INFORMATION RETRIEVAL VIA UNIVERSAL SOURCE CODING

A Thesis
Presented to
The Academic Faculty

by

Soo Hyun Bae

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
November 2008

INFORMATION RETRIEVAL
VIA UNIVERSAL SOURCE CODING

Approved by:

Dr. Biing-Hwang Juang, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Russell M. Mersereau
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ghassan Al-Regib
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Linda M. Wills
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Thrasyvoulos N. Pappas
School of Electrical Engineering and Computer Science
Northwestern University

Date Approved: 5 November 2008

Soli Deo Gloria

ACKNOWLEDGEMENTS

Conducting research and writing a dissertation was an arduous but exciting journey for me. Although the journey ends up with receiving a Ph.D degree, I cannot complete the journey without acknowledging support from many people.

Throughout the countless hours of research, Dr. Fred Juang has encouraged me to explore unknown areas and taught me how to discover a new knowledge. He has always patiently waited for me to produce *the results* and showed me by example what a model researcher should be. He deserves the highest credit and my sincere acknowledgment.

I sincerely thank Dr. Thrasos Pappas for his gracious and insightful help whenever I requested it. He steered my research in a more valid direction in regards to human perception and guided me to understand what is practically valuable research. His saying about *belief* still becomes strong motivation to climb the mountain up to the *summit*.

I wish to acknowledge Dr. Russell Mersereau, Dr. Ghassan Al-Regib, and, Dr. Linda Wills who served as members of my dissertation committee, for their many helpful comments and suggestions. Special appreciation is given to Hewlett-Packard Laboratories and Dr. Ronald Schafer for their generous support on this research.

I also thank Dr. Sean Ramprashad at DoCoMo Communications Labs USA for his careful guidance in perceptual audio coding and quantization theory. From him, I learned to dig more deeply in understanding observed results. I also thank Dr. Ananth Grama at Purdue University and Dr. James Wang at Pennsylvania State University for providing their implementations of image coding and image retrieval system. The comparisons with them were very helpful in objectively evaluating the

algorithms proposed in this dissertation.

I consider myself fortunate to have many friends that contributed directly or indirectly to my studies. I would like to thank my group members: Woojay Jeon, Qiang Fu, Antonio Moreno, Dwi Mansjur, Ted Wada, Gaofeng Yue, Rungsun Munkong, Enrique Arnuncio, Yong Zhao, Umair Altaf, Shigeki Miyabe, and Sunghwan Shin for their friendship and fruitful collaborations. I greatly appreciate Nawaf Almoosa with whom I conducted some fundamental experiments. I would also like to acknowledge many friends from the Center for Signal and Image Processing (CSIP): Toygar Akgun, Kerkil Choi, Jinwoo Kang, Martin Tobias, Dihong Tian, Seokchul Kwon, Sibel Yaman, Hyungjoon Kim, Sam Li, and Volkan Cehver. As well as bringing joy and laughter to my life, they gave me many lessons on signal processing theories. Also, I am especially indebted to Changsoon Choi and Sanghyun Seo, who forced me to work hard from the time of application to the time of graduation.

My love and gratitude go to my family and in-laws for their endless love and prayer. Especially, I deeply appreciate my father. Since I was in elementary school, he has taught me computer programming, digital logic theory, and circuit theory. He has been my role model as an electrical engineer. I would like to thank the Billy Davis family and friends in FBA Sanctuary Orchestra, especially Richard Rose and Jeff Cranfill, for their prayer. I thank Dennis Brittingham for proofreading this dissertation. Of course, I am responsible for any weaknesses or inaccuracies that remain in this dissertation.

Last but not least, I would like to express much gratitude and love to my wife, Naeri Kim. She has prayed for me, advised me on oral presentation, first proofread my papers, helped to manage image databases, and even inspired me with many research ideas. Without her patience and love, I would not have been able to complete this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This dissertation explores the intersection of information retrieval and universal source coding techniques and studies an optimal multidimensional source representation from an information theoretic point of view. Previous research on information retrieval particularly focus on learning probabilistic or deterministic source models based on primarily two different types of source representations, e.g., fixed-shape partitions or uniform regions. We study the limitations of the conventional source representations on capturing the semantics of the given multidimensional source sequences and propose a new type of primitive source representation generated by a universal source coding technique. We propose a multidimensional incremental parsing algorithm extended from the Lempel-Ziv incremental parsing and its three component schemes for multidimensional source coding. The properties of the proposed coding algorithm are exploited under two-dimensional lossless and lossy source coding. By the proposed coding algorithm, a given multidimensional source sequence is parsed into a number of variable-size patches. We call this methodology a parsed representation.

Based on the source representation, we propose an information retrieval framework that analyzes a set of source sequences under a linguistic processing technique and implemented content-based image retrieval systems. We examine the relevance of the proposed source representation by comparing it with the conventional representation of visual information. To further extend the proposed framework, we apply a probabilistic linguistic processing technique to modeling the latent aspects of a set of documents. In addition, beyond the symbol-wise pattern matching paradigm employed in the source coding and the image retrieval systems, we devise a robust pattern matching that compares the first- and second-order statistics of source patches. Qualitative and quantitative analysis of the proposed framework justifies the superiority of the proposed information retrieval framework based on the parsed representation. The proposed source representation technique and the information retrieval frameworks encourage future work in exploiting a systematic way of understanding multidimensional sources that parallels a linguistic structure.

CHAPTER I

INTRODUCTION AND BACKGROUND

The widespread deployment of digitizing systems has greatly accelerated the accumulation of digital information. The need for prompt and intelligent access to stored information has bolstered a wide range of research in the field of informatics, especially content-based image retrieval (CBIR). A particular challenge arises from the fact that a major portion of the accumulated information is represented in its raw digital form, such as photographic images, and is in general difficult to organize in a semantically sensible structure for easy access. A text-based database is conventionally organized based on a list of (automatically extracted) keywords and in their alphabetical order; in contrast, the organization of a picture database is not as straightforward, unless text annotation, which requires intensive labor and may not be sufficiently comprehensive, is available. Hence, analyzing multidimensional data and extracting features from them for the purpose of organization and retrieval is a challenging task.

