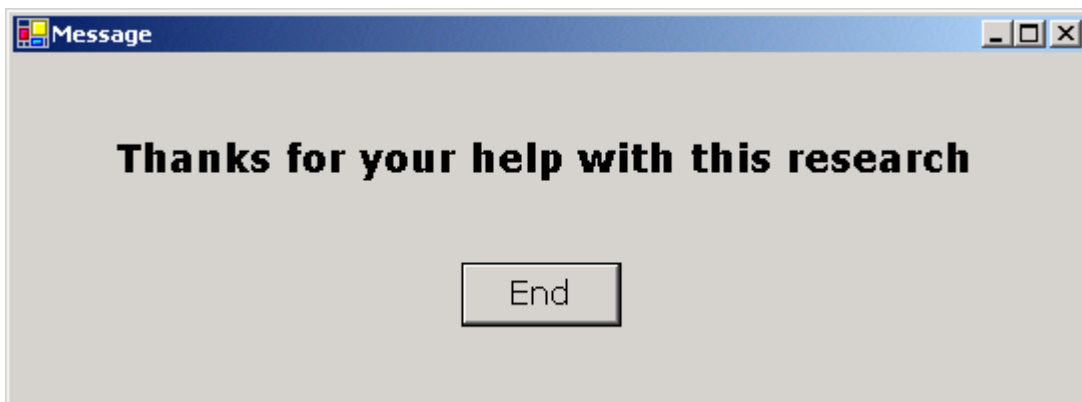


For each image displayed, please click one of the two radio button based on where you are satisfied with image's eye color the criteria displayed in the title (i.e. slightly blue eyes in case shown above) or not

If the “More” button is enabled, there are more images to be evaluated and clicking on it will open those images.

If “End Color” button is enabled, there are no more images for this particular eye color. Click it.

Step 3: Repeat step 2 until the following window opens which indicates that experiment is over.



Click “End” button.

Thanks for your participation.

Figure F.5: Instructions to Train the New Random Proportional Method with “Not”

Please follow the following instructions

Note: Whenever you want to abort the experiment, click “Cancel” button.

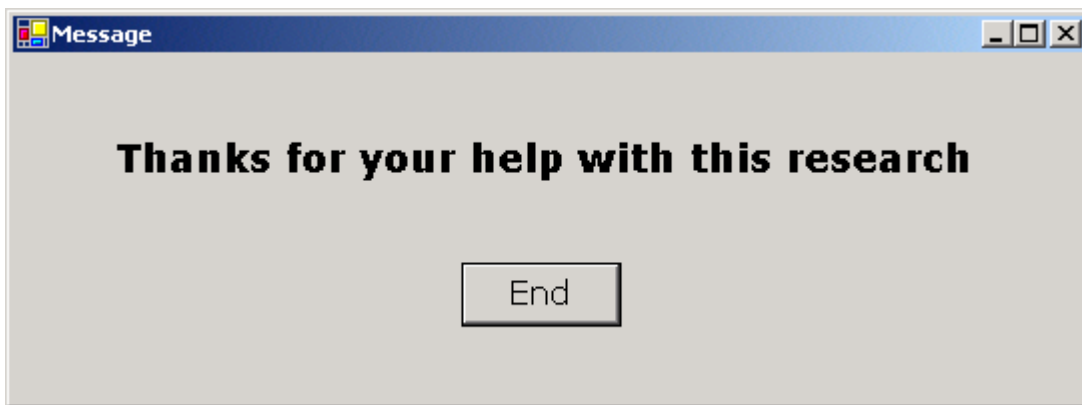
Step 1: Select the desktop icon labeled “Experiment 3”

For each image displayed, select one of the radio buttons indicating whether the image meets the criteria displayed at the top of the page, or if the image would be better described by a more or less intense description or image does not have eye color displayed on the title.

If the “More” button is enabled, there are more images to be evaluated and clicking on it will open those images.

If “End Color” button is enabled, there are no more images for this particular eye color. Click it.

Step 3: Repeat step 2 until the following window opens which indicates that experiment is over.



Click “End” button.

Thanks for your participation.

APPENDIX G

Stored Procedure to Initialize New Random Proportional Weights

Appendix G contains the SQL Server stored procedure used to initialize the Eye Color table values for the new random proportional method. The procedure is executed in the SQL Server Enterprise Manager window prior to executing queries using the system.

```
CREATE PROCEDURE [Initialize_NProp_Random]
as

declare @id as int,@color as varchar(20)
declare @vala as float, @valb as float, @counter as int, @looper as int

DECLARE color_weight CURSOR
for
select ID,color from color
OPEN color_weight

FETCH NEXT FROM color_weight
into @ID,@color

        set @looper = 0
        set @counter = 2
        set @vala= rand()

        if @vala < 0.02
        Begin
                set @vala = 0.02
        End

        if @vala > 0.96
        Begin
                set @vala = 0.96
        End
        set @valb= ( 1.0 - @vala) / 2.0
```

```
WHILE @@FETCH_STATUS = 0
BEGIN

    IF @looper = 0
    BEGIN
        IF @counter = 2
        BEGIN
            print 'ID'
            print @id
            print 'color'
            print @color
            update color
            set weight=@vala
            where ID=@id and color=@color
            FETCH NEXT FROM color_weight
            into @ID,@color
        END

        IF @counter = 1
        BEGIN
            print 'ID'
            print @id
            print 'color'
            print @color
            update color
            set weight= @valb
            where ID=@id and color=@color
            FETCH NEXT FROM color_weight
            into @ID,@color
        END

        IF @counter = 0
        BEGIN
            print 'ID'
            print @id
            print 'color'
            print @color
            update color
            set weight=@valb
            where ID=@id and color=@color
            FETCH NEXT FROM color_weight
            into @ID,@color
        END
    END
END
```

```
END

set @counter = @counter - 1

IF @counter < 0
BEGIN
    set @counter = 2
    set @vala= (rand() )

    if @vala < 0.02
    Begin
        set @vala = 0.02
    End

    if @vala > 0.96
    Begin
        set @vala = 0.96
    End

    set @valb= (1.0 - @vala) / 2.0
    set @looper = @looper + 1
END
END

IF @looper = 1
BEGIN
    IF @counter = 2
    BEGIN
        print 'ID'
        print @id
        print 'color'
        print @color
        update color
        set weight=@valb
        where ID=@id and color=@color
        FETCH NEXT FROM color_weight
        into @ID,@color
    END

    IF @counter = 1
    BEGIN
        print 'ID'
        print @id
```



```

    print 'color'
    print @color
    update color
    set weight= @vala
    where ID=@id and color=@color
    FETCH NEXT FROM color_weight
    into @ID,@color
END

```

```

IF @counter = 0
BEGIN
    print 'ID'
    print @id
    print 'color'
    print @color
    update color
    set weight=@valb
    where ID=@id and color=@color
    FETCH NEXT FROM color_weight
    into @ID,@color
END

```

```

set @counter = @counter - 1
IF @counter < 0
BEGIN
    set @counter = 2
    set @valb= (rand())

```

```

    if @vala < 0.02
    Begin
        set @vala = 0.02
    End

```

```

    if @vala > 0.96
    Begin
        set @vala = 0.96
    End

```

```

set @vala= (1.0 - @valb) / 2.0
set @looper = @looper + 1

```

```

END
END

```

```
IF @looper = 2
BEGIN
  IF @counter = 2
  BEGIN
    print 'ID'
    print @id
    print 'color'
    print @color
    update color
    set weight=@valb
    where ID=@id and color=@color
    FETCH NEXT FROM color_weight
    into @ID,@color
  END

  IF @counter = 1
  BEGIN
    print 'ID'
    print @id
    print 'color'
    print @color
    update color
    set weight= @valb
    where ID=@id and color=@color
    FETCH NEXT FROM color_weight
    into @ID,@color
  END

  IF @counter = 0
  BEGIN
    print 'ID'
    print @id
    print 'color'
    print @color
    update color
    set weight=@vala
    where ID=@id and color=@color
    FETCH NEXT FROM color_weight
    into @ID,@color
  END
END
```

```

    set @counter = @counter - 1
    IF @counter < 0
    BEGIN
        set @counter = 2
        set @vala= ( rand() )
        if @vala < 0.02
        Begin
            set @vala = 0.02
        End

        if @vala > 0.96
        Begin
            set @vala = 0.96
        End

        set @valb= (1.0 - @vala) / 2.0
        set @looper = @looper + 1
    END
END
IF @looper = 3
BEGIN
    Set @looper = 0
    set @vala= (rand() )
    if @vala < 0.02
    Begin
        set @vala = 0.02
    End

    if @vala > 0.96
    Begin
        set @vala = 0.96
    End
    set @valb= (1.0 - @vala) / 2.0
END
END

CLOSE color_weight
DEALLOCATE color_weight
GO

```

APPENDIX H

Stored Procedure to Retrieve all Images for given Eye Color

Appendix H contains the SQL Server stored procedure used to retrieve all images for the given color. The input to the stored procedure is the eye color.

```
CREATE PROCEDURE [FetchDirectID]
@color as varchar(10)
AS
SET ANSI_NULLS ON
exec('select * from Person P,Color C where
      P.ID=C.ID and C.Color='+'''+@color+''')
GO
```

APPENDIX I

Stored Procedure to Retrieve Images for New Random Proportional Method based on the Query

Appendix I contains the SQL Server stored procedure used to retrieve images for the given color and in between the modifier range for New Random Proportional Method. The input to the stored procedure is SQL query, modifier and eye color. It interprets the fuzzy modifiers and translates them into SQL queries.

```
CREATE    PROCEDURE [Fetch_Data]
@query as varchar(10),
@s1 as  varchar(50),
@s2 as  varchar(50)

AS

SET ANSI_NULLS ON
Declare @high as float,
@low as float

if @s2 = 'Blue' or
@s2='Green' or
@s2='Brown'
begin
    select @high =high from Range where modifier=@s1
    select @low= low from Range where modifier=@s1

    exec('select * from Person P,Color C where
        P.ID=C.ID and C.weight <= '+@high+' and C.weight >= '+@low+' and
        C.Color='+'''+@s2+''')
End
GO
```

