IMPOSING GRAPH STRUCTURES ON SPACE-TIME
DENSITIES FOR INDEXING AND RETRIEVAL

by

ULUKBEK IBRAEV

A Dissertation submitted to the
Graduate School-New Brunswick
Rutgers, The State University of New Jersey
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
Graduate Program in Communication, Information and Library Studies
written under the direction of
Paul B. Kantor
and approved by

____________________________
____________________________
____________________________
____________________________

New Brunswick, New Jersey
October, 2005
fMRI is a brain imaging technique that allows us to see traces of neural activation caused by cognitive processes in the human brain, with a few seconds delay. The development of the fMRI technique, with its relatively high temporal and spatial resolution and non-invasive nature, opened a new era in functional neuroimaging. We can expect rapid growth of fMRI databases and with that the need to organize, index, store and retrieve this data.

The focus of this research is to explore using graph structures to represent cognitive processes in a brain and to assess their effectiveness for fMRI retrieval. The idea that it might be possible to represent dynamic brain images by graphs representing 4D space-time structures, where nodes represent activated brain areas and edges represent interactions between them, was put forward in a series of proposals by Paul Kantor and
Stephen Hanson, the third of which was funded by the NSF, supporting the work reported here. In the present work we did not attempt to validate that concept in terms of the specific space time localization of the activated brain regions reported by the graph. Instead, we seek to validate it in the context of query by image content. In that context we faced two problems: extracting the graph, and assessing the retrieval performance. The author has proposed the idea of linking time slices by a correlation measure (maximum correlation graph). The validation was in terms of the ROC curves for every possible retrieval case.

The results showed that the correlation-linking approach provides better retrieval performance than an alternative based on an anatomical representation of the brain (Brodmann vector). The Brodmann vector-based approach performed better than random guessing in 85% of all cases and the correlation-based approach performed better than the random guessing in 98% of all cases. This was much better than chance. Another method of link finding (minimum distance graph) was also shown to give performance comparable to the correlation-based links and performed better than chance in 98% of all cases. The results of the experiments supported the proposed approach for fMRI data retrieval.
I would like to express my thanks to my advisor Paul Kantor, who kept on encouraging, guiding and helping me during my many years of Ph.D. studies. Without his constant support, constructive criticism and valuable suggestions I would not be able to finish this dissertation. I was very lucky to have a mentor who was always there for his students.

Thanks to my committee members, Prof. Tefko Saracevic, Prof. Sven Dickinson, and Prof. Gheorghe Muresan. Their timely suggestions and insights pushed me to do a better job in writing this dissertation. Thanks to my family, my father, my grandmother, and my brother, for their sacrifices and continuous support from overseas. Special thanks to my mother, who always dreamed about this day. Thanks to my fiancee, Jing Sun, for her encouragement and unconditional love. I would also like to express my gratitude to Nicu Cornea, for his invaluable help. It was a pleasure working with him on the Brain Images project. I thank Ali Shokoufandeh and Diego Macrini for generously allowing me to use their graph matching code. Finally, I would like to thank my colleagues at Alexandria Project Laboratory. Writing this dissertation would not be an enjoyable experience without them.
Dedication

To my Mother, Saginbubu Idrisova.
# Table of Contents

Abstract ................................................................................................................................. ii  
Acknowledgments .............................................................................................................. iv  
Dedication ............................................................................................................................ v 
List of Tables ....................................................................................................................... xi  
List of Figures ...................................................................................................................... xii  

Chapter 1: Introduction ......................................................................................................... 1  
1.1 Functional Magnetic Resonance Images (fMRI) ..................................................... 1  
1.2 Content-Based fMRI Retrieval ............................................................................. 2  
1.3 Graph Structures ..................................................................................................... 3  
1.4 Human Brain Activity .............................................................................................. 4  
1.5 Research Questions ................................................................................................. 4  
1.6 Organization of This Thesis .................................................................................... 7  

Chapter 2: Content-Based Image Retrieval ......................................................................... 10  
2.1 GEMINI .................................................................................................................... 11  
2.2 Feature Extraction .................................................................................................... 14  
2.3 Color .......................................................................................................................... 14  
2.4 Texture ....................................................................................................................... 16  
2.5 Shape .......................................................................................................................... 18  
2.5.1 Region Based Shape ........................................................................................... 22
2.5.2 Point Based Shape Descriptors .......................................................... 24
2.5.3 Skeleton Based Shape Descriptors .................................................. 28
2.5.4 Shape and Medical Images ............................................................... 29

Chapter 3: Image Retrieval Systems ......................................................... 31
3.1 QBIC .................................................................................................. 31
3.2 Photobook ....................................................................................... 33
3.3 VisualSEEk ...................................................................................... 34
3.4 NeTra ............................................................................................... 34
3.5 MARS .............................................................................................. 35
3.6 PicHunter ......................................................................................... 36
3.7 Blobworld ....................................................................................... 37
3.8 Search Engine for 3D Models ........................................................... 38
3.9 Marie ............................................................................................... 40
3.10 Hybrid Systems That Use Text and Image ........................................ 42
   3.10.1 Image Labeling ........................................................................ 43
   3.10.2 Region Naming ........................................................................ 44
3.11 Current Research Issues ................................................................. 46

Chapter 4: Time-Space Densities and fMRI ............................................... 48
4.1 Time-Space Densities ........................................................................ 48
4.2 Basic Physics of fMRI ...................................................................... 49
4.3 Brain Activity and fMRI ................................................................. 56
   4.3.1 Brain Activity .......................................................................... 56
   4.3.2 The BOLD Effect .................................................................... 57
9.2 Activation Maps ........................................................................................................ 107

Chapter 10: Content-Based fMRI Retrieval Using Brodmann Vectors .......... 111
  10.1 Conceptual Framework for Matching Brodmann Vectors .......... 111
  10.2 Results of the Brodmann Vector Matching ......................... 113
  10.3 Using ROC Curves to Represent Retrieval Performance ........ 118

Chapter 11: Content-Based fMRI Dataset Retrieval Using Minimum Distance
  Graphs ................................................................................................................ 126
  11.1 Conceptual Framework for Using Minimum Distance Graphs ..... 126
  11.2 Results of the MDG Based fMRI Matching ......................... 129
  11.3 ROC Curves for MDG Matching ......................................................... 133

Chapter 12: Content-Based fMRI Retrieval Using Maximum Correlation
  Graphs ................................................................................................................ 139
  12.1 Conceptual Framework for Using Maximum Correlation
      Graphs ........................................................................................................... 139
  12.2 Results of the MCG Based fMRI Matching ......................... 141
  12.3 ROC Curves for MCG Matching ......................................................... 146

Chapter 13: Concluding Discussion on fMRI Matching ......................... 151
  13.1 Comparison of the three fMRI Matching Heuristics .......... 151
  13.2 Limitations of the Research .......................................................... 155

Chapter 14: Conclusion ......................................................................................... 157
  14.1 A Brief Summary of Our Experiments .................................. 157
  14.2 Significance of the Research and Directions for Further Study .... 160
List of Tables

Table 7.1: Brief summary of heuristics used for content-based fMRI retrieval. ................................................................. 91

Table 8.1: Description of fMRI experiments in the repository. ......................... 99

Table 10.1: ROC curve areas ranked in decreasing order. Only the top 15 queries are shown. ............................................................... 123

Table 11.1: ROC areas for MDG retrieval ranked in decreasing order. Only the top 15 queries are shown. ............................................................. 137

Table 12.1: ROC areas for MCG retrieval ranked in decreasing order. Only the top 15 queries are shown. ............................................................. 149

Table 13.1: Mean, standard deviation, and standard error of the mean for areas under the ROC curve. ................................................................. 152

Table 13.2: Number of datasets for each condition in the collection. ............... 153

Table 13.3: Mean, standard deviation, and the standard error of the mean of ROC areas for fMRI matching heuristics, ordered by condition. .... 154
List of Figures

Figure 1.1: Flow chart of the path from psychological experiment to data retrieval. ................................................................. 7

Figure 2.1: Color histogram of a fictitious sunset photograph. ......................... 15

Figure 2.2: a) 3×3 image with four colors; b) co-occurrence matrix. ............... 17

Figure 2.3: Tangent lines of a curve. ................................................................. 19

Figure 2.4: Patches generated by a set of boundary fragments for a cone sketch. Arrows indicate locations of the fragment endpoints. Fragments are shown in canonical positions in the center of the template squares. ................................................................. 21

Figure 2.5: The initial image is repeatedly convolved with Gaussians to produce the set of scale space images (shown on the left). Adjacent Gaussian images are subtracted to produce the difference of Gaussian images (shown on the right). .................................................. 25

Figure 2.6: Shock types, courtesy of Siddiqi et al. (1999). ............................... 27

Figure 2.7: Shock graph of a simple hammer shape. Color to shock type mapping is shown in the right column (1\textsuperscript{st} order shock is yellow, 4-th order is red). ................................................................. 29

Figure 3.1: Hierarchical tree for text and regions (Barnard et al., 2003). ......... 42

Figure 4.1: Resonance frequency of hydrogen nucleus in magnetic field. ......... 50

Figure 4.2: Spatial encoding in the gradient field. Direction of the magnetic
field is in the Z-axis direction, shown at the bottom. Resonance
frequencies for different locations in magnetic field are shown at the
top. ................................................................. 52

Figure 4.3: Isolated plane of tissue exposed to the gradient field. .......... 53

Figure 4.4: Single slice of tissue is examined many times, each time with a
gradient oriented in different direction. .............................................. 54

Figure 4.5: Physiological changes during brain activation. The fMRI imaging is
based on the lower three blocks. The arrows inside blocks indicate
the direction and magnitude of the change. ........................................ 57

Figure 4.6: Schematic description of the BOLD effect. There is an initial dip of
1-3 sec. followed by a ramp of 3-8 sec. After the end of the
stimulation, the signal undershoots the baseline and resolves in
about 20 sec. ..................................................................................... 59

Figure 4.7: Plot of the double-gamma HRF as defined in FSL. ............... 60

Figure 5.1: Overview of the main methods for fMRI data analysis. ............ 61

Figure 7.1: Building a Brodmann vector. Each Brodmann area that has
activated voxels will have a corresponding non-zero component in
the Brodmann vector. ...................................................................... 84

Figure 7.2: The Minimum Distance Graph (MDG). (a) Given nodes and edge
weights (Euclidean distances); (b) Dijkstra’s shortest path algorithm
is used to build the MDG by connecting nodes so that the signal
path from root to any node is minimal. ............................................. 87

Figure 7.3: The Maximum Correlation Graph (MCG). (a) Given a set of nodes
and edges with weights (correlation coefficients); (b) Dijkstra’s shortest path algorithm is used to build the MCG. The result is that the sum of correlations along any path from root is maximal.

Figure 7.4: The DAG matching algorithm. (a) Given a query and a model graph, (b) form bipartite graph, in which the edge weights are the pair-wise node similarities. Then, (c) compute a maximum matching and add the best edge to the solution set. Finally, (d) split the graphs at the matched nodes, and (e) recursively descend.

Courtesy of Macrini (2003).

Figure 9.1: Typical Precision-Recall curve.

Figure 9.2: $t$-map for one Recall experiment subject. Red areas represent brain regions with statistically significant activations.

Figure 10.1: Hit matrix for the recall data. Subjects are ordered along the X and Y axes, right to left, and from top to bottom. Similarity to color mapping is at the top.

Figure 10.2: Hit matrix for the morality data. Subjects are ordered along the X and Y axes, right to left, and from top to bottom. There is a visible pattern that shows some experimental conditions are similar.

Figure 10.3: Hit matrix for the event perception data. Subjects are ordered along the X/Y axes, right to left, and from top to bottom.

Figure 10.4: Hit matrix for the entire repository. Similarities are represented by color intensities (yellow – high similarity, blue – low similarity).

Figure 10.5: ROC curve for three queries from top, middle and bottom of the
ranked ROC areas list. ................................................................. 121

Figure 10.6: Precision plotted as a function of recall for a Brodmann vectors. ... 122

Figure 10.7: Histogram of frequencies for the areas under ROC curve for 986
queries. Most queries have the ROC area above 0.5 (random
guessing). ....................................................................................... 124

Figure 11.1: Hit matrix for the recall data using MDG matching. It appears
there is no visible pattern. Mapping from similarity to color is at
the top. .................................................................................................. 128

Figure 11.2: Hit matrix for the morality data using MDG matching. It appears
there is no visible pattern. Color mapping is at the top. .............. 130

Figure 11.3: Hit matrix for the event perception data using MDG matching.
Again, there is no visible pattern. .................................................. 131

Figure 11.4: Hit matrix for all experiments using MDG matching. It appears
that morality data is strongly separated from other experiments. .... 132

Figure 11.5: ROC curves for three queries with the top, the median and the
worst performances. ................................................................. 135

Figure 11.6: Precision as a function of recall for MDG matching. ............... 136

Figure 11.7: Histogram of frequencies for areas under ROC curve for 928
MDG queries. Most queries have ROC areas above 0.5 (random
guessing). ....................................................................................... 138

Figure 12.1: Hit matrix for the recall data, using MCG matching. .............. 142

Figure 12.2: Hit matrix for the morality data, using MCG matching. ............ 143

Figure 12.3: Hit matrix for the event perception data, using MCG matching. ..... 144
Figure 12.4: Hit matrix for all data using MCG matching. Morality data is strongly separated from other experiments. ........................................ 145

Figure 12.5: ROC curves using MCG matching for the same three subjects as in Figure 11.5. ...................................................................................... 147

Figure 12.6: Precision as a function of recall for MCG matching. ................. 148

Figure 12.7: Histogram of frequencies for areas under the ROC curve for 928 MCG queries. ...................................................................................... 150
Chapter 1: Introduction

1.1 Functional Magnetic Resonance Images (fMRI)

Recent years have seen an enormous increase in the size of digital data collections worldwide. However, we cannot use or access these collections unless there are mechanisms to efficiently organize, index and retrieve this type of data. A variety of algorithms and systems have been developed for information retrieval (IR) (Baeza-Yates et al., 1999). The focus has been mainly on textual representation of information, but today digital data collections are not just limited to textual information. We need systems that can efficiently search for images, music, video and other types of multimedia data. One important type of multimedia data is time-sequenced three-dimensional (3D) images. This type of multimedia data comes from a wide variety of sources. One rapidly developing source of time-sequenced 3D images is functional Magnetic Resonance Imaging (fMRI).
fMRI is a brain imaging technique that allows us to see traces of neural activation caused by cognitive processes in the human brain, in almost real time. The development of the fMRI technique, with its relatively high temporal and spatial resolution and non-invasive nature, opened a new era in functional neuroimaging (Kim et al., 1997). Therefore, we can expect rapid growth of fMRI databases and the need to organize, index, store, retrieve and analyze the data.

1.2 Content-Based fMRI Retrieval

A typical fMRI experiment produces a series of 200 to 400 three-dimensional brain images acquired at intervals of a few second. The size of the series can be several dozens of megabytes. There are thousands of researchers doing fMRI experiments all over the world. The total amount of fMRI data collected around the world could be in the range of thousands of terabytes per imaging facility, and growing. Sharing of this data would benefit brain scientists all around the world. For instance, large public fMRI databases containing shared data can accelerate data mining of associations between structures and functions of the human brain (Megalooikonomou et al., 1999). Also, the data sharing will benefit thousands of researchers by permitting them to reuse fMRI data that has been analyzed and reported in the neuroimaging literature (Fox and Lancaster, 2001). Unfortunately, only a small fraction of the data is currently made available to other researchers (Van Horn and Gazzaniga, 2002). One of the main problems in sharing data is the challenge of content-based indexing and retrieval of fMRI data. For instance, we were unable to find references to studies that applied techniques developed for 2D image
retrieval, which rely on features such as color, shape, and texture, to time sequenced fMRI data. It appears that there might be several reasons why techniques for 2D-image retrieval are not effective in the case of fMRI data. All fMRI images have similar color distribution, shape and texture since they are all images of human brains. A more serious problem with fMRI is that scanned images are noisy and often corrupted by motion (Salli et al, 2000). It appears to be very difficult to separate noise from cognitive processes in the “thinking” human brain. Finally, our understanding of cognitive processes in the human brain is incomplete. Content-based retrieval of fMRI images requires efficient and robust methods that can successfully address these challenges.

1.3 Graph Structures

In this context, the interesting and important research questions are: how should we compare two fMRI series, what features to extract, how to define a distance function between fMRI series, what kind of indices provide for the efficient and robust content-based retrieval of fMRI data.

All of these questions are quite hard and there are no definitive answers yet. However, graph structures have proved to be promising for Image Retrieval in two dimensions (Shokoufandeh et al., 1999) and three dimensions (Sundar et al., 2003). Imagine that a closed curve boundary in plane is set on fire. The set of points that are reached simultaneously by two or more fire fronts is called the shape’s medial axis. Shokoufandeh et al. (1999) proposed to use the medial axes to build a two-dimensional skeleton of a shape, a so called “shock” graph, and use it for shape matching. Sundar et
al. (2003) further developed this idea by using “medial” surfaces to construct skeletons of three-dimensional objects that can be used for shape matching. Thus, reducing two and three dimensional objects to graphs has been shown effective for some indexing and retrieval.

### 1.4 Human Brain Activity

There are two fundamental principles to the brain’s functional organization: functional specialization and functional integration (Friston, 2004). Functional specialization means that there are various clusters of neurons in brain responsible for different tasks. For example, one region of brain processes input from our visual sensors and another region processes motor movements. Functional integration means that different parts of the brain interact with each other to complete a task. Thus, many parts of the brain exchange signals and work together to complete a cognitive task.

If we could see this chain-like process of interactions in the brain, it might resemble a graph structure whose nodes are specific areas in the brain and whose edges are interaction pathways.

### 1.5 Research Questions

Suppose that we have a collection of fMRI datasets of human subjects performing different cognitive tasks. Users would like to search fMRI collections based on semantics of the data. Unfortunately, at present, there is no theory for extracting semantics from the
fMRI data. One way to address this problem is to seek low-level features that correlate with the high-level semantics. For example, it is generally accepted in the brain sciences, that a neural activity in the brain cause changes in the Blood Oxygen Level Dependant (BOLD, See Chapter 4) signal, which is recorded in the MRI scanner. Thus, it appears that the low-level feature, BOLD signal, correlates with the high-level semantics, neural activity. Almost all fMRI data analysis methods are based on this property of the BOLD signal (GLM, ICA, PCA, etc.).

The General Linear Model (GLM) based approaches use the correlation between stimulus and the BOLD signal as a low-level feature representing the semantics of the data. However, it appears to us that this static snapshot of the brain activity contains no temporal information about the course of the neural activity. Thus, it is good for identifying functional specialization, but not functional integration.

It appears to us that the correlations of the BOLD signal with a delayed stimulus time series (lagged correlations) may reveal the functional integration in the brain. We argue that the functional integration resembles a graph structure whose nodes are specific areas in the brain and whose edges are interaction pathways. In this thesis, we explore using graph structures to represent cognitive processes in a brain solely in terms of their effectiveness for fMRI retrieval.

We propose linking brain areas (nodes) based on correlation measures (leading to what we call the maximum correlation graph). In this method, the areas with the strongest correlations will be connected. An alternative link finding method is based on the Euclidean distance (which we call the minimum distance graph). The brain areas with the shortest Euclidean separations are linked in this approach.
The specific research questions investigated in this thesis are:

1) Can graph-based representations of the high-level semantics of the fMRI data provide for effective data retrieval? This is operationalized by proposing two complete graph-based methods and assessing their performance.

2) Do the Brodmann areas (classification of brain regions based on their appearance in a light microscope, see Chapter 7) provide meaningful structures for defining features in fMRI retrieval? This is operationalized by asking specifically: How effective is the representation of fMRI data by Brodmann vectors, which capture the level of activation in Brodmann areas, for fMRI retrieval?

To assess the question whether complex, graph-based approaches are needed in this setting we ask:

3) Do simpler representations based on anatomical structure of the brain provide for effective data retrieval? This is operationalized by proposing a method based on the level of activation in Brodmann areas and assessing its performance.

4) Do the graph-based approaches provide a better representation of the high-level semantics than methods based on anatomical representation of the brain? This is investigated by comparing their performance across nearly 1000 instances of real fMRI data.
1.6 Organization of This Thesis

We propose answers to all of these questions and assess them using nearly 1000 real instances of fMRI data. Our analysis involves reviewing the complex path from psychological experiment to data retrieval.

Figure 1.1: Flow chart of the path from psychological experiment to data retrieval.

Figure 1.1 gives a diagram that explains how the parts of this thesis come together. At each stage a key method that we used is indicated, with an asterisk to indicate if it is a novel technique. Also, the diagram shows a Chapter or Section where the key method is defined.
This process begins with the collection of fMRI data about the “thinking” brain. The physics of this is reviewed in Chapter 4. The data is processed to determine its relation to stimuli (Chapter 5). Then lagged correlations are computed as explained in Section 7.1.

The centers of mass of contiguous clusters from lagged correlation maps are extracted (Section 7.3). These centers represent nodes of graphs. In this thesis, we propose two new graph-based heuristics for content-based fMRI data retrieval. One heuristic connects centers of nodes based on the minimum Euclidean distance in space (MDG). This is motivated by the idea that signals in the brain propagate with finite speed and, therefore, nodes close in space are somewhat more likely to be related. The other heuristic connects centers of nodes based on their correlation with each other (MCG). This is motivated by the idea that nodes with strong correlation are more likely to be related. We use these approaches to connect nodes of the graphs, as described in Sections 7.3 and 7.4. We use a DAG matching algorithm (Section 7.5) to compute similarity between the graphs. Data retrieval experiments are conducted to find the most “similar” fMRIs for each heuristic (Chapters 10, 11, and 12). Also, ranked retrieval measures are computed for the heuristics (Chapters 10, 11, and 12). The performance of fMRI data retrieval is evaluated and compared across methods in Chapter 13. Finally, the conclusions about effectiveness and validity of the proposed approaches are drawn in Chapter 14.

Note that, in order to make this thesis valuable to future researchers, we have gone into detail on several parts that fall outside the focus of information science, because we have introduced new methods at other stages. For example, we reviewed
fMRI data analysis methods in Chapter 5 and graph matching techniques in Chapter 6.
Chapter 2:

Content-Based Image Retrieval

Two major research communities, Information Retrieval and Computer Vision, have contributed to the Multimedia Information Retrieval since the 1970’s. These two communities study Multimedia IR from two different perspectives, the former text-based and the latter image-based.

Text-based Image Retrieval originated in late 1970’s (Chang et al., 1980). It was popular at that time to annotate images with text and then use text-based information retrieval systems to perform image retrieval. However, as the size of image collection grows the amount of labor required to manually annotate images becomes impractical. Also, there is the problem of subjectivity of human perception. The same image content may be perceived quite differently by different annotators. Imprecision resulting from this may cause mismatches in the retrieval processes (Rui et al., 1997).

The emergence of the large-scale image collections in the early 90’s made the two difficulties (labor and subjectivity) faced by the manual annotation approach more visible. Content-based Image Retrieval was proposed to overcome these difficulties (Faloutsos et al., 1994). “Content-based” means that instead of being manually annotated
with text-based keywords, images would be indexed by their own visual content features, such as color, texture, shape, etc. The query is specified not as a set of keywords but as a query image. The system is asked to retrieve images in a database that are “most similar” to the query image (or exemplar) in terms of the specific features used for indexing.

In this Chapter, we review literature on content-based image retrieval. Also, we will review the low-level features used for indexing and retrieval of images. In particular, we discuss color, shape and texture. Although, approaches discussed here are not applied in this thesis, they can potentially be used in the fMRI data retrieval.

2.1 GEMINI

The problem is to design fast searching algorithms that will search a database of objects and locate objects that “match” a query object, exactly or approximately. Of course, the similarity of, or the distance between, two objects has to be quantified. Generally, we rely on a domain expert to supply such a distance function $D(I_1,I_2)$ (Baeza-Yates and Ribeiro-Neto, 1999). For example, suppose that our database contains fingerprints. The domain expert, in this case a fingerprint analyst, will specify a distance function so that the fingerprints are considered similar when they match as judged by an expert. Greater values of the distance function $D(I_1,I_2)$ correspond to greater dissimilarity between images.