In this dissertation, we present a new comprehensive framework for information retrieval that can effectively deal with multidimensional discrete sources and bridge the gap between the sources and human semantics. Specifically, we are interested in a universal source coding algorithm as a source characterization and a feature extraction technique. Since Shannon opened the realm of information theory [72], many universal source coding algorithms have been proposed. Some of which are prevalent in many data compression applications. The algorithm that is of our interest is the Lempel-Ziv incremental parsing algorithm (LZ78) [97], which parses the given discrete source into distinct phrases while constricting the dictionary. Since the statistical distribution of the given source is implicitly embedded into the dictionary, in the proposed information retrieval systems, we use the dictionary as a rich resource for analyzing the given sources. However, LZ78 can deal only with a one-dimensional source sequence, it is desired to develop such a source coding technique that can encode a multidimensional source. Based on the universal source coding technique, we will consider feature extraction algorithms and their applications to information retrieval problems. In this dissertation, we focus on image retrieval applications, but the underlying source characterization technique can be applied to arbitrary dimensional sources, e.g., audio signals in one dimension, video sequences in three dimensions, images in two dimensions.

Our goal in this dissertation can be divided into four key parts:

- We develop a new universal source coding technique for a multidimensional finite discrete sequence and evaluate the performance of its lossless and lossy implementations.

- We develop an image retrieval framework that uses the features extracted by the multidimensional universal source coding algorithm, implement image retrieval systems based on the proposed framework, and evaluate the performance of the systems.

- We validate the source representation induced by the universal source coding technique under a visual information analysis.

- We develop a probabilistic framework for information retrieval by formulating robust pattern matching into a semantic analysis model.

To tackle the aforementioned two main problems, universal source coding and image retrieval, an abundance of literature has been reported for the last decades. In this dissertation section, we briefly review prior work on the image retrieval problems, followed by various universal source coding techniques.

1.0.1 Related work in Image Retrieval

The basic structure of an image retrieval system consists of three fundamental components: query processing, visual feature extraction, and similarity measure. According to the Merriam-Webster dictionary [53], *query* is defined as *a question in the mind*. Although we assume the query in one's mind precisely points to the target object, a rendered query may contain some distortions caused by many sources, for example, dialect or accent in spoken query, misleading textual expression in text query, and incorrect choice of image example in image query. Such query distortions need to be distilled by query processing so as to bring out the true intent embedded in the query expression. In the processing of visual feature extraction, images stored in a database are analyzed, and numeric descriptors capturing specific visual characteristics, called *features* [11, Sec. 1.4], are extracted. After an image retrieval system constructs feature databases for the stored images, the similarity between the feature of each image and the feature extracted from an issued query is computed in a domain where semantic similarities of whole or specific images can be relevantly defined. Although these three components are described separately here, they are required to be coherently devised or implemented according to the types of input queries, characteristics of target images, and computational constraints. Finally, the image retrieval system sorts the retrieved images according to the computed similarities and renders them on an output device. A schematic overview of the basic structure of an image retrieval system is depicted in Figure 1.

Generally, image retrieval systems are categorized into three groups by their modality of queries:

1. *Text-based image retrieval*: The query in one's mind is represented as keywords, phrases, or sentences. Systems in this category process the given query to reduce semantic distortions and inflections and extract keywords from phrases or sentences. Then, the system searches for a match between the processed

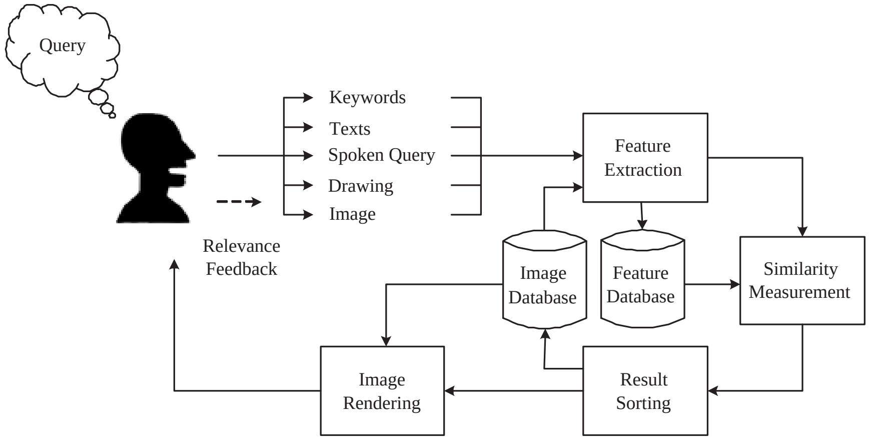


Figure 1: A schematic overview of an image retrieval system.

keywords and the keywords associated with each image in the database. This is adopted in many popular image search engines, e.g., Google and Yahoo!.

2. *Content-based image retrieval*: This type of system accepts imagery queries, such as hand drawing and exemplary images, that convey the full or a part of the desired images. The first processing of the system is to extract appropriate features from the issued query and to map them onto a semantic feature space where a similarity between the desired semantic concept and those concepts each image contains can be computed. The systems for hand drawing and exemplary images are widely called query-by-sketch systems and query-by-example systems, respectively.

3. *Interaction-based image retrieval*: Systems in this group are capable of interacting with the users for several purposes, e.g., clarifying the given query, narrowing down the search range of database, and requesting feedback from the rendered images (so called relevance-feedback). A comprehensive overview can be found in [29, 70, 94].

In addition to the aforementioned system groups with single modality queries, there have been composite image retrieval systems that are capable of multimodal queries.

For example, those systems reported in [16, 39, 45] can process such queries that are rendered both in text and in image, and the system proposed in [59] was designed to deal with audiovisual queries. In this dissertation, we limit our focus on the CBIR systems.