APPENDIX J

Stored Procedure to Retrieve Images for Direct Rating Method based on the Query

Appendix J contains the SQL Server stored procedure used to retrieve images for the given color and in between the modifier range for Direct Rating Method. The input to the stored procedure is SQL query, modifier and eye color. It interprets the fuzzy modifiers and translates them into SQL queries.

```
CREATE PROCEDURE [Fetch_Direct_Data]
@query as varchar(10),
@s1 as varchar(50),
@s2 as varchar(50)

AS

SET ANSI_NULLS ON
Declare @high as float,
@low as float

if @s2 = 'Blue' or
@s2='Green' or
@s2='Brown'
begin
select @high =high from Range where modifier=@s1
select @low= low from Range where modifier=@s1

exec('select * from Person P,ColorDirect C where
P.ID=C.ID and C.weight <= '+@high+' and C.weight >= '+@low+' and
C.Color='+'''+@s2+''')
end

GO
```

APPENDIX K

Stored Procedure to Retrieve Images for New Random Proportional Method with

“Not” Modifier based on the Query

Appendix K contains the SQL Server stored procedure used to retrieve images for the given color and in between the modifier range for New Random Proportional Method with “Not”. The input to the stored procedure is SQL query, modifier and eye color. It interprets the fuzzy modifiers and translates them into SQL queries.

```
CREATE PROCEDURE [Fetch_Data_Second]
@query as varchar(10),
@s1 as varchar(50),
@s2 as varchar(50)

AS

SET ANSI_NULLS ON
Declare @high as float,
@low as float

if @s2 = 'Blue' or
@s2='Green' or
@s2='Brown'
begin
select @high =high from Range where modifier=@s1
select @low= low from Range where modifier=@s1

exec('select * from Person P,ColorSecond C where
P.ID=C.ID and C.weight <= '+@high+' and C.weight >= '+@low+' and
C.Color='+'''+@s2+''')

End
Go
```

APPENDIX L

Stored Procedure to Update Weight for New Random Proportional Method

Appendix L contains the SQL Server stored procedure used to update the weights in the database to reflect the feedback given by the user after the query evaluation for the new random proportional method. The input to the stored procedure is image ID, eye color, more_less criteria and modifier.

```
CREATE          PROCEDURE [Update_Data]
@ID as int,
@color as varchar(50),
@more_less as varchar(50),
@modifier as varchar(50)

AS

SET ANSI_NULLS ON
declare @threshold as float,@existing_weight as float,@new_weight as float

if @color = 'Blue' or
   @color='Green' or
   @color='Brown'
begin
    select @threshold =threshold from Range where modifier=@modifier
    select @existing_weight=Weight from Color where ID=@ID and Color=@color
    set @new_weight=@existing_weight

    if @more_less = 'meets'
    begin
        if @existing_weight < @threshold
        begin
            set @new_weight = @existing_weight + .015
        end
    end
end
```



```

if @existing_weight > @threshold
begin
    set @new_weight = @existing_weight - .015
end
end
else
begin
if @more_less = 'more'
begin
    if @modifier = 'very'
    begin
        if @existing_weight > @threshold
        begin
            set @new_weight = @existing_weight - .015
        end
        if @existing_weight < @threshold
        begin
            set @new_weight = @existing_weight + .015
        end
    end
    end
    else
    begin
        set @new_weight = @existing_weight + .015
    end
end
end
else
begin
if @modifier='slightly'
begin
    if @existing_weight > @threshold
    begin
        set @new_weight = @existing_weight - .015
    end
    if @existing_weight < @threshold
    begin
        set @new_weight = @existing_weight + .015
    end
end
end
    else
    begin
        set @new_weight = @existing_weight - .015
    end
end
end
end

```

```

end

update Color set weight=@new_weight where ID=@ID and color=@color

end

if @color = 'Broad' or
   @color = 'Average' or
   @color='Narrow'
begin
  select @threshold =threshold from Range where modifier=@modifier
  select @existing_weight=Weight from Face where ID=@ID and Face=@color
  set @new_weight=@existing_weight

  if @more_less = 'meets'
  begin
    if @existing_weight < @threshold
    begin
      set @new_weight = @existing_weight + .015
    end
    if @existing_weight > @threshold
    begin
      set @new_weight = @existing_weight - .015
    end
  end
end
else
begin
  if @more_less = 'more'
  begin
    if @modifier = 'very'
    begin
      if @existing_weight > @threshold
      begin
        set @new_weight = @existing_weight - .015
      end
      if @existing_weight < @threshold
      begin
        set @new_weight = @existing_weight + .015
      end
    end
  end
  else
  begin
    set @new_weight = @existing_weight + .015
  end
end
end

```

```
end
end
else
begin
  if @modifier='slightly'
  begin
    if @existing_weight > @threshold
    begin
      set @new_weight = @existing_weight - .015
    end
    if @existing_weight < @threshold
    begin
      set @new_weight = @existing_weight + .015
    end
  end
  end
  else
  begin
    set @new_weight = @existing_weight - .015
  end
end
end
update Face set weight=@new_weight where ID=@ID and Face=@color
end
GO
```

APPENDIX M

Stored Procedure to Update Weights for Direct Rating Method

Appendix M contains the SQL Server stored procedure used to update the weights in the database to reflect the feedback given by the user after the query evaluation for the direct rating method. For example, if user said that eyes are slightly blue then 0.2 weight is stored in the database for that image in ColorDirect table. The input to the stored procedure is image ID, eye color and modifier.

```
CREATE          PROCEDURE [Update_Data_Direct]
@ID as int,
@color as varchar(50),
@modifier as varchar(50)

AS

SET ANSI_NULLS ON
declare @new_weight as float
IF @color = 'Blue' or
   @color='Green' or
   @color='Brown'
begin
    if @modifier = 'not'
    begin
        set @new_weight = 0
    end

    if @modifier = 'slightly '
    begin
        set @new_weight = 0.2
    end

    if @modifier = 'medium '
```

```
begin
    set @new_weight = 0.52
end

if @modifier = 'very '
begin
    set @new_weight = 0.85
end

update ColorDirect set weight=@new_weight where ID=@ID and color=@color
end
GO
```

APPENDIX N

Stored Procedure to Update Weights for New Random Proportional Method with

“Not” Modifier

Appendix N contains the SQL Server stored procedure used to update the weights in the database to reflect the feedback given by the user after the query evaluation for the new random proportional method with “Not”. The input to the stored procedure is image ID, eye color, more_less criteria and modifier.

```
CREATE          PROCEDURE [Update_Data_Second]
@ID as int,
@color as varchar(50),
@more_less as varchar(50),
@modifier as varchar(50)

AS

SET ANSI_NULLS ON
declare @threshold as float,@existing_weight as float,@new_weight as float
if @color = 'Blue' or
   @color='Green' or
   @color='Brown'
begin
    select @threshold =threshold from Range where modifier=@modifier
    select @existing_weight=Weight from ColorSecond
        where ID=@ID and Color=@color
    set @new_weight=@existing_weight

    if @more_less ='nocolor'
    begin
        set @new_weight = @existing_weight - .025
    end
end
```

```
else
begin
  if @more_less = 'meets'
  begin
    if @existing_weight < @threshold
    begin
      set @new_weight = @existing_weight + .015
    end

    if @existing_weight > @threshold
    begin
      set @new_weight = @existing_weight - .015
    end
  end
else
begin
  if @more_less = 'more'
  begin
    if @modifier = 'very'
    begin
      if @existing_weight > @threshold
      begin
        set @new_weight = @existing_weight - .015
      end

      if @existing_weight < @threshold
      begin
        set @new_weight = @existing_weight + .015
      end
    end
  end
else
begin
    set @new_weight = @existing_weight + .015
  end
end
else
begin
  if @modifier='slightly'
  begin
    if @existing_weight > @threshold
    begin
      set @new_weight = @existing_weight - .015
```


APPENDIX O

Stored Procedure to Insert Weight for Images to maintain History

Appendix O contains the SQL Server stored procedure used to insert the Blue eye Color Weight, Green Eye Color Weight and Brown Eye Color Weight of the each image in the table in order to record the history of the user's response for the direct rating method.

```
CREATE PROCEDURE [Insert_Hist_Direct]

@ID as int,
@ValueBlue as Decimal (3,2),
@ValueGreen as Decimal(3,2),
@ValueBrown as Decimal(3,2)

AS

SET ANSI_NULLS ON

insert into HistColorDirect (ID,BlueW, GreenW, BrownW)
        values(@ID,@ValueBlue,@ValueGreen,@ValueBrown)

GO
```

APPENDIX P

Stored Procedure to Update Weights for Test Data

Appendix P contains the SQL Server stored procedure used to update the weights in the database to reflect the feedback given by the user after the query evaluation for both the methods in the testing phase. The input to the stored procedure is image ID, eye color, UserResponse(value '0' means satisfied, Value '1' means not satisfied) and testMethod (value 'Random' means for New Random Proportional Method, Value 'Direct' means for Direct Rating Method).

```
CREATE PROCEDURE [Update_Test]
@ID as int, @color as varchar(50),
@UserResponse as varchar(50),
@testMethod as varchar(50)

AS
SET ANSI_NULLS ON
if @testMethod = 'Random'
begin
    if @color = 'Blue' or @color='Green' or @color='Brown'
    begin
        update TestRandom set weight=@UserResponse where ID=@ID and color=@color
    end
end

if @testMethod = 'Direct'
begin
    if @color = 'Blue' or @color='Green' or @color='Brown'
    begin
        update TestDirect set weight=@UserResponse where ID=@ID and color=@color
    end
end
GO
```

APPENDIX Q

Comparison charts

Appendix Q contains charts to show comparison for final membership weights between Direct Rating and New Random Proportional for Brown and Green eye color.