The case is well illustrated by an example in Baeza-Yates and Ribeiro-Neto (1999). Suppose that our multimedia database consists of graphs of time series, such as yearly stock price movements, with one price per day. These can be thought of as images.
Similarity of time series can be defined in many ways depending on the intended use. For example, one can define two time series to be similar if they follow the same trend although the scale might be different. That is, two series are considered similar if they go up and down simultaneously. But, one might define time series to be similar if one follows the other with a one day delay. For example, a stock broker might use this knowledge to predict the prices of similar stocks for a gain. Sometimes, we might be interested just in the average values of stocks over a period of time. Then the similarity could be defined as the difference between average prices over a time interval. If the goal is to find the best price match, the distance function between two series S and Q can be defined as the Euclidean distance

\[ D(S,Q) \equiv \left( \sum_{i=1}^{n} (S[i] - Q[i])^2 \right)^{1/2} \quad (2.1) \]

where \( S[i] \) stands for the value of stock S on the \( i \)-th day. Then a typical query could be something like “give me stocks that are closest to the stock of Microsoft”, for example. Clearly, if our database consists of N objects it will be necessary to compute N values of the distance function, one for each object in the database. This could be very time consuming. The GEMINI (GEneric Multimedia object IndexIng) approach is supposed to overcome this problem (Baeza-Yates and Ribeiro-Neto, 1999). It is based on two steps, each of which tries to avoid the disadvantages of sequential scanning:

a) a “quick-and-dirty” test to discard quickly the vast majority of non-qualifying objects (false alarms are allowed);

b) the use of spatial access methods to achieve faster than sequential searching.
For instance, to apply a “quick-and-dirty” test to the database of time series, we might characterize the sequence with a single number, which will help us to discard many non-qualifying sequences. The average stock price over the year is an example of such a number. Numbers that contain some information about multimedia objects are referred to as “features”. Using a good feature, it is possible to apply a quick test and discard many objects with just a single numerical comparison for each object. Using one feature is good, using several features might be even better because it may reduce the number of false alarms at the cost of making the quick-and-dirty test a bit more expensive (Baeza-Yates and Ribeiro-Neto, 1999).

With GEMINI, the first step is to determine the distance measure between two objects. Having decided on the distance measure, the next step is to find some features that can lower-bound/preserve it. Essentially this means that we need features that carry a lot of information about the corresponding objects and have good discriminatory power (Baeza-Yates and Ribeiro-Neto, 1999). The “lower bounding” property means that none of the qualified objects are discarded although false positives are allowed. In the case of image databases, objects are images and features are those extracted from image content, such as color, texture, shape, position, etc. The process of computing features from image content is called “feature extraction”. Using these features, many non-qualifying objects in a database can be quickly discarded. As long as the features are lower-bounding, this process never discards qualifying objects. Features do not depend on the query and can be pre-computed when objects are added to a database, making the quick-and-dirty tests blindingly fast. Computationally expensive distance functions are computed only for the remaining objects, significantly reducing the complexity.
2.2 Feature Extraction

Feature extraction is one of the most important stages in content-based image retrieval. The performance of the retrieval process depends on the qualities of selected features. Features can be classified as general or domain specific. Color, texture, and shape are general while domain-specific may be defined for X-rays, DNA sequences, etc. General features can be used in most applications. Once a feature is selected (e.g., color), there is no single best representation for a given feature, but rather multiple representations characterizing the feature from different perspectives (Rui et al., 1997). In the next sections we will review the use of general features in content-based image retrieval.

2.3 Color

As it was mentioned previously, a feature may have multiple representations. The color histogram is perhaps the most commonly used color feature representation. To compute a \(k\)-color histogram, where \(k\) is the number of colors, we need to distribute all pixels into \(k\) bins. Each bin in the color histogram records the percentage of pixels that are most similar to that color. The number of bins \(k\) is arbitrary and depends on the application. The more bins, the better similarity in terms of color distribution. However, there is a trade-off. Using large values of \(k\) significantly increase the search time since more comparisons are required.
Figure 2.1, following Baeza-Yates and Ribeiro-Neto (1999), shows such a histogram of a fictitious photograph of a sunset: there are many red, pink, orange, and purple pixels, but only a few white and green ones.

![Color histogram of a fictitious sunset photograph.](image)

Figure 2.1: Color histogram of a fictitious sunset photograph.

We can view the $k$-histogram as a vector with $k$ components. One method to measure the distance between two histograms $X$ and $Y$ is given by

$$D^2(X, Y) = (\bar{x} - \bar{y})' A (\bar{x} - \bar{y}) = \sum_{i}^{k} \sum_{j}^{k} a_{ij} (x_i - y_i)(x_j - y_j)$$

(2.2)

where the superscript $t$ indicates matrix transposition, and the color-to-color similarity matrix $A$ has entries $a_{ij}$ which describe the similarity between color $i$ and color $j$ (Baeza-Yates and Ribeiro-Neto, 1999).

However, histograms are fixed-size structures and they cannot achieve a balance between expressiveness and efficiency. Sometimes only a small fraction of the bins contains significant information. For example, picture of a desert landscape mostly contains blue and yellow pixels, making a finely quantized histogram inefficient. On the
other hand, a coarsely quantized histogram would be inadequate for pictures with a multitude of colors, such as a costume party or carnival. Therefore, some type of descriptor is needed that is flexible enough to represent a wide range of pictures.

Rubner et al. (2000) proposed variable-size descriptions of distributions called *signatures*. A *signature* is a set of the main clusters of a histogram, each represented by a single point (the cluster center) together with a weight that represents the size of that cluster. Note that simple images have short signatures and complex images have long signatures. Specifically, the authors proposed to use an Earth Mover’s Distance (EMD) to measure the distance between two distributions. The EMD between two distributions is the least amount of work needed to transform one distribution into the other. For example, given two distributions, one can be viewed as a mass of earth properly spread in space, the other as a collection of holes. The EMD measures the least amount of work needed to fill the holes with the earth. The unit of work is defined as transporting a unit of earth by a unit of ground distance. The EMD is more robust than other histogram matching techniques, e.g., Equation (2.2), because it can operate on variable-size representations of distributions, which avoid quantization and other binning problems.

### 2.4 Texture

Texture refers to a visual pattern that has properties of homogeneity that do not result from the presence of only a single color or intensity (Smith & Chang, 1996). It is a property of all real surfaces, such as buildings, grass, fabric, etc.
Haralick et al. (1973) proposed a co-occurrence matrix representation of texture feature. The textual feature was based on statistics computed from the relative frequency distribution which described how often one gray tone appeared in a specified spatial relationship to another gray tone on the image. Certainly, this can easily be extended to color images. For example, Figure 2.2 (a) shows a 3×3 image with four colors. The co-occurrence matrix for this image is shown in Figure 2.2 (b). Element \((i, j)\) of this matrix contains the total number of pairs such that the pixels are adjacent and the first pixel of the pair has color \(i\) and the second pixel has color \(j\). Note that each image pixel can have up to eight adjacent neighbors, two horizontal, two vertical and four diagonal. Later on, many other researchers followed in their footsteps and proposed enhanced versions of this approach (Gotlieb et al., 1990).

Tamura et al. (1978) were motivated by the psychological studies in human visual perception of texture and proposed some new texture feature representations. Based on
what appeared to be important in the psychological studies, they identified six visual texture properties: coarseness, contrast, directionality, line likeness, regularity and roughness. Notice that all six texture representations that were proposed are visually meaningful. This was a strong point for their adoption in image retrieval.

2.5 Shape

The ability to retrieve images by shape is one of the basic requirements that is proposed for retrieval (Eakins and Graham, 1999). Shape seems to be a well defined concept, unlike texture. For example, Biederman (1987) found empirical evidence that humans primarily recognize objects by their shape.

Rui, She and Huang (1996) proposed that a useful shape feature representation should satisfy the following four conditions:

a) Robustness to transformation – the representation must be invariant to translation, rotation, and scaling of shapes, and to the starting point used in defining the boundary sequence.

b) Robustness to noise – the representation must be robust to spatial discretization noise.

c) Feature extraction efficiency – feature vectors should be efficiently computable.

d) Feature matching efficiency – the distance metric must require a very small computational cost.
There are two categories of shape feature representations: boundary-based and region based. The boundary based representations use only outer boundary of the shape while region based representations use entire region of the shape.

The Fourier descriptor is a representative of the boundary-based algorithms. The examples of early work can be found in Zahn et al. (1972) and Persoon et al. (1977). A point moving along the boundary can be specified by some function of one parameter, the length of the path $S$ from the origin $O$. We can normalize this parameter so that its sum over the entire curve is equal to $2\pi$ and call this new parameter $t = 2\pi s / L$, where $L$ is the length of the entire curve. Suppose that we define a function $\Phi(t)$ that gives the angular variation between the tangent lines at the origin $O$ and position $t$ as illustrated in Figure 2.3. This function is real, continuous and periodic with period $2\pi$. Thus, it can be transformed into a Fourier series:

$$\Phi(t) = \sum_{k=0}^{\infty} a_k \exp(-jkt) \quad (2.3)$$
The set of coefficients $a_i$ is called the Fourier Descriptors. Fourier Descriptors are invariant to translation, rotation and scaling. Thus, they are good candidates for shape representation.

The Fourier Descriptors are in the class of global descriptors that capture entire shape of the object. Sclaroff and Pentland (1995) introduced a more powerful type of global descriptor called modal analysis. In modal analysis, shapes are represented by canonical deformations from some prototype object. For example, imagine that the collection of sample points in the image is attached by springs to an elastic body. Under the load exerted by these springs, the elastic body will deform to match the shape outlined by the points. In this case, the object’s shape can be described in terms of the eigenvectors of the prototype object’s stiffness matrix. However, the major limitation of this method is that the procedure of attaching virtual springs between the sample points and the surface of the elastic body implicitly imposes a standard parameterization on the data. The modal analysis avoids this problem by using a Gaussian interpolation function.

Given a collection of $m$ sample points $x_i$ from an image, the Gaussian basis function is

$$g_i(x) = e^{-\frac{(x-x_i)^2}{2\sigma^2}}$$  \hspace{1cm} (2.4)$$

where $x_i$ is the function’s center and $\sigma$ is its standard deviation. For each image, the method starts with the collection of sample points and uses them as nodes in building a finite element model of the shape. It can be visualized as constructing a model of the shape by covering each sample point with a Gaussian blob of rubbery material. Given the segmentation information, one can fill in interior areas and trim away material that extends outside of the shape.
The eigenmodes (eigenvectors) $\phi_i$ of the finite element model give the description of the shape and its deformations. They are also called *mode shape vectors* because they describe how each mode deforms the shape by displacing original locations

$$X_{\text{Deformed}} = X + a\phi_i \quad (2.5)$$

where $a$ is a scalar. The first three eigenmodes are the rigid body modes of translation and rotation. The rest are non-rigid modes.

![Diagram](image)

*Figure 2.4: Patches generated by a set of boundary fragments for a cone sketch. Arrows indicate locations of the fragment endpoints. Fragments are shown in canonical positions in the center of the template squares.*

The main weakness of global descriptors is that they require perfect segmentation to be useful, and this is impossible. Since then, powerful new techniques have been developed that capture local shape (Nelson and Selinger, 1998; Shokoufandeh et al.,
The cubist movement in art, pioneered by Picasso and Braque, inspired Nelson and Selinger (1998) to develop a new shape representation successfully used in object recognition. They distinguished two essential elements in the cubism movement, fragmentary but distinctive parts that serve to key the concepts and other non-distinctive features that serve to verify an overall expression.

Based on this idea, they represented the visual appearance of an object as a loosely structured combination of a number of local context regions keyed by distinctive parts. A local context region is an image patch surrounding the key part and containing representations of other key features intersecting the patch. Each boundary fragment (curve) found in an image was considered as a key feature. Figure 2.4 shows the set of patches generated by boundary fragments for a cone. The arrows show the locations of the fragment endpoints or diameter. Note that this representation is redundant in that local contexts generated from large curves may contain all or most of the curves in an object.

Computation of the shape representation consists of several steps. For every image, the boundary extraction algorithm selects 20 or so boundaries as keys. From these keys patches are generated and stored in the database. Each context patch contains information about the size, location and orientation of the object that produced it relative to the key curve.

### 2.5.1 Region Based Shape

The “moment invariants” method is a representative of a region-based shape descriptor. Using nonlinear combinations of geometric moments, Hu (1962) derived a set of seven
moments that have desirable properties of being invariant under image translation, rotation and scaling. These region based moments can be used as the shape features. His work spawned many improved versions. For example, Teague (1980) introduced Zernike moments that allow independent moment invariants to be constructed to an arbitrarily high order.

The most successful region-based shape descriptor is a recent technique called “shape context” (Belongie et al., 2002), which is gaining popularity. Shapes are represented by a set of points sampled from the shape contours. These points are not required to be landmarks or curvature extrema. Larger sample sizes give better approximations to the underlying shape. Each sample point is associated with a shape descriptor, called the “shape context”, which describes the distribution of the remaining points relative to it. Consider a set of vectors originating from a point to all other sample points. The set of vectors expresses the configuration of the entire shape relative to the reference point. The full set of vectors as a shape descriptor is much too detailed. The authors proposed the shape context as a more robust and compact descriptor. Let $p_i$ be a point on the shape. The shape context of $p_i$ is a coarse histogram $h_i$ of the relative coordinates of the remaining $n-1$ sample points,

$$h_i(k) = \# \{ q \neq p_i : (q - p_i) \in \text{bin}(k) \} \quad (2.6)$$

where $(q - p_i)$ is the distance between points $q$ and $p_i$.

Then object recognition is done in two stages. First, correspondences between points on the two shapes are computed. Corresponding points on the two shapes will have similar shape contexts. Therefore, the correspondence problem can be cast as an optimal assignment problem. In the second stage, the correspondences are used to estimate the
transformation that best aligns the two shapes. The dissimilarity measure between the two shapes is equal to the sum of matching errors between corresponding points, together with a term measuring the magnitude of the transformation.

2.5.2 Point Based Shape Descriptors

There is an increasingly popular class of shape descriptors that is neither region nor boundary based, but rather point-based (Lowe, 2004; Carneiro and Jepson, 2004; Schmid and Mohr, 1997). Lowe (2004) developed the Scale Invariant Feature Transform (SIFT) approach that uses “keypoints” for image matching. First, it detects locations that are invariant to scale change of the image by searching for stable features, also called keypoints, across all possible scales, using a continuous function of scale known as scale space. The scale space of an image is defined as a function \( L(x, y, \sigma) \), produced by the convolution of the input image \( I(x, y) \) with a variable-scale Gaussian

\[
    L(x, y, \sigma) = G(x, y, \sigma) \ast I(x, y) \quad (2.7)
\]

where \( \ast \) is the convolution operation in \( x \) and \( y \), and

\[
    G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2 + y^2) / 2\sigma^2} \quad (2.8)
\]

Note that \( x, y \) and \( \sigma \) are axes of the scale space.

The stable keypoint locations are detected by using scale-space extrema (maxima and minima) in the difference of Gaussian function convolved with the image, \( D(x, y, \sigma) \). It is computed from the difference of two nearby scales separated by a constant multiplicative factor \( k \),

\[
    D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \ast I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.9)
\]
Note that $D$ can be computed by simple image subtraction of smoothed images $L$.

Figure 2.5 sketches the construction of $D(x, y, \sigma)$. The initial image is incrementally convolved with Gaussians to produce images separated by a constant factor $k$ in scale space (shown stacked in the left column). Adjacent image scales are subtracted to produce the difference of Gaussian images shown stacked in the right column.

![Diagram showing the construction of $D(x, y, \sigma)$](image)

*Figure 2.5: The initial image is repeatedly convolved with Gaussians to produce the set of scale space images (shown on the left). Adjacent Gaussian images are subtracted to produce the difference of Gaussian images (shown on the right).*

To detect the local maxima (minima) of $D(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and its nine neighbors in the scale above and below. It is selected as a keypoint only if it is larger (smaller) than all of these neighbors.
Schmid and Mohr (1997) used corners and vertices in an image as “interest points” to extract local descriptors. The set of differential invariants were used as local descriptors. For instance, a database consists of a set \( \{ M_k \} \) of models. Each model \( M_k \) is defined by the vectors of invariants \( \{ V_j \} \) calculated at the interest points \( j \) of the model image, where

\[
V[0..3] = \begin{bmatrix}
L \\
L_i L_i \\
L_i L_j L_j \\
L_{ii}
\end{bmatrix}
\]  
(2.10)

and \( L_i \) are the elements of the “local jet”. Note that the vector is given in tensorial notation – so called Einstein summation convention. The local jet of order \( N \) at a point \( X(x, y) \) is defined by

\[
J^N[I](X, \sigma) = \{ L_{i_1 \ldots i_n}(X, \sigma) \} | (X, \sigma) \in I \times R^+ ; n = 0, \ldots, N \}
\]  
(2.11)

where \( L_{i_1 \ldots i_n}(X, \sigma) \) is the convolution of image \( I \) with the Gaussian derivatives \( G_{i_1 \ldots i_n}(X, \sigma) \) and \( i_k \in \{ x, y \} \). Note that the first component of \( V \) represents the Gaussian filtered image function \( L(x, y) \), the second component the square of the gradient magnitude and the fourth the Laplacian. Indeed, for a two-dimensional image

\[
L_i L_i = \sum_{i=x,y} L_i L_i = L_x^2 + L_y^2 \\
L_{ij} = \sum_{i=x,y} L_{ii} = L_{xx} + L_{yy}
\]  
(2.12)

where \( L_x \) denotes partial derivative \( \frac{\partial L}{\partial x} \).
During a storage process, each vector $V_j$ is added to the database with a link to the model $m$ for which it was computed. Basically, the simplest database is a table of pairs $(V_j, m)$.

Carneiro and Jepson (2004) also used interest points detected in an image to extract local descriptors. However, they used a different set of local descriptors. For each local image descriptor they formed a feature vector $f_i = f(x) = [m, \theta, \sigma, \nu]$, where $x$ is the interest point location, $m$ is the model identification from which this feature was extracted, $\theta$ is the main orientation, $\sigma$ is the scale, and the vector $\nu$ contains the feature values. Here the vector $\nu = \rho e^{i\Phi}$ is the vector of amplitudes $\rho$ and phases $\Phi$ of bandpass filter responses.

![Figure 2.6: Shock types, courtesy of Siddiqi et al. (1999).](image-url)
2.5.3 **Skeleton Based Shape Descriptors**

There is also an exciting work on using “shock graphs” for shape matching (Siddiqi et al., 1999). Let X be the open interior of a simple closed curve. Suppose that the boundary of the curve is set on fire and the evolution of the fire front is observed. The medial axis of the curve, denoted M(X), is defined as the set of points reached simultaneously by two or more fire fronts. Curves can be classified into “shocks” (singularities) based on the variation of the radius function along the medial axis. Figure 2.6 illustrates the four types of shocks.

At a first-order shock the radius function varies monotonically, the same as for a protrusion. At a second-order shock the radius function achieves a strict local minimum so that the medial axis is disconnected if the shock is removed. It can be visualized as a neck. At a third-order shock the radius function is constant along the medial axis, as for a bend with parallel sides. At a fourth-order shock the radius function achieves a strict local maximum, e.g. when the evolving curve annihilates into a single point or a seed. Shocks of the same type are colored in the same color giving the encoding or coloring of the shocks.

Finding the medial axes of a shape gives its skeleton. The skeleton together with the shock encoding can be used to produce a shape’s descriptor called a “shock graph”. Figure 2.7 illustrates the process of constructing a shock graph for a simple shape.

Each shock type gives birth to a node in a shock graph. Nodes are colored according to their shock types, which are shown in the right column (e.g. 4th order shock is red). A root node is labeled with S and leaf nodes are labeled with L.
Shokoufandeh et al. (1999) successfully applied shock graphs in the domain of 2D silhouette matching. Sundar et al. (2003) extended this work by successfully applying shock graphs to 3D shape matching.

Figure 2.7: Shock graph of a simple hammer shape. Color to shock type mapping is shown in the right column (1st order shock is yellow, 4-th order is red).

### 2.5.4 Shape and Medical Images

Shape descriptors were successfully applied for searching in medical images as well. For example, Korn et al. (1996) proposed using the “pattern spectrum” of a shape for fast searching in medical image databases. The pattern spectrum is a shape-size descriptor developed by Maragos (1989). In the simplest terms, the pattern spectrum is the size...
distribution of the area of a shape painted by a brush (primitive element) when the size of its foot is varied. It summarizes whether an image has much fine detail, or little detail. They applied the pattern spectrum based shape matching algorithm to two dimensional images of synthetic tumor shapes.

Petrakis and Faloutsos (1997) proposed an Attributed Relational Graph (ARG) based algorithm for searching in medical image databases. They assumed that all images contain a number of common shapes that can be segmented into “labeled” objects. In a medical database, common objects could correspond to anatomical structures, such as heart, lungs, kidney, backbone, etc. There can be “unlabeled” objects in images as well. The ARG captures the information about properties of the objects and relationships between them. Each image in a database has a corresponding ARG. Thus, the matching is done between the ARG of a query image and the ARGs stored in the database.
Chapter 3:  
Image Retrieval Systems

Since the early 90’s many content-based image retrieval systems, both commercial and research, have been built. Most of them support one or more of the following options (Chang et al., 1998):

a) interactive browsing  
b) query by example  
c) search by features and sketches  
d) text based query  
e) subject navigation with customized image categories

In this Chapter, we discuss the existing content-based image retrieval systems. The URLs to the online demos are given for those that we were able to find and verify.

3.1 QBIC

Query By Image Content (QBIC) is the first commercial content-based image retrieval system (Faloutsos et al., 1994). Given that semantic features are outside the capability of
current machine vision technology, the QBIC developers selected the properties of color, texture, and shape for feature extraction, because they have broad, intuitive applicability. Users can ask queries on any of the above properties, or a Boolean combination of the above. All queries are “approximate” queries.

For instance, if a user is interested in retrieving a beach scene, he or she can form the query as one of the color distribution (e.g. 35% white, 65% blue area coverage) and textures, such as presence of sand textures (Faloutsos et al., 1994). In addition to this type of queries, QBIC also supports two other ways of specifying a query (Faloutsos et al., 1994):

a) “query by example” – the user can choose one of the displayed images and ask for images similar to the selected one; this is closely related to the concept of relevance feedback.

b) “direct query” – the user specifies the desired color/shape/texture directly, e.g., by picking colors from a palette on the screen (termed multi-color picker), or drawing a sketch with the mouse.

The two modes of operation may be used interchangeably within the same session.

The color feature used in QBIC is the $k$-element color histogram, where $k=256$ or $k=64$ is the number of colors. The distance function (dissimilarity measure) was computed according to Equation 2.2. For shape features, they used as features the area, circularity, eccentricity, major axis orientation, etc., for a total of 20 features (Faloutsos et al., 1994). The texture features are modifications of the coarseness, contrast, and directionality features proposed in Tamura et al. (1978). We discussed the shape and texture features in Chapter 2. QBIC uses “lossy” transformations that introduce false hits,
such as using the average of red, green and blue color components to do a “quick-and-dirty” test. This doesn’t create false negatives and allows the use of fast multidimensional indexing methods, by paying a small post-processing cost to eliminate the false hits. An online QBIC demo is available at http://wwwqbic.almaden.ibm.com (checked August 24, 2005).

3.2 Photobook

Photobook is a set of interactive tools for browsing and searching images and image sequences (Pentland et al., 1996). There are three types of Photobook: one that allows search based on appearance (Appearance Photobook), one that can be searched by 2D shape (Shape Photobook), and one that allows search based on texture (Texture Photobook). Users can also search Photobook using text annotation information in a database, and directly by image content. Users most frequently employ it by selecting one of the currently displayed images, and asking it to sort the entire set of images in terms of their similarity to the selected image. Each of the three photobooks can be efficiently employed in many diverse areas. Pentland et al. (1996) suggest several promising applications. For instance, applications for Appearance Photobook as applied to face databases include customs, security and criminal investigation. Applications of the Shape Photobook include searching catalogs of consumer goods, such as hand tools. The Texture Photobook could be used to search texture patches in the design and decorating industries. An online demo is available at http://vismod.www.media.mit.edu/vismod/demos/photobook/index.html (checked August 24, 2005).
3.3 VisualSEEk

VisualSEEk is a prototype system for searching by visual features (Smith and Chang, 1996a). The VisualSEEk system is used by diagramming spatial arrangements of color regions. The system then searches for images containing the most similar arrangements of similar regions. For instance, this enables a user to submit a sunset query as “red-orange region color on top and blue or green region at the bottom” as a sketch. VisualSEEk decomposes each image into regions, which have feature properties, such as color, and spatial properties such as size, location and relationships to other regions. In this way, images are compared by comparing their regions. The decomposition relies on the representation of color regions by color sets. A key issue here is the choice of color space and how it is partitioned. For example, each image pixel can be represented as a 3-D vector \( V(r, g, b) \) in the RGB (red, green, blue) color space. Thus, RGB color space groups vectors \( V \) into clusters (Smith and Chang, 1996a).