Since the inception of a CBIR system by IBM QBIC [25] in the commercial domain and MIT Photobook [55, 64] in a research domain, a number of CBIR systems have been devised based on various feature extraction techniques and similarity measures. In most of the CBIR systems, visual features are formed from small chunks of visual information by typically two types of techniques: fixed-block partition [10,15,23,32,37,56,60,66,75,77–79,90] and image segmentation [26,41,61,81,91]. In the fixed-block partition technique, on the one hand, processing of imagery data involves a square of pixels. Once a given image is partitioned into a number of blocks, pixel statistics, such as color histogram and texture configuration, are estimated from each individual block. Some of the CBIR systems based on a fixed-block partition train the image codebook by vector quantization (VQ) so as to construct a visual "lexicon" that captures visual "semantics" under some minimum average distortion criteria [15,32,37,77–79,90]. Since many natural objects are not tessellations of square blocks, a fixed-block partition is usually inferior in representing an image of natural objects. Conversely, image segmentation techniques are employed for extracting object or sub-object regions in a given image. However, it is widely recognized that machine segmentation of images is still a long way from mimicking the way a human would identify real world objects. A more comprehensive study on feature extraction techniques for CBIR can be found in [21]. Note that references above are only examples of related work in image retrieval, not meant to be exhaustive. Extensive surveys of CBIR can be found in [17, 69, 74, 80].

1.0.2 Related work in Universal Source Coding

The recent proliferation of universal lossless data compression is undoubtedly due to the seminal papers [96, 97] by Lempel and Ziv under their theoretical underpinning of [46, 95]. The most remarkable benefit of their codes, which differentiate themselves from other universal lossless data compression methods (e.g., Huffman coding [31] and arithmetic coding [67]), is that without any prior knowledge of the statistical distribution of the given source, their algorithms asymptotically achieve a source rate approaching the entropy of the source. There have been three main streams of subsequent research on the Lempel-Ziv algorithm since then. One stream is to develop algorithms of higher coding efficiency [54, 86]. Another stream has worked on lossy compression algorithms that incorporate a distortion measure for the underlying pattern matching schemes. Instead of reconstructing the *exact* source symbols, these algorithms generate *approximate* symbols with a substantially reduced amount of coding resources. Following initial attempts made by Morita [57] and Steinberg [76] without addressing the issue of asymptotic optimality, there have been a series of studies along this line [3, 40, 42, 43, 87, 88, 92]. It is worth noting that Łuczak *et al.* demonstrated in [51] that a straightforward lossy extension of the Lempel-Ziv algorithm cannot achieve the optimal rate-distortion but the *generalized Shannon entropy*. The third direction is to design algorithms that are capable of dealing with higher dimensional discrete sources since many discrete data in a variety of media processing applications are naturally arranged as multidimensional arrays. To the best of our knowledge, the first attempt was made by Lempel and Ziv [47], followed by Sheinwald *et al.* [73]. In [47], a given two-dimensional source is linearized to fit for the use of the one-dimensional coding scheme, as many subsequent research efforts do. Nevertheless, coding optimality in a higher dimensional space has not been extensively studied.

A substantial amount of literature has also been written on lossy source coding for higher dimensional sources, mainly aiming at image compression applications [2, 6, 12, 18, 24, 65]. One promising lossy two-dimensional source coding algorithm, called two-dimensional pattern matching compression (2DPMC), was proposed in [1]. Using additional lossless compression algorithms on top of the lossy scheme, the 2DPMC was demonstrated to have a coding performance similar to JPEG [33,63] image compression with an affordable complexity. The central theme of the above effort lies in the idea of approximate pattern matching, which is essential not only in data compression but also in other media processing applications. However, most of the previous image coding research has a limited scope on pattern matching, and further understanding of multidimensional patterns and the performance of such matching algorithms is needed.

In this dissertation, while we are interested in the investigation of efficient source coding algorithms based on approximate pattern matching, here we are motivated by the ability of universal coding algorithms in adaptively capturing the source statistics (or model) through the use of a "dictionary." We consider existing lossy coding algorithms, such as JPEG [33], JPEG2000 [34], and H.264 [36], rather mature in achieving their objectives in coding efficiency. The main cue to understanding the importance of pattern matching based universal source coding is that it effectively identifies approximate repetitiveness of subsets of source symbols since the occurrence pattern has great potential for understanding and analyzing the given source, and that it compresses the source with an affordable amount of coding resources.

1.0.3 Organization of the dissertation

A detailed study on LZ78 is provided and a multidimensional incremental parsing algorithm is presented in Chapter 2. Three component schemes of the proposed parsing algorithm along the line of the three components of LZ78 are provided. The parsing algorithm is implemented into lossless and lossy image compression whose performance is compared with that of existing compression algorithms based on pattern matching. In Chapter 3, an information retrieval framework based on the source representation generated by the incremental parsing is proposed and implemented into content-based image retrieval systems. In Chapter 4, a probabilistic framework for information retrieval by formulating robust pattern matching into a semantic analysis model is presented. An in-depth analysis of trained model is also provided in the same chapter. In the two types of image retrieval systems proposed in Chapters 3 and 4, the set of query images for evaluating their performance are a portion of a given image database. In Chapter 5, we evaluate the noise robustness of the retrieval systems with perturbed query images generated from various types of distortions. Finally, in Chapter 6, we summarize the contributions to and conclusions of our research and discuss possible avenues of future work.

CHAPTER II

MULTIDIMENTIONAL INCREMENTAL PARSING FOR UNIVERSAL SOURCE CODING

2.1 Introduction

A multidimensional incremental parsing algorithm for multidimensional discrete sources, as a generalization of the Lempel-Ziv incremental parsing algorithm, is investigated in this chapter. We address three essential component schemes in designing multidimensional incremental parsing for universal source coding: the construction of a hierarchical structure for multidimensional source coding, the augmentation of dictionary, and maximum decimation matching. Next, we propose two distortion functions for maximum decimation approximate pattern matching in order to design a lossy source coding scheme. We then apply the proposed schemes in the design of lossless/lossy image compression algorithms. The performance of our algorithm is compared to that of the Lempel-Ziv-Welch (LZW) algorithm [86], its lossy extensions, and the 2DPMC. Note that the 2DPMC originally consists of three compression components, a pattern matching, an enhanced runlength coding, and an arithmetic coding. In this dissertation, the 2DPMC with only the pattern matching scheme is considered for a fair comparison.