Figure Q.1: Direct Rating (No Green) vs. New Random Proportional

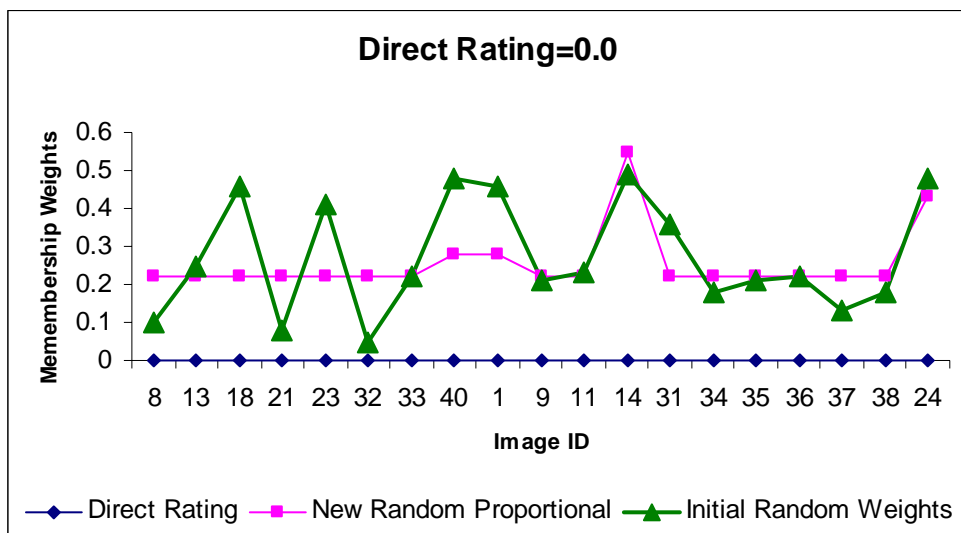


Figure Q.2: Direct Rating (Slightly Green) vs. New Random Proportional

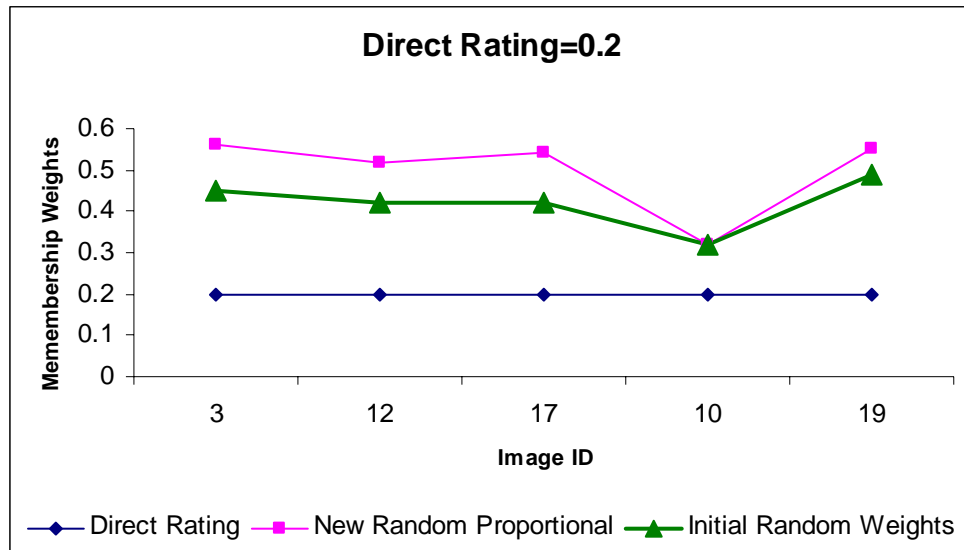


Figure Q.3: Direct Rating (Medium Green) vs. New Random Proportional

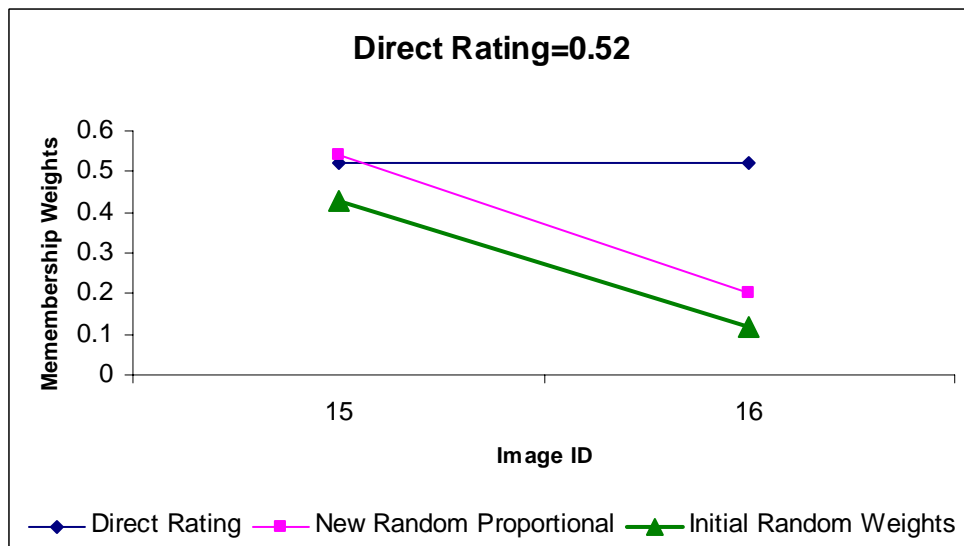


Figure Q.4: Direct Rating (No Brown) vs. New Random Proportional

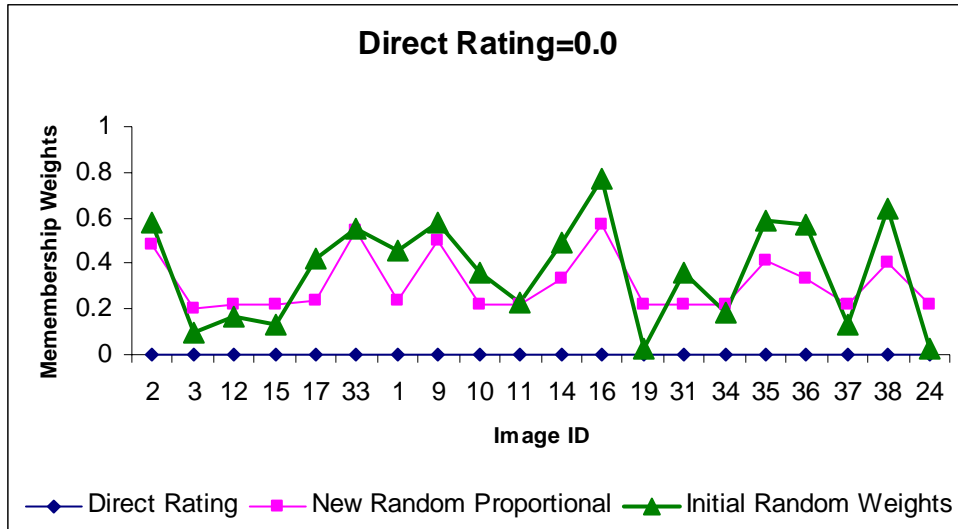


Figure Q.5: Direct Rating (Medium Brown) vs. New Random Proportional

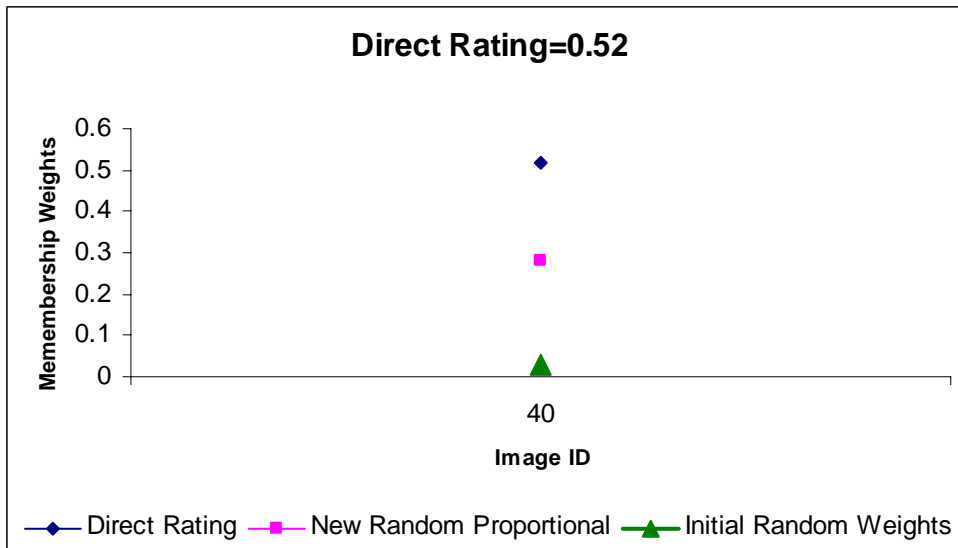
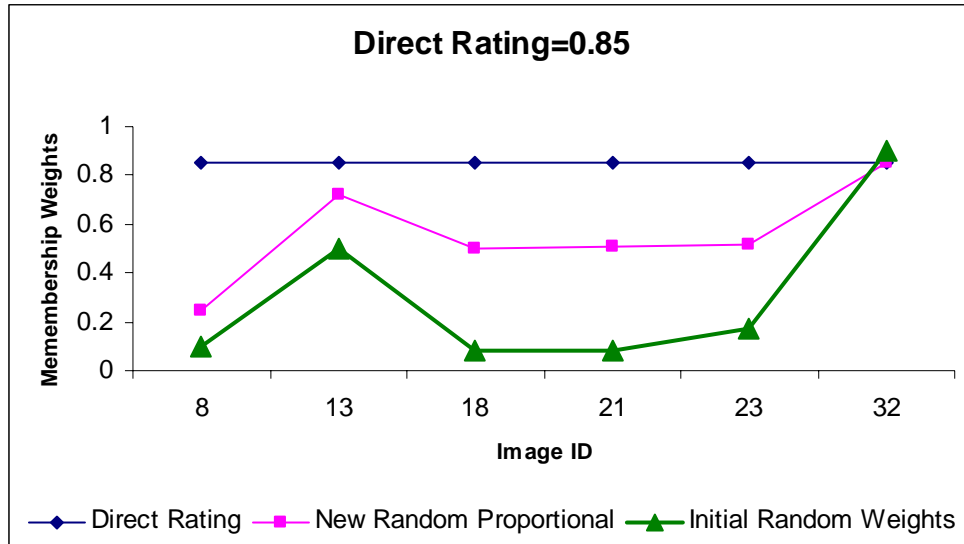