3.4 NeTra

NeTra is a prototype image retrieval system that uses color, texture, shape and spatial location information in segmented image regions to search an image database (Ma and Manjunath, 1999). The novel aspect of NeTra is that it uses a robust automated image segmentation algorithm that allows object or region-based search. Image segmentation improves the quality of image retrieval when images contain multiple complex objects. Images are segmented into homogeneous regions as they are added to the database, and
image attributes representing each of these regions are computed. In addition to image segmentation, other important components of the system include an efficient color representation, and indexing of color, texture and shape features for search and retrieval. An online demo of NeTra is available at http://vision.ece.ucsb.edu/netra/ (checked August 24, 2005).

3.5 MARS

The Multimedia Analysis and Retrieval System (MARS) is a prototype image retrieval system developed at University of Illinois at Urbana-Champaign (Mehrotra et al., 1997). An image is represented as a collection of automatically extracted low-level image features (e.g., color, texture, shape and layout) and human-generated descriptions of the image. Color features are represented in the HSV (hue, saturation, value) space which is chosen for its de-correlated and uniform coordinates. It is represented as an 8x8 2D histogram over the HS coordinates and the V coordinate is dropped since it is easily affected by lighting conditions. The texture feature (coarseness, contrast, and directionality) are a modified version of the texture features developed in Tamura et al. (1978). The “coarseness” measures granularity of the texture (fine vs. coarse) and is represented by a coarseness histogram. The contrast represents the distribution of luminance in the image. A single number represents this component.

The shape features of an object in an image are represented by its boundary. The boundary of an object is represented using a Modified Fourier Descriptor (Rui et al., 1996) and the Euclidean distance in Fourier space is used to measure similarity based on
shape of objects. Finally, MARS supports representation of color and texture layouts in an image. The image is split into 5x5 subimages to extract the layout features. Color and texture features are extracted from each subimage. To compute similarity based on layout, similarity is computed with respect to each subimage and then a weighted sum is used to compute the layout distance (Mehrotra et al., 1997). An online demo is available at http://www-db.ics.uci.edu/pages/demos/index.shtml (checked August 24, 2005).

3.6 PicHunter

PicHunter is an image retrieval system based on a simple Bayesian probability estimation framework (Cox et al., 2000). PicHunter uses simple Bayesian probability rules to respond to the user’s relevance feedback. A typical search session consists of a query followed by repeated relevance feedback. The system displays a set of images and the user takes an action, observed by the system. The probability distribution of image relevance is recomputed at each iteration of the session, based on the history of images displayed so far and user actions. That is each image $T_i$ in the database has a certain probability of being a target image $T$ and this probability is re-evaluated at each iteration according to a simple Bayesian rule

$$P(T = T_i \mid H_t) = \frac{P(H_t \mid T = T_i)P(T = T_i)}{P(H_t)} \quad (3.1)$$

where $P(T = T_i \mid H_t)$ is a “posterior” probability that image $T_i$ is the target, given the observed history $H_t$, $P(H_t \mid T = T_i)$ is the likelihood of observing history $H_t$ given that the target is $T_i$, and $P(T = T_i)$ is “a priori” probability that $T_i$ is a target image. Note that
the canonical choice assigns an \textit{a priori} probability equal to $1/n$ to all images, where $n$ is the number of images in the database.

The history $H_i$ consists of sets of images displayed $D_i$ and user actions $A_i$, so that $H_i = \{D_1, A_1, ..., D_i, A_i\}$. It is assumed that the user’s judgment of image similarity can be captured by a small number of image features. Psychophysical experiments directed the selection of the image features and similarity distance function. The original version of PicHunter used 18 global image features, such as the color histogram, mean color saturation of an image, the median intensity, image width, image height, global contrast, etc. The current version incorporates spatial distribution of colors, in addition to color histogram. This knowledge about the user’s judgment helps to direct the probability estimates to the target image over repeated relevance feedback.

### 3.7 Blobworld

Blobworld is an image retrieval framework developed at the University of California at Berkeley (Carson et al., 2002). The framework is based on the segmentation of images into a small set of regions that are coherent in color and texture, and querying using the properties of these regions. The image regions roughly correspond to objects or parts of objects present in the image. Thus, the querying is at the level of objects which is hoped to be semantically more meaningful than using global image properties.

After the segmentation is complete, the color and texture properties of each segmented region are computed. The user can access the regions directly to see the segmentation of the submitted query image and specify which aspects of the image are
important to the query. When the query results are returned, the user can also see the internal representation of each retrieved image.

A typical image retrieval session with the Blobworld system starts with a user submitting a query to the segmentation/feature extraction algorithm to see its Blobworld representation. The user selects the blobs (regions) that need to be matched and specifies the relative importance of the blob features. The system then ranks retrieved images according to score and returns the best matches, indicating for each image which set of blobs provided the highest score. This information helps the user to refine the query. After examining the retrieved results, the user can adjust the weighting of the blob features or/and specify new blobs to match and resubmit the refined query. The online demo is available at http://elib.cs.berkeley.edu/photos/blobworld (checked August 24, 2005).

### 3.8 Search Engine for 3D Models

The Search Engine for 3D Models is a system developed at Princeton University for indexing and retrieval of 3D models available on the Web (Funkhouser et al, 2003). The system searches its database of models based on shape similarity. Feature extraction for shapes in 3D is far more complicated than for 2D images. Indeed, the 1D boundary contours for 2D shapes have a natural arc parameterization, while 3D surfaces do not. Therefore, common shape descriptors for 2D images can not be extended to 3D surfaces.

The system uses a novel shape-descriptor based on spherical harmonics. The idea is to decompose a 3D model into a collection of functions defined on a set of concentric
spheres. The first step in the decomposition is to rasterize images into a $2R \times 2R \times 2R$ voxel grid, assigning a voxel a value of 1 if it is within one voxel width of a polygonal surface. Otherwise, the voxel is assigned a value of 0. The value of $R$ is around 32, which provides adequate granularity. Note that each voxel of a $64 \times 64 \times 64$ grid can be stored in one bit, with value of 0 or 1. The model is moved so that the center of mass is at the point $(R, R, R)$.

A binary real-valued function can be defined on the set of voxel grid points and expressed in spherical coordinates:

$$f_r(\theta, \phi) = f(r, \theta, \phi) = \text{Voxel}(r \sin \theta \cos \phi + R, r \cos \theta + R, r \sin \theta \sin \phi + R)$$ (3.2)

where $r \in [0, R]$, $\theta \in [0, \pi]$, and $\phi \in [0, 2\pi]$. Restricting to the different radii gives a collection of spherical functions $\{f_1, f_2, \ldots, f_R\}$. Each of these functions can be expressed as a sum of its different frequencies using spherical harmonics:

$$f_r(\theta, \phi) = \sum_m f_r^m(\theta, \phi) = \sum_{n=-m}^{m} a_{mn} \sqrt{\frac{(2m+1)(m-n)!}{4\pi(m+n)!}} P_{mn}(\cos \theta) e^{in\phi}$$ (3.3)

The collection of function’s frequencies is defined as its signature. Combining these different signatures over different radii gives a two-dimensional rotationally invariant \textit{spherical harmonics descriptor} for the 3D model. The system compares two 3D models by computing the Euclidean distance between their spherical harmonics descriptors.

The system supports queries based on 3D sketches, 2D sketches, 3D models, and/or text keywords. That is, any type of query can be supplemented with text keywords. 3D models can be loaded from a file or selected from the returned results by choosing the “find similar models” button. With the 2D sketch option, user can submit
three sketches of the query object: front view, side view and top view. The online demo is available at http://shape.cs.princeton.edu (checked August 24, 2005).

3.9 Marie

The Marie project has concentrated on the retrieval of captioned images of the kind found in picture libraries and on the Internet (Rowe et al., 1999). The idea behind the system is that images are easier to understand with the descriptive text in addition to the content-based feature extraction.

The system starts by collecting the images and captions. Images on the Internet can easily be spotted in HTML markup language. However, the symbolic graphics, of no interest to the index are also stored the same way as GIF and JPEG photographs. A useful system must be able to distinguish them. It has been shown that seven quickly-found image features are sufficient enough to distinguish photos with 70% recall, ratio of photos found to all photos, and 70% precision, ratio of actual photos in those found, in a random test of Web pages (Rowe and Frew, 1998). These features are size, squareness, number of distinct colors, fraction of pure colors (white, black, pure gray, red, green, and blue), color variation between neighbor pixels, variety of colors, and use of common non-photograph words, such as “button” or “logo”, in the name of the image file. Words commonly associated with photographs, for example, are names, locations, etc. Based on these seven features, a decision is made whether the image is a photograph or a symbolic graphics.
Then the system looks for captions around each identified photograph. Another seven input classifier is applied to decide if a text is a caption. Its input parameters are: distance in lines from the caption candidate to the photograph, number of other candidates at the same distance, strength of emphasis (e.g. italics), appropriateness of candidate length, use of common words of caption, number of identical words between candidate and either image file name or its non-graphic substitute, and fraction of the words having at least one physical-object sense.

Found captions are parsed and indexed. The system continues by doing basic image processing on found photographs. The image is split into small irregular regions based on color similarity, and regions are iteratively merged using color and texture similarity until 50-100 regions remain. Then 17 key properties, developed from survey of a variety of captioned images, are calculated. These properties include color, texture, density, horizontality and verticality, boundary shape, and boundary strength. Eventually, the results of a caption and image analysis are integrated based on the “linguistic focus” of the caption and the “visual focus” of the image. The linguistic focus represents the most important words derived in the linguistic and statistical analysis of a caption. Visual foci are region sets with five characteristics: (1) they are large; (2) their center of gravity is near the image center; (3) they do not touch the edges of the image; (4) their boundary has good contrast; and (5) they are maximally different from non-focus regions of the image.

The Marie system supports querying the indexed information by parsed English queries or by key phrases. The strength of the system is in its integrated approach to
image retrieval using clues from linguistics, image analysis, presentation layout and mathematical analysis.

![Hierarchical tree for text and regions (Barnard et al., 2003).](image)

**Figure 3.1: Hierarchical tree for text and regions (Barnard et al., 2003).**

### 3.10 Hybrid Systems That Use Text and Image

All the image retrieval systems that we looked at so far do searching based on low level image properties, such as color and/or texture distribution. However, users are interested in the semantics of the image content. For example, a user searching for a zebra image will not be satisfied by images with plausible histograms (Barnard et al., 2005).

In recent years, there was a new wave of multimedia retrieval systems that use both text and image data to try and do automatic annotation (image labeling). Predicting
words associated with images is called automatic annotation, while predicting words associated with image regions is called region naming (Barnard et al., 2003). The developers of these systems assume that solving the object recognition task will allow semantically meaningful image retrieval. Indeed, such systems would automatically annotate images and allow users to access images via keywords.

### 3.10.1 Image Labeling

Barnard et al. (2003) developed several models for image labeling. In one of the models, images and co-occurring text are generated by nodes arranged in a hierarchical tree structure, as in Figure 3.1. Every image is in some cluster and each cluster is represented by a path from a leaf to the root. Nodes close to the root are shared by many clusters, and nodes closer to the leaf are shared by fewer clusters. This means that nodes close to the root tend to generate items (words and regions) shared by a large number of clusters. Nodes closer to the leaves generate items more specific to small number of clusters.

Since each image is in a given cluster, it is generated by the nodes on that path. Therefore, a document is modeled by a sum over the clusters, weighted by the probability that the document is in the cluster. The Equation 3.4 describes the process for generating the set of observations $D$ associated with a document $d$:

$$p(D|d) = \sum_c p(c) \prod_{w \in W} \left[ \sum_l p(w|l,c) p(l|d) \right] \prod_{b \in B} \left[ \sum_l p(b|l,c) p(l|d) \right]$$  \hspace{1cm} (3.4)

where $c$ are clusters, $w$ are the words in document $d$, $b$ are the image regions in document $d$, and $l$ are levels. $D$ is the set of observations for the document, such that $D = W \cup B$,
where $W$ is the set of words for the document and $B$ is the set of regions (blobs) for the document. In other words, an observation means that a word $W$ was observed with a blob $B$ in a document. The exponents are introduced to normalize for differing numbers of words and blobs in each image. $N_{w,d}$ is the number of words in document $d$ and $N_w$ is the maximum number of words in any document.

The image labeling is done by assuming that the new document is presented with a set of observed blobs $B$. The model computes $p(w | B) \propto p(w, B)$ for each word $w$ in its vocabulary:

$$p(w | B) \propto \sum p(c) p(w | c) p(B | c)$$  \hspace{1cm} (3.5)

Suppose that word “tiger” frequently co-occurs with orange and black regions. Then the probability estimation (Equation 3.5) means that new images with orange and black regions will have higher estimated probability of containing a tiger.

### 3.10.2 Region Naming

A different image retrieval system is being developed by Wachsmuth et al. (2003) that can capture part/whole hierarchy among the words. For instance, assume that consistently leg and top regions are referred together as a table. Then one could capture the image part/whole structure as word relations in a lexicon, instead of treating leg and top as two labels for the same object.

The system segments images into regions. The result is a region adjacency graph, where nodes represent homogenous regions and edges capture region adjacency. Then the
The qualitative shape of each region is encoded by its shock graph (see Section 2.4). The set of all regions from training images are clustered according to a distance function that measures the similarity of two shock graphs. Objects with similar shapes are clustered together. This means that it is likely that objects belonging to the same class are clustered together, e.g. hammers.

The system uses a probabilistic translation model to find the correspondence between words and image regions. Probabilistic translation models produce the translation string $e$ given the source string $f$, e.g. $e$ in English and $f$ in French, by maximizing the probability $P(e|f)$. The system predicts words from image regions taking $f$ as the words in the text and $e$ as the regions in the image.

Segmentation can cause an object to be divided into a collection of regions. Multiple regions can be handled by adding to the sequence of regions in each image the regions resulting from all possible merges of image regions. The probability is given by:

$$ P(w|r) = \frac{e}{(L+1)^M} \prod_{j=1}^{M} \sum_{a_j=0}^{L} t(w_j | r_{a_j}) $$

where $r$ is a region, $w$ is a word, $e$ is a constant, $M$ is the number of words in English, $t$ defines a translation table, $a$ is an alignment that maps each region to one of the English words, and $L$ denotes the total number of segmented and merged regions in an image. In practice the number of possible mergers may be intractable. Therefore, the system uses an iterative bootstrapping strategy that filters out hypothetically meaningful merges (merges that increase the quality of the translation).
3.11 Current Research Issues

Humans tend to use high level concepts in everyday life. Similarly, what the user expects from a database is a semantically meaningful answer to a query (Santini and Jain, 1998). However, what current computer vision techniques can automatically extract from images are mostly low-level features. In constrained applications, such as human face and fingerprint recognition, it is possible to link the low-level features to high level concepts, faces or fingerprints (Rui and Chang, 1997). However, in the general setting, there is no direct link from low-level features to high-level concepts.

The semantic gap is not the only problem with current image retrieval systems. Breiteneder and Eidenberger (2000) summarized drawbacks to current retrieval systems:

a) Complicated interfaces – casual users are overwhelmed by the demand for a definitive answer on similarity, selection of features, and, often necessary, provision of weights.

b) Unsatisfactory querying performance – systems use the distance function to calculate the similarity. This process is very slow and response times in the range of minutes may occur for large databases.

c) Low result quality – using only general features for all types of images leads to low quality retrieval results.

In this context our present research does not address these user interface and usability issues. We are concerned to develop technologies that automatically recognize semantically related fMRI images. We shall see relations of the methods proposed here to
several of these earlier approaches.
Chapter 4:
Time-Space Densities and fMRI

In this Chapter, we review the basic physics of the Magnetic Resonance Imaging (MRI). We discuss the use of this technology for studying brain functionality. We also review the “chemistry” of the brain activations. Specifically, we expose the relationship between a neural activity and a change in the Blood Oxygen Level Dependent (BOLD) signal, which is recorded in the fMRI scanner. Finally, we describe the shape of the typical BOLD signal generally accepted in the neuroscience community. In this work, we rely on the correlation between a neural activity and the BOLD signal. In our computations, we assume that the shape of the BOLD signal conforms to the one described here.

4.1 Time-Space Densities

Technically any two-dimensional images, such as a photograph or a painting, can be abstracted as a function \( F \) of two variables that maps coordinates \( X \) and \( Y \) to a complex variable called color, \( F(X, Y) \rightarrow \text{Color} \) (Duda et al., 1973). Quite similarly, three-dimensional (3D) images are functions of three variables: \( F(X, Y, Z) \rightarrow \text{Color} \). In the case of grayscale images, the variable \( \text{Color} \) is defined as a real number and the function
\(F(X, Y, Z)\) defines a scalar field or density field of the variable \textit{Color} at each point \((X, Y, Z)\). If the variable \textit{Color} is a vector with three components \((\text{hue}, \text{saturation}, \text{value})\) then function \(F(X, Y, Z)\) defines a vector field. Although it is quite common to refer to 3D brain images as volumes, we will use the term “density” to emphasize the fact that we are concerned with 3D images that define a density function at each point. One may view time-sequenced densities as functions of four variables, three variables for space and one variable for time. Thus, a time-sequenced density is a function \(F(X, Y, Z, T) \rightarrow \text{Color}\).

A functional Magnetic Resonance Image (fMRI) is an example of a time-space density. fMRI data consist of 3D images of the brain taken every two or three seconds over a period of time, typically several hundred seconds. However, the process by which the supply of fresh blood to regions of the brain becomes a density field is somewhat complex. In the first (physical) step, the amount of oxygenated blood is measured. In the second (statistical) step, this data is related to cognition and perception. In this description, we will follow Oldendorf et al. (1988).

4.2 Basic Physics of fMRI

Any magnet placed in an external magnetic field aligns itself in the external magnetic field so that its north pole points toward the south pole of the external magnetic field. The most familiar example of this phenomenon is a compass needle in the magnetic field of the Earth. If we tap the tip of the needle at rest with our finger, it will start an oscillatory movement around its original orientation until it will finally stop at rest position due to mechanical friction. The frequency of the needle’s oscillation is called its natural
frequency. The natural frequency depends on the characteristics of the needle and is proportional to the strength of the external magnetic field (Oldendorf et al, 1988). For example, if the strength of the external field doubles, the natural frequency doubles as well. This is true for any magnet, not just a compass needle. The location of a magnet, in a magnetic field with known gradient, can be obtained by measuring the magnet’s oscillation frequency since this is determined by the field strength (see Figure 4.1).

![Figure 4.1: Resonance frequency of hydrogen nucleus in magnetic field.](image)

Electrical currents create magnetic fields. In a straight piece of wire carrying electrical current, the magnetic field has the shape of rings surrounding the wire. If we bend the wire into a circular loop the magnetic field will morph into circular rings passing through the loop. The shape of this magnetic field is similar to the shape of the magnetic field of a natural magnet and has north and south poles.
Living tissue has no compass needles. However, it contains billions of small magnets created by the nuclei of certain atoms. Hydrogen atoms have the strongest magnetic nucleus, in ratio to mass, and make up two thirds of the atoms in living tissues. We may consider that the nucleus of the hydrogen atom contains a positively charged proton “spinning around its own center”. This spinning movement of the positive charge produces a magnetic field similar to the field produced by electrical current in a circular loop (in fact, the proton’s magnetism is a quantum mechanical phenomenon, but it can be visualized this way). It is the hydrogen atoms in tissue water and body fat that are imaged in MRI (Oldendorf et al., 1988).

Now, suppose there is a magnetic field throughout the brain. A hydrogen nucleus stimulated with an electromagnetic field, at a frequency equal to the natural frequency at which hydrogen nucleus oscillates in magnetic field, will exhibit a resonance effect. Therefore, this frequency is sometimes called the nucleus’ resonance frequency. Also, it is often called the Larmor frequency, named after the British physicist Joseph Larmor. For example, the Larmor frequency of the hydrogen nucleus in a magnetic field of one Tesla is equal to 42.6 million cycles per second (42.6 MHz). Just as we use a finger tap to stimulate the compass needle, the nucleus of the hydrogen atom can be stimulated by electromagnetic waves generated by the MRI scanner. The hydrogen nucleus absorbs energy from the electromagnetic wave and starts oscillating if the frequency of the wave is equal to the Larmor frequency of the nucleus. During the oscillation process, the nucleus re-emits energy in the form of electromagnetic waves with its resonance frequency until it stops due to the exhaustion of absorbed energy. The absorption of energy by the magnetic atomic nucleus is called nuclear magnetic resonance.
Figure 4.2: Spatial encoding in the gradient field. Direction of the magnetic field is in the Z-axis direction, shown at the bottom. Resonance frequencies for different locations in magnetic field are shown at the top.

The signal from an individual nucleus is too weak to measure. However, the antenna of an MRI scanner can detect the signal from many millions of nuclei. An MRI scanner uses an ordinary radio wave transmitter and receiver to stimulate nuclei and to detect the re-emitted signal for imaging inside the body. Detecting the re-emitted signal is not enough to define the spatial location of tissue. To isolate a slice of tissue, the MRI scanner has to create a magnetic gradient. The continuous magnetic field with the field strength changing (gradient) along the arrow in Figure 4.2 is applied. In the presence of the gradient field, hydrogen atoms in the body become spatially encoded because there is a known mapping between their resonance frequency and their position in the gradient field (Figure 4.2).
Transmitting pulses of radio waves with various frequencies can stimulate different slices of tissue that are perpendicular to the gradient field. For example, exposing the entire body to the pulses of radio signal with frequency of 42.6 MHz will stimulate only hydrogen atoms lying within the plane of tissue where the strength of magnetic field is equal to 1 Tesla, shown as dashed line in Figure 4.2. Hydrogen nuclei in the stimulated slice will re-emit the absorbed energy in form of electromagnetic signal that can be detected by the radio antenna of the MRI scanner. The strength of the emitted radio signal is proportional to the number of hydrogen atoms in the slice. Hydrogen atoms in the adjacent slices are not stimulated and, hence, emit no radio signal. Note that
the radio antenna of the MRI scanner can be used for both sending electromagnetic pulses stimulating hydrogen nuclei and detecting the nuclei’s re-emitted signals.

This is yet not enough for imaging since the signal intensity would simply indicate the average signal from all regions of the slice. After the slice of tissue has been isolated and stimulated, the main gradient field, which is usually in the head-to-toe Z-axis direction, and the radio-frequency excitation are turned off. A second gradient is then created at right angles to the Z-axis in the X-Y plane. Figure 4.3 shows a new magnetic field, in the X-axis direction, applied to a slice of tissue. Hydrogen atoms in the slice that were oscillating with one frequency are now in magnetic fields with differing strengths. Nuclei resonate at new frequencies that depend on their position in the new gradient field. Thus, the stimulated slice of tissue is sliced into parallel strips such that nuclei along any one strip are oscillating at the same frequency. Resonance frequencies for each slice are

Figure 4.4: Single slice of tissue examined many times, each time with a gradient oriented in different direction.
shown at the top in Figure 4.3. Measuring the intensity of the signal at different frequencies the concentration of hydrogen atoms along all lines in the slice can be obtained.

After the emitted radio signal strengths have been measured, the process is repeated again. The main gradient along the Z-axis is restored and the same slice of tissue is excited by the short burst of radio wave. Note that it is necessary to restore this gradient after each measurement so that the same slice is excited by the radio wave. Then the main gradient is turned off and the second gradient in the X-Y plane is applied, but this time in a slightly different direction. The hydrogen concentration in a new set of parallel strips is measured (see Figure 4.4). This process is repeated many times during the course of imaging one slice of tissue, so that by the end of the scan, the slice has been cut into many sets of parallel lines each crossing the plane at a different angle. When this is complete the total density along many lines passing through the slice is known, and then the techniques of Computerized Tomography (CT) can be applied. The same process is repeated for other slices of tissue and the Fourier transform is applied to reconstruct the MRI image from this data.

functional Magnetic Resonance Imaging is a new use of existing MRI technology that has been adapted to obtain brain images sensitive to the change of local blood flows. It was realized (Ogawa et al., 1990) that the presence of paramagnetic substances in the bloodstream could act as vascular markers giving useful contrast. Ogawa et al. (1990) showed, in experiments with rodents, that the oxygenation level of blood could be used as natural contrast agent because deoxyhemoglobin is more paramagnetic than oxyhemoglobin. Neural activity creates an imbalance between oxygen intake and blood
flows that can be measured with the use of MRI techniques. Ogawa et al. (1992) showed experimentally that when subjects viewed bright light it caused a change in the hemoglobin level in visual cortex that could be detected by MRI imaging. This development led to the so-called Blood Oxygenation Level Dependent (BOLD) contrast techniques used for functional mapping studies of human brain (Turner et al., 1998).