In the next section we briefly review the Lempel-Ziv incremental parsing rule (LZ78) for one-dimensional lossless source coding. The design of multidimensional incremental parsing (MDIP) for lossless source coders is discussed and its three essential component schemes are suggested in Section 2.3. We provide two distortion functions for approximate pattern matching in order to implement universal lossy source coders for two-dimensional images in Section 2.4. A performance comparison of the suggested implementations with other pattern matching based source coders is provided in Section 2.5, and a discussion and final remarks are found in Section 2.6.

2.2 Review of Lempel-Ziv Incremental Parsing Code

An essential operation of the LZ78 algorithm is to parse the given source sequence into a number of distinct phrases and to construct a dictionary containing the previously registered patterns of symbols. Let $\mathbf{X} = \{X_i\}_{i=1}^{n}$ be a stationary ergodic sequence taking values from a finite alphabet $\mathcal{A}$ with cardinality $|\mathcal{A}| < \infty$. The LZ78 starts with an empty dictionary $\mathbb{D} = \emptyset$, finds the dictionary index j at which the encoder gives the longest match, and augments the dictionary with the last parsed phrase $\mathbb{D}_j$ appended with the next source symbol X_i at the k^{th} coding epoch, denoted by $\mathbb{D}_j \circ X_i$. It then transmits the codeword $\{j,\, X_i\}$ corresponding to the index and the symbol.

The LZ78 has three main parsing steps: pattern matching, codeword assignment, and dictionary augmentation. Since it is aiming at reconstructing the given symbols without information loss, it searches the longest dictionary entry that is *exactly* matched with the source symbols at the corresponding coding point. Then, to transmit the codeword to the decoder at each coding epoch, it spends $\lceil \log_2 \Gamma \rceil + \lceil \log_2 |\mathcal{A}| \rceil$ bits, where $\lceil x \rceil$ denotes the least integer not smaller than x, and Γ corresponds to the number of dictionary entries, or equivalently same as the number of distinct phrases. Thus, the total length of the code is

$$\mathcal{L}(\mathbf{X}) = \Gamma \cdot (\lceil \log \Gamma \rceil + \lceil \log |\mathcal{A}| \rceil). \tag{1}$$

Throughout this dissertation, $\log x = \log_2 x$. Often the decoder can efficiently synchronize the construction of the dictionary and hence the encoder is allowed to spend only $\lceil \log k \rceil + \lceil \log |\mathcal{A}| \rceil$ bits for each phrase because the number of dictionary entries at k^{th} coding epoch is precisely k. Therefore, $\mathcal{L}(\mathbf{X})$ is reduced to

$$\mathcal{L}(\mathbf{X}) = \sum_{k=1}^{\Gamma}(\lceil \log k \rceil + \lceil \log |\mathcal{A}| \rceil). \tag{2}$$

It is known that for a given stationary ergodic source, the average number of bits per symbol $\mathcal{L}(\mathbf{X})/n$ approaches the entropy of the given source $H(\mathbf{X})$ as $n \to \infty$ [14].

2.3 Design of Multidimensional Incremental Parsing

As mentioned in Section 2.1, a majority of research on multidimensional universal source coding tried to generate a one-dimensional source sequence from a given multidimensional source with the aid of a scanning scheme so that the LZ78 or equivalent coding algorithms could be employed. For example, in [47], the Peano-Hilbert plane-filling curve is used for this purpose. However, it is not clear that such a linearization method represents the most suitable method for analyzing the local property of a given source. In this section, we propose a generalized incremental parsing rule for multidimensional universal lossless source coding and introduce three essential component schemes, which are the counterparts of the three essential procedures in the LZ78 algorithm.

Suppose that we are given an m-dimensional closed convex paralleloid $\mathbf{X}$, m-paralleloid in short,

$$\mathbf{X} = \{X_{x_1,\cdots,x_m} : 0 \leq x_i < n_i,\ i = 1, \cdots, m,\ (x_1, \cdots, x_m) \in \mathbb{Z}^m\}, \tag{3}$$

where $X_{x_1,\cdots,x_m}$ takes a value from a finite alphabet $\mathcal{A}$ and n_i denotes the number of elements along the i^{th} axis y_i. Let $\vec{x}$ denote the m-dimensional index vector $(x_1, \cdots, x_m)^T$. $\mathbf{X}(\vec{x})$ denotes the symbol at $\vec{x}$ in the paralleloid, and $\mathbf{X}(\vec{0})$ denotes the symbol at the origin, presumably the top-left symbol if $m = 2$. Examples of a two- and a three-dimensional paralleloids are provided in Figure 2.

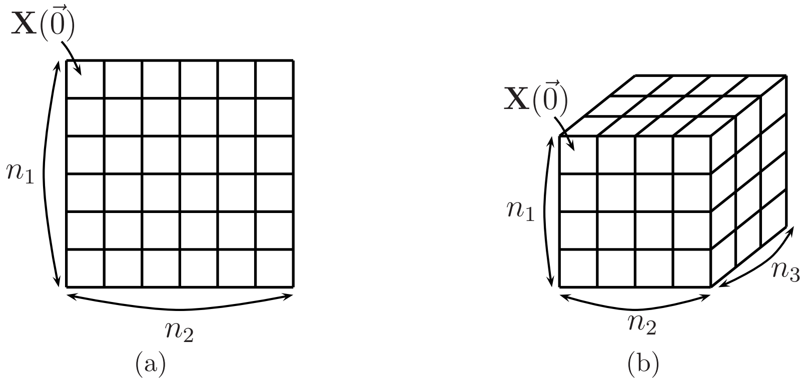


Figure 2: Examples of a two- and a three-dimensional paralleloids, sized of (n_1, n_2) and (n_1, n_2, n_3), respectively.