Figure Q.6: Direct Rating (Very Brown) vs. New Random Proportional

APPENDIX R

Fuzzy Relational Database Prototype Code

Appendix R contains the source code used to create and run the User Graphical Interface for the prototype. The source code attached below ran to train the fuzzy relational database for the new random proportional method. The source code for the User Graphical Interface to train the database for Direct Rating and New Random Proportional as well as for testing methods can be referenced from the application itself as there is little difference between them. The original code that is referenced in [1] is updated for the evaluation.

```
'Fall 2005
'Shweta Sanghi
'This form is the first form of the prototype. It randomly evokes forms.

Public Class main
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
    End If
```

```

        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form
    Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents Button1 As System.Windows.Forms.Button
    Friend WithEvents Button2 As System.Windows.Forms.Button
    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
    Me.Label1 = New System.Windows.Forms.Label
    Me.Button1 = New System.Windows.Forms.Button
    Me.Button2 = New System.Windows.Forms.Button
    Me.SuspendLayout()
    '
    'Label1
    '
    Me.Label1.Font = New System.Drawing.Font("Verdana", 18.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
    Me.Label1.ImageAlign = System.Drawing.ContentAlignment.TopCenter
    Me.Label1.Location = New System.Drawing.Point(120, 88)
    Me.Label1.Name = "Label1"
    Me.Label1.Size = New System.Drawing.Size(440, 64)
    Me.Label1.TabIndex = 0
    Me.Label1.Text = "Welcome to Fuzzy Relational Database for Images
Retrieval"
    Me.Label1.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter
    '
    'Button1
    '
    Me.Button1.Font = New System.Drawing.Font("Verdana", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
    Me.Button1.Location = New System.Drawing.Point(144, 240)
    Me.Button1.Name = "Button1"
    Me.Button1.Size = New System.Drawing.Size(104, 32)
    Me.Button1.TabIndex = 4
    Me.Button1.Text = "Ok"
    '
    'Button2
    '
    Me.Button2.Font = New System.Drawing.Font("Verdana", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
    Me.Button2.Location = New System.Drawing.Point(384, 240)

```



```

Me.Button2.Name = "Button2"
Me.Button2.Size = New System.Drawing.Size(104, 32)
Me.Button2.TabIndex = 5
Me.Button2.Text = "Cancel"
'
'main
'
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.ClientSize = New System.Drawing.Size(656, 341)
Me.Controls.Add(Me.Button2)
Me.Controls.Add(Me.Button1)
Me.Controls.Add(Me.Label1)
Me.Name = "main"
Me.Text = "Welcome to Fuzzy Relational Database for Images
Retrieval"
Me.ResumeLayout(False)

End Sub

#End Region
Dim modifier1 As Integer
Dim modifier2 As Integer
Dim modifier3 As Integer
Dim CanUpdate As Integer

Private Sub main_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

Randomize()
'Int((upperbound - lowerbound + 1) * Rnd() + lowerbound)

modifier1 = CInt(Int((3 * Rnd() + 1)))

Select Case modifier1
Case 1
modifier2 = CInt(Int((2 * Rnd() + 2)))
If modifier2 = 2 Then
modifier3 = 3
Else
modifier3 = 2
End If

Case 3
modifier2 = CInt(Int((2 * Rnd() + 1)))
If modifier2 = 2 Then
modifier3 = 1
Else
modifier3 = 2
End If

Case 2
modifier2 = 3

```

```

        modifier3 = 1
    End Select

    CanUpdate = 1
    'MessageBox.Show("Modifier values 1: ", modifier1)
    'MessageBox.Show("Modifier 2:", modifier2)
    'MessageBox.Show("Modifier 3:", modifier3)
End Sub

' cancel button
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Me.Dispose()
    Me.Close()
End
End Sub

'ok button
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

    'evoke the form according to the value in modifier1 variable
    If (modifier1 = 1) Then
        Dim f As slightlyBlue
        f = New slightlyBlue
        f.initialize(modifier1, modifier2, modifier3, CanUpdate)
        Me.Hide()
        f.Show()
    ElseIf (modifier1 = 2) Then
        Dim f As MediumBlue
        f = New MediumBlue
        f.initialize(modifier1, modifier2, modifier3, CanUpdate)
        Me.Hide()
        f.Show()
    Else
        Dim f As VeryBlue
        f = New VeryBlue
        f.initialize(modifier1, modifier2, modifier3, CanUpdate)
        Me.Hide()
        f.Show()
    End If

End Sub
End Class

'Database Research Summer 2004
'Karen Joy and Smita Dattatri
'This form invokes the stored procedure "Fetch Data" which interprets the
Fuzzy query and
'returns the set of rows to be displayed on the User Interface.This form
also invokes a

```

```
'stored procedure called Update_data which updates the database depending
on the feedback
'given by the user.
```

```
'Spring, 2005
'Debra Duke
'Updated to add multiple page viewing functionality.
```

```
'Fall 2005
'Shweta Sanghi
'Updated the functionality to train for slightly blue
```

```
Imports System.IO
Imports System.Data.SqlClient
```

```
Public Class slightlyBlue
    Inherits System.Windows.Forms.Form
```

```
#Region " Windows Form Designer generated code "
```

```
Public Sub New()
    MyBase.New()
```

```
'This call is required by the Windows Form Designer.
InitializeComponent()
```

```
'Add any initialization after the InitializeComponent() call
```

```
End Sub
```

```
'Form overrides dispose to clean up the component list.
Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
    If disposing Then
        If Not (components Is Nothing) Then
            components.Dispose()
        End If
    End If
    MyBase.Dispose(disposing)
End Sub
```

```
'Required by the Windows Form Designer
Private components As System.ComponentModel.IContainer
```

```
'NOTE: The following procedure is required by the Windows Form
Designer
```

```
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
```

```
Friend WithEvents Panell As System.Windows.Forms.Panel
Friend WithEvents Button2 As System.Windows.Forms.Button
Friend WithEvents Button1 As System.Windows.Forms.Button
Friend WithEvents Labell As System.Windows.Forms.Label
```

```

Friend WithEvents Button3 As System.Windows.Forms.Button
Friend WithEvents Panel2 As System.Windows.Forms.Panel
<System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
    Me.Panel1 = New System.Windows.Forms.Panel
    Me.Button3 = New System.Windows.Forms.Button
    Me.Button1 = New System.Windows.Forms.Button
    Me.Button2 = New System.Windows.Forms.Button
    Me.Label1 = New System.Windows.Forms.Label
    Me.Panel2 = New System.Windows.Forms.Panel
    Me.Panel1.SuspendLayout()
    Me.SuspendLayout()
    '
    'Panel1
    '
    Me.Panel1.Controls.Add(Me.Button3)
    Me.Panel1.Controls.Add(Me.Button1)
    Me.Panel1.Controls.Add(Me.Button2)
    Me.Panel1.Dock = System.Windows.Forms.DockStyle.Bottom
    Me.Panel1.Location = New System.Drawing.Point(0, 637)
    Me.Panel1.Name = "Panel1"
    Me.Panel1.Size = New System.Drawing.Size(872, 32)
    Me.Panel1.TabIndex = 2
    '
    'Button3
    '
    Me.Button3.Anchor = System.Windows.Forms.AnchorStyles.Bottom
    Me.Button3.Font = New System.Drawing.Font("Verdana", 9.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
    Me.Button3.Location = New System.Drawing.Point(136, 0)
    Me.Button3.Name = "Button3"
    Me.Button3.Size = New System.Drawing.Size(104, 24)
    Me.Button3.TabIndex = 4
    Me.Button3.Text = "More"
    '
    'Button1
    '
    Me.Button1.Anchor = System.Windows.Forms.AnchorStyles.Bottom
    Me.Button1.BackColor = System.Drawing.Color.Transparent
    Me.Button1.Font = New System.Drawing.Font("Verdana", 9.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
    Me.Button1.Location = New System.Drawing.Point(288, 0)
    Me.Button1.Name = "Button1"
    Me.Button1.Size = New System.Drawing.Size(224, 24)
    Me.Button1.TabIndex = 1
    Me.Button1.Text = "End Slightly Blue Color"
    '
    'Button2
    '
    Me.Button2.Anchor = System.Windows.Forms.AnchorStyles.Bottom

```

```

        Me.Button2.BackColor = System.Drawing.Color.Transparent
        Me.Button2.Font = New System.Drawing.Font("Verdana", 9.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.Button2.Location = New System.Drawing.Point(552, 0)
        Me.Button2.Name = "Button2"
        Me.Button2.Size = New System.Drawing.Size(104, 24)
        Me.Button2.TabIndex = 3
        Me.Button2.Text = "Cancel"
    '
    'Label1
    '
        Me.Label1.Font = New System.Drawing.Font("Verdana", 14.25!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.Label1.Location = New System.Drawing.Point(16, 0)
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New System.Drawing.Size(832, 24)
        Me.Label1.TabIndex = 4
        Me.Label1.Text = "Label1"
        Me.Label1.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter
    '
    'Panel2
    '
        Me.Panel2.Location = New System.Drawing.Point(0, 32)
        Me.Panel2.Name = "Panel2"
        Me.Panel2.Size = New System.Drawing.Size(864, 600)
        Me.Panel2.TabIndex = 5
    '
    'slightlyBlue
    '
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.ClientSize = New System.Drawing.Size(872, 669)
        Me.Controls.Add(Me.Panel2)
        Me.Controls.Add(Me.Label1)
        Me.Controls.Add(Me.Panel1)
        Me.Name = "slightlyBlue"
        Me.Text = "Query Results"
        Me.WindowState = System.Windows.Forms.FormWindowState.Maximized
        Me.Panel1.ResumeLayout(False)
        Me.ResumeLayout(False)

End Sub

#End Region
Dim Query As String
Dim string2 As String
Dim modifier As String
Dim color_face As String
Dim ds As New DataSet
Dim al As New ArrayList

```

```

Dim nextPix As Integer
Dim remainingPix As Integer

Dim modifier1 As Integer
Dim modifier2 As Integer
Dim modifier3 As Integer
Dim CanUpdate As Integer

'Connection String used to connect to the database.
'CSC-LAB=Name Of the Computer
'DBResearch=Name of the database

'2/16/05: Name of the Computer changed to reflect working on my
local machine (DEBS), Debra
Dim con As New System.Data.SqlClient.SqlConnection("data
source=vikas;initial catalog=DBResearch;user
id=dbresearch;password=dbresearch")

'Event handling procedure for the 'Update' button

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim I As Integer
    Dim more_less As String
    'con.Open()
    'This block of code checks which check boxes were clicked by the
user and for which
    'person in order to update the corresponding rows in the
database.For each person the
    'stored procedure Update_Data is invoked.
    For I = 0 To al.Count - 1
        If (al(I).meets_criteria.Checked() = True) Then
            more_less = "meets"
        ElseIf (al(I).more.Checked() = True) Then
            more_less = "more"
        ElseIf (al(I).less.Checked() = True) Then
            more_less = "less"
        End If

        Dim daptr = New SqlDataAdapter
        daptr.SelectCommand = New SqlCommand
        daptr.SelectCommand.CommandType = CommandType.StoredProcedure
        daptr.SelectCommand.CommandText = "dbo.Update_Data"
        daptr.SelectCommand.Connection = con
        daptr.SelectCommand.Parameters.Add("@ID", al(I).ID)
        daptr.SelectCommand.Parameters.Add("@color", color_face)
        daptr.SelectCommand.Parameters.Add("@more_less", more_less)
        daptr.SelectCommand.Parameters.Add("@modifier", modifier)
        con.Open()
        Try
            daptr.SelectCommand.ExecuteNonQuery()