4.3 Brain Activity and fMRI

This section exposes the importance of fMRI imaging in understanding of the human brain function. Here, we review the “chemistry” of the human brain activation. Also, we give the description of the BOLD effect and its importance for fMRI imaging. Finally, we look at one popular parametric model describing the BOLD effect. In our description we follow Thirion (2003).

4.3.1 Brain Activity

The human brain contains around $10^{12}$ neurons. Their activity supports all the cognitive, sensory and motor processes of the body. Neurons carry information and exchange it through their synapses. The information is encoded in the form of the depolarization of the neuron’s membrane. It is exchanged at the level of synapses through the release of neurotransmitters (ions of potassium $K^+$ and sodium $Na^+$) that causes the depolarization of the next neuron. This phenomenon is also called “firing”. Recovery from firing
requires repackaging of neurotransmitter, a process that consumes Adenosine Triphosphate (ATP).

The ATP consumption requires a continuous supply of oxygen and glucose, supplied by the Cerebral Blood Flow (CBF). CBF increases substantially close to the areas of neural activity and used for assessing brain activity by Positron Emission Tomography (PET).

![Diagram of physiological changes during brain activation](image)

*Figure 4.5: Physiological changes during brain activation. The fMRI imaging is based on the lower three blocks. The arrows inside blocks indicate the direction and magnitude of the change.*

### 4.3.2 The BOLD Effect

Oxygen metabolism, measured in term of cerebral metabolic rate of oxygen (CMRO₂), increases much less than CBF during brain activation. As the result of this imbalance
between CBF and CMRO₂, there is a substantial drop in oxygen extraction and a corresponding drop in the deoxyhemoglobin content of the venous blood. Deoxyhemoglobin (dHb) is paramagnetic, and, therefore, the MRI signal is sensitive to this change. The presence of deoxyhemoglobin reduces the MRI signal at rest, while activation induces a slight increase of the MRI signal, known as the BOLD effect.

However, the real picture is significantly more complex. The BOLD signal change depends on the combined changes in Cerebral Blood Volume (CBV), CBF, and CMRO₂. Figure 4.5 gives a summary of the physiological processes during brain activation, adopted from Thirion (2003). The arrows inside the blocks indicate the direction and the magnitude of the change. For example, oxygen extraction decreased, while blood flow significantly increased. Note that the detailed explanation of these processes is unknown.

Figure 4.6 gives a description of a typical BOLD signal, adopted from Thirion (2003). The main features of this description are accepted by many authors. However, there are minor differences concerning initial dip and the post stimulation undershoot. The description concerns more the shape of the stimulation than the values of the change because they depend on the stimulation, the acquisition sequence and the scanner. The BOLD effect includes the initial delay of 1-3 sec. after the stimulus, followed by the signal increase of 3-8 sec. before the signal reaches its peak. After the end of the stimulus, the signal declines and often undershoots the original baseline. It takes about 20 sec. for undershoot to resolve. There is a variant of this model with an initial dip of 1-2 sec. at the beginning of the stimulation.
Figure 4.6: Schematic description of the BOLD effect. There is an initial dip of 1-3 sec. followed by a ramp of 3-8 sec. After the end of the stimulation, the signal undershoots the baseline and resolves in about 20 sec.

It is commonly accepted that the BOLD signal actually reflects the neuronal activity. However, the quantitative relationship between them has not been established yet. The shape of the BOLD effect is often referred to as a *hemodynamic response function* (HRF). There are many heuristic models to describe the hemodynamic response. Some of them model dip, some model undershoot or both. However, since the shape of the HRF is unknown, they are all just heuristics.

In this thesis, we use an fMRI data analysis package, called FMRIB Software Library (Smith et al., 2001), to compute the activation maps. We choose FSL over SPM because it is faster and has much better logging and batch mode processing capabilities. To comply with the generally accepted HRF model, we set the shape of the HRF model
in FSL as the difference of two gamma functions. Equation 4.2 gives the formula for the HRF function used in FSL

\[ h(T) = \frac{\lambda_1 (\lambda_1 T)^{K_1-1}}{(K_1-1)!} e^{-\lambda_1 T} - C \frac{\lambda_2 (\lambda_2 T)^{K_2-1}}{(K_2-1)!} e^{-\lambda_2 T} \]  
(4.2)

where \( \lambda_1 = \lambda_2 = 0.05 \), \( K_1 = 6 \), \( K_2 = 16 \), and \( C = 1/6 \).

Figure 4.7 plots this HRF function. Note that the HRF is the difference of two gamma functions, which models the undershoot but not the initial dip. Also, note that the slopes of the HRF are not symmetrical.

Figure 4.7: Plot of the double-gamma HRF as defined in FSL.
Chapter 5:

Analysis of fMRI Data

There is an enormous amount of research on fMRI data analysis. The existing analytic methods can be characterized into two groups: hypothesis-driven and exploratory methods (Thirion, 2003). Figure 5.1, adopted from Thirion (2003), gives the overview of the major existing methods.

Figure 5.1: Overview of the main methods for fMRI data analysis.
The hypothesis-driven methods assume a certain form of response to the experimental stimulation (Peterson et al., 1999a). Therefore, the response function can be parameterized and the model parameters can be estimated from experiments. The final step of this methodology is a statistical test that is interpreted as presence or absence of activation. Mostly, these methods are voxel-based and decide whether a particular voxel was activated by a given stimulus. The voxel is a basic element of 3D volume, analogous to the pixel (picture element) for a 2D image. MRI scanners generate output as 3D images, called volumes, consisting of thousands of voxels. The higher the resolution of an image the more voxels it has.

The potential weakness of the voxel-based methods is that they postulate a certain form of the response function, which may prove to be incorrect. On the positive side, they can give unambiguous answer to questions, such as “Does a particular voxel correlate, with the assumed response, to the given stimuli?” and give the probability of giving a false statement.

Exploratory methods (Peterson et al, 1999) search for patterns present in the dataset and ask how these patterns are structured in time and space. This kind of approach considers all voxels simultaneously, so in some sense it is multivariate. The idea is to identify by exploration what response patterns are present in the data and, if possible, to label them into known patterns (heartbeat, respiration, head motion, drift of the magnet, etc.). The ultimate goal of these methods is to identify patterns that can be generalized across different subjects or datasets.

In the next sections, we review the most important representatives of both types of fMRI analysis methods. We pay special attention to the General Linear Model (GLM),
which is at the heart of many popular fMRI tools (SPM, FSL, etc.). In this thesis, we use the GLM based approach for computing brain activations.

5.1 The General Linear Model

The General Linear Model (GLM) is implemented in several fMRI analysis tools, such as the Statistical Parametric Mapping (SPM) software (Friston et al., 1994). We will summarize the steps of this model, because almost all fMRI literature refers to it.

This summary is adopted from Frackowiack et al. (2004). Suppose that we conducted an experiment where we measured a response variable \( Y \) (such as oxygenation at a particular voxel) at \( N \) times. Let \( Y_i \) be the value of variable \( Y \) at time \( i=1,\ldots,N \). We assume that variable \( Y_i \) is to some degree random and denote it by a capital letter. Also, suppose that we have \( K \) \((K<N)\) explanatory variables, each measured without an error. Let \( x_{ij} \) be the value of explanatory variable \( j=1,\ldots,K \) at time \( i=1,\ldots,N \). For instance, an explanatory variable can be a dummy variable indicating the levels of an experimental factor.

The General Linear Model assumes that the response variable \( Y_i \) can be expressed as a linear combination of the explanatory variables plus random noise:

\[
Y_i = x_{i1}B_1 + x_{i2}B_2 + \ldots + x_{ik}B_k + \epsilon_i \tag{5.1}
\]

Here, the \( B_i \) are unknown parameters corresponding to a particular explanatory variable. The noise is an independent normal random variable with zero mean and variance \( \sigma^2 \), so
it can be written $\varepsilon_i \approx N(0, \sigma^2)$. Linear models with other noise distributions are called **generalized linear models**.

The general linear model can be easily expressed using matrix notation. For instance, consider writing Equation 5.1 for each observation:

\[
Y_1 = x_{11}B_1 + x_{12}B_2 + \ldots + x_{1K}B_K + \varepsilon_1
\]

\[
Y_2 = x_{21}B_1 + x_{22}B_2 + \ldots + x_{2K}B_K + \varepsilon_2
\]

\[\ldots\]

\[
Y_N = x_{N1}B_1 + x_{N2}B_2 + \ldots + x_{NK}B_K + \varepsilon_N
\]

The same in equivalent matrix form:

\[
\begin{pmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_N
\end{pmatrix} =
\begin{pmatrix}
x_{11} & x_{12} & \ldots & x_{1K} \\
x_{21} & x_{22} & \ldots & x_{2K} \\
\vdots & \vdots & \ddots & \vdots \\
x_{N1} & x_{N2} & \ldots & x_{NK}
\end{pmatrix}
\begin{pmatrix}
B_1 \\
B_2 \\
\vdots \\
B_K
\end{pmatrix} +
\begin{pmatrix}
\varepsilon_1 \\
\varepsilon_2 \\
\vdots \\
\varepsilon_N
\end{pmatrix}
\]

which in matrix notation is:

\[
Y = XB + \varepsilon \quad (5.2)
\]

where $Y$ is the column vector of observations, $\varepsilon$ is the column vector of noise terms, and $B$ is the column vector of model parameters. The $N \times K$ matrix $X$ is called the **design matrix**. It has one row per observation and one column per model parameter. For example, suppose that our model had two parameters, in other words two explanatory variables. The first variable could be a visual stimulation, such as presenting a picture of a geometrical object. The second variable could be an auditory stimulation, such as playing music. Our explanatory variables have two values, zero when there is no stimulus and one when the stimulus is on. The sample design matrix for this model with five brain scans (observations) could look like this:
\begin{pmatrix}
0 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
1 & 1 \\
\end{pmatrix}

(5.3)

The design matrix gives the values of each explanatory variable at each observation time. There is a column for each explanatory variable and a row for each observation time in the design matrix. For example, Equation 5.3 shows that the visual condition (the first column) was on and auditory condition (the second column) was off at the time of the second observation. Note that the number of stimuli and their timings are controlled by the experimenter and depends on his/her experimental goals. The design matrix is almost a complete description of the model. This is where our experimental knowledge about the expected signal is quantified.

Suppose that we want to solve Equation 5.2 with zero noise. If the number of parameters $K$ is equal to the number of observations $N$ and the determinant of the design matrix is non-zero, then there is one unique solution. There are infinitely many solutions when $K$ is greater than $N$. However, it generally can not be solved if $K$ is less than $N$. To understand this intuitively, imagine that $K = 1$ and $N = 10$. In this case, Equation 5.2 without noise would give ten different equalities that can not be satisfied simultaneously.

Typically, the number of observations $N$ is greater than 300, and the number of parameters $K$ is chosen to be less than $N$. Thus, Equation 5.2 can not be solved and some kind of best fit estimation is required. This can be done by the least squares method.
Let $\overline{B} = [\overline{B}_1, ..., \overline{B}_K]^T$ be a set of parameter estimates. Then the fitted values are $\overline{Y} = [\overline{Y}_1, ..., \overline{Y}_N]^T = X\overline{B}$. This gives residual errors $e = [e_1, ..., e_N]^T = Y - \overline{Y} = Y - X\overline{B}$. The residual sum of squares $S = \sum_{i=1}^{N} e_i^2 = e^T e$ is the sum of squared differences between the actual and fitted values and measures the fit of the model with the estimated parameters. The least squares method minimizes the residual sum of squares. This sum is minimized when all partial derivatives $\frac{\partial S}{\partial B_i}$ are zero. It can be shown that this is equivalent to finding $B$ that settles the normal equations:

$$X^TY = (X^TX)\overline{B} \quad (5.4)$$

### 5.2 Principal Component Analysis

Principal Component Analysis (PCA) need not relate observations to stimuli. It is often performed using the Singular Value Decomposition (SVD) techniques. The SVD decomposes the data into mutually orthogonal spatio-temporal components. Suppose that the dataset has $L$ voxels and $N$ time frames (observations), forming an $L \times N$ matrix. Then the SVD of $X$ is:

$$X = U\Sigma V^T \quad (5.5)$$

where $U$ and $V$ are $L \times L$ and $N \times N$ orthogonal matrices, and $\Sigma$ is a $L \times N$ matrix with non-zero elements only on its diagonal.
One can perform SVD on a raw dataset to explore it. However, this is rarely done. Usually, the space of interest relative to the $N \times K$ design matrix $Y$ is defined and SVD is performed on $XY^T$.

5.3 Independent Component Analysis

SVD finds spatial and temporal components that are mutually orthogonal. On the other hand, Independent Component Analysis (ICA) finds statistically independent components either in the spatial or the temporal domain but not both. Usually ICA is used with a spatial independence rather than with a temporal independence criterion. The reason is that there are many more voxels than time points, for assessing statistical independence.

In the case of spatial ICA, the dataset $X$ is a set of images $X(t)$ and is viewed as a superposition of independent images $S$, called “sources”. In other words,

$$X^T = MS + E \quad (5.6)$$

where $M$ is the “mixing” matrix and $E$ is a residual noise term. Generally, $X, M, S,$ and $E$ are $L \times N$, $N \times K$, $K \times L$, and $N \times L$ matrices respectively. $K$ is the unknown parameter, the number of independent sources, $L$ is the number of voxels, and $N$ is the number of observations (images). Thus, the solution to the problem is

$$S = U(X - E)^T \quad (5.7)$$

where the $K \times N$ matrix $U$ is the “unmixing” matrix and can be viewed as an inverse of the mixing matrix $M$.

Note that in case of temporal ICA

$$X = M'S' + E' \quad (5.8)$$
where $M$, $S$, and $E$ are $L \times K$, $K \times N$, and $L \times N$ matrices respectively.

The fundamental principle in ICA is that the independent components must be nongaussian for ICA to be possible (Hyvarinen and Oja, 2000). Mutual information is a measure of the dependence between random variables. One approach to ICA estimation is minimization of mutual information. However, the theory of ICA is outside the scope of this thesis. Interested readers may refer to Hyvarinen and Oja (2000) for good description of ICA theory and algorithms.

The matrix $S$ can be viewed as a set of activation maps corresponding to the different independent effects present in the dataset. ICA is one of the exploratory techniques widely used for the study of fMRI data.

### 5.4 Clustering

Clustering techniques have been used with fMRI data as an alternative approach. The main assumption here is that voxels are activated according to some rule and they can be classified into several types of similar activation. Voxels with similar activation are assigned to a cluster, where the cluster center represents the temporal behavior of a “typical” voxel in the cluster. Clustering techniques support isolating areas with similar activation and asking whether two voxels or cluster have similar activation (Goutte et al., 1999).

Several studies have been conducted applying clustering methods directly to the fMRI time series, using the fuzzy $K$-means algorithm (Baumgartner et al., 1998; Moser et al., 1999). The difference between standard $K$-means and fuzzy $K$-means algorithms is
that in the former case the cluster membership is either zero (not-member) or one (member). In the latter case the membership is fuzzy and can have any value between 0 and 1 depending on the strength of the membership. Filzmoser et al. (1999) used hierarchical clustering methods on fMRI data. However, due to the high noise level, clustering results on raw fMRI data were unsatisfactory, and did not classify voxels into meaningful clusters of activation.

Goutte et al. (1999) explored clustering on the cross-correlation between stimulus and time series rather than raw fMRI time series. The cross-correlation function at a voxel $i$ is defined as

$$y_i(t) = \frac{1}{N} \sum_{u=1}^{N} x_i(u)S(u-t)$$

where $x_i(u)$ is the observed value of voxel $i$ in image $u$, $N$ is the number of images, $S(u)$ is the stimulus function (value of stimulus at time of scanning image $u$), and $t$ is an integer delay that varied between $-T$ and $T$, where $T$ is the period of the stimulus. Note that the cross-correlation was defined only at intervals equal to time of repetition.

In this thesis, we use “lagged” correlations, which are related to the concept of cross-correlations. The time of repetition, time between adjacent scans, is usually around 2 seconds. It is likely that 2 second intervals are too long for cognitive processes. Therefore, we use lagged correlations which are defined at positive intervals much smaller than time of repetition.

Dimitriadou et al. (2004) compared the performance of several clustering algorithms applied to fMRI data. Particularly, they compared three types of algorithms: hierarchical, crisp (neutral gas, self-organizing maps, hard competitive learning, $K$-means, maximin-distance, CLARA) and fuzzy ($K$-means, fuzzy competitive learning).
The quality of clustering voxels into a cluster containing activated voxels was evaluated on two measures: the correlation coefficient and the weighted Jaccard coefficient. The correlation coefficient is an intuitive measure which is simply the correlation between the time course of the cluster center and the stimulus time course convolved with the HRF. The Jaccard coefficient is a measure that takes into account activated voxels in the cluster (true positives $a$), non-activated voxels in the cluster (false negatives $b$) and activated voxels not in the cluster (false positives $c$). We refer to the cluster with activated voxels as the cluster here. The weighted Jaccard coefficient was used due to a small number of true positives compared to the total number of voxels in fMRI. Equation 5.10 gives the formula for the weighted Jaccard coefficient.

$$J_c = \frac{a + \frac{1}{P(a)}}{a + \frac{1}{P(a)} + b + \frac{1}{P(b)} + c + \frac{1}{P(c)}}$$  \hspace{1cm} (5.10)$$

where $P(a) = (a + c) * (a + b) / n^2$, $P(b) = (b + d) * (a + b) / n^2$ and $P(c) = (a + c) * (c + d) / n^2$ are the probabilities that a voxel is true positive, false negative and false positive respectively, $d$ is the number of truly negative voxels and $n$ is the total number of voxels in the dataset. For instance, a truly positive voxel must be in the activated cluster, with probability $P_1 = (a + c) / n$, and be activated, with probability $P_2 = (a + b) / n$. This gives $P(a) = P_1 * P_2 = (a + c) * (a + b) / n^2$. We assume that each voxel has equal probability of being activated/non-activated.

The drawback of this study is that the performance of the algorithms was compared on fully artificial (mathematical) and synthesized (hybrid) fMRI datasets. It is not known whether the results of the comparison would hold for real fMRI data.
Chapter 6:

Graph-Based Methods

Graph matching is one of the fundamental techniques in computer vision and pattern recognition (Massaro & Pelillo, 2003). Graphs provide a means by which abstract pictorial descriptions can be matched to one another (Luo et al., 2001). There has been great deal of effort over the past decades to devise efficient and robust algorithms for graph matching. Two radically distinct approaches have emerged. The first framework casts the matching problem as a pure graph-theoretic problem, for which a solid theory and powerful algorithms have been developed (the maximum clique problem). In the second approach, the matching problem is viewed as one of objective function minimization. In this case, an objective function is minimized and its minimizers correspond to solutions of the graph matching problem (Massaro & Pelillo, 2003).

In this Chapter, we define the problem of graph matching. Then we proceed by reviewing representatives of both graph-theoretic and optimization approaches to the graph matching problem. Also, we discuss an algorithm developed by Shokoufandeh et al. (2001), which we use in this thesis to match graphs extracted from the fMRI data.
6.1 Notations and Definitions

Before going into the details we need to introduce some notations and definitions following Pelillo et al. (1999). Let $G=(V, E)$ be a graph (undirected graph), where $V$ is a set of vertices and $E$ is a set of edges. The number of vertices in $V$ is called the order of $G$. The size of $G$ is a number of edges in $E$. Two vertices $u, v \in V$ are adjacent (denoted $u \sim v$) if they are connected by an edge. A graph with directed edges is called directed. The structure of a graph $G$ can be represented by a $|V| \times |V|$ adjacency matrix $D$, such that

$$
D_{ab} = \begin{cases} 
  w_{ab}, & \text{if } (a, b) \in E \\
  0, & \text{otherwise}
\end{cases}
$$

(6.1)

where $w_{ab}$ is the weight of the edge between vertices $a$ and $b$. A path is any sequence of distinct vertices $v_0v_1\ldots v_n$, such that for all $i=1,2,\ldots,n$, $v_{i-1} \sim v_i$, the length of the path is $n$. The path is called a cycle if $v_0 = v_n$. A graph is called connected if every pair of vertices is connected by a path. The distance between two vertices $u$ and $v$ is the length of the shortest path connecting them. A subset of vertices $C$ is called a clique if all of its vertices are mutually adjacent. A clique is called maximal if it is not contained in any larger clique and called maximum if it is the largest clique in the graph. An isomorphism between two graphs $G_1=(V_1, E_1)$ and $G_2=(V_2, E_2)$ is any bijection $\phi: V_1 \rightarrow V_2$ such that $(i, j) \in E_1 \iff (\phi(i), \phi(j)) \in E_2$, for all $i, j \in V_1$. Two graphs are called isomorphic if there exists an isomorphism between them. In particular, the maximum common subgraph problem consists of finding the largest isomorphic subgraphs of $G_1$ and $G_2$. 
A connected graph with no cycles is called a tree. If a tree has a distinguished vertex designated as a root, then it is called a rooted tree. The level of a vertex \( u \) in a rooted tree, \( \text{lev}(u) \), is the length of the path connecting the root to \( u \). If vertices \( u \) and \( v \) are adjacent and \( \text{lev}(v) - \text{lev}(u) = +1 \), we say that \( u \) is the parent of \( v \) and \( v \) is a child of \( u \). Trees have the interesting property that any two vertices are connected by a unique path.

The association graph derived from two graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \) is the undirected graph \( G = (V, E) \), where \( V = V_1 \times V_2 \) and

\[
E = \{(i, h), (j, k) \in V \times V : i \neq j, h \neq k, (i, j) \in E_1 \iff (h, k) \in E_2\} \quad \text{(Massaro et al., 2003)}.
\]

### 6.2 Graph Matching

It has been proven that all maximal (maximum) cliques in \( G \) are in one-to-one correspondence with maximal (maximum) common subgraph isomorphisms between \( G_1 \) and \( G_2 \) (Barrow and Burstall, 1976). This result establishes equivalence between the graph matching problem and the maximum clique problem. The maximum clique problem is known to be \( \mathcal{NP} \)-complete. However, powerful heuristics have been developed which efficiently find good approximate solutions and there are many classes of graphs for which the problem is solvable in polynomial time (Pelillo et al., 1999).

Gold and Rangarajan’s (1996) graduated assignment algorithm is an example of such an optimization framework. The weighted graph matching problem is defined as follows. Given two undirected graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \) which may be sparse and whose edges may take real values, find the match matrix \( M \) such that the following objective function is minimized:
subject to $\forall a \sum_{i=1}^{A} M_{ai} \leq 1$, $\forall a \sum_{i=1}^{A} M_{ai} \leq 1$, $\forall ai M_{ai} \in \{0,1\}$. Graphs $G_1$ and $G_2$ have $A$ and $I$ vertices respectively and

$$C_{aij} = \begin{cases} 0, & \text{if either } (a,b) \notin E_1 \text{ or } (i,j) \notin E_2 \\ c((a,b),(i,j)), & \text{otherwise} \end{cases} \tag{6.3}$$

The matrix $M$ shows which vertices in the two graphs match. For instance,

$$M_{ai} = \begin{cases} 1, & \text{if vertex } a \in V_1 \text{ matches to vertex } i \in V_2 \\ 0, & \text{otherwise} \end{cases} \tag{6.4}$$

The function $c(\ldots)$ is a measure of “compatibility” between the edges of the two graphs.

Note that the intuition behind Equation 6.2 is to find a best match between the vertices of the two graphs so that their topologies are most similar.

The weighted graph matching problem formulated above is NP-complete for many definitions of the function $c(\ldots)$. Thus, only approximate solutions can be found. Gold and Rangarajan (1996) proposed a graduated assignment method to find such solutions. Given an initial value $M^0$, the objective function can be expanded via a Taylor series:

$$F(M) \approx -\frac{1}{2} \sum_{a} \sum_{i} \sum_{b} \sum_{j} M_{ai}^0 M_{bj}^0 C_{aij} - \sum_{a} \sum_{i} \frac{\partial F}{\partial M_{ai}} \bigg|_{M=M^0} (M_{ai} - M_{ai}^0) \tag{6.5}$$

The algorithm starts with some initial feasible value for $M$. Then it does a first-order Taylor series expansion by taking the partial derivative. The algorithm finds new values corresponding to the current assignment. Then it takes the resulting $M$ and substitutes
back in Equation 6.5, repeating the process until $M$ converges or a threshold number of iterations is reached.