We now define the multidimensional *sequence patch* $\mathbf{X}(\vec{x}; \vec{a})$

$$\mathbf{X}(\vec{x}; \vec{a}) = \{X_{\bar{x}_1, \cdots, \bar{x}_m} : x_i \leq \bar{x}_i < x_i + a_i, \, i = 1, \cdots, m\}, \tag{4}$$

with an integer vector $\vec{a} = (a_1, \cdots, a_m)^T \in \mathbb{Z}^m$. Thus, in $\mathbf{X}(\vec{x}; \vec{a})$, $\vec{x}$ denotes the top-left corner, called an *anchor point*, and $\vec{a}$ denotes the dimension of the subset. In the previous example of a two-dimensional paralleloid, $\mathbf{X}(\vec{0})$ is equivalent to $\mathbf{X}(\vec{0}; (1, 1))$.

Before proceeding to the detailed algorithm, let us define two complementary sets, $\mathbf{E}$ and $\mathbf{E}^c$. If an encoder outputs any codeword for symbol $\mathbf{X}(\vec{x})$, then the $\vec{x}$ is in $\mathbf{E}$. Thus, when an encoder starts generating codewords for a given source, all $\vec{x}$ is in $\mathbf{E}^c$, not in $\mathbf{E}$. Now, we define the decimation field $\mathbf{F}(\vec{x})$ as follows:

$$\mathbf{F}(\vec{x}) = \begin{cases} 0, & \vec{x} \in \mathbf{E} \\ 1, & \vec{x} \in \mathbf{E}^c. \end{cases} \tag{5}$$

Also,

$$\mathbf{F}(\vec{x}; \vec{a}) = \{F_{\bar{x}_1, \cdots, \bar{x}_m} : x_i \leq \bar{x}_i < x_i + a_i, \, i = 1, \cdots, m\}. \tag{6}$$

Obviously, $\mathbf{F}(\vec{x})$ is equivalent to $\mathbf{F}(\vec{x}; (1, 1))$ as in the previous example.

Now, we discuss the principles of a multidimensional suffix, which define how a multidimensional patch is constructed. Suppose that we are given an m-dimensional paralleloid symbol patch with size $n_1 \times \cdots \times n_m$ in the space formed by m orthogonal axes, $y_1, \cdots, y_m$. As an example of a two-dimensional patch shown in Figure 3, the patch is prescribed to be constructed from a one-symbol patch.It then grows along the axis y_1 by appending corresponding suffixes until it becomes an $n_1 \times 1$ patch. The patch now grows along the axis y_2 by appending an $n_1 \times 1$ suffix to form an $n_1 \times n_2$ patch. In general, this procedure is repeated until the constructed patch becomes an $n_1 \times \cdots \times n_m$ patch. This principle of a multidimensional suffix helps one to avoid any *suffix ambiguity* for a given patch. With the help of this principle, one can easily construct a multidimensional suffix tree. For an example of a one-dimensional suffix tree for the LZ78, the reader is referred to [68, Ch. 2].

Now, we are in a position to comment on the multidimensional growing database model, the *dictionary* in short, which is a set of multidimensional patches. Given a dictionary $\mathbb{D}$ and a patch $\mathbf{X}(\vec{x}; \vec{a})$, we define the dictionary $\mathbb{D} \Leftarrow \mathbf{X}(\vec{x}; \vec{a})$ as the dictionary $\mathbb{D}$ augmented by $\mathbf{X}(\vec{x}; \vec{a})$. We define two additional operations $| \cdot |$ and $[\cdot]$. $|\mathbb{D}|$ denotes the number of elements of $\mathbb{D}$, and $|\mathbb{D}_j|$ refers to the number of symbols of the j^{th} patch. Also, $[\mathbb{D}_j]$ corresponds to the area vector $\vec{a}$ of the j^{th} patch element.

2.3.1 Maximum Decimation Matching

As reviewed in Section 2.2, the LZ78 consists of three major schemes: pattern matching, codeword assignment, and dictionary augmentation. We refer to the counterpart of pattern matching in the multidimensional incremental parsing code as *maximum decimation matching* (MDM). The decimation level of a match is defined as the number of symbols that can be encoded with the given match. Since it is probable that some symbols could be already encoded at a previous coding epoch, each decimation level is less than or equal to the number of symbols of each match. An MDM scheme searches the match, at which the decimation level for the given source sequence is maximized. We identify two categories of MDM: absolute MDM and approximate MDM.

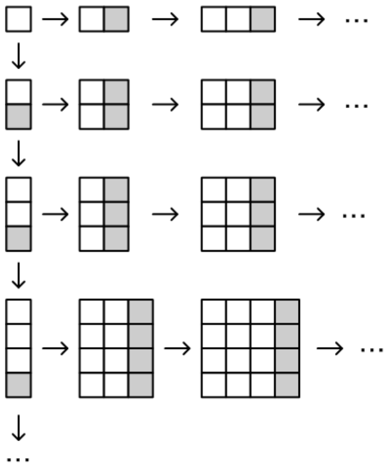
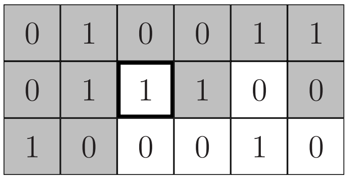


Figure 3: An $n_1 \times n_2$ patch is constituted from a single symbol patch by appending suffixes. The shaded sequences correspond to the suffix of the patch.