```

```

    Catch ex As Exception
        MessageBox.Show("Failed to execute query")
    End Try

    con.Close()

Next

    Dim slightlyGreen As slightlyGreen
    slightlyGreen = New slightlyGreen
    slightlyGreen.initialize(modifier1, modifier2, modifier3,
CanUpdate)
    Me.Dispose()
    slightlyGreen.Show()

End Sub

    'This function is invoked when the form is loaded.The stored
procedure "Fetch_Data" is
    'invoked and the fuzzy modifiers are passed as parameters to this
stored procedure.
    'This stored procedure interprets the fuzzy modifiers and returns the
result set that
    ' matches the criteria.

Private Sub slightlyBlue_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
    Dim I As Integer
    Dim Count As Integer
    Dim cmd As New System.Data.SqlClient.SqlCommand("Fetch_Data",
con)

    cmd.CommandType = CommandType.StoredProcedure

    cmd.Parameters.Add(New SqlParameter("@query", SqlDbType.VarChar,
100))
    cmd.Parameters("@query").Value = Query
    cmd.Parameters.Add(New SqlParameter("@s1", SqlDbType.VarChar,
100))
    cmd.Parameters("@s1").Value = modifier
    cmd.Parameters.Add(New SqlParameter("@s2", SqlDbType.VarChar,
100))
    cmd.Parameters("@s2").Value = color_face
Try
    Dim da As New System.Data.SqlClient.SqlDataAdapter(cmd)
    da.Fill(ds)
Catch ex As Exception
    MessageBox.Show(ex.ToString)
End Try

```

```

'displays the number of rows returned as a result of query (use
for debugging)
'MessageBox.Show(ds.Tables(0).Rows().Count)

If color_face = "Green" Or color_face = "Blue" Or color_face =
"Brown" Or color_face = "green" Or color_face = "blue" Or color_face =
"brown" Then
    Labell.Text() = "To what extent do you agree these people
have " + modifier + " " + color_face + " Eyes?"
Else
    Labell.Text() = "People With " + modifier + " " + color_face
+ " Faces"
End If

If (ds.Tables(0).Rows().Count) <> 0 Then
    Count = ds.Tables(0).Compute("COUNT(ID)", "")

    'The following block of code creates a new object of ImageSet
for each
    'record in the which result set which contains the image and
the Check Boxes
    'associated with it. It also reads the image in binary
format from the database
    'and displays it appropriately.

For I = 0 To Count - 1
    Dim S As New ImageSet(color_face)
    Dim bits As Byte() = CType(ds.Tables(0).Rows(I).Item(2),
Byte())

    Dim memorybits As New MemoryStream(bits)
    Dim bitmap As New Bitmap(memorybits)

    S.picture.Image = bitmap
    al.Add(S)
    al(I).ID = ds.Tables(0).Rows(I).Item(0)
Next I

    'variable indicating the first picture to be displayed on a
page
    nextPix = 0
    'variable indicating the remaining pictures to be displayed
remainingPix = al.Count

    placeComponents(nextPix)

    updateButtons()

    ' Labell.Text() = "People With Slightly Blue Eyes"

    'Changes the labels on the Panel depending on the query.

Else

```



```

        Me.Button3.Enabled = False
        MessageBox.Show("No images match search criteria. Please
Press End Button for next search")
    End If
End Sub

'The following block of code places the components on the Group Box
and
'then places the group box on the panel to be displayed.
Private Sub placeComponents(ByVal Position1 As Integer)

    Dim X As Integer, Y As Integer
    Y = 8
    Dim I As Integer

    'limit the display to 6 images
    Dim lastPix As Integer

    'determining the last picture for the current page
    If remainingPix <= 6 Then
        lastPix = Position1 + remainingPix - 1
    Else : lastPix = Position1 + 5
    End If

    For I = Position1 To lastPix
        Dim G As New GroupBox
        G.Location() = New Point(8 + X, Y)
        G.Size() = New Size(268, 295) 'was (288,350)

        al(I).picture.Location = New Point(50, 24)
        al(I).picture.size = New Size(200, 200) 'was (224,120)

        al(I).meets_criteria.location = New Point(32, 225) 'was (32,
250)
        al(I).meets_criteria.size = New Size(176, 24)

        al(I).more.size = New Size(176, 24)
        al(I).more.location = New Point(32, 250) 'was (32, 275)

        al(I).less.size = New Size(176, 24)
        al(I).less.location = New Point(32, 275) 'was (32,300)

        G.Controls.Add(al(I).picture)
        G.Controls.Add(al(I).meets_criteria)
        G.Controls.Add(al(I).more)
        G.Controls.Add(al(I).less)
        Panel2.Controls.Add(G)

        'Allows to display multiple rows of records.
        X = 300 + X
        If ((I Mod 6) = 2) Then

```

```

        Y = 295 + 5 + Y
        X = 0
    End If
Next

'update the number of picutes left to be displayed
If remainingPix > 6 Then
    remainingPix = remainingPix - 6
Else
    remainingPix = 0
End If

'update first picture on next page
nextPix = nextPix + 6
End Sub

'set up buttons depending on number of images in query result
Private Sub updateButtons()
    If al.Count <= 6 Or remainingPix = 0 Then
        Me.Button3.Enabled = False
        Me.Button1.Enabled = True
    Else
        Me.Button3.Enabled = True
        Me.Button1.Enabled = False
    End If
End Sub

Public Sub initialize(ByVal m1 As Integer, ByVal m2 As Integer, ByVal
m3 As Integer, ByVal cu As Integer)
    Query = "select * from person where color.color = 'blue';"
    modifier = "Slightly"
    color_face = "Blue"

    modifier1 = m1
    modifier2 = m2
    modifier3 = m3
    CanUpdate = cu
End Sub

'Event handling for 'Cancel' button
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    con.Close()
    Me.Dispose()
End
End Sub

'Event handling for 'More' button

```

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    'clear existing pictures
    Panel2.Controls.Clear()

    'build and display next page
    placeComponents(nextPix)
    Panel2.Show()

    updateButtons()
End Sub

```

```

'-----
'The class ImagesSet is a user defined class which has the image
'and the check boxes as its components.

```

```

Public Class ImageSet
    Public picture As PictureBox
    Public meets_criteria As RadioButton
    Public more As RadioButton
    Public Less As RadioButton
    Public ID As Integer

    Public Sub New(ByVal color_face As String)
        picture = New PictureBox
        meets_criteria = New RadioButton
        more = New RadioButton
        Less = New RadioButton
        meets_criteria.Text = "Meets The Criteria"
        meets_criteria.Checked = True
        'Changes the labels on the Check Boxes depending on the
query.
        If color_face = "Green" Or color_face = "green" Then
            more.Text = "More Green"
            Less.Text = "Less Green"
        End If
        If color_face = "Brown" Or color_face = "brown" Then
            more.Text = "More Brown"
            Less.Text = "Less Brown"
        End If
        If color_face = "Blue" Or color_face = "blue" Then
            more.Text = "More Blue"
            Less.Text = "Less Blue"
        End If
        If color_face = "Broad" Or color_face = "broad" Then
            more.Text = "More Broad"
            Less.Text = "Less Broad"
        End If
        If color_face = "Narrow" Or color_face = "narrow" Then
            more.Text = "More Narrow"
            Less.Text = "Less Narrow"
        End If
    End Sub
End Class

```

```

        End If
        ID = 0

    End Sub

End Class

'Database Research Summer 2004
'Karen Joy and Smita Dattatri
'This form invokes the stored procedure "Fetch Data" which interprets the
Fuzzy query and
'returns the set of rows to be displayed on the User Interface.This form
also invokes a
'stored procedure called Update_data which updates the database depending
on the feedback
'given by the user.

'Spring, 2005
'Debra Duke
'Updated to add multiple page viewing functionality.

'Fall 2005
'Shweta Sanghi
'Updated the functionality to train the prototype

Imports System.IO
Imports System.Data.SqlClient

Public Class slightlyGreen
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
    End Sub

```

```

        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form
    Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents Panel1 As System.Windows.Forms.Panel
    Friend WithEvents Button2 As System.Windows.Forms.Button
    Friend WithEvents Button1 As System.Windows.Forms.Button
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents Button3 As System.Windows.Forms.Button
    Friend WithEvents Panel2 As System.Windows.Forms.Panel
    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
    Me.Panel1 = New System.Windows.Forms.Panel
    Me.Button3 = New System.Windows.Forms.Button
    Me.Button1 = New System.Windows.Forms.Button
    Me.Button2 = New System.Windows.Forms.Button
    Me.Label1 = New System.Windows.Forms.Label
    Me.Panel2 = New System.Windows.Forms.Panel
    Me.Panel1.SuspendLayout()
    Me.SuspendLayout()
    '
    'Panel1
    '
    Me.Panel1.Controls.Add(Me.Button3)
    Me.Panel1.Controls.Add(Me.Button1)
    Me.Panel1.Controls.Add(Me.Button2)
    Me.Panel1.Dock = System.Windows.Forms.DockStyle.Bottom
    Me.Panel1.Location = New System.Drawing.Point(0, 637)
    Me.Panel1.Name = "Panel1"
    Me.Panel1.Size = New System.Drawing.Size(872, 32)
    Me.Panel1.TabIndex = 2
    '
    'Button3
    '
    Me.Button3.Anchor = System.Windows.Forms.AnchorStyles.Bottom
    Me.Button3.Font = New System.Drawing.Font("Verdana", 9.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
    Me.Button3.Location = New System.Drawing.Point(176, 0)
    Me.Button3.Name = "Button3"
    Me.Button3.Size = New System.Drawing.Size(104, 24)
    Me.Button3.TabIndex = 4
    Me.Button3.Text = "More"
    '
    'Button1

```