Luo and Hancock (2001) developed a maximum-likelihood framework for graph matching. Their method treats the graph to be matched (the data graph) as observed data and considers the set of correspondences with some available model (the model graph) as hidden variables. Let $G_D = (V_D, E_D)$ and $G_M = (V_M, E_M)$ be data and model graphs respectively. The goal of matching is to associate vertices in the data graph with the vertices in the model graph. The state of match is represented by a function $f : V_D \rightarrow V_M$.

For instance, $f(a) = b$ means that a vertex $a$ in the data graph is matched to a vertex $b$ in the model graph. Let $S$ be a $|V_D| \times |V_M|$ matching matrix whose elements are assignment variables such that

$$S_{ab} = \begin{cases} 1, & \text{if } f(a) = b \\ 0, & \text{otherwise} \end{cases} (6.6)$$

The main idea of this method is to find the matrix of assignment variables that maximizes the conditional likelihood of the observed data graph given the available model graph. In other words, the algorithm seeks the matrix of assignment variables which satisfies the following condition:

$$S = \arg \max_S P(G_D | G_M, S) (6.7)$$

There is a possibility that any vertex of the data graph may be in correspondence with any vertex of the model graph. This involves summing probabilities over the set of possible correspondences. Thus,

$$P(G_D | G_M, S) = \prod_{a \in V_D, b \in V_M} p(x_a | y_b, S) (6.8)$$
where $p(x_a \mid y_b, S)$ is the probability that vertex $a$ in data graph is in correspondence with the vertex $b$ in model graph.

The final likelihood function in Luo and Hancock’s (2001) model is

$$P(G_D \mid G_M, S) = \prod_{a \in V_D, b \in V_M} \sum_{P_e} \left[ \frac{1}{P(x_a)} \right]^{\psi_{a|b}} \exp \left[ \ln \frac{1 - P_e}{P_e} \sum_{c \in V_D, d \in V_M} D_{ac} M_{bd} S_{ed} \right]$$

(6.9)

where $P_e$ is the probability of the assignment $f(a) = b$ being in error, $D_{ac}$ and $M_{bd}$ are adjacency matrices for data graph and model graph respectively. This model is solved by the Expectation Maximization (EM) algorithm. The EM algorithm iterates between the interleaved expectation and maximization steps until convergence is reached. The expectation step involves updating the a posteriori probabilities of the missing data using the current parameter estimates. The model parameters are recomputed to maximize the expectation value of the data likelihood in the expectation step.

Pelillo et al. (1999) tried to unify both of these graph matching frameworks for matching hierarchical structures. They argued that most graph matching literature focused on “flat” problems in the sense that there is no partial ordering imposed on the data. However, in many practical problems data is organized in hierarchical manner, e.g. trees. Matching such representations is very important in pattern recognition applications.

As we mentioned before, there is a one-to-one correspondence between the maximal (maximum) common subtree isomorphism and maximal (maximum) clique in the corresponding association graph. Pelillo et al. (1999) generalize this result for the tree association graph (TAG) by using a path-string notation. The path-string of two vertices $x_0$ and $x_n$, joined by the unique path $x_0x_1\ldots x_{n-1}x_n$, is the string $\text{str}(x_0,x_n) = s_1s_2\ldots s_n$ on the alphabet $\{-1, +1\}$ where for all $i=1,2,\ldots,n$, $s_i = \text{lev}(x_i) - \text{lev}(x_{i-1})$. The difference
between association graph and TAG is that for any two vertices \((u, w)\) and \((v, z)\) in TAG \((u, w) \sim (v, z) \iff str(u, v) = str(w, z)\).

Now the problem of matching two trees has been reduced to finding a maximum clique in the TAG. Let \(G\) be an arbitrary graph of order \(n\). Given a subset of vertices \(C\), its characteristic vector \(x^c\) is defined as:

\[
x^c_i = \begin{cases} 
1/|C|, & \text{if } u_i \in C \\
0, & \text{otherwise}
\end{cases}
\]

(6.10)

where \(|C|\) is the cardinality of \(C\), and \(u_i\) is a vertex in graph \(G\).

Pelillo et al. (1999) use the modified Motzkin-Straus theorem (Bomze, 1997) to solve the maximum clique problem. This theorem establishes a connection between global maximizers of the quadratic function \(f(X) = X^TAX + \frac{1}{2}X^TX\), where \(A\) is the adjacency matrix of \(G\), and the maximum clique of \(G\). It states that a subset of vertices \(C\) of a graph \(G\) is a maximum clique if and only if its characteristic vector \(x^c\) is a global maximizer of the function \(f\). Thus, this approach unifies graph-theoretic and optimization frameworks in graph matching.

Horaud and Sossa (1995) approached graph matching by indexing. They argue that when a query graph must be matched to a collection of many model graphs, the problem becomes intractable because the complexity grows significantly with the number of features. Thus, an indexing process is necessary when the matching process has to select among many possible objects.

Let \(G_1\) and \(G_2\) be graphs with an equal number of vertices. They are isomorphic if and only if there exists a permutation matrix \(P\) satisfying \(A_2 = PA_1P^{-1}\), where \(A_1\) and
$A_2$ are adjacency matrices of the two graphs. Thus, there are two ways to decide if two graphs are isomorphic. First, one can find the permutation matrix that satisfies this condition. Alternatively, one can find an algebraic characterization of the adjacency matrix that is invariant under a similarity transformation of the adjacency matrix. Luckily, there are polynomials which characterize a graph unambiguously up to an isomorphism. One such polynomial is the second immanantal polynomial associated with the Laplacian matrix of a graph. The elements of the Laplacian matrix $L(G)$ of a binary graph (a graph whose adjacency matrix has only binary values) are defined as:

$$
L_{ij}(G) = \begin{cases} 
  d_i, & \text{if } i = j \\
  -1, & \text{if there is an edge between vertices } i \text{ and } j \\
  0, & \text{otherwise}
\end{cases}
$$

(6.11)

where $d_i$ is the number of graph edges meeting at vertex $i$ and is called the degree of the vertex $i$. The second immanantal polynomial associated with $n \times n$ Laplacian matrix $L(G)$ can be written in generic form as:

$$
d_2(xI - L(G)) = c_0(L(G))x^n - c_1(L(G))x^{n-1} + \ldots + (-1)^n c_n(L(G))
$$

(6.12)

The coefficients of this polynomial are integers and can be computed as ($n$ is the number of vertices and $m$ is the number of edges of the graph):

$$
\begin{align*}
  c_0(L(G)) &= n - 1 \\
  c_1(L(G)) &= 2m(n - 1) \\
  &\vdots \\
  c_k(L(G)) &= \sum_{X \in Q_m} \left( \sum_{i=1}^n L_{ii} \det(L(G)\{X\}(i)) - \det(L(G)\{X\}) \right)
\end{align*}
$$

(6.13)

where $L_{ii}$ is a diagonal term of $L(G)$ and $Q_m$ is the set of all the $C^k_n$ strictly increasing sequences of size $k$ ($2 \leq k \leq n$) obtained from the set $\{1, 2, \ldots, n\}$. A necessary (but not
sufficient) condition for two graphs to be isomorphic is that they have the same second immanantal polynomial.

Horaud and Sossa’s (1995) algorithm uses this property of the polynomial to quickly search the database of graphs for a match. The algorithm employs a clever technique to disassemble graphs into smaller components for subgraph matching. It also uses a hashing technique to speed up the matching process.

Graph structures have been successfully used for indexing images. For example, Shokoufandeh et al. (1999) applied shock graphs for indexing and matching of two-dimensional silhouettes. The shock graph is constructed by finding shocks or singularities of the curve, by an evolution process in the plane. The boundary of the object is found and then it is deformed inward with a known velocity. Singularity points of the deformation are called shocks and form the nodes of the shock graph. Nodes are ordered according to their times of formation. The authors developed an algorithm for matching two shock graphs based on both topological structure and geometric structure. The feature used for matching graphs is based on the sum of largest eigenvalues of the graph’s adjacency matrix. The sum is computed for each subtree rooted at the nodes of the graph to preserve the topology of the graph. Sundar et al. (2003) extended this work by applying graph structures for indexing and retrieving three-dimensional volumetric objects. They have computed a skeletal graph, directly from the 3-D object, which contained topological information about the object and was used for subsequent indexing.
Chapter 7:

Brodmann Vectors and Graph Building

Heuristics

In this Chapter, we give the definition of time “lagged” correlations and explain why we introduced them. Also, we propose three new heuristics for content-based fMRI retrieval: Brodmann vectors, Minimum Distance Graph (MDG) and Maximum Correlation Graph (MCG). We give the underlying assumptions and the construction steps for each heuristic. In Section 7.5, we define the node similarity function used for graph matching in this thesis.

7.1 Lagged Correlations

One simple way to construct a graph from fMRI data is to threshold the image so that only voxels with relatively high intensity remain and then cluster those voxels into centroids. Essentially, the sequence of fMRI images is reduced to the timed sequence of centroids. Each centroid has spatial location and time when it was observed. It can be imagined that centroids are located in 4-D space (space and time). Assume for a moment
that centroids may be related to other centroids that are one step later in time. A naive choice is to compute the Euclidean time-space distance between centroids and connect those that are adjacent in time and have the shortest spatial separation. This heuristic imposes a graph structure on the fMRI sequence that could potentially be used for indexing and retrieval. However, it is unlikely that this naive heuristic will be effective for our purposes.

The problem with the naive approach is that fMRI images are very noisy and thresholding voxel intensities will pick up both activations and random noise. Also, a human brain may have “random thoughts” even when it is at rest. Thus, we need to find regions in the brain that correspond to stimuli. One common way to find such activations is to compute the correlation between voxel time series and stimulus time series (convolved with the hemodynamic response function). Thresholding the voxels by requiring a probability of activation significantly different from that for noise will give activation regions that correlate with the stimuli.

This approach gives a static map, or snapshot of activated regions in a brain. Now, based on our current knowledge of brain sciences, it appears that there are two fundamental principles to the brain’s functional organization: functional specialization and functional integration (Friston, 2004). Functional specialization means that there are various clusters of neurons in the brain responsible for different tasks. For example, one region of the brain processes input from our visual sensors and another region processes motor movements. However, this is not enough to explain processing of complex stimuli. There is also functional integration, which means that different parts of the brain interact
with each other to complete a task. It has been quite difficult to identify functional integrations in human brain.

To return to graph building, it is not enough to identify static activations in the brain. The correlation map of activations identifies areas of functional specialization. It tells us that some region in the brain was activated during a task. However, it does not tell us about interaction between activated regions, in processing the task. Suppose that when a subject is shown specific visual stimuli he/she has to press a particular button. This task involves two parts: processing the visual stimulus and pressing the button. There should be some causal chain of activations. It is unreasonable to think that the subject’s hand will press the button before he/she “realizes” that the specific visual stimulus has been presented. Thus, conventional logic tells us that there should be functional integration between at least two regions of the brain responsible for visual and motor processing.

Assuming that the form of the hemodynamic response function (HRF) is known, one can locate regions of the brain that are activated immediately following the presentation of the stimuli. However, this approach cannot identify parts of brain that are consistently activated with a certain delay after the presentation of the stimulus. Identifying these “delayed regions” requires computing “lagged” correlations. That is, we need to identify voxels that have stronger correlation with the stimulus \( n \times D \) seconds later after the stimulus onset time. \( D \) is a duration of a basic time lag in seconds, \( n \) is the number of the time lag and will take values 0, 1, 2, … up to some reasonable number \( M \).

We, therefore, compute \( M \) lagged correlations and identify voxels based on their probability of being significantly different from noise. After clustering active voxels in each correlation image, there are \( M \) time sequenced correlation images with activation
clusters. Notice that clusters in the correlation image $n$ are always activated $n*D$ seconds later than those that respond first to the presenting stimulus.

### 7.2 Brodmann Vectors

The activation clusters appear all over the brain. One way to label brain regions is by Brodmann areas. Brodmann (1909) classified brain regions based on their cytoarchitecture, in other words, their appearance under a light microscope. Sometimes, there is a link between the microscopic appearance of a region and its function. However, there is no necessary connection between Brodmann areas and brain functions.

The human brain has 47 Brodmann areas. Brodmann areas can span both hemispheres. To establish a connection between the voxels of an fMRI image and Brodmann areas, the image must be normalized to a template brain (e.g. MNI or Talairach) (Collins, 1994). Normalization is a process in which the image is transformed (rotated, scaled, translated, etc.) to fit the standard canonical brain, such as Talairach or Montreal Neurological Institute (MNI) templates. In general, the Brodmann areas obtained by using different atlases are not precise due to tremendous variation in individual brains.

Given the map of activated areas in a “thinking” brain, we can determine what percentage of each Brodmann area is active. The level of activation in a particular Brodmann area can be computed by dividing the number of active voxels in that area by the size of the area. This way, we can build a 47 component vector. Each component of this vector gives the level of activation in the corresponding Brodmann area. Now, we
know that Brodmann areas span both hemispheres. Even if the same Brodmann area is activated in two subjects, the activations may be in different hemispheres. Thus, it appears that we should to distinguish between right and left halves of a Brodmann area.

The size of the vector is doubled to separate information about activations in both the left and right halves of Brodmann areas. We will refer to this modified or “doubled” vector as a “Brodmann vector”. Figure 7.1 sketches the construction of a Brodmann vector.

![Figure 7.1: Building a Brodmann vector. Each Brodmann area that has activated voxels will have a corresponding non-zero component in the Brodmann vector.](image)

Suppose there are two Brodmann vectors that have value 0.3 at component 35. This does not mean that two subjects have similar activation in Brodmann area 35. In fact, it is quite possible that there are no overlapping voxels in the activations for two subjects. This might be a problem. On the other hand, it appears that simply counting voxel overlaps is not an effective way to compare activation maps. In particular, computing voxel overlap
requires perfect normalization, which is not feasible due to large variation in size and shape of individual brains. Hence, voxels with the same coordinates might correspond to functionally different areas, making voxel overlaps ineffective.

It is not clear yet whether Brodmann areas are related to functionality. However, Brodmann vectors can potentially be used in matching “similar” fMRIs. The fMRI datasets can be represented by Brodmann vectors derived from the activation maps that show clusters of activated areas. Let $B_1$ and $B_2$ be the Brodmann vectors for two different fMRIs. The normalized similarity is defined as the inner product of two Brodmann vectors

$$S(B_1, B_2) = \frac{B_1 \cdot B_2}{\|B_1\| \|B_2\|} \quad (7.1)$$

Higher values of $S$ represent greater similarity. We call this technique the “Brodmann vectors” approach and use it for fMRI matching in this thesis. This choice is motivated by the success of the cosine similarity measure in text retrieval applications.

### 7.3 Minimum Euclidean Distance as a Graph Building Principle

The weakness of the Brodmann vectors approach is that it requires perfect normalization to a canonical brain (MNI or Talairach), which is impossible to achieve. Small changes in the normalization process can alter voxel memberships from one Brodmann area to another. On the other hand, graph matching algorithms are quite stable under the presence of small noise. Therefore, we conjecture that graph based methods might be more effective than the simple Brodmann vectors approach.
There are a number of ways to construct nodes and edges of the graph. However, a preference is given to graph construction heuristics grounded in our current knowledge of human brain functionality.

The output of computing lagged correlations is a timed sequence of correlation maps (t-maps) with clusters of activations. We use the fMRIB Software Library (FSL) (Smith et al., 2001) package, run in batch mode, to generate a set of lagged correlation maps. That is, we get a sequence of lagged correlation maps by running FSL for each time lag. It appears reasonable to select centers of mass of the activation clusters as nodes of the graph. Equation 7.2 gives the center of mass for a cluster of \( N \) objects

\[
\bar{C} = \frac{\sum_{i=1}^{N} m_i \bar{x}_i}{\sum_{i=1}^{N} m_i} = \frac{1}{M} \sum_{i=1}^{N} m_i \bar{x}_i \quad (7.2)
\]

where \( m_i, \bar{x}_i, \) and \( M \) are the mass and coordinates of object \( i \) and total mass of the cluster, respectively. We will use the strength of the correlation (\( t \)-value) as voxel mass in this thesis. The node gets the sum of \( t \)-values over its voxels in its mass.

The decision as to which nodes to connect is not obvious. Presumably, signals in the brain are transmitted via chemicals, with a finite propagation speed. Hence, it is somewhat likely that activation clusters close in space are related. If so, the Euclidean distance can be used to decide which nodes should be connected. Two clusters in consecutive time lagged t-map are connected if their centroids have minimum Euclidean separation in space. We will refer to the graph built with this heuristic as the Minimum Distance Graph. We need to initialize the graph in the first t-map.
We assume that when a stimulus is presented, some specific areas of the brain are activated. Then some signal propagates from this area to other places that subsequently get activated. Thus, our model of interaction looks like a tree with a root, and with branches that follow the pattern of activation in the brain. It is possible that a particular brain region receives signals via several paths simultaneously. However, we do not allow such interactions in our model for sake of simplicity. Notice that our problem of finding the interaction tree is equivalent to the problem of finding the shortest path from a single source. Indeed, only nodes in adjacent lags are connected, if the Euclidean distance between them is minimal. Hence, the path from the root to any node is minimal.

Figure 7.2: The Minimum Distance Graph (MDG). (a) Given nodes and edge weights (Euclidean distances); (b) Dijkstra’s shortest path algorithm is used to build the MDG by connecting nodes so that the signal path from root to any node is minimal.
Thus, our problem has been reduced to one of finding the shortest path from a single source (root). Dijkstra’s shortest path algorithm gives a polynomial time solution to this problem (Cormen et al., 1990). The only remaining question is how to choose the root node. We hypothesized that interaction propagates from some area of the brain that was activated by the stimulus. Therefore, it is likely that this area will have strong correlation with the stimulus. Based on this, the minimum heuristic selects that node with the strongest correlation (t-value) at time lag 0 (no delay) as a root node. In building paths, all nodes at time lag 0 will be connected to the root node and the weights of the edges will be set to zero so that they all have equal chance to be connected to nodes in the next lag.

Figure 7.2 shows the steps to build a Minimum Distance Graph (MDG) for a fictitious graph. Note that node 1 is a root. It must have the largest t-value among all nodes at time lag 0 to be a root. Edge weights, equal to Euclidean distances, are shown next to each edge. Our goal is to minimize the signal path from the root to all nodes. Dijkstra’s shortest path algorithm is used to minimize this path. Note that node 2, in time lag 0, is given an equal chance to connect to any nodes in the next time lag (its distance to the root being artificially set to 0).

7.4 Maximum Correlation as a Graph Building Principle

Our ultimate goal is to connect activations that are related to each other. Our alternative heuristic for connecting nodes is based on this idea. We hypothesize that related
activations should strongly correlate with each other. Therefore, the heuristic connects centroids based on the strength of their correlations.

There are two natural choices in determining the strength of the correlation. One approach is based on a value of the correlation coefficient between the central voxel of a cluster and the stimulus time series. The central voxel is the voxel at the cluster’s center of mass. Hence, clusters with the largest positive $t$-values at the center are connected with each other. In this case, connected centroids do not necessarily correlate with each other but correlate with the stimulus time series. An alternative approach is to actually compute between-cluster correlations. Computing correlations between the original time series of the centers of clusters in adjacent time lag correlation maps measures how well these clusters correlate.

Let $x_1(t)$ and $x_2(t)$ be the time series of two cluster centers (the centers of mass). Then the correlation between the two cluster centers is computed as

$$r = \frac{\sum_{i=1}^{N} (x_1(t) - \mu_1)(x_2(t) - \mu_2)}{N\sigma_1\sigma_2}$$

where $N$ is the number of observations (fMRI scans), $\mu_1$ and $\mu_2$ are the mean values for the first and the second time series respectively, $\sigma_1$ and $\sigma_2$ are the standard deviations for the first and the second time series respectively.

According to our hypothesis, the nodes with the strongest correlations should be connected. We will refer to the graph built with this heuristic as the Maximum Correlation Graph. As before, we assume that a stimulus activates some sensory area of the brain and the activation spreads from there to other areas. Hence, the pattern of activation is a tree. In selecting the root of the tree we apply the same reasoning as for the
minimum distance graph. That is, a node with the strongest \( t \)-value in the initial lag map (lag 0) is selected as the root. As in the previous heuristic, all the nodes in initial lag are given equal chance by connecting them to the root node with edge weights of zero.

Of course, this problem can be reduced to the shortest path from a single source problem as well. Since, at every step, the heuristic connects nodes with the maximum correlation, the sum of correlation values along any path is maximal. The problem can be reduced to the shortest path problem by minimizing the sum of \((1 - \text{correlation})\) along any path from the root. It is easy to see that if this sum is minimal along some path, the sum of correlations along this path is maximal. Hence, the heuristic can use Dijkstra’s shortest path algorithm to build the tree using \((1 - \text{correlation})\) as edge weights.

\[
\begin{align*}
\text{Lag 0} & \quad 1 \quad 2 \\
\text{Lag 1} & \quad 3 \quad 4 \quad 5 \\
\text{Lag 2} & \quad 6 \\
\end{align*}
\]

\[
\begin{align*}
\text{Time} \quad \downarrow
\end{align*}
\]

**Figure 7.3:** The Maximum Correlation Graph (MCG). (a) Given a set of nodes and edges with weights (correlation coefficients); (b) Dijkstra’s shortest path algorithm is used to build the MCG. The result is that the sum of correlations along any path from root is maximal.
Figure 7.3 gives an example of building a Maximum Correlation Graph (MCG) for a fictitious graph. Given the set of nodes and edges with weights equal to the correlation coefficients, we use Dijkstra’s shortest path algorithm to connect the nodes with the strongest correlations. Dijkstra’s algorithm minimizes the sum of complemented weights (1 – weight). Hence, the outcome is that the sum of correlations along any path from root is maximal. Note that node 1 at time lag 0 is a root, which means it has the largest t-value among all nodes at lag 0. Also, node 2 is given equal chance to connect to nodes in the next time lag by artificially setting its correlation with the root to 0.

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Features</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brodmann Vector</td>
<td>Brodmann Areas</td>
<td>Activation overlaps</td>
<td>Builds a vector consisting of activation and BA overlaps</td>
</tr>
<tr>
<td></td>
<td>(BA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum Distance</td>
<td>Euclidean</td>
<td>Distance along any</td>
<td>Connects nodes with minimum Euclidean distance in consecutive frames</td>
</tr>
<tr>
<td></td>
<td>distance</td>
<td>path from root is</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>minimal</td>
<td></td>
</tr>
<tr>
<td>Maximum Correlation</td>
<td>Correlation</td>
<td>Sum of correlations</td>
<td>Connects nodes with maximum correlation in consecutive frames</td>
</tr>
<tr>
<td></td>
<td></td>
<td>along any path</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from root is maximal</td>
<td></td>
</tr>
</tbody>
</table>

*Table 7.1: Brief summary of heuristics used for content-based fMRI retrieval.*
Table 7.1 shows a brief summary of the three fMRI matching heuristics discussed in this thesis. Note that Brodmann vector matching requires good brain normalization while the two graph-based heuristics should be more stable under presence of small displacements or changes in anatomy.

### 7.5 Computing Graph Similarity

Finally, with fMRI sequences represented by trees, we need a matching rule. Shokoufandeh et al. (2001) developed an algorithm to match noisy Directed Acyclic Graphs (DAG).

The main idea of the algorithm is to cast the structural matching problem as a set of bipartite graph matching problems. A similarity matrix for graphs is computed in the process. Each entry of this matrix is a result of “domain-dependent” node similarity $\delta(u, v)$ and a measure of the nodes’ topological similarity. The node similarity function measures pairwise similarity of the node’s attributes and the structure of the DAG rooted at that node. The best pairwise node correspondence obtained in this step is used to split the graph in two. The split yields two smaller DAGs to match and the algorithm recursively proceeds in a greedy fashion. Figure 7.4 shows the steps of the algorithm.