In the absolute MDM, at each coding epoch, the encoder finds the match at an arbitrary anchor location from the given source sequence. It inevitably involves high complexity for computing decimation levels at a large number of anchor points, while it gives the highest decimation level and requires the least number of coding epochs. Since the anchor point at each coding epoch in the absolute MDM is arbitrary, the encoder needs to spend additional bits for transmitting the anchor location information to the decoder. On the other hand, in the approximate MDM, the encoder finds the maximum decimation match at a fixed anchor point, which is specified by a predetermined heuristic scheme. Since both the encoder and the decoder can estimate each anchor point, this scheme does not require any additional bits for the anchor location.

For a better understanding of this, we consider an example of a two-dimensional binary source ($|\mathcal{A}| = 2$). Suppose that we are given a 3×6 binary source as depicted in Figure 4 and the encoder employs a raster scanning scheme to determine anchor points. At each coding epoch, the encoder attempts to find matches from the dictionary with a distortion function. Let $\rho : \mathcal{A} \times \mathcal{A} \rightarrow [0, \infty)$ be a nonnegative distortion function, and $\rho(x, y)$ equals zero if $x = y$ and one otherwise.



0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	0

Figure 4: Example of two-dimensional incremental parsing for the binary source. The encoder follows the approximate MDM criterion. The shaded area corresponds to those where matches are found and previously coded. The symbol at the anchor point (1, 2) is specified in a bold box.

Let us define the distortion function for the given two patches $\mathbf{X}$ and $\mathbf{Y}$ as follows:

$$\rho(\mathbf{X}, \mathbf{Y}) = \frac{1}{|\mathbf{X}|} \sum_{\vec{x} \in [\mathbf{X}]} \rho(\mathbf{X}(\vec{x}), \mathbf{Y}(\vec{x})). \tag{7}$$

The set of indices, at which the dictionary entry satisfies the following criterion, is obtained as

$$\mathbf{H} = \{j \mid \rho\left(\mathbb{D}_j,\, \mathbf{X}(\Delta; [\mathbb{D}_j])\right) = 0,\ \text{for } 0 \leq j < |\mathbb{D}|\}, \tag{8}$$

where Δ corresponds to the current anchor point. The index of the maximum decimation match is

$$k_{\max} = \underset{k \in \mathbf{H}}{\operatorname{argmax}} \left\{ \sum_{\vec{x} \in \mathbf{F}(\Delta; [\mathbb{D}_k])} \mathbf{F}(\vec{x}) \right\}. \tag{9}$$

Consider that the encoder now attempts to find a match at the anchor point $\Delta = (1, 2)$ in the example. By (7) and (8), the encoder finds the relevant dictionary indices $\{1, 6, 7\}$. Among the three matches, $\mathbb{D}_6$ is the maximum decimation match because the decimation levels of the three matches are 1, 2, and 1 for $\mathbb{D}_1$, $\mathbb{D}_6$, and $\mathbb{D}_7$, respectively. It then transfers the index of the match using $\lceil \log |\mathbb{D}| \rceil$ bits with the set of augmentative symbols using overhead bits. However, different from the original one-dimensional Lempel-Ziv algorithm, it is required to have additional schemes for transmitting the set of augmentative symbols in multidimensional cases.

Table 1: Anchor points, match indices, and dictionary entries generated at each coding epoch from the example of binary source in Figure 4.

Coding Epoch	Anchor Point	Match Index	Dictionary Index	Dictionary Entry
0	(0,0)	null	0	0
1	(0,1)	null	1	1
2	(0,2)	0	2	$\begin{bmatrix}0\\1\end{bmatrix}$
			3	$\begin{bmatrix}0&0\end{bmatrix}$
3	(0,3)	2	4	$\begin{bmatrix}0\\1\\0\end{bmatrix}$
			5	$\begin{bmatrix}0&1\\1&0\end{bmatrix}$
4	(0,4)	1	6	$\begin{bmatrix}1\\0\end{bmatrix}$
			7	$\begin{bmatrix}1&1\end{bmatrix}$
5	(0,5)	6	8	$\begin{bmatrix}1\\0\\0\end{bmatrix}$
6	(1,0)	5	9	$\begin{bmatrix}0&1&1\\1&0&0\end{bmatrix}$

2.3.2 Hierarchical Structure of Multidimensional Incremental Parsing

Remember that the LZ78 encoder generates a symbol phrase formed by appending the next source symbol to the last parsed phrase and augments the dictionary with the phrase. It then transmits one augmentor sequence at a rate proportional to the size of the alphabet. This is for adapting the dictionary to a given source along the sequence so that it achieves a longer match with the dictionary entries at remaining coding epochs. For an m-dimensional source sequence, the encoder is prescribed to generate m augmentative entries along all the m axes at each coding epoch. In the previous example, at the third coding epoch, the match found is “0” and the encoder augmented the dictionary with the two entries, $[0\ 1]^T$ and $[0\ 0]$.

However, if the number of dimension is high and the found match is considerably large, the encoder needs to transmit the necessary number of bits for the augmentative symbols. For instance, the given source is of three dimensions with a size of $10 \times 10 \times 10$ and the encoder finds the maximum decimation match with a size of

$3 \times 3 \times 3$ at the current anchor point. In this case, as generally depicted in Figure 5, it obviously has three augmentative sets of symbols that have nine symbols along one axis, respectively. Thus, at this epoch, without encoding, the encoder would need to transmit 27 symbols using $\lceil 27 \cdot \log |\mathcal{A}| \rceil$ bits as well as the dictionary index. Obviously, the augmentation symbols need proper coding. We identify a general approach to resolving this problem: one should employ an (m-1)-dimensional incremental parsing code to generate codewords for the augmentative set of symbols in the m-dimensional incremental parsing code. Repeatedly, the (m-1)-dimensional algorithm is to be operated with the help of the (m-2)-dimensional algorithm and so on. Therefore, we hence suggest that the m-dimensional incremental parsing code be hierarchically structured on a basis of all the lower dimensional parsing algorithms. This procedure is reminiscent of marginalization of the probability space.