```

    Me.Button1.Anchor = System.Windows.Forms.AnchorStyles.Bottom
    Me.Button1.BackColor = System.Drawing.Color.Transparent
    Me.Button1.Font = New System.Drawing.Font("Verdana", 9.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
    Me.Button1.Location = New System.Drawing.Point(328, 0)
    Me.Button1.Name = "Button1"
    Me.Button1.Size = New System.Drawing.Size(200, 24)
    Me.Button1.TabIndex = 1
    Me.Button1.Text = "End Slightly Green Color"
    '
    'Button2
    '
    Me.Button2.Anchor = System.Windows.Forms.AnchorStyles.Bottom
    Me.Button2.BackColor = System.Drawing.Color.Transparent
    Me.Button2.Font = New System.Drawing.Font("Verdana", 9.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
    Me.Button2.Location = New System.Drawing.Point(584, 0)
    Me.Button2.Name = "Button2"
    Me.Button2.Size = New System.Drawing.Size(104, 24)
    Me.Button2.TabIndex = 3
    Me.Button2.Text = "Cancel"
    '
    'Label1
    '
    Me.Label1.Font = New System.Drawing.Font("Verdana", 14.25!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
    Me.Label1.Location = New System.Drawing.Point(16, 0)
    Me.Label1.Name = "Label1"
    Me.Label1.Size = New System.Drawing.Size(832, 24)
    Me.Label1.TabIndex = 4
    Me.Label1.Text = "Label1"
    Me.Label1.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter
    '
    'Panel2
    '
    Me.Panel2.Location = New System.Drawing.Point(0, 32)
    Me.Panel2.Name = "Panel2"
    Me.Panel2.Size = New System.Drawing.Size(864, 600)
    Me.Panel2.TabIndex = 5
    '
    'slightlyGreen
    '
    Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
    Me.ClientSize = New System.Drawing.Size(872, 669)
    Me.Controls.Add(Me.Panel2)
    Me.Controls.Add(Me.Label1)
    Me.Controls.Add(Me.Panel1)

```

```

    Me.Name = "slightlyGreen"
    Me.Text = "Query Results"
    Me.WindowState = System.Windows.Forms.FormWindowState.Maximized
    Me.Panell1.ResumeLayout(False)
    Me.ResumeLayout(False)

End Sub

#End Region
Dim Query As String
Dim string2 As String
Dim modifier As String
Dim color_face As String
Dim ds As New DataSet
Dim al As New ArrayList
Dim nextPix As Integer
Dim remainingPix As Integer

Dim modifier1 As Integer
Dim modifier2 As Integer
Dim modifier3 As Integer
Dim CanUpdate As Integer

'Connection String used to connect to the database.
'CSC-LAB=Name Of the Computer
'DBResearch=Name of the database

'2/16/05: Name of the Computer changed to reflect working on my
local machine (DEBS), Debra
Dim con As New System.Data.SqlClient.SqlConnection("data
source=vikas;initial catalog=DBResearch;user
id=dbresearch;password=dbresearch")

'Event handling procedure for the 'Update' button

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim I As Integer
    Dim more_less As String
    'con.Open()
    'This block of code checks which check boxes were clicked by the
user and for which
    'person in order to update the corresponding rows in the
database.For each person the
    'stored procedure Update_Data is invoked.
    For I = 0 To al.Count - 1
        If (al(I).meets_criteria.Checked() = True) Then
            more_less = "meets"
        ElseIf (al(I).more.Checked() = True) Then
            more_less = "more"
        ElseIf (al(I).less.Checked() = True) Then

```

```

        more_less = "less"
    End If

    Dim daptr = New SqlDataAdapter
    daptr.SelectCommand = New SqlCommand
    daptr.SelectCommand.CommandType = CommandType.StoredProcedure
    daptr.SelectCommand.CommandText = "dbo.Update_Data"
    daptr.SelectCommand.Connection = con
    daptr.SelectCommand.Parameters.Add("@ID", al(I).ID)
    daptr.SelectCommand.Parameters.Add("@color", color_face)
    daptr.SelectCommand.Parameters.Add("@more_less", more_less)
    daptr.SelectCommand.Parameters.Add("@modifier", modifier)
    con.Open()
    Try
        daptr.SelectCommand.ExecuteNonQuery()
    Catch ex As Exception
        MessageBox.Show("Failed to execute query")
    End Try

    con.Close()

Next

Dim slightlyBrown As slightlyBrown
slightlyBrown = New slightlyBrown
slightlyBrown.initialize(modifier1, modifier2, modifier3,
CanUpdate)
Me.Dispose()
slightlyBrown.Show()

End Sub

'This function is invoked when the form is loaded.The stored
procedure "Fetch_Data" is
'invoked and the fuzzy modifiers are passed as parameters to this
stored procedure.
'This stored procedure interprets the fuzzy modifiers and returns the
result set that
' matches the criteria.

Private Sub slightlyGreen_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
    Dim I As Integer
    Dim Count As Integer
    Dim cmd As New System.Data.SqlClient.SqlCommand("Fetch_Data",
con)

    cmd.CommandType = CommandType.StoredProcedure

    cmd.Parameters.Add(New SqlParameter("@query", SqlDbType.VarChar,
100))

```



```

cmd.Parameters("@query").Value = Query
cmd.Parameters.Add(New SqlParameter("@s1", SqlDbType.VarChar,
100))
cmd.Parameters("@s1").Value = modifier
cmd.Parameters.Add(New SqlParameter("@s2", SqlDbType.VarChar,
100))
cmd.Parameters("@s2").Value = color_face
Try
    Dim da As New System.Data.SqlClient.SqlDataAdapter(cmd)
    da.Fill(ds)
Catch ex As Exception
    MessageBox.Show(ex.ToString)
End Try

'displays the number of rows returned as a result of query (use
for debugging)
'MessageBox.Show(ds.Tables(0).Rows().Count)

If color_face = "Green" Or color_face = "Blue" Or color_face =
"Brown" Or color_face = "green" Or color_face = "blue" Or color_face =
"brown" Then
    Labell.Text() = "To what extent do you agree these people
have " + modifier + " " + color_face + " Eyes?"
Else
    Labell.Text() = "People With " + modifier + " " + color_face
+ " Faces"
End If

If (ds.Tables(0).Rows().Count) <> 0 Then
    Count = ds.Tables(0).Compute("COUNT(ID)", "")

    'The following block of code creates a new object of ImageSet
for each
    'record in the which result set which contains the image and
the Check Boxes
    'associated with it. It also reads the image in binary
format from the database
    'and displays it appropriately.

For I = 0 To Count - 1
    Dim S As New ImageSet(color_face)
    Dim bits As Byte() = CType(ds.Tables(0).Rows(I).Item(2),
Byte())

    Dim memorybits As New MemoryStream(bits)
    Dim bitmap As New Bitmap(memorybits)

    S.picture.Image = bitmap
    al.Add(S)
    al(I).ID = ds.Tables(0).Rows(I).Item(0)
Next I

```

```

page      'variable indicating the first picture to be displayed on a
nextPix = 0
'variable indicating the remaining pictures to be displayed
remainingPix = al.Count

placeComponents(nextPix)

updateButtons()

Labell1.Text() = "To what extent do you agree these people
have " + modifier + " " + color_face + " Eyes?"

'Changes the labels on the Panel depending on the query.
'If color_face = "Green" Or color_face = "Blue" Or color_face
= "Brown" Or color_face = "green" Or color_face = "blue" Or color_face =
"brown" Then
'Labell1.Text() = "People with " + modifier + " " + color_face
+ " Eyes"
'Else
' Labell1.Text() = "People With " + modifier + " " +
color_face + " Faces"
'End If
Else
Me.Button3.Enabled = False
MessageBox.Show("No images match search criteria. Please
Press End Button for next search")
End If
End Sub

'The following block of code places the components on the Group Box
and
'then places the group box on the panel to be displayed.
Private Sub placeComponents(ByVal Position1 As Integer)

Dim X As Integer, Y As Integer
Y = 8
Dim I As Integer

'limit the display to 6 images
Dim lastPix As Integer

'determining the last picture for the current page
If remainingPix <= 6 Then
    lastPix = Position1 + remainingPix - 1
Else : lastPix = Position1 + 5
End If

For I = Position1 To lastPix
    Dim G As New GroupBox
    G.Location() = New Point(8 + X, Y)

```

```

G.Size() = New Size(268, 295) 'was (288,350)

al(I).picture.Location = New Point(50, 24)
al(I).picture.size = New Size(200, 200) 'was (224,120)

250) al(I).meets_criteria.location = New Point(32, 225) 'was (32,
al(I).meets_criteria.size = New Size(176, 24)

al(I).more.size = New Size(176, 24)
al(I).more.location = New Point(32, 250) 'was (32, 275)

al(I).less.size = New Size(176, 24)
al(I).less.location = New Point(32, 275) 'was (32,300)

G.Controls.Add(al(I).picture)
G.Controls.Add(al(I).meets_criteria)
G.Controls.Add(al(I).more)
G.Controls.Add(al(I).less)
Panel2.Controls.Add(G)

'Allows to display multiple rows of records.
X = 300 + X
If ((I Mod 6) = 2) Then
    Y = 295 + 5 + Y
    X = 0
End If
Next

'update the number of picutes left to be displayed
If remainingPix > 6 Then
    remainingPix = remainingPix - 6
Else
    remainingPix = 0
End If

'update first picture on next page
nextPix = nextPix + 6
End Sub

'set up buttons depending on number of images in query result
Private Sub updateButtons()
    If al.Count <= 6 Or remainingPix = 0 Then
        Me.Button3.Enabled = False
        Me.Button1.Enabled = True
    Else
        Me.Button3.Enabled = True
        Me.Button1.Enabled = False
    End If
End Sub

```

```

    Public Sub initialize(ByVal m1 As Integer, ByVal m2 As Integer, ByVal
m3 As Integer, ByVal cu As Integer)
        Query = "select * from person where color.color = 'green';"
        modifier = "Slightly"
        color_face = "Green"

        modifier1 = m1
        modifier2 = m2
        modifier3 = m3
        CanUpdate = cu
    End Sub

    'Event handling for 'Cancel' button
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        con.Close()
        Me.Dispose()
    End Sub
End Sub

    'Event handling for 'More' button
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        'clear existing pictures
        Panel2.Controls.Clear()

        'build and display next page
        placeComponents(nextPix)
        Panel2.Show()

        updateButtons()
    End Sub

    '-----
    'The class ImagesSet is a user defined class which has the image
    'and the check boxes as its components.