Macrini (2003) improved the original DAG matching algorithm of Shokoufandeh et al. (2001). We have extended and modified the publicly available C++ implementation of the algorithm to handle our data. For instance, we extended the base DAG classes with our own fMRI Graph class.
The DAGs are matched on topological structure and node similarity. Therefore, we have defined a new “domain-dependent” node distance function. The node similarity is simply the complement of the node distance. Equation 7.4 gives the definition of the node distance function used in this thesis.
\[
D = C \left[ \frac{d(v,u)^2}{\sigma_r^2} + \frac{(t_{avg}^v - t_{avg}^u)^2}{\sigma_i^2} + \frac{(n_v - n_u)^2}{\sigma_n^2} \right]
\]  
(7.4)

where \(d(i,j), t_{avg}^i, \) and \(n_i\) are the Euclidean distance between nodes \(i\) and \(j\), the average \(t\)-value of node \(i\), and the size of node \(i\), respectively. Constant \(C\) was experimentally set to 1/6 to keep the values of node distance function below 1. The average \(t\)-value is the sum of \(t\)-values divided by the node’s size. Terms are normalized by the variance of distance, average \(t\)-value and size \(\sigma_r^2, \sigma_i^2, \) and \(\sigma_n^2\) computed over all nodes of all graphs in our collection. The normalization gives some assurance that each part of distance function is treated equally.

In case of fMRI graphs, the mass of a node is a sum of \(t\)-values over its voxels. Hence, the node distance function depends on the Euclidean distance and differences in size and correlation (average \(t\)-value). Note that the node distance is zero for identical nodes and greater than zero for distinct nodes.
Chapter 8:

Research Questions and Data

In this Chapter, we discuss the problems in indexing and retrieving fMRI datasets. Although users want to search fMRI collections based on semantics, there is no theoretical foundation for extracting semantics from the fMRI data. One approach is to extract low-level features that correlate with high-level semantics and use them for data retrieval.

In the previous chapter, we discussed three potential heuristics for content-based fMRI data retrieval. Here, we define the specific research questions explored in this thesis. Also, we describe the collection of fMRI data used for testing our research questions.

8.1 Research Questions

fMRI is one of the rapidly growing types of important multimedia data. However, it appears that there are no effective content-based retrieval algorithms to access fMRI
collections. Unfortunately, the methods developed in image retrieval are not easy to adapt to fMRI data.

An alternative way to index into the collection is to assign keywords to each fMRI and do textual search on keywords. However, the manual assignment of keywords is labor intensive and suffers from subjectivity of the human perception. Also, in many cases, we simply may not possess the needed description of fMRI experiments. Hence, it is very important to develop content-based methods for indexing and retrieval of fMRI data.

Based on previous research, it appears that the human brain functionality conforms to two fundamental principles: functional specialization and functional integration. Each brain region specializes in processing specific types of stimuli. Processing a mental task requires interaction of the specialized areas. We propose to represent this as a 4D graph in space and time, where nodes are the specialized areas and edges are interactions between the areas. In this context, the fMRI retrieval problem becomes a graph matching problem, where powerful techniques have been developed.

A review of the graph based methods showed that graphs were successfully applied to 2D and 3D image retrieval. The graph based methods are invariant to rotation and scaling. They are also stable under presence of small noise, which is perfect for noisy fMRI data.

The focus of this research is to explore using graph structures to represent cognitive processes in a brain and to assess their effectiveness for fMRI retrieval. Suppose that we have a collection of fMRI scans of human subjects performing different
cognitive tasks. Given a “query” fMRI, our ultimate goal is to be able to retrieve fMRIs where different subjects perform similar/same tasks.

In the previous chapter, three possible heuristics were proposed for content-based fMRI data retrieval. This thesis will investigate the effectiveness of a “query by example” information retrieval for fMRI data. Our ultimate goal is to retrieve all the “similar” fMRIs in the collection, given a query fMRI. By “similar” we mean fMRIs where subjects performed the same cognitive tasks (e.g. studying pictures of faces, solving moral or emotional dilemmas, etc). Our research questions are:

1) Do the Brodmann areas provide meaningful structures for defining features in fMRI retrieval? This is operationalized by asking specifically: How effective is representation of fMRI data by Brodmann vectors, for information retrieval in fMRI?
2) Can graph structures be used for query-by-example retrieval of fMRI data? This is operationalized by proposing two complete graph-based methods and assessing their performance.

So, we hypothesize that it is reasonable to use Brodmann areas as features for fMRI retrieval. But also, we conjecture that graph structures (perhaps, because they are stable to noise and imperfections in brain motion correction and normalization processes) provide another method for retrieving similar fMRI images.

One could use precision, recall or some combination of precision and recall to measure the performance of IR. We will use ROC curves and precision-recall curves for evaluation of our heuristics.
8.2 fMRI Data

fMRI data produced by The Novel Indexing and Retrieval of Dynamic Brain Images (BRIM) project is utilized in this research. BRIM is a multidisciplinary project being carried out at Rutgers University and funded by National Science Foundation. The principal investigators for the project are Dr. Paul Kantor and Dr. Stephen Hanson. The goal of the BRIM project is to develop cutting edge mathematical and computational models for the indexing and retrieval of fMRI data. Another goal of BRIM is to develop a Napster-like information retrieval system that allows cognitive scientists and physicians to search distributed databases by “query by example” technique (presenting an fMRI sequence as a query and asking to search for similar fMRI sequences).

Some of the search tools developed in the BRIM project are based on the idea of dynamic brain activation. It is assumed that when something happens in the brain, neurons exchange electrical signals leaving behind complex activation patterns that are revealed by fMRI brain imaging technique. However, the time resolution of current imaging techniques does not allow one to see electrical signals traveling across the brain. Therefore, the project also aimed at improving time resolution through EEG interpolation of fMRI data. However, in the work reported here, resolution is “improved” by computing correlations at several lags.

The majority of the fMRI datasets available in the BRIM project have been used for testing and evaluation of heuristics proposed in this thesis. The collection contains over one thousand fMRI time sequences where subjects perform different tasks. Table 8.1
shows a brief description of all available experiments in the collection. The time of repetition is the interval between successive fMRI scans.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Stimuli</th>
<th>Number of subjects</th>
<th>Duration</th>
<th>Number of scans</th>
<th>Time of repetition of scans</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oddball (visual/auditory)</td>
<td>Pictures/words</td>
<td>4</td>
<td>410 sec</td>
<td>205</td>
<td>2 sec</td>
</tr>
<tr>
<td>Event perception (study/house)</td>
<td>Study video/ house video</td>
<td>40</td>
<td>310 sec / 155 sec</td>
<td>210 to 305/110 to 155</td>
<td>1 or 1.5 sec</td>
</tr>
<tr>
<td>Recall (study/try/recall)</td>
<td>Faces, locations, and objects</td>
<td>9 (4 runs)</td>
<td>918 sec / 185 sec</td>
<td>510 or 103</td>
<td>1.8 sec</td>
</tr>
<tr>
<td>Morality (M+, M-, E+, e-)</td>
<td>Moral dilemmas</td>
<td>25 (12 runs)</td>
<td>300 sec</td>
<td>150</td>
<td>2 sec</td>
</tr>
</tbody>
</table>

*Table 8.1: Description of fMRI experiments in the collection.*

### 8.2.1 Oddball Experiments

Subjects in oddball experiments watched a series of stimuli and were asked to respond, by pressing a button, when an “oddball” appeared. An “oddball” is a stimulus unlike the others in the series. Oddball experiments are two types: visual and auditory. In a visual
session, the oddball was a picture of George Bush (oddball) and non-targets were pictures of circles. The stimulus sequence consisted of a series of fixation screens (a cross) and stimulus screens. The sequence ran: fixation-stimulus-fixation-stimulus etc. The location of oddballs in the sequence was not periodic.

In an auditory session, the stimulus was auditory, not visual. The oddball was a low pitched tone and the non-targets were high pitched tones. For the auditory sequence, a pause without sound preceded each stimulus. The sequence ran: pause-stimulus-pause-stimulus etc.

The stimulus sequence was the same for subjects across the session type. However, the sequences for visual and auditory sessions were different. In both visual and auditory sessions, subjects were presented a sequence of two hundred stimulations (non-targets or oddball). The stimulus duration was 1.01 sec. and the fixation/pause duration was 1.01 sec. Each experiment lasted for 410 seconds and produced 205 fMRI images, acquired every two seconds (the time of repetition – TR).

8.2.2 Event Perception Experiments

In the event perception experiments, subjects watched two videos: a “study video” and a “house video”. The “study video” shows a man studying in a room and the “house video” shows a geometrical object moving on the screen. The study and house videos were 310 seconds and 155 seconds long respectively. Depending on the time of repetition used for each subject, the study video condition had a number of scans varying from 210 to 305 and the house video condition had a number of scans varying from 110 to 155. Each
video was shown twice to each subject, representing two different modes. In the “active” mode, subjects watched videos and were asked to press a button any time they think that “an event has happened”. The definition of events was very general, such as closing a book, the object bouncing from the border of the screen, etc. In the “passive” mode, subjects simply watched the video without taking any actions. There were 40 subjects in this experiment.

8.2.3 Recall Experiments

During the recall study, each subject was scanned in four different sessions/runs. The first three runs each had two distinct phases. In the first phase, subjects studied pictures of faces, locations and geometrical objects (the “study” phase). These stimuli (conditions) were coded as “study faces”, “study locations”, and “study objects”. In the second phase, subjects were asked to try to recall either faces, or locations, or objects (the “try” phase). Subjects verbally communicated the recalled images. The conditions in this phase were coded as “try faces”, “try locations”, and “try objects”. The fourth run had only one phase called the “recall” phase. In this phase, subjects were asked to do free recall of pictures from all three categories (faces, locations, and objects) in any order. We coded these conditions as “recall faces”, “recall locations”, and “recall objects”. The repetition time for all runs was 1.8 seconds. The first three runs had 510 volumes (brain images) and the fourth run had 103 volumes. The experimental duration was 408 seconds for the first three runs and 185 seconds for the fourth run. Nine subjects participated in the experiment. There are 9 (subjects) x 4 (runs) = 36 fMRI datasets in the recall study.
8.2.4 Morality Experiments

In the morality study, subjects were presented with a series of practical dilemmas (Greene et al., 2004). There were 60 dilemmas in the pool. The exact description of dilemmas is available online at http://www.neuron.org/cgi/content/full/44/2/389/DC1. These dilemmas were divided into “moral” and “nonmoral” categories based on the responses of pilot participants. The moral dilemmas were coded as M+ and nonmoral as M-. Two independent coders evaluated dilemmas using criteria intended to capture the difference between the intuitively “up close and personal” (more emotional) and the intuitively less personal (less emotional). The personal/emotional condition was coded as E+ and impersonal/nonemotional condition as e-. Hence, the dilemma coded as M+E+ represents a moral personal dilemma (condition).

The dilemmas were presented on a visual display projected into the scanner in a series of 12 runs of five trials each. Each dilemma was presented as text in a series of three screens. The first two screens described the scenario and the last posed a question about the appropriateness of an action one might perform in that scenario (e.g. pushing a person off the bridge). Subjects read at their own pace, pressing a button to advance from the first screen to the second, and from the second to the third screen. After reading the last screen, subjects were asked to respond by pressing one of two buttons, “appropriate” or “inappropriate”. Subjects were limited to a maximum of 46 seconds to read all three screens and respond.

The intertrial interval (ITI) lasted for a minimum of 14 seconds (seven images) in each trial. This is intended to allow the hemodynamic response to return to the baseline
after each trial. During the ITI, subjects viewed a fixation cross. Stimuli were presented and behavioral responses collected using stimulus presentation software.

We had to remove data from some of runs from the morality experiment because they lacked either fMRI data or stimulus information. However, we were able to include a large portion of the morality experiments in the collection. In total, we included 25 morality subjects with 12 runs each. The total number of fMRI datasets for the morality data was $25 \times 12 = 300$. Hence, our collection of test data included 380 fMRI time series.
Chapter 9:

Experimental Design

We employed quantitative methods to conduct the Information Retrieval (IR) experiments and to investigate the performance of fMRI matching heuristics. The statistical methods employed in processing the data included the General Linear Model and linear regression. For the work on IR, signal detection theory and receiver operating characteristic (ROC) curves (Egan, 1975) were used to investigate the detection rate and false alarm rate for different heuristics.

In the following sections, we discuss the general idea of the experimental design and define the variables used in the analyses.

9.1 Precision and Recall

Many IR experiments base their evaluations on recall and precision. Recall is defined as the ratio of relevant objects retrieved to the total number of relevant objects in the collection:
Recall = \frac{\text{Number of relevant documents retrieved}}{\text{Number of relevant documents in collection}} \quad (9.1)

In case of ranked outputs, the number of relevant documents retrieved must be modified to mean the number of relevant documents retrieved before the cutoff point. If the cutoff point be \( n \), the number of relevant documents retrieved \( g \) is a function of \( n \), that is \( g(n) \). This is a monotonically increasing function of \( n \). Thus, the recall at cut-off point \( n \) is

\[
\text{Recall}(n) = \frac{g(n)}{G} \quad (9.2)
\]

where \( G \) is the number of relevant documents in the collection.

Precision is defined as the ratio between the number of relevant documents retrieved and the total number of documents retrieved:

\[
\text{Precision} = \frac{\text{Number of relevant documents retrieved}}{\text{Number of documents retrieved}} \quad (9.3)
\]

Similar to the recall, the number of relevant documents can be changed to the number of relevant documents before the cutoff point for ranked outputs. Hence, the precision becomes a function \( g(n)/n \), of the cutoff point at \( n \) documents retrieved. That is, precision at cut-off point \( n \) is

\[
\text{Precision}(n) = \frac{g(n)}{n} \quad (9.4)
\]

From the definitions of precision and recall, it is clear that both of these measures do not distinguish between a system that returns relevant documents at the top of the returned set and a system that retrieves relevant documents at the bottom of the returned
set. Indeed, the values of precision and recall for such systems are identical. However, a user would favor a system that returns relevant documents at the top of the list.

One could use different cut-off points for different conditions to address this problem. For example, we could choose the cut-off point equal to the number of relevant documents, the so-called $R$-precision.

![Typical precision-recall curve](image)

*Figure 9.1: Typical Precision-Recall curve.*

The values of recall and precision obtained by changing cut-off points can be used to plot precision as a function of recall, the so-called precision-recall curve. This curve captures the trade-off between the precision and recall. For every value of recall it gives the corresponding value of precision. Each point on the precision-recall curve corresponds to a distinct cut-off point $n$. That is, for every point on the curve, its coordinates are: $x = \text{Recall} (n)$ and $y = \text{Precision} (n)$. 
Figure 9.1 shows the typical precision-recall curve with classical concave shape. It shows the trade-off between precision and recall. Trying to increase recall, usually introduces more non-relevant documents into the returned list, thus, reducing the precision. And the wise versa, trying to increase precision typically reduces recall by removing relevant documents from the returned list. The ultimate goal is to improve both precision and recall, i.e. move the curve up and to the right. The area under the precision-recall curve gives the measure of performance called *average precision*. Average precision is defined as

\[
P_{\text{avg}} = \frac{1}{R} \sum_{i=1}^{n} \text{Precision}(i) \cdot r(i) \tag{9.5}
\]

where \(n\) is the number of documents in the returned list, \(R\) is the total number of relevant documents in the collection and function \(r(i)\) is equal to 1 if the \(i\)-th document in the list is relevant, and equal to 0 otherwise. In other words, the average precision is the sum of precisions at each relevant document in the returned list divided by the number of relevant documents in the collection.

The higher values of average precision mean better performance. However, this measure is not intuitive to interpret.

### 9.2 Activation Maps

The FMRIB Software Library (FSL) (Smith et al., 2001) is a software package that can compute activation maps for fMRI data using the General Linear Model. Activation maps are also called \(t\)-maps because they give \(t\)-statistics values for each voxel of the fMRI image. For example, Figure 9.1 shows the \(t\)-map computed by FSL for one subject of the
Recall experiment. Red areas are brain regions with statistically significant response to the stimulus. That is, these areas have strong correlation with the stimulus time series. In this case, the stimulus was studying pictures of faces.

Figure 9.2: t-map for one Recall experiment subject. Red areas represent brain regions with statistically significant activations.

By default, FSL reports voxels as statistically significant or activated if their correlation with the stimulus is so great that uncorrelated (pure noise) data would reach this level only 5% of the time, adjusted for the number of voxels. The correction step is necessary because fMRI images have a large number of voxels (around $10^5$). Thus, even with the probability of error $p=0.05$, there will be $0.05 \times 10^5 = 5000$ voxels activated by chance. There are two accepted ways to adjust the probability: the Bonferroni correction and the cluster-level correction. The Bonferroni correction states that the probability $p$ divided by the number of voxels gives the corrected probability. The corrected probability is used subsequently to threshold significantly correlated voxels. The cluster-level correction adjusts the probability for each cluster separately, depending on the
number of voxels in the cluster (Frackowiak et al., 2004). The Bonferroni correction is stricter than the cluster-level correction. Note that FSL uses the cluster-level correction.

The activation image ($t$-map) is a 3-D volume where the value of each voxel is its $t$-value (a measure of the strength of its correlation with the stimulus). The correlation is between the voxel time series and the stimulus time series. The 3-D image of a $t$-map is presented here in the 2-D plane as a sequence of slices from bottom to top, reading right to left (bottom of the brain is at the top left corner).

The FSL software also groups activated voxels into contiguous clusters (if they touch at a face, edge or corner) and can compute transformation matrices for normalizing brains. Under normalization a brain image is mapped by rotation, scaling, translation, etc. to fit a canonical standard brain, such as the Montreal Neurological Institute (MNI) template brain. This is needed to compare activations across subjects, since individual brains have somewhat different shapes and sizes.

The transformation matrix is a 4x4 matrix where the rotation matrix is in rows and columns 0, 1, and 2, the translation vector is in the right column and the bottom row is always 0,0,0,1.

$$
\begin{pmatrix}
x' \\
y' \\
z' \\
1
\end{pmatrix} = \begin{pmatrix}
r_{00} & r_{01} & r_{02} & t_0 \\
r_{10} & r_{11} & r_{12} & t_1 \\
r_{20} & r_{21} & r_{22} & t_2 \\
0 & 0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
x \\
y \\
z \\
1
\end{pmatrix}
$$

(9.3)

For example, Equation 9.3 shows how to use the transformation matrix to transform a point $(x, y, z)$ into a new point in normalized space.

We employed FSL to compute $t$-maps and lagged $t$-maps for 100, 200, 300, 400 and 500 millisecond delays for every subject, condition and run in our fMRI repository.
We limited lags to 500 milliseconds due to computational constraints. The computations took a month to run continuously on three Pentium 4, 1.8 GHz, and 1.5 GB RAM Linux boxes for the data in the collection.

Features such as Brodmann vectors and nodes of graphs were extracted from the FSL generated t-maps and lagged t-maps. We will discuss this in more detail in the next three chapters. Graphs were normalized to the standard MNI brain using Equation 9.3 prior to the matching. As we mentioned before, the relevance of the image in the retrieval experiments was based on the stimulus (condition). That is, the retrieved fMRI was judged relevant if the stimulus was the same for both the query fMRI and the retrieved fMRI. For example, if the query had “Study Face” condition, all images in the collection with “Study Face” condition were considered relevant.
Chapter 10:

Content-Based fMRI Retrieval Using

Brodmann Vectors

In this Chapter, we report the results of the retrieval experiments that we conducted using the Brodmann vector approach. This Chapter has the following structure. First, we describe the procedure we used for extracting Brodmann vectors from the collection of fMRI data in Section 10.1. Then, we discuss the retrieval experiments and the results in terms of the hit matrices in Section 10.2. The performance of a system can be characterized by its ROC curve and its area. We define the ROC curve and discuss its properties in Section 10.3. Also, in the same section, we discuss the retrieval performance of the heuristic in terms of the ROC curves and precision-recall curves.

10.1 Conceptual Framework for Matching Brodmann Vectors

From the repository of fMRI data, we have generated 986 Brodmann vectors by the following procedures:
1) For each condition of the experiment, we obtained a \( t \)-map using the FSL package. For example, each “recall” experiment had 9 conditions: three “study” conditions (study face, study location, and study object), three “try to recall” conditions (try face, location, and object), and three “free recall” conditions (recall face, location and object).

2) We repeated this process for each experiment, subject and run. That is, “recall” experiments had 9 subjects and each subject had 3 runs for the study and the try conditions and 1 run for the recall condition. Thus, we generated (9 subjects × 3 runs × 6 conditions + 9 subjects × 1 run × 3 conditions) = 189 \( t \)-maps for recall experiments.

3) FSL \( t \)-maps were thresholded so that only 1% of the voxels in the standard MNI brain have \( t \)-values above the threshold. We consider these voxels as activated. The purpose of thresholding in this way is that the number of activated voxels extracted from each \( t \)-map is the same. The fraction 1% was suggested by the project director.

4) For each \( t \)-map, we computed the overlaps of activated voxels with Brodmann areas, to obtain components of the Brodmann vectors (See Section 7.1). Thus, we built 986 Brodmann vectors, one for each \( t \)-map in the repository.

One may hypothesize that Brodmann areas are related to brain functionality. Based on this idea we applied Brodmann vectors for information retrieval (IR) in fMRI. We have computed the cosine of the angle between Brodmann vectors, as the measure of their similarity. We used the similarity scores to conduct leave-one-out retrieval
experiments. The results of these experiments were visualized using ROC curves discussed later.

Figure 10.1: Hit matrix for the recall data. Subjects are ordered along the X and Y axes right to left, and from top to bottom. Similarity to color mapping is at the top.

10.2 Results of the Brodmann Vector Matching

The Brodmann vector matching and retrieval of fMRI data is based on the selected similarity measure, the cosine of the angle between vectors. To the retrieval, we have computed pairwise similarities for all 986 Brodmann vectors. In other words, we have
computed a 986×986 symmetric similarity matrix, in which each entry is the similarity of corresponding vectors. We call this matrix the “hit matrix”.

The hit matrix is very easy to display visually. Figure 10.1 shows the hit matrix for Brodmann vectors of the “recall” data. The vectors are ordered along the $X/Y$ axes, from right to left (ordered by stimulus, then subject and then run), and from top to bottom. Pixel intensity in the hit matrix represents the similarity value for the corresponding pair of experiments (labels are not shown on the image). The more similar are the Brodmann vectors for the experiments, the brighter the pixel intensity. From Figure 10.1, it appears that the study conditions, in which subjects studied pictures of faces, locations and objects, form a pretty bright square. This means that Brodmann vectors for the study conditions are quite similar to each other in terms of our similarity measure, cosine of the angle between vectors.

In recall conditions, subjects were asked to freely recall pictures one of faces, locations or objects that they remember. It appears that recall conditions also form a visible square, although it is not as bright as for the study conditions. Note that in the hit matrix, pixels on a diagonal from bottom left corner to top right corner represent matching a Brodmann vector with itself, and are always equal to one (bright line along this diagonal).

There were 189 $t$-maps for the recall data in our repository. We generated 736 $t$-maps for the morality data. Figure 10.2 gives the hit sub-matrix for the morality data.

Experiments are ordered along $X/Y$ axis, from right to left (ordered by stimulus, subject and run), and from top to bottom. It appears that there is a visible pattern of evenly spaced bright squares in the hit matrix. However, this pattern is not as easy to
interpret as the one for the recall data. Certainly, we would like to see large boxes of bright areas as in the previous hit matrix. However, the hit matrix for the morality data has off-diagonal similarities that we did not see in the other matrix. This might indicate that there is some interesting property of the data that Brodmann vectors have a problem separating. We have since discovered that these bright off-diagonal lines represent within subject similarity for this task.

Figure 10.2: Hit matrix for the morality data. Subjects are ordered along the X and Y axes, right to left, and from top to bottom. There is a visible pattern that shows some experimental conditions are similar.
Overall, it appears that the Brodmann vector is not the best choice for categorizing the morality data. We note that morality experiment is far more complex in its design than recall experiment. In morality experiment, subjects are asked to resolve difficult personal and emotional moral dilemmas. In contrast, subjects simply studied and recalled pictures in the recall experiment.

![Hit matrix for the event perception data](image)

*Figure 10.3: Hit matrix for the event perception data. Subjects are ordered along the X/Y axes, right to left, and from top to bottom.*

Next we consider the event perception data. There were 53 event perception experiments in the repository. The design of this experiment is more complex than for recall but less complex than for morality. During this experiment, subjects watched a video of either a student studying in the classroom (Study video) or a colored ball bouncing around on the screen (House video). They were asked to press a button
whenever they thought “a new event” happened. Figure 10.3 gives the hit matrix for the event perception data. It appears that there is no visible pattern. Quantitative tests confirm that Brodmann vectors performed worst on the event perception data.

![Hit matrix for the event perception data.](image)

**Figure 10.3:** Hit matrix for the event perception data. It appears that there is no visible pattern. Quantitative tests confirm that Brodmann vectors performed worst on the event perception data.