2.3.3 Dictionary Augmentation

Although there is no parameterized probability model associated with any variable-to-fixed length code such as the LZ78, as previously addressed, the number of encoded bits per source symbol asymptotically achieves the source entropy for a stationary ergodic source. This is due to the parsing of the source symbols, which is the fundamental behavior of a universal variable-to-fixed length code. Such a code for one-dimensional sequences parses a given source sequence into variable length phrases and then assigns a codeword that has a fixed length. Basically the parsing followed by the dictionary augmentation incorporates the estimation of the probability of the given sequence by taking advantage of the frequency of occurrences of source sequences.

For a more thorough understanding of the estimation of source probability estimation, let us consider the construction of a suffix tree. Because of the principle of the multidimensional suffix provided earlier in this section, one can represent the dictionary by a multidimensional sequence tree in a tractable fashion.

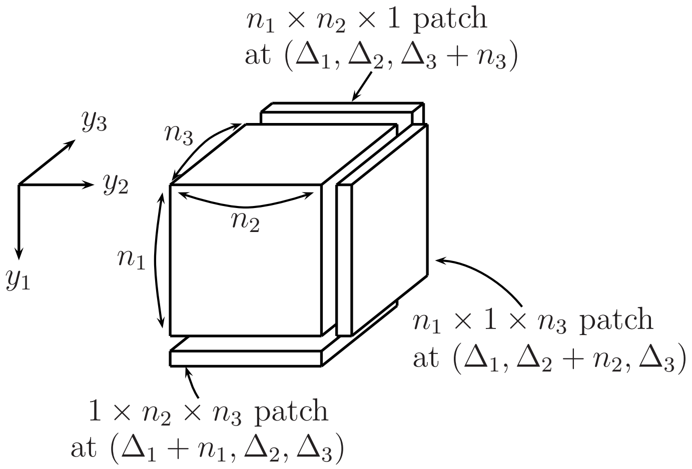

$n_1 \times n_2 \times 1$ patch
at $(\Delta_1, \Delta_2, \Delta_3 + n_3)$
y_3
n_3
y_2
n_2
n_1
y_1
$n_1 \times 1 \times n_3$ patch
at $(\Delta_1, \Delta_2 + n_2, \Delta_3)$
$1 \times n_2 \times n_3$ patch
at $(\Delta_1 + n_1, \Delta_2, \Delta_3)$

Figure 5: Three-dimensional symbol patch with a size of $n_1 \times n_2 \times n_3$ has three augmentative set of symbols that is to be appended into the dictionary. The sequence space is formed by three orthogonal axes, y_1, y_2, and y_3. The anchor point is at $(\Delta_1, \Delta_2, \Delta_3)$.

Let π be a node corresponding to an m-dimensional patch in a suffix tree, the root node π_0 has the null patch, and $\gamma_d(\pi)$ denotes the number of descendent nodes at π. According to the principles of the multidimensional suffix, each patch can append symbols along the y_1 axis first, and then along the y_2 and the other axes sequentially. Thus, for example, a $2 \times 2 \times 1$ patch in three-dimensional source symbols can have descendent nodes with $2 \times 3 \times 1$ or $2 \times 2 \times 2$ patches. However, it is not allowed to have descendent nodes with $3 \times 2 \times 1$ because the patch grew along the y_1 axis. The number of new augmentative symbols along y_i axis is $\prod_{k=1}^{i-1} n_k$, and the maximum number of descendent nodes at node π is

$$\gamma_m(\pi) = \sum_{i=c_m}^{m} |\mathcal{A}|^{\prod_{k=1}^{i-1} n_k}, \tag{10}$$

where $c_m = \min\{c \mid n_i = 1 \text{ for } 1 \leq c < i < m+1,\ c \in \mathbb{Z},\ i \in \mathbb{Z}\}$ is the index of axis to which the patch can append symbols along. At the k^{th} coding epoch, $\Gamma_m(k)$ is the total number of augmented nodes, and $\Gamma_f(k)$ is the total number of fully augmented nodes at which $\gamma_d(\pi)$ is equivalent to $\gamma_m(\pi)$. When a new node π' is appended to the dictionary $\mathbb{D}$, the encoder finds the deepest node π_j, which can only have π' as a descendent node. After appending the given node to $\mathbb{D}$, it then updates the variables

$\gamma_d(\pi')$, $\gamma_m(\pi')$, $\gamma_d(\pi_j)$, $\Gamma_m(k)$, and $\Gamma_f(k)$. The complete algorithm of the construction of a multidimensional suffix tree is presented in Algorithm 1.

Algorithm 1 Construction of a multidimensional suffix tree

```
**Initialize:** $\Gamma_m(0) \leftarrow 1$, $\Gamma_f(0) \leftarrow 0$, $k \leftarrow 0$, $\mathbb{D} \leftarrow \emptyset$
**Input:** a new node $\pi'$
$k \leftarrow k + 1$
Find the deepest node $\pi_j$
$\mathbb{D} \Leftarrow \pi'$ under $\pi_j$
$\gamma_m(\pi') \leftarrow$ by (10), $\gamma_d(\pi') \leftarrow 0$
$\gamma_d(\pi_j) \leftarrow \gamma_d(\pi_j) + 1$, $\Gamma_m(k) \leftarrow \Gamma_m(k) + 1$
**if** $\gamma_d(\pi_j)$=$\gamma_m(\pi_j)$ **then**
  $\Gamma_f(k) \leftarrow \Gamma_f(k) + 1$
**end if**
```

The suffix tree corresponding to the example of a two-dimensional binary source given in Figure 4 is provided in Figure 6.

When a new patch π' is required to be appended to $\mathbb{D}$, it is obvious that π' can be a descendent of one of [0], $[0\ 1]^T$, [0 0], $[0\ 1\ 0]^T$, $[0\ 1; 1\ 0]$, $[0\ 1; 1\ 0; 1\ 1]$, [1], $[1\ 0]^T$, [1 1], and $[1\ 0\ 0]^T$. One can notice that no node can be a descendent of the root node because it is fully augmented. The number of candidate nodes, one of which will be an ascendent node of π', is easily computed by $\Gamma_m(k) - \Gamma_f(k)$, denoted by $\Gamma_p(k)$. Finally, the probability of a π' given the concurrent $\mathbb{D}$ is

$$P(\pi' \mid \mathbb{D}) = P(\pi' \mid \pi_{\Gamma_m - 1}, \cdots, \pi_1, \pi_0) = \frac{1}{\Gamma_p(k)} \quad . \tag{11}$$

In this example, $P(\pi' \mid \mathbb{D}) = 1/10$.