    Public Class ImageSet
        Public picture As PictureBox
        Public meets_criteria As RadioButton
        Public more As RadioButton
        Public Less As RadioButton
        Public ID As Integer

        Public Sub New(ByVal color_face As String)
            picture = New PictureBox
            meets_criteria = New RadioButton
            more = New RadioButton

```

```

Less = New RadioButton
meets_criteria.Text = "Meets The Criteria"
meets_criteria.Checked = True
'Changes the labels on the Check Boxes depending on the
query.
If color_face = "Green" Or color_face = "green" Then
    more.Text = "More Green"
    Less.Text = "Less Green"
End If
If color_face = "Brown" Or color_face = "brown" Then
    more.Text = "More Brown"
    Less.Text = "Less Brown"
End If
If color_face = "Blue" Or color_face = "blue" Then
    more.Text = "More Blue"
    Less.Text = "Less Blue"
End If
If color_face = "Broad" Or color_face = "broad" Then
    more.Text = "More Broad"
    Less.Text = "Less Broad"
End If
If color_face = "Narrow" Or color_face = "narrow" Then
    more.Text = "More Narrow"
    Less.Text = "Less Narrow"
End If
ID = 0

End Sub

End Class
End Class

'Database Research Summer 2004
'Karen Joy and Smita Dattatri
'This form invokes the stored procedure "Fetch Data" which interprets the
Fuzzy query and
'returns the set of rows to be displayed on the User Interface.This form
also invokes a
'stored procedure called Update_data which updates the database depending
on the feedback
'given by the user.

'Spring, 2005
'Debra Duke
'Updated to add multiple page viewing functionality.

'Fall 2005
'Shweta Sanghi
'Updated the functionality to train the prototype

Imports System.IO

```

```
Imports System.Data.SqlClient

Public Class slightlyBrown
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form
    Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents Panell1 As System.Windows.Forms.Panel
    Friend WithEvents Button2 As System.Windows.Forms.Button
    Friend WithEvents Button1 As System.Windows.Forms.Button
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents Button3 As System.Windows.Forms.Button
    Friend WithEvents Panel2 As System.Windows.Forms.Panel
    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
    Me.Panell1 = New System.Windows.Forms.Panel
    Me.Button3 = New System.Windows.Forms.Button
    Me.Button1 = New System.Windows.Forms.Button
    Me.Button2 = New System.Windows.Forms.Button
    Me.Label1 = New System.Windows.Forms.Label
    Me.Panel2 = New System.Windows.Forms.Panel
    Me.Panell1.SuspendLayout()
    Me.SuspendLayout()
    '

```

```

'Panell
,
Me.Panell.Controls.Add(Me.Button3)
Me.Panell.Controls.Add(Me.Button1)
Me.Panell.Controls.Add(Me.Button2)
Me.Panell.Dock = System.Windows.Forms.DockStyle.Bottom
Me.Panell.Location = New System.Drawing.Point(0, 637)
Me.Panell.Name = "Panell"
Me.Panell.Size = New System.Drawing.Size(872, 32)
Me.Panell.TabIndex = 2
,
'Button3
,
Me.Button3.Anchor = System.Windows.Forms.AnchorStyles.Bottom
Me.Button3.Font = New System.Drawing.Font("Verdana", 9.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
Me.Button3.Location = New System.Drawing.Point(96, 0)
Me.Button3.Name = "Button3"
Me.Button3.Size = New System.Drawing.Size(104, 24)
Me.Button3.TabIndex = 4
Me.Button3.Text = "More"
,
'Button1
,
Me.Button1.Anchor = System.Windows.Forms.AnchorStyles.Bottom
Me.Button1.BackColor = System.Drawing.Color.Transparent
Me.Button1.Font = New System.Drawing.Font("Verdana", 9.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
Me.Button1.Location = New System.Drawing.Point(272, 0)
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(208, 24)
Me.Button1.TabIndex = 1
Me.Button1.Text = "End Slightly Brown Color"
,
'Button2
,
Me.Button2.Anchor = System.Windows.Forms.AnchorStyles.Bottom
Me.Button2.BackColor = System.Drawing.Color.Transparent
Me.Button2.Font = New System.Drawing.Font("Verdana", 9.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
Me.Button2.Location = New System.Drawing.Point(552, 0)
Me.Button2.Name = "Button2"
Me.Button2.Size = New System.Drawing.Size(104, 24)
Me.Button2.TabIndex = 3
Me.Button2.Text = "Cancel"
,
'Label1
,

```

```

        Me.Label1.Font = New System.Drawing.Font("Verdana", 14.25!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.Label1.Location = New System.Drawing.Point(16, 0)
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New System.Drawing.Size(832, 24)
        Me.Label1.TabIndex = 4
        Me.Label1.Text = "Label1"
        Me.Label1.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter
    '
    'Panel2
    '
    Me.Panel2.Location = New System.Drawing.Point(0, 32)
    Me.Panel2.Name = "Panel2"
    Me.Panel2.Size = New System.Drawing.Size(864, 600)
    Me.Panel2.TabIndex = 5
    '
    'slightlyBrown
    '
    Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
    Me.ClientSize = New System.Drawing.Size(872, 669)
    Me.Controls.Add(Me.Panel2)
    Me.Controls.Add(Me.Label1)
    Me.Controls.Add(Me.Panel1)
    Me.Name = "slightlyBrown"
    Me.Text = "Query Results"
    Me.WindowState = System.Windows.Forms.FormWindowState.Maximized
    Me.Panel1.ResumeLayout(False)
    Me.ResumeLayout(False)

End Sub

#End Region
Dim Query As String
Dim string2 As String
Dim modifier As String
Dim color_face As String
Dim ds As New DataSet
Dim dsHst As New DataSet 'varibale storing dataset which is the
result of the query
Dim dsID As New DataSet 'varibale storing dataset which is the
result of the query
Dim al As New ArrayList
Dim nextPix As Integer
Dim remainingPix As Integer

Dim modifier1 As Integer
Dim modifier2 As Integer
Dim modifier3 As Integer
Dim CanUpdate As Integer

```



```

'Connection String used to connect to the database.
'CSC-LAB=Name Of the Computer
'DBResearch=Name of the database

'2/16/05: Name of the Computer changed to reflect working on my
local machine (DEBS), Debra
Dim con As New System.Data.SqlClient.SqlConnection("data
source=vikas;initial catalog=DBResearch;user
id=dbresearch;password=dbresearch")

'Event handling procedure for the 'Update' button

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim I As Integer
    Dim more_less As String
    Dim CountHist As Integer
    Dim CountID As Integer
    'con.Open()
    'This block of code checks which check boxes were clicked by the
user and for which
    'person in order to update the corresponding rows in the
database.For each person the
    'stored procedure Update_Data is invoked.
    For I = 0 To al.Count - 1
        If (al(I).meets_criteria.Checked() = True) Then
            more_less = "meets"
        ElseIf (al(I).more.Checked() = True) Then
            more_less = "more"
        ElseIf (al(I).less.Checked() = True) Then
            more_less = "less"
        End If

        Dim daptr = New SqlDataAdapter
        daptr.SelectCommand = New SqlCommand
        daptr.SelectCommand.CommandType = CommandType.StoredProcedure
        daptr.SelectCommand.CommandText = "dbo.Update_Data"
        daptr.SelectCommand.Connection = con
        daptr.SelectCommand.Parameters.Add("@ID", al(I).ID)
        daptr.SelectCommand.Parameters.Add("@color", color_face)
        daptr.SelectCommand.Parameters.Add("@more_less", more_less)
        daptr.SelectCommand.Parameters.Add("@modifier", modifier)
        con.Open()
        Try
            daptr.SelectCommand.ExecuteNonQuery()
        Catch ex As Exception
            MessageBox.Show("Failed to execute query")
        End Try

        con.Close()

```

```

Next

'increase the update value
CanUpdate = CanUpdate + 1

Dim md As Integer

If (CanUpdate < 4) Then
    If (CanUpdate = 2) Then
        md = modifier2
    ElseIf (CanUpdate = 3) Then
        md = modifier3
    End If

    Select Case md
        Case 1
            Dim slightlyBlue As slightlyBlue
            slightlyBlue = New slightlyBlue
            slightlyBlue.initialize(modifier1, modifier2,
modifier3, CanUpdate)
            Me.Dispose()
            slightlyBlue.Show()

        Case 2
            Dim MediumBlue As MediumBlue
            MediumBlue = New MediumBlue
            MediumBlue.initialize(modifier1, modifier2,
modifier3, CanUpdate)
            Me.Dispose()
            MediumBlue.Show()

        Case 3
            Dim VeryBlue As VeryBlue
            VeryBlue = New VeryBlue
            VeryBlue.initialize(modifier1, modifier2, modifier3,
CanUpdate)
            Me.Dispose()
            VeryBlue.Show()
    End Select
    'evoke forms
Else
    'Update the history table and then evoke the message learn
screen
    ''Writing record into the history table

    '' Fetching all the id's of images
    Dim cmdID As New
System.Data.SqlClient.SqlCommand("FetchImageID", con)

    cmdID.CommandType = CommandType.StoredProcedure

Try

```

```

        Dim da As New System.Data.SqlClient.SqlDataAdapter(cmdID)
        da.Fill(dsID)
    Catch ex As Exception
        MessageBox.Show(ex.ToString)
    End Try
    ' ' MessageBox.Show("table count images",
dsID.Tables(0).Rows().Count)

    If (dsID.Tables(0).Rows().Count) <> 0 Then
        CountID = dsID.Tables(0).Compute("COUNT(ID)", "")

        'The following block of code creates a new object of
ImageSet for each
        'record in the which result set which contains the image
and the slide bar
        'associated with it. It also reads the image in binary
format from the database
        'and displays it appropriately.