![Hit matrix for the entire repository.](image)

**Figure 10.4:** Hit matrix for the entire repository. Similarities are represented by color intensities (yellow – high similarity, blue – low similarity).

An interesting question is how these experiments compare to each other. We computed pairwise similarity for all 986 experiments (Brodmann vectors) in the repository. Figure 10.4 gives the hit matrix for all of the experiments together.

![Hit matrix for the entire repository.](image)
Experiments are ordered along the $X$ and $Y$ axes, right to left (ordered by the stimulus, then subject, and then run), and from top to bottom. It appears that Brodmann vectors perform well for recall experiments. There is also the noted interesting pattern of off-diagonal similarities for event perception. Finally, Brodmann vectors performed poorly on the event perception data.

The hit matrix is good to visually inspect data for observing interesting patterns and properties. However, it does not give a good quantitative measure of the performance. In the next section, we will try to establish a quantitative measure of the retrieval performance for the Brodmann vectors heuristic.

10.3 Using ROC Curves to Represent Retrieval Performance

The above results can be better summarized by using Receiver Operating Characteristic (ROC) curves (Egan, 1975). The ROC curve shows the chance of correctly “detecting” right condition, as a function of the probability of false alarms. Next we give a brief explanation of the ROC curve and two of its basic concepts, detection rate and false alarm rate, following Ng (1999).

Suppose that we have a simple burglar alarm made of two cans near the entrance door, with one can on the top of the other. Whenever, a thief breaks in, the two cans will fall, so the burglar sets off the alarm. However, sometimes it may not work, for example, when a patient thief pushes the door very slowly. Suppose it works only eight times out of ten in average. Also, sometimes when there are trucks passing by the road in front of the house, the two cans may also fall. Hence, the trucks may set off the alarm as well.
Let’s say this happens three times out of ten in average. According to the terminology of receiver operating characteristic, the *detection rate* of this system is 80%, while the *false alarm rate* is 30%.

If we start using three cans instead of two, it will be more difficult for a thief to break in without setting off the alarm. Suppose the alarm works on nine of ten break-ins. However, when a truck passes by it will set off the alarm seven times out of ten. Thus, the detection rate of the system rises from 80% to 90%, but the false alarm rate rises from 30% to 70%.

The more cans we use, the higher is the detection rate as well as the false alarm rate. For instance, if we balance five cans instead of three, it is almost impossible for a thief to break in without setting off the alarm. However, whenever a truck passes by, it will also set off the alarm without fail. In other words, the detection rate is 100% and the false alarm rate is also 100%. On the contrary, if there are no cans near the door, the detection rate is 0 and so is the false alarm rate.

If we plot all the pairs of the detection and false alarm rates on a graph, where detection rate is on the vertical axis and false alarm rate is on the horizontal axis, we will get a curve, the detection rate as a function of the false alarm rate. For good alarms, the curve is concave because it is easier for a thief to set off the alarm than for a truck to set off the alarm in the beginning, and it is easier for a truck to set off the alarm than a thief, at the end.

Translating the above metaphor to our situation, the burglar alarm is the retrieval function, based on Brodmann vectors heuristic in this case, which seeks to decide whether an experimental condition is “similar” to the query condition or not. When the
function correctly retrieves the same condition, we consider it as detection. On the contrary, when the function retrieves a different condition, we consider it as a false alarm.

The area under the ROC curve has a special meaning. The performance of a system can be approximately characterized by the ROC curve area. The area under the ROC curve of the ideal system is 1. Hence, the closer the ROC area is to 1, the closer the system’s performance to the ideal. On the contrary, the closer the area is to 0, the closer is its performance to the absolute worst. The diagonal line from the origin to point (1, 1) represents random guessing. In other words, this is the ROC curve of a system that randomly selects each object in the collection as similar, or non-similar with equal probability. Therefore, the detection rate and the false alarm rate for this system are always equal. The ROC area of a system doing random guessing is equal to 0.5.

We have conducted information retrieval experiments on the fMRI data using the similarity matrix computed above. The leave-one-out scheme was used for IR. The idea of the leave-one-out scheme in IR is to remove one object from the set and use it as a query. Hence, we removed each experiment (of 986 in our collection) and used it as a query to retrieve “similar” experiments from the set of remaining 985 experiments. We assume as the definition of “truth” that “similar” means those where subjects perform the same type of cognitive task. Effectively, the meaning of similar is equivalent to “same condition” in our case. For instance, let the query dataset be one where a subject studies pictures of faces. We want to retrieve every experiment in our repository where a subject studies pictures of faces. In other words, we want to retrieve every experiment with the “study face” condition/stimulus.
For each query, datasets in the repository were ranked in the decreasing order of their similarity scores. That is, experiments were ranked in the decreasing order of the cosine of the angle between Brodmann vector corresponding to the dataset and the Brodmann vector of the query dataset. Thus, experiments ranked high in the list are similar to the query in terms of their Brodmann vectors.

**ROC Curves for Brodmann Vectors**

![ROC Curves for Brodmann Vectors](image)

*Figure 10.5: ROC curve for three queries from the top, the middle and the bottom of the ranked ROC areas list.*

We repeated the leave-one-out procedure for every dataset in the repository. That is, in turn, we removed each dataset from the repository and used it as a query for retrieval. The ROC curve and the area under the ROC curve were computed for each query. This was used to build a list of ROC curve areas, ranked in decreasing order, for all 986 queries. The performance for a particular query can be examined looking at the
corresponding ROC curve. For example, Figure 10.5 shows the ROC curves for three queries, from the top, middle and bottom of the list of ROC curve areas.

![Precision versus Recall Curve for Brodmann Vectors](image)

*Figure 10.6: Precision plotted as a function of recall for Brodmann vectors.*

Both the detection rate and the false alarm rate vary from 0 to 1, where 1 means 100%. Next to the curves are legends that tell the subject name, condition and run number. The curve with the red colored triangles is for the subject coded as SUBJ10 studying pictures of faces in his run 1. The blue diamonds represent the subject LM210703 making a decision about a moral and emotional dilemma at his run 9. Finally, the curve with the green triangles is for the subject az_29 watching a video of a person studying in a classroom at his run 1.
Precision-recall curves give another way to visually assess the performance of the queries. For example, Figure 10.6 plots the precision as a function of recall for the same three queries as in Figure 10.5. Clearly, it shows that the Brodmann vectors heuristic performed well for the curve with red colored crosses, but performed poorly for curves with blue diamonds and green triangles. One could make the same conclusions based on the ROC curves. The advantage of the ROC curve over the precision-recall curve is that the area under the ROC curve characterizes the performance of the system. The area under the precision-recall curve has no clear meaning.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Subject, Run</th>
<th>ROC area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study Faces</td>
<td>SUBJ10, run1</td>
<td>0.981</td>
</tr>
<tr>
<td>Study Locations</td>
<td>SUBJ4, run2</td>
<td>0.978</td>
</tr>
<tr>
<td>Study Locations</td>
<td>SUBJ7, run3</td>
<td>0.975</td>
</tr>
<tr>
<td>Study Locations</td>
<td>SUBJ7, run1</td>
<td>0.974</td>
</tr>
<tr>
<td>Study Faces</td>
<td>SUBJ10, run3</td>
<td>0.972</td>
</tr>
<tr>
<td>Study Faces</td>
<td>SUBJ13, run3</td>
<td>0.971</td>
</tr>
<tr>
<td>Study Locations</td>
<td>SUBJ17, run2</td>
<td>0.970</td>
</tr>
<tr>
<td>Study Faces</td>
<td>SUBJ17, run2</td>
<td>0.969</td>
</tr>
<tr>
<td>Study Locations</td>
<td>SUBJ8, run3</td>
<td>0.969</td>
</tr>
<tr>
<td>Study Faces</td>
<td>SUBJ13, run2</td>
<td>0.968</td>
</tr>
<tr>
<td>Study Faces</td>
<td>SUBJ8, run2</td>
<td>0.964</td>
</tr>
<tr>
<td>Study Locations</td>
<td>SUBJ4, run3</td>
<td>0.961</td>
</tr>
<tr>
<td>Study Locations</td>
<td>SUBJ8, run1</td>
<td>0.959</td>
</tr>
<tr>
<td>Study Locations</td>
<td>SUBJ4, run1</td>
<td>0.957</td>
</tr>
<tr>
<td>Study Faces</td>
<td>SUBJ4, run3</td>
<td>0.955</td>
</tr>
</tbody>
</table>

*Table 10.1: ROC curve areas ranked in decreasing order. Only the top 15 queries are shown.*

The steeper the ROC curve, the larger is the area under it, and the better is the retrieval performance. Indeed, a steeper ROC curve means that the detection rate is high for low false alarm rates.
We want the performance of our Brodmann vectors heuristic to be better than random guessing. In terms of ROC curves, this translates to an area under ROC curve for nearly every query being greater than the area of ROC curve for random guessing. As noted before, the ROC curve area of the random guessing is equal to 0.5. Table 10.1 gives the 15 queries with the largest areas under the ROC curve.

It appears that they all have very high values of the ROC curve area. This indicates that our Brodmann-vector-based matching heuristic performed very well for these queries. However, it is necessary to look at how the heuristic performed for all
queries. Figure 10.7 gives the histogram of the frequency distribution for areas under ROC curve for 986 queries.

No good retrieval system should have its value of area under the ROC curve below 0.5, random guessing. It appears, from Figure 10.7, that Brodmann vectors based heuristic performed better than random guessing for a substantial majority of queries. In fact, it performed better than 0.5 in 835 cases out of 986, which is 85% of all queries.
Chapter 11:

Content-Based fMRI Dataset Retrieval Using

Minimum Distance Graphs

In this Chapter, we report the retrieval experiments that we conducted using minimum distance graphs (MDG). First, we describe the procedure we used for extracting MDGs from the collection in Section 11.1. Next, in the Section 11.2, we report the retrieval experiments and discuss the retrieval performance in terms of the hit matrices. Finally, in Section 11.3, we compute ROC curves and precision-recall curves. Also, Section 11.3 discusses the performance of the MDG heuristic in terms of the ROC curve and its area.

11.1 Conceptual Framework for Using Minimum Distance Graphs

We used the fMRI data in our collection to generate 928 minimum distance graphs. The number of graphs was lower than the number of Brodmann vectors due to the difference in selecting activated voxels. We thresholded voxels based on the probability of error $p=0.05$ (See Section 9.2) for the Minimum Distance Graph (MDG). Some of the $t$-maps had no voxels with a correlation above the threshold. Hence, we could not build valid
graphs for these $t$-maps. Note that previously we used 1% of the standard brain voxels as the threshold for Brodmann vectors, which guaranteed that every $t$-map had an equal number of thresholded voxels. The following are the steps of the procedure to generate Minimum Distance Graph:

1) For each condition of the experiment, we employed FSL to generate a correlation map ($t$-map). Furthermore, we ran FSL in batch mode to generate lagged correlation maps ($t$-maps) at the following time lags: 100, 200, 300, 400, and 500 milliseconds.

2) We repeatedly computed $t$-maps and lagged $t$-maps for each experimental condition, subject and run. At this step, we generated a total of 5,568 $t$-maps because each experiment, subject, condition and run had exactly 6 $t$-maps (basic $t$-map and 5 lagged $t$-maps). This is the reason why we had to limit our computations to 500 millisecond lags. Even this number of $t$-maps took us a month to compute continuously using three Pentium 4 class computers.

3) By default, FSL reports voxels as statistically significant if their correlation with a stimulus is so great that uncorrelated data would reach this level only 5% of the time, adjusted for the number of voxels (See Section 9.2). We extracted those voxels reported significantly activated, using FSL at this default probability, from each $t$-map, including the lagged $t$-maps. These voxels were grouped into contiguous clusters of activation using FSL.

4) We refer to a basic $t$-map as time lag 0 and lagged $t$-maps at 100, 200, 300, 400, 500 milliseconds as time lag 1, 2, 3, 4, and 5 respectively. Thus, for each subject, condition and run we have a sequence of $t$-maps (volumes) at lags 0, 1, 2, 3, 4,
and 5. For every sequence of \( t \)-maps, we computed the centers of mass of the contiguous clusters (Equation 7.2) in its volumes. Thus, we reduced every fMRI dataset in the collection to a set of nodes, where the centers of mass represented nodes.

5) The Minimum Euclidean Distance heuristic, described in Section 7.2, was used to build MDGs from these sets of nodes.

Figure 11.1: Hit matrix for the recall data using MDG matching. It appears there is no visible pattern. Mapping from similarity to color is at the top.
Our hypothesis was that signals in a brain propagate with a finite speed. Therefore, we imagine that brain areas that are activated are more likely to cause other nearby areas to be activated. For instance, imagine that a stimulus activates visual cortex and the activation spreads out from there to other areas. Our hypothesis is that we might try to trace this spread of activation in our Minimum Distance Graph. Thus, we hypothesized that we could reduce fMRI data to a MDG and use it for content-based fMRI retrieval.

### 11.2 Results of the MDG Based fMRI Matching

Following the procedure in Section 11.1, we have computed 928 MDGs for available fMRI data. The main idea here is to reduce fMRI to its MDG representation and do graph matching on MDGs. We used the improved version (Macrini, 2003) of the DAG matching algorithm developed by Shokoufandeh et al. (2001) for matching MDGs extracted from fMRI data. The code was modified to handle fMRI data. Also, we introduced the new domain-dependant node distance function defined in Equation 7.4.

We used the DAG matching algorithm to compute pairwise similarities for each pair of graphs in a collection of 928 MDGs extracted from fMRI data. In other words, we computed a 928×928 hit matrix, in which each entry is the similarity score of corresponding MDGs.

Figure 11.1 shows the hit matrix for MDG matching applied to recall data. Experiments are ordered along X and Y axis, from right to left, top to bottom. The
similarity-to-color mapping is shown at the right top, blue – low similarity and yellow – high similarity.

It appears that there are no visible patterns in the data. This is different from the Brodmann vectors based matching. Brodmann vectors were successful at separating different conditions of the recall data, especially the study conditions.

Figure 11.2: Hit matrix for the morality data using MDG matching. It appears there is no visible pattern. Color mapping is at the top.
Let us now take a look at the hit matrix for morality data. As we remember from the previous chapter, the Brodmann vector heuristic was not able to separate distinct conditions for this data.

Figure 11.2 gives the hit matrix for running the MDG heuristic on morality data. Subjects are ordered along X and Y axis, right to left, top to bottom. The similarity mapping is embedded in the picture. It appears that this heuristic is also not able to separate distinct conditions.

![Hit matrix for the event perception data using MDG matching.](image)

Figure 11.3: Hit matrix for the event perception data using MDG matching. Again, there is no visible pattern.

Finally, we will look at the results of MDG matching for event perception data. The event perception experiments are less demanding, in terms of cognitive processing, than morality experiment but require more effort than simple recall experiments.
Figure 11.4: Hit matrix for all experiments using MDG matching. It appears that morality data is strongly separated from other experiments.

However, note that the recall and event perception designs are somewhat similar because they both require visual processing. Remember that subjects studied pictures in recall experiments and watched video in event perception.

Figure 11.3 gives the hit matrix for MDG matching for the event perception data. Brodmann vectors performed very poorly on this data. It appears that MDG matching is no different, at least visually.
However, it is more interesting to see how results for different conditions compare with each other. Figure 11.4 gives the hit matrix for MDG matching applied to the entire collection. Experiments are ordered along $X$ and $Y$ axis, right to left, top to bottom. Similarity to color mapping is show at the top.

The most striking thing is that it appears that the morality data are strongly separated from other experiments. This means that MDGs generated from the morality data are much more similar to each other than to graphs of the other experiments.

Figure 11.4 also shows that MDGs for the recall and the event perception data are quite similar to each other. Perhaps, this is explained by the fact that both recall and event perception data require intensive processing in visual cortex. The morality data is different since it involves brain areas responsible for emotional and analytical decision making. We were not able to achieve this level of qualitative and functionally plausible data separation using Brodmann vectors.

The hit matrices are useful for visually inspecting data for interesting patterns and properties. Now we turn to quantitative measures of performance. In the next section, we will look at ROC curves and other measures of performance.

### 11.3 ROC Curves for MDG Matching

The hit matrix for MDGs was used to conduct query-by-example IR experiments. We used the leave-one-out design. The idea of leave-one-out design is to remove one object from the collection and use it as a query.
The minimum distance heuristic builds an MDG from a series of \( t \)-maps (\( t \)-maps at time lags 0, 1…5) that are computed for a particular subject being exposed to a stimulus. Hence, there is a direct mapping between that MDG and the fMRI for that subject.

Following the design, we removed each MDG from the 928 in the collection and used it as a query to retrieve “similar” MDGs from the set of remaining 927 graphs. Since there is a direct mapping between MDGs and fMRI files, this is the same as searching for fMRIs that are similar to an fMRI of a graph used as a query. In other words, given a query fMRI we want to retrieve “similar” fMRIs in the repository. “Similar” means those images where subjects perform the same type of cognitive task. Note that the meaning of similar is equivalent to the same condition in our case. For example, suppose that a query fMRI is the one where a subject studies pictures of faces. Hence, our goal is to retrieve every fMRI in the repository where a subject studies pictures of faces. In short, we want to retrieve every fMRI experiment with the study face condition/stimulus.

For each query, we constructed a list that included the fMRI experiment’s name and its similarity to the query. The similarity scores were computed using the graph matching algorithm (Shokoufandeh et al., 2001). The list was ranked in the decreasing order of the similarity scores. This is the same as the ranked list of retrieved fMRIs, because it can be cut-off at any desired point.

We repeated the leave-one-out retrieval for each data set in the collection. That is, in turn, we removed each data set from the collection and used it as a query. Thus, we generated 928 ranked lists, one for each query. The ROC curve and the area under the ROC curve were computed for each query. The performance for a particular query can be
examined looking at the corresponding ROC curve. Figure 11.5 gives the ROC curves for the three MDG queries, with the top, the median and the worst performance.

![ROC curves for MDG matching](image)

*Figure 11.5: ROC curves for three queries with the top, the median and the worst performances.*

The red colored cross is for run 1 of the subject az_21 watching a video of a ball bouncing on the screen. This condition is coded as “house video” in the event perception experiment. The MDG heuristics performed very well for this query. This means that it was able to detect and retrieve other fMRIs where a subject had the same condition (house video). The area under the ROC curve was 0.973 for this query, very close to the ideal.

The MDG performed not so well for other conditions. The blue colored diamond is for the subject MD290903, making decisions on a moral but not emotional dilemma at run 6. The subject SUBJ17 in the recall experiment was studying pictures of locations at
run 3. The ROC curve areas were 0.639 and 0.389 for subjects SUBJ17 and MD290903 respectively.

**Precision versus Recall curve for MDG matching**

![Precision versus Recall curve for MDG matching](image)

*Figure 11.6: Precision as a function of recall for MDG matching.*

Let us take a look at the precision-recall curve for the same three queries. Figure 11.6 shows precision as a function of recall. Clearly, it confirms our conclusions that MDG performed better the query represented by red crosses than for the other two curves. It appears that the similar datasets were retrieved early in the returned list for the query represented by red crosses. MDG performed better for blue diamonds query than for the green triangles query, because majority of the similar fMRI datasets were at the top of the returned list for blue diamonds.
We want the performance of our MDG heuristic to be better than random guessing. In terms of ROC curves, this means that area under ROC curve for every query should be greater than 0.5. We computed the ROC areas for all queries and built a big list of ROC areas, ranked in decreasing order. Table 11.1 gives ROC areas for the 15 queries with the largest areas under the ROC curve for the MDG based retrieval.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Subject, Run</th>
<th>ROC area</th>
</tr>
</thead>
<tbody>
<tr>
<td>House video</td>
<td>az_21, run1</td>
<td>0.973</td>
</tr>
<tr>
<td>House video</td>
<td>az_30, run1</td>
<td>0.962</td>
</tr>
<tr>
<td>House video</td>
<td>az_22, run1</td>
<td>0.958</td>
</tr>
<tr>
<td>House video</td>
<td>az_31, run1</td>
<td>0.955</td>
</tr>
<tr>
<td>House video</td>
<td>az_40, run1</td>
<td>0.942</td>
</tr>
<tr>
<td>Recall Locations</td>
<td>SUBJ8, run4</td>
<td>0.936</td>
</tr>
<tr>
<td>Study video</td>
<td>az_19, run1</td>
<td>0.933</td>
</tr>
<tr>
<td>House video</td>
<td>az_33, run1</td>
<td>0.932</td>
</tr>
<tr>
<td>Study Locations</td>
<td>SUBJ8, run2</td>
<td>0.924</td>
</tr>
<tr>
<td>House video</td>
<td>az_20, run1</td>
<td>0.923</td>
</tr>
<tr>
<td>House video</td>
<td>az_19, run1</td>
<td>0.921</td>
</tr>
<tr>
<td>Study Locations</td>
<td>SUBJ16, run1</td>
<td>0.921</td>
</tr>
<tr>
<td>Study Locations</td>
<td>SUBJ16, run2</td>
<td>0.919</td>
</tr>
<tr>
<td>Study Objects</td>
<td>SUBJ8, run1</td>
<td>0.917</td>
</tr>
<tr>
<td>Try Objects</td>
<td>SUBJ16, run2</td>
<td>0.917</td>
</tr>
</tbody>
</table>

Table 11.1: ROC areas for MDG retrieval ranked in decreasing order. Only the top 15 queries are shown.

There are two things worth noticing in Table 11.1. First, all 15 queries have very high values of the ROC areas indicating that our MDG based matching heuristic performed very well for these queries. Second, it appears that MDG retrieval performed very well for the event perception data, that is, the house and study video conditions. This is in contrast to the Brodmann vectors approach which performed not well on the event perception data. Also, we note that visual examination of hit matrix for event perception did not reveal how well MDG performs on this data.
Figure 11.7: Histogram of frequencies for areas under ROC curve for 928 MDG queries. Most queries have ROC areas above 0.5 (random guessing).

The other important question is how well heuristic performed for all queries. Figure 11.7 gives the histogram of frequency distribution for areas under ROC curve for 928 MDG queries.

No valid retrieval system should have the value of ROC area less than 0.5, the ROC area of random guessing. It appears, from Figure 11.7, that the MDG-based heuristic performed better than random guessing for almost all queries. In fact, MDG performed better than random guessing in 912 cases out of 928, which is 98% of all queries.
Chapter 12:
Content-Based fMRI Retrieval Using
Maximum Correlation Graphs

In this Chapter, we report the retrieval experiments that we conducted using maximum
correlation graphs. First, we describe the procedure we used for extracting MCGs from
the collection in Section 12.1. We report the retrieval experiments and discuss the
retrieval performance in terms of the hit matrices in Section 12.2. In Section 12.3, we
assess the retrieval performance using ROC curves and precision-recall curves. This is
followed by the discussion of the performance in terms of the area under the ROC curve.

12.1 Conceptual Framework for Using Maximum Correlation Graphs

The number of Maximum Correlation Graphs (MCG) was the same as the number of
MDGs, 928. The number of graphs was lower than the number of Brodmann vectors due
to the difference in the method used to select activated voxels. For the MCG, a threshold
was set based on significance of the correlation with the stimulus. Therefore, some $t$-
maps had no voxels with values above the threshold. We were not able to build MCGs
for such \( t \)-maps. So we decided to exclude fMRI images, whose \( t \)-maps have no active voxels, from our analysis. Note that for Brodmann vectors, a threshold was set so that 1\% of the voxels in a standard MNI brain have \( t \)-values higher than the threshold. This guaranteed that every \( t \)-map had an equal number of selected voxels. The following are the steps to build a Minimum Correlation Graph:

1) We used FSL to generate a \( t \)-map for each condition of an experiment. Also, we ran FSL in batch mode to compute lagged \( t \)-maps at the following time lags: 100, 200, 300, 400, and 500 milliseconds for each condition.

2) Step 1 was repeated for every experiment in the repository. At this step, we obtained 5,568 \( t \)-maps because every possible combination of subject, condition and run yields a basic \( t \)-map and 5 lagged \( t \)-maps.

3) By default, FSL reports voxels as statistically significant if their correlation with a stimulus is so great that uncorrelated data would reach this level only 5\% of the time, corrected for the number of voxels (See Section 9.2). We extracted those voxels reported significantly activated, using FSL at this probability, from each \( t \)-map, including the lagged \( t \)-maps. These voxels were grouped into contiguous clusters of activation using FSL.