        For I = 0 To CountID - 1
            'Find weights regarding eye color of the each images
            Dim cmdImage As New
System.Data.SqlClient.SqlCommand("FetchHist", con)

            cmdImage.CommandType = CommandType.StoredProcedure
            cmdImage.Parameters.Add(New SqlParameter("@ID",
SqlDbType.Int))
            cmdImage.Parameters("@ID").Value =
dsID.Tables(0).Rows(I).Item(0)

            Try
                Dim da As New
System.Data.SqlClient.SqlDataAdapter(cmdImage)
                da.Fill(dsHst)
            Catch ex As Exception
                MessageBox.Show(ex.ToString)
            End Try
            'MessageBox.Show("table count",
dsHst.Tables(0).Rows().Count)

            ' ' writing into the history table
            If (dsHst.Tables(0).Rows().Count) <> 0 Then
                CountHist = dsHst.Tables(0).Compute("COUNT(ID)",
"")

                'The following block of code creates a new object
of ImageSet for each record in the
                'which result set which contains the image and
the Check Boxes associated with it.
                'It also reads the image in binary format from
the database and displays it appropriately.

```

```

        'increment counter to record the number of each
type of query being executed
        Dim increment As SqlClient.SqlCommand
        increment = New SqlClient.SqlCommand
        increment.CommandType =
CommandType.StoredProcedure
        increment.CommandText = "dbo.Insert_Hist"
        increment.Connection = con

        increment.Parameters.Add(New SqlParameter("@ID",
SqlDbType.Int))
        increment.Parameters("@ID").Value =
dsHst.Tables(0).Rows(0).Item(0)

        increment.Parameters.Add(New
SqlParameter("@ValueBlue", SqlDbType.Decimal))
        increment.Parameters("@ValueBlue").Value =
dsHst.Tables(0).Rows(0).Item(2)

        increment.Parameters.Add(New
SqlParameter("@ValueBrown", SqlDbType.Decimal))
        increment.Parameters("@ValueBrown").Value =
dsHst.Tables(0).Rows(1).Item(2)

        increment.Parameters.Add(New
SqlParameter("@ValueGreen", SqlDbType.Decimal))
        increment.Parameters("@ValueGreen").Value =
dsHst.Tables(0).Rows(2).Item(2)

        con.Open()
        Try
            increment.ExecuteNonQuery()
        Catch ex As Exception
            MessageBox.Show(ex.ToString)
        End Try

        dsHst.Clear()
        con.Close()

    End If

    con.Close()

Next I

Else
    MessageBox.Show("No images match your criteria")
End If

'' end of creating history

Dim MessageLearn As MessageLearn

```

```

        MessageLearn = New MessageLearn
        Me.Dispose()
        MessageLearn.Show()
    End If

End Sub

'This function is invoked when the form is loaded.The stored
procedure "Fetch_Data" is
'invoked and the fuzzy modifiers are passed as parameters to this
stored procedure.
'This stored procedure interprets the fuzzy modifiers and returns the
result set that
' matches the criteria.

Private Sub slightlyBrown_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
    Dim I As Integer
    Dim Count As Integer
    Dim cmd As New System.Data.SqlClient.SqlCommand("Fetch_Data",
con)

    cmd.CommandType = CommandType.StoredProcedure

    cmd.Parameters.Add(New SqlParameter("@query", SqlDbType.VarChar,
100))
    cmd.Parameters("@query").Value = Query
    cmd.Parameters.Add(New SqlParameter("@s1", SqlDbType.VarChar,
100))
    cmd.Parameters("@s1").Value = modifier
    cmd.Parameters.Add(New SqlParameter("@s2", SqlDbType.VarChar,
100))
    cmd.Parameters("@s2").Value = color_face
    Try
        Dim da As New System.Data.SqlClient.SqlDataAdapter(cmd)
        da.Fill(ds)
    Catch ex As Exception
        MessageBox.Show(ex.ToString)
    End Try

    'displays the number of rows returned as a result of query (use
for debugging)
    'MessageBox.Show(ds.Tables(0).Rows().Count)

    If color_face = "Green" Or color_face = "Blue" Or color_face =
"Brown" Or color_face = "green" Or color_face = "blue" Or color_face =
"brown" Then
        Labell.Text() = "To what extent do you agree these people
have " + modifier + " " + color_face + " Eyes?"
    Else

```

```

        Labell1.Text() = "People With " + modifier + " " + color_face
+ " Faces"
    End If

    If (ds.Tables(0).Rows().Count) <> 0 Then
        Count = ds.Tables(0).Compute("COUNT(ID)", "")

        'The following block of code creates a new object of ImageSet
for each
        'record in the which result set which contains the image and
the Check Boxes
        'associated with it. It also reads the image in binary
format from the database
        'and displays it appropriately.

    For I = 0 To Count - 1
        Dim S As New ImageSet(color_face)
        Dim bits As Byte() = CType(ds.Tables(0).Rows(I).Item(2),
Byte())

        Dim memorybits As New MemoryStream(bits)
        Dim bitmap As New Bitmap(memorybits)

        S.picture.Image = bitmap
        al.Add(S)
        al(I).ID = ds.Tables(0).Rows(I).Item(0)
    Next I

    'variable indicating the first picture to be displayed on a
page
    nextPix = 0
    'variable indicating the remaining pictures to be displayed
remainingPix = al.Count

    placeComponents(nextPix)

    updateButtons()

    'Labell1.Text() = "People With Slightly Brown Eyes"

    'Changes the labels on the Panel depending on the query.
    'If color_face = "Green" Or color_face = "Blue" Or color_face
= "Brown" Or color_face = "green" Or color_face = "blue" Or color_face =
"brown" Then
        'Labell1.Text() = "People with " + modifier + " " + color_face
+ " Eyes"
        'Else
        ' Labell1.Text() = "People With " + modifier + " " +
color_face + " Faces"
        'End If
    Else
        Me.Button3.Enabled = False

```

```

        MessageBox.Show("No images match search criteria. Please
Press End Button for next search")
    End If
End Sub

'The following block of code places the components on the Group Box
and
'then places the group box on the panel to be displayed.
Private Sub placeComponents(ByVal Position1 As Integer)

    Dim X As Integer, Y As Integer
    Y = 8
    Dim I As Integer

    'limit the display to 6 images
    Dim lastPix As Integer

    'determining the last picture for the current page
    If remainingPix <= 6 Then
        lastPix = Position1 + remainingPix - 1
    Else : lastPix = Position1 + 5
    End If

    For I = Position1 To lastPix
        Dim G As New GroupBox
        G.Location() = New Point(8 + X, Y)
        G.Size() = New Size(268, 295) 'was (288,350)

        al(I).picture.Location = New Point(50, 24)
        al(I).picture.size = New Size(200, 200) 'was (224,120)

        al(I).meets_criteria.location = New Point(32, 225) 'was (32,
250)
        al(I).meets_criteria.size = New Size(176, 24)

        al(I).more.size = New Size(176, 24)
        al(I).more.location = New Point(32, 250) 'was (32, 275)

        al(I).less.size = New Size(176, 24)
        al(I).less.location = New Point(32, 275) 'was (32,300)

        G.Controls.Add(al(I).picture)
        G.Controls.Add(al(I).meets_criteria)
        G.Controls.Add(al(I).more)
        G.Controls.Add(al(I).less)
        Panel2.Controls.Add(G)

        'Allows to display multiple rows of records.
        X = 300 + X
        If ((I Mod 6) = 2) Then
            Y = 295 + 5 + Y

```

```

        X = 0
    End If
Next

'update the number of picutes left to be displayed
If remainingPix > 6 Then
    remainingPix = remainingPix - 6
Else
    remainingPix = 0
End If

'update first picture on next page
nextPix = nextPix + 6
End Sub

'set up buttons depending on number of images in query result
Private Sub updateButtons()
    If al.Count <= 6 Or remainingPix = 0 Then
        Me.Button3.Enabled = False
        Me.Button1.Enabled = True
    Else
        Me.Button3.Enabled = True
        Me.Button1.Enabled = False
    End If
End Sub

Public Sub initialize(ByVal m1 As Integer, ByVal m2 As Integer, ByVal
m3 As Integer, ByVal cu As Integer)
    Query = "select * from person where color.color = 'brown';"
    modifier = "Slightly"
    color_face = "Brown"

    modifier1 = m1
    modifier2 = m2
    modifier3 = m3
    CanUpdate = cu
End Sub

'Event handling for 'Cancel' button
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    con.Close()
    Me.Dispose()
End
End Sub

'Event handling for 'More' button

```



```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    'clear existing pictures
    Panel2.Controls.Clear()

    'build and display next page
    placeComponents(nextPix)
    Panel2.Show()

    updateButtons()
End Sub

```

```

'-----
'The class ImagesSet is a user defined class which has the image
'and the check boxes as its components.

```

```

Public Class ImageSet
    Public picture As PictureBox
    Public meets_criteria As RadioButton
    Public more As RadioButton
    Public Less As RadioButton
    Public ID As Integer

    Public Sub New(ByVal color_face As String)
        picture = New PictureBox
        meets_criteria = New RadioButton
        more = New RadioButton
        Less = New RadioButton
        meets_criteria.Text = "Meets The Criteria"
        meets_criteria.Checked = True
        'Changes the labels on the Check Boxes depending on the
query.
        If color_face = "Green" Or color_face = "green" Then
            more.Text = "More Green"
            Less.Text = "Less Green"
        End If
        If color_face = "Brown" Or color_face = "brown" Then
            more.Text = "More Brown"
            Less.Text = "Less Brown"
        End If
        If color_face = "Blue" Or color_face = "blue" Then
            more.Text = "More Blue"
            Less.Text = "Less Blue"
        End If
        If color_face = "Broad" Or color_face = "broad" Then
            more.Text = "More Broad"
            Less.Text = "Less Broad"
        End If
        If color_face = "Narrow" Or color_face = "narrow" Then
            more.Text = "More Narrow"
            Less.Text = "Less Narrow"
        End If
    End Sub
End Class

```

```

        End If
        ID = 0

    End Sub

End Class

'Fall 2005
'Shweta Sanghi
'This form thanks the user for help and quits. It also means that
' the update operation was successful.

Public Class MessageLearn
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form
Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents Button2 As System.Windows.Forms.Button
    <System.Diagnostics.DebuggerStepThrough(> Private Sub
InitializeComponent()
        Me.Label1 = New System.Windows.Forms.Label
        Me.Button2 = New System.Windows.Forms.Button

```

```

    Me.SuspendLayout()
    '
    'Label1
    '
    Me.Label1.Font = New System.Drawing.Font("Verdana", 14.25!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
    Me.Label1.Location = New System.Drawing.Point(56, 32)
    Me.Label1.Name = "Label1"
    Me.Label1.Size = New System.Drawing.Size(440, 32)
    Me.Label1.TabIndex = 0
    Me.Label1.Text = "Thanks for your help with this research"
    '
    'Button2
    '
    Me.Button2.Font = New System.Drawing.Font("Verdana", 12.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
    Me.Button2.Location = New System.Drawing.Point(216, 96)
    Me.Button2.Name = "Button2"
    Me.Button2.Size = New System.Drawing.Size(88, 32)
    Me.Button2.TabIndex = 3
    Me.Button2.Text = "End"
    '
    'MessageLearn
    '
    Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
    Me.ClientSize = New System.Drawing.Size(536, 165)
    Me.Controls.Add(Me.Button2)
    Me.Controls.Add(Me.Label1)
    Me.Location = New System.Drawing.Point(5000, 7000)
    Me.Name = "MessageLearn"
    Me.Text = "Form1"
    Me.ResumeLayout(False)

End Sub

#End Region

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Me.Dispose()
    Me.Close()
End
End Sub

End Class

```