4) We refer to a basic \( t \)-map as time lag 0 and lagged \( t \)-maps at 100, 200, 300, 400, 500 milliseconds as time lag 1, 2, 3, 4, and 5 respectively. Thus, for each subject, condition and run we have a sequence of \( t \)-maps (volumes) at lags 0, 1, 2, 3, 4, and 5. For every sequence of \( t \)-maps, we computed the centers of mass of the contiguous clusters (Equation 7.2) in its volumes. Thus, every fMRI dataset in the
collection was reduced to a set of nodes, where the centers of mass represented nodes.

5) The Maximum Correlation heuristic, described in Section 7.3, was used to build MCGs from these sets of nodes.

Note that Steps 1-4 are identical to steps needed to build the Minimum Distance Graph. So these computations were done once and reused. The real difference in building an MDG and an MCG is in Step 5. A different heuristic is used here to connect the nodes of a graph.

Ultimately, we want to connect activated areas that are related to each other. Following this goal, we hypothesized that related activations should strongly correlate with each other. Therefore, the Maximum Correlation heuristic connects nodes based on the strength of their correlations with each other.

In this model, we computed actual correlations between nodes. The main idea was that connecting nodes with the strongest correlations in an MCG could reveal functional integration in a brain. We hypothesized that we could use the MCG representation of fMRI data for content-based fMRI retrieval.

12.2 Results of the MCG Based fMRI Matching

Following the steps described in Section 12.1, we computed 928 MCG representations for fMRI files. The main idea was to represent fMRIs as MCGs and do graph matching on MCGs. We used the improved version (Macrini, 2003) of the DAG matching
algorithm developed by Shokoufandeh et al. (2001) for graph matching. The code was modified to handle MCGs. We also replaced the domain-dependent node distance function with the one defined in Equation 7.4.

![Hit matrix for the recall data, using MCG matching.](image)

*Figure 12.1: Hit matrix for the recall data, using MCG matching.*

The DAG matching algorithm was used to compute pairwise similarities for pairs of graphs extracted from fMRI data in the collection. As a result, we obtained a 928×928 hit matrix, in which each entry was the similarity score of corresponding MCGs.
Figure 12.1 shows the hit matrix for MCG matching on the recall data. Experiments are ordered along the $X$ and $Y$ axes from right to left (ordered by the stimulus, subject and run), and from top to bottom. The similarity-to-color mapping is shown at the right top, blue – low similarity and yellow – high similarity.

Visually, Figure 12.1 appears very similar to the hit matrix for the MDG matching. Perhaps, the reason is that both types of graph have exactly the same sets of vertices. The only real difference is in their topological structure, because they have
different sets of edges. The set of edges depends on the heuristic, while the set of nodes
does not.

Figure 12.2 is the hit matrix for the morality data, using the MCG matching. Although, the hit matrices for the morality data look the same with MDG and MCG
matching, they are not. We will see this in the next section when we look at the ROC
curves.

The hit matrix for the event perception data does not reveal any interesting
pattern. We can see it in Figure 12.3, which give the hit matrix for MCG matching.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{hit_matrix}
\caption{Hit matrix for the event perception data, using MCG matching.}
\end{figure}

Finally, Figure 12.4 gives the hit matrix for all data together. Again, the hit matrix
looks similar to the one of the MDG matching. It also shows the strong separation of the
morality data from the other experiments (oddball and event perception). It appears that both graph-based heuristics were able to pick up differences in brain activations for solving moral dilemmas and for watching pictures and videos.

Figure 12.4: Hit matrix for all data using MCG matching. Morality data is strongly separated from other experiments.
12.3 ROC Curves for MCG Matching

The big hit matrix for MCG matching, shown in Figure 12.4, was used to conduct query-by-example IR experiments. We kept the design of the retrieval experiments the same, leave-one-out. In leave-one-out design, an object is removed from the collection and used as a query.

The maximum correlation heuristic builds an MCG from a series of $t$-maps ($t$-maps at time lags 0, 1…5) computed for a subject in an experiment. There is a direct mapping between the MCG and the fMRI file for the subject.

According to the experimental design, we removed each MCG from the 928 in the collection and used it as a query to retrieve “similar” MCGs from the set of remaining 927 graphs. There is a direct mapping between the MCG and the fMRI file. Thus, this is equivalent to searching for fMRIs that are similar to the fMRI of the graph used as a query. That is, given a query fMRI we retrieve “similar” fMRIs from the repository. fMRIs are considered similar if subjects perform the same type of cognitive task. In other words, fMRI images are similar if they have the same condition.

For each query, we constructed a list with experiment names and their similarity scores. The list was ranked in the decreasing order of similarity scores. Note that this is equivalent to a ranked list of retrieved results for the query, because it can be cut-off at any desired point.

We repeated the leave-one-out retrieval for each dataset in the collection. Thus, we generated 928 ranked lists, one for each query. The ROC curve and the area under the ROC curve were computed for all queries. The performance of a query can be visually
examined by its ROC curve. In Figure 12.5, we plotted ROC curves of the three fMRI datasets plotted in Figure 11.5 for comparison purposes.

![ROC curves for MCG matching](image)

Figure 12.5: ROC curves using MCG matching for the three subjects as in Figure 11.5.

Although, the ROC curves look the same, they are not. The areas under ROC curve for the subjects’ az_21, SUBJ17 and MD290903 were 0.974, 0.705 and 0.389 respectively for MCG matching. Compare this to 0.973, 0.639, and 0.389 for the same subjects using MDG matching. It appears that the maximum correlation heuristic performed slightly better in the middle ranks than did the minimum distance heuristic.

To confirm our conclusions, we plotted the precision-recall curves for the same three queries in Figure 12.6.
Figure 12.6: Precision as a function of recall for MCG matching.

It shows that the similar datasets appeared early in the returned list for the query with red colored crosses (az_21, House Active). Also, the performance was better for curve with blue diamonds than for curve with green triangles. It appears that the majority of the relevant documents were at the top of the returned list for blued diamonds.

We put together ROC areas for all the queries to obtain a big table of ROC areas and ranked it in decreasing order. Table 12.1 gives the ROC areas for the 15 queries with the largest areas under the ROC curve for the MCG based retrieval.

All 15 queries have very high values of ROC areas indicating that the MCG based heuristic performed very well for these queries. MCG performed very well on the event perception data, as did MDG. However, note that they performed best for different
subjects, although, some subjects are in both lists. Both MCG and MDG performed well on data where the Brodmann vector approach performed poorly (the event perception data). Visual examination of the hit matrix for the event perception data did not show that MCG did well on this data.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Subject, Run</th>
<th>ROC area</th>
</tr>
</thead>
<tbody>
<tr>
<td>House video</td>
<td>az_21, run1</td>
<td>0.974</td>
</tr>
<tr>
<td>House video</td>
<td>az_22, run1</td>
<td>0.959</td>
</tr>
<tr>
<td>House video</td>
<td>az_31, run1</td>
<td>0.956</td>
</tr>
<tr>
<td>House video</td>
<td>az_30, run1</td>
<td>0.938</td>
</tr>
<tr>
<td>Study video</td>
<td>az_19, run1</td>
<td>0.936</td>
</tr>
<tr>
<td>Recall Locations</td>
<td>SUBJ13, run4</td>
<td>0.927</td>
</tr>
<tr>
<td>Recall Locations</td>
<td>SUBJ8, run4</td>
<td>0.926</td>
</tr>
<tr>
<td>Study Objects</td>
<td>SUBJ7, run3</td>
<td>0.926</td>
</tr>
<tr>
<td>Study Objects</td>
<td>SUBJ16, run1</td>
<td>0.925</td>
</tr>
<tr>
<td>Study Objects</td>
<td>SUBJ14, run3</td>
<td>0.925</td>
</tr>
<tr>
<td>Study Faces</td>
<td>SUBJ14, run3</td>
<td>0.925</td>
</tr>
<tr>
<td>Study Objects</td>
<td>SUBJ10, run3</td>
<td>0.923</td>
</tr>
<tr>
<td>Study Objects</td>
<td>SUBJ14, run2</td>
<td>0.920</td>
</tr>
<tr>
<td>House video</td>
<td>az_19, run1</td>
<td>0.920</td>
</tr>
<tr>
<td>House video</td>
<td>az_33, run1</td>
<td>0.919</td>
</tr>
</tbody>
</table>

Table 12.1: ROC areas for the MCG retrieval ranked in decreasing order. Only the top 15 queries are shown.

Let us now look at how MCG performed for all queries. Figure 12.7 gives the histogram of frequency distribution for the ROC areas of 928 MCG queries.

No valid system should have value of ROC area less than 0.5, the ROC area of random guessing. Figure 12.7 shows that the MCG-based heuristic performed better than random guessing for almost all queries. MCG performed better than random guessing in 98% of queries (910 cases out of 928).
Figure 12.7: Histogram of frequencies for areas under the ROC curve for 928 MCG queries.
Chapter 13:
Concluding Discussion on fMRI Matching

In this Chapter, we give further discussion about the retrieval performance of the three heuristics. Specifically, we compare their average performances across fMRI data corresponding to different cognitive tasks. We show that the MDG and MCG based methods performed much better than the Brodmann vector approach, especially for the event-perception and recall data. Finally, we discuss the limitations of this research.

13.1 Comparison of the three fMRI Matching Heuristics

We used ROC curves (Receiver Operating Characteristic curve, see Chapter 10 for details) to investigate the performance for three content-based fMRI matching heuristics (Brodmann vectors, MDG and MCG). The reason we did not use the precision-recall curves is because in our context, the retrieval performance is equivalent to success in the task of grouping together fMRI images corresponding to the same kind of brain activity, and this is best modeled as a categorization task. The ROC curves express performance of the system as a quantitative measure, the area under the ROC curve.
Although, our heuristics are very simple, they are remarkably powerful. As we have shown in Chapter 10, Chapter 11 and Chapter 12, all three methods perform a good deal better than a system that does random guessing. The Brodmann Vectors heuristic performed better than random guessing in 835 cases out of 986 (85% of all queries). The MDG and MCG heuristics beat random guessing in 912 cases out of 928 (98% of all queries) and in 910 cases out of 928 (98% of all queries) respectively. It appears that both of the graph-based heuristics, MDG and MCG, performed much better than the Brodmann vectors based method. MDG slightly outperformed MCG, by 2 cases, which is surely not statistically significant.

Another way to express the comparison is by the mean values and standard deviations of ROC areas. Table 13.1 gives the mean and standard deviation of the area under ROC curve over all the cases, for each method.

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>Standard Error of The Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brodmann vectors</td>
<td>0.59</td>
<td>0.13</td>
<td>0.004</td>
</tr>
<tr>
<td>Minimum Distance</td>
<td>0.66</td>
<td>0.11</td>
<td>0.004</td>
</tr>
<tr>
<td>Maximum Correlation</td>
<td>0.66</td>
<td>0.11</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Table 13.1: Mean, standard deviation, and standard error of the mean for areas under the ROC curve.

It appears that both MCG and MDG easily outperformed Brodmann vectors method. Note that MCG and MDG have smaller standard deviations, which indicates that they have less variation in their performance. The “standard deviation of the mean” or the “standard error of the mean” is the standard deviation of the original distribution divided
by the square root of the sample size (the number of cases the mean is based upon). Thus, the larger the sample size, the smaller the standard error of the mean. Since there are quite a few cases here, these differences are seen to be significant. For example, the two sample $t$-test for means with unequal variances between the Brodmann vector and the MCG heuristics is statistically significant (two tailed, probability $p < 0.0001$).

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Condition</th>
<th>Number of relevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event perception</td>
<td>House video</td>
<td>27</td>
</tr>
<tr>
<td>Event perception</td>
<td>Study video</td>
<td>25</td>
</tr>
<tr>
<td>Morality</td>
<td>M-e-</td>
<td>241</td>
</tr>
<tr>
<td>Morality</td>
<td>M+E-</td>
<td>243</td>
</tr>
<tr>
<td>Morality</td>
<td>M+E+</td>
<td>252</td>
</tr>
<tr>
<td>Recall</td>
<td>Recall Faces</td>
<td>9</td>
</tr>
<tr>
<td>Recall</td>
<td>Recall Locations</td>
<td>9</td>
</tr>
<tr>
<td>Recall</td>
<td>Recall Objects</td>
<td>9</td>
</tr>
<tr>
<td>Recall</td>
<td>Study Faces</td>
<td>27</td>
</tr>
<tr>
<td>Recall</td>
<td>Study Locations</td>
<td>27</td>
</tr>
<tr>
<td>Recall</td>
<td>Study Objects</td>
<td>27</td>
</tr>
<tr>
<td>Recall</td>
<td>Try Faces</td>
<td>27</td>
</tr>
<tr>
<td>Recall</td>
<td>Try Locations</td>
<td>27</td>
</tr>
<tr>
<td>Recall</td>
<td>Try Objects</td>
<td>27</td>
</tr>
</tbody>
</table>

*Table 13.2: Number of datasets for each condition in the collection.*

Perhaps, this is explained by the fact that MCG and MDG include temporal information about the course of the activation. The Brodmann vectors method is based on the static snapshot of the “thinking” brain (basic $t$-map), while MCG and MDG include temporal information encoded in lagged correlations (lagged $t$-maps). It is possible that two datasets with distinct graphs may have exactly the same Brodmann vectors if they have identical basic $t$-maps. Thus, graphs are in some sense a richer representation than the Brodmann vector.
Perhaps, another way to explain the better performance of MCG and MCG over the Brodmann vector is that they are invariant to rotation and translation and, thus, to small displacements or changes in the brain anatomy. The edges of MDG and MCG depend on the Euclidean distances and correlations between nodes respectively. If we would turn the brain upside down or move it, the distances between nodes and the correlations would not change. Thus, we would get the same graph even if the brain was turned upside down or moved. This is not true for the Brodmann vector, because even small changes in the brain anatomy cause changes in its components.

<table>
<thead>
<tr>
<th>Data</th>
<th>Heuristic</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>Standard Error Of The Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>Brodmann Vectors</td>
<td>0.74</td>
<td>0.18</td>
<td>0.0131</td>
</tr>
<tr>
<td>Recall</td>
<td>Minimum Distance</td>
<td>0.83</td>
<td>0.07</td>
<td>0.0052</td>
</tr>
<tr>
<td>Recall</td>
<td>Maximum Correlation</td>
<td>0.83</td>
<td>0.07</td>
<td>0.0051</td>
</tr>
<tr>
<td>Morality</td>
<td>Brodmann Vectors</td>
<td>0.54</td>
<td>0.04</td>
<td>0.0014</td>
</tr>
<tr>
<td>Morality</td>
<td>Minimum Distance</td>
<td>0.61</td>
<td>0.05</td>
<td>0.0019</td>
</tr>
<tr>
<td>Morality</td>
<td>Maximum Correlation</td>
<td>0.61</td>
<td>0.05</td>
<td>0.0019</td>
</tr>
<tr>
<td>Event perception</td>
<td>Brodmann Vectors</td>
<td>0.73</td>
<td>0.14</td>
<td>0.0224</td>
</tr>
<tr>
<td>Event perception</td>
<td>Minimum Distance</td>
<td>0.85</td>
<td>0.08</td>
<td>0.0135</td>
</tr>
<tr>
<td>Event perception</td>
<td>Maximum Correlation</td>
<td>0.83</td>
<td>0.09</td>
<td>0.0152</td>
</tr>
</tbody>
</table>

*Table 13.3: Mean, standard deviation, and the standard error of the mean of ROC areas for fMRI matching heuristics, ordered by condition.*

The number of similar datasets for each condition was quite different. Table 13.2 gives the number of similar datasets for each condition. Queries for the morality data have many more relevant fMRI datasets than queries for the recall or event perception data. For instance, the M+E+ (moral and emotional) condition in the morality data has
252 relevant datasets, while the “study faces” condition of the recall data has only 27 relevant datasets.

Table 13.3 gives the mean and standard deviation of ROC areas for all three heuristics ordered by condition. Note how well MDG and MCG performed in average on the recall and the morality data. Standard errors of the mean show that the differences in performance between the graph-based methods and the Brodmann vector approach are statistically significant for all conditions. However, as we mentioned before, this might be due to the large number of cases. Clearly, it appears that the most difficult data to retrieve was morality for all three heuristics. We propose that this is not surprising because morality experiments require subjects to make difficult emotional and moral decisions. However, even on morality problems, the graph-based methods beat the Brodmann vector method by a large margin.

The performances of MDG and MCG are pretty comparable. MCG performed slightly better on the recall data, while MDG slightly outperformed MCG on the morality and event perception data.

### 13.2 Limitations of the Research

We have proposed that representation by Brodmann vectors, MDG and MCG can effectively address the problem of information retrieval for fMRI data. The results of the experiments, which we conducted to test our ideas, support our suggestions. However, this does not imply that the average performance will be the same for other types of fMRI experiments. For instance, we saw from Table 12.3 that the average performance of
heuristics depended on data. It appears that all three approaches do better on fMRI experiments with simpler tasks.

We do not have a good scheme to automatically classify data into different categories in such a way that for queries in the same category the performance will be the same. However, it is encouraging that MDG and MCG performed almost equally well on both the recall and event perception data. This gives us hope that it is feasible to build a robust classification scheme for fMRI data. Eventually, we may be able to predict how well our heuristics will perform for a particular collection of data.

Another limitation is that we only used time lags up to 500 milliseconds due to the computational constraints. However, complex cognitive task may take more than 500 milliseconds to complete. This might be one of the reasons why MDG and MCG did not perform as well on the morality data, which is quite complex, as did on the recall and event perception data.

There are some other choices which were somewhat arbitrary, but had to be made to complete this work. These include: 1) the specific node distance function; 2) not allowing signal merges in the MDG or the MCG; 3) allowing edges only between nodes in adjacent time lags; etc.

All of these are somewhat arbitrary simplifications of the model. In each case, exploring more general models would permit us to optimize the models, with no decrease in performance.

In particular, the graph-based models here, which attempt to capture a space-time progression of brain activity, seem to hold real promise for retrieval of dynamic brain images, by their content alone.
Chapter 14:

Conclusion

In this Chapter, we discuss the conclusions that we drew from our retrieval experiments with the three potential heuristics. It appears that all three heuristics performed much better than by chance. The graph-based methods significantly outperformed the Brodmann vector approach on all available data. We argue that this validates our research hypothesis on the feasibility of using graph structures and anatomical representations of the brain for content-based fMRI data retrieval. Thus, it appears that graph structures provide a better representation of dynamics of neural activity than static Brodmann vectors. We argue that there are several reasons for this: a) graph-based methods include temporal information unlike the Brodmann vector approach; b) graph-based methods are invariant to small displacements in the brain anatomy.

14.1 A Brief Summary of Our Experiments

fMRI is a brain imaging technique that allows us to see traces of neural activation caused by cognitive processes in the human brain, with a few seconds delay. The development of
the fMRI technique, with its relatively high temporal and spatial resolution and non-invasive nature, opened a new era in functional neuroimaging. The amount of fMRI data generated at the research centers around the world is growing rapidly. There is an urgent need to organize, index, store and retrieve this data.

However, as yet there is no theory for content-based fMRI retrieval. It appears that there are several reasons for this. One major problem is that scanned images are noisy and often corrupted by motion. It is very difficult to separate random noise from cognitive processes in the human brain. More fundamentally, our understanding of cognitive processes in human brain is incomplete.

This research investigates the potential for effective content-based IR in fMRI data. We tried to identify good features for retrieving “similar” fMRI images in a “query-by-example” setting. Based on our understanding of human brain functionality, we established working assumptions to guide in feature extraction, and explored three fMRI matching heuristics:

1) Brodmann Vectors: assume that Brodmann areas are related to functionally distinct areas of brain and use this for content-based retrieval.

2) Minimum Distance Graph: Signals in a brain propagate with finite speed. Therefore, hypothesize that one can trace the topology of these signals in a MDG. This reduces the content-based fMRI retrieval problem to MDG matching.

3) Maximum Correlation Graph: Brain functionality is based on two principles: functional specialization and functional integration. We hypothesized that MCG can reveal the functional integration of brain areas working together. If valid, this would reduce the IR problem to one of MCG matching.
The idea that it might be possible to represent dynamic brain images by graphs representing 4D space time structures was put forward in a series of proposals by Kantor and Hanson, the third of which was funded by the NSF, supporting the work reported here. In the present work we do “not” attempt to validate that concept in terms of the specific space time localization reported by the graph. Instead, we seek to validate it in the context of query by image content. In that context we face two problems: extracting the graph, and assessing the performance.

We have used the ROC curve to measure the performance of the fMRI matching heuristics: Brodmann vectors, Minimum Distance, and Maximum Correlation. Using fMRI data produced in the BRIM project (see Chapter 8 Section 8.2) as test data, we found that the three methods are good candidates for content-based fMRI retrieval. In particular, MDG and MCG heuristics performed better than random guessing in 912 cases out of 928 (98% of all queries) and in 910 cases out of 928 (98% of all queries) respectively. Even the worst of the three, the Brodmann vectors method, performed better than random guessing in 835 cases out of 986 (85% of all queries). The average values of the ROC area were 59%, 66% and 66% for Brodmann vectors, MDG and MCG respectively. MDG and MCG performed exceptionally well on the more complex recall and morality data, with average values of the ROC area up to 85%±8% (standard deviation 0.08) and 83%±7% (standard deviation 0.07) respectively (See Table 13.3). Note how close this is to the ROC area of an ideal system, which is equal to 1.

The author proposed the idea of linking time slices by a correlation measure. The results showed that the correlation linking approach provides better retrieval performance
than an alternative based on an anatomical representation of the brain. Another method of link finding (minimum distance) was also shown to give performance comparable to the correlation-based links.

14.2 Significance of the Research and Directions for Further Study

As yet there is no theory for content-based indexing and retrieval of fMRI data. In this research, we have identified three similarity measures, Brodmann areas, Minimum Distance and Maximum Correlation, which show great promise in successfully assessing the “similarity” of fMRI experiments. We have demonstrated how to use these features for content-based fMRI data retrieval. We find that the performance of the three proposed methods is much better than random guessing, especially MDG and MCG.

Since the three proposed heuristics for content-based fMRI retrieval seem valid (but of course, are not the final answer), we do have a basic set of methods for effective fMRI matching. We expect that the performance of the proposed heuristics on new data will be comparable with levels reported here.

In this research, we limited our largest time lags to 500 milliseconds. However, it is possible that processing a complex stimulus, like a moral dilemma, takes a longer time. This might be one of the reasons why MDG and MCG performed worse on the morality data than on the recall and event perception data. This research can be extended by computing additional time lags. After all, it is just a matter of computational time. Perhaps, MDG and MCG built on larger numbers of time lags will be able to retrieve the
morality data with the same impressive performance that they show here on the recall and event perception data.

Another challenging direction for researchers is to define more sophisticated node distance functions and graph building heuristics. We believe that our research shows the potential of using such methods for content-based fMRI retrieval.

Also, it will be important to test these heuristics on new data. We need to establish which differences in the data cause the differences in performance, and perhaps fine tune the methods to handle other types of data. It is possible that, based on the findings of this analysis, one can build a data classification scheme. Perhaps, different types of data will require different retrieval methods or perhaps even fusion of several methods.

Specifically, we saw that on some data the MDG performed better and on other data the MCG performed better. It is an interesting topic for future research to see whether fusion of these retrieval methods could improve on the performance of each method alone, or whether the level of performance can be anticipated so that the most appropriate technique can be automatically selected for a certain database.
Bibliography


Curriculum Vita

Ulukbek Ibraev

EDUCATION

Ph.D., Information Science, October 2005
Rutgers University, New Brunswick, NJ

Master of Science, Computer Science, January 2002
Rutgers University, New Brunswick, NJ

Master of Library Science, May 1999
Rutgers University, New Brunswick, NJ

Bachelor of Science, Computational Physics, May 1996
Kyrgyz State National University, Bishkek, Kyrgyzstan

RESEARCH

• Novel Indexing and Retrieval of Dynamic Brain Images
  Graduate Assistant, September 2002 – June 2005

• Semantic Dimensionality and Effective Data Fusion in Information Retrieval
  Graduate Assistant, September 1999 - May 2002

PUBLICATIONS


WORK EXPERIENCE

Graduate Assistant, September 2002 – June 2005
Rutgers University – New Brunswick

Teaching Assistant, Fall 2002.
Programming for Information Technology & Informatics Majors I (04:547:201)
Rutgers University, New Brunswick

Programmer, Summer 2001
CheckFree Corporation, Newark, NJ

Programmer, Summer 1998
Ovid Technologies Inc., New York, NY

Price Waterhouse LLP, Bishkek, Kyrgyzstan