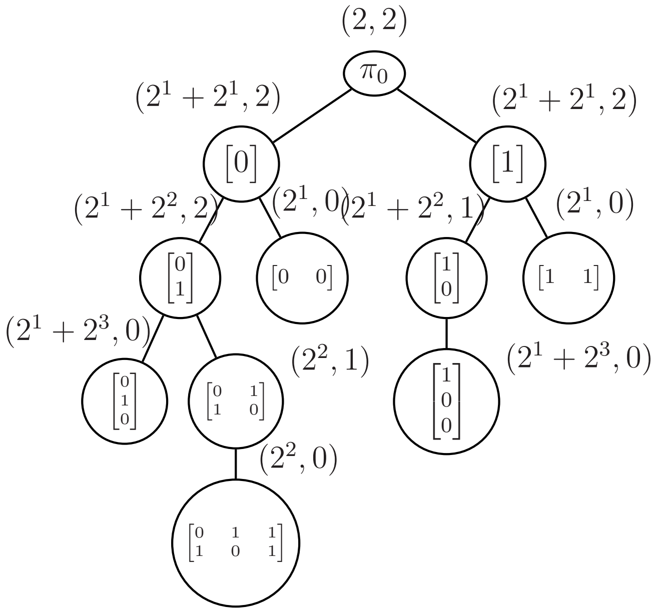

$(2,2)$
π_0
$(2^1+2^1,2)$
$(2^1+2^1,2)$
$[0]$
$[1]$
$(2^1+2^2,2)$
$(2^1,0)$
$(2^1+2^2,1)$
$(2^1,0)$
$\begin{bmatrix}0\\1\end{bmatrix}$
$\begin{bmatrix}0 & 0\end{bmatrix}$
$\begin{bmatrix}1\\0\end{bmatrix}$
$\begin{bmatrix}1 & 1\end{bmatrix}$
$(2^1+2^3,0)$
$(2^2,1)$
$(2^1+2^3,0)$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$(2^2,0)$
$\begin{bmatrix}0 & 1 & 1\\1 & 0 & 1\end{bmatrix}$

Figure 6: The suffix tree constructed by the two-dimensional binary source given in Figure 4. As can be seen, this is equivalent to the tree-structured dictionary of Table 1. The pair of numbers next to each node is $(\gamma_m(\pi), \gamma_d(\pi))$. At the current coding epoch, $\Gamma_m(k)$ is 11, and $\Gamma_f(k)$ is 1 because of the root node π_0.

Finally, the probability of the given multidimensional source $\mathbf{X}$ is computed as

$$\begin{aligned} P(\mathbf{X}) &= p(\pi_\Gamma \mid \pi_{\Gamma-1}, \cdots, \pi_0) p(\pi_{\Gamma-1}, \cdots, \pi_0) \\ &= p(\pi_\Gamma \mid \pi_{\Gamma-1}, \cdots, \pi_0)\, p(\pi_{\Gamma-1} \mid \pi_{\Gamma-2}, \cdots, \pi_0) \\ &\quad \cdots p(\pi_2|\pi_1, \pi_0)\, p(\pi_1|\pi_0)\, p(\pi_0) \\ &= \prod_{k=1}^{\Gamma} \frac{1}{\Gamma_p(k)} \quad , \end{aligned} \tag{12}$$

where Γ is the total number of distinct patches for the given $\mathbf{X}$. Consequently, the self information of $\mathbf{X}$ is estimated as

$$I(\mathbf{X}) = -\log_2 \prod_{k=1}^{\Gamma} \frac{1}{\Gamma_p(k)} \quad \text{(bits)}. \tag{13}$$

Summarizing all three essential component schemes, we now look at the entire algorithm for the multidimensional incremental parsing code. For a given source $\mathbf{X}$, the encoder first finds the maximum decimation match at the current anchor point Δ. It transmits codewords for the match to a decoder. While encoding the augmentative patches with lower dimensional parsing codes, the encoder augments the dictionary with the patches. The complete algorithm is given in Algorithm 2.

Algorithm 2 Multidimensional incremental parsing for lossless source coding

```
Input: source sequence $\mathbf{X}$, source dimension $\vec{n} = [n_1 \cdots n_m]^T$
Output: encoded bit sequence $\mathbb{B}$
  $\Delta \leftarrow \vec{0}$, $\mathbb{D} \leftarrow \emptyset$, $\mathbf{F}(\cdot) = 1$
  while $\Delta < \vec{n}$ do
      $\mathbf{H} \leftarrow$ eq. (8)
      if $\mathbf{H}$ is not null then {Maximum Decimation Matching}
          $\mathbb{B} \leftarrow k_{\max}$ by eq. (9), $\vec{a} \leftarrow [\mathbb{D}_{k_{\max}}]$
      else
          $\mathbb{B} \leftarrow \{\text{null}\}$, $\vec{a} \leftarrow \vec{1}$
      end if
      for $i = 1$ to $m$ do
          $a_i \leftarrow a_i + 1$
          Encode new symbols in $\mathbf{X}(\Delta; \vec{a})$ by Algorithm 2 {Hierarchical Encoding}
          $\mathbb{D} \Leftarrow \mathbf{X}(\Delta; \vec{a})$ by Algorithm 1 {Dictionary Augmentation}
      end for
      $\mathbf{F}(\Delta; [\mathbb{D}_{k_{\max}}]) = 0$ {Decimation Field Update}
      Move $\Delta$ somewhere $\mathbf{F}(\Delta) = 1$
  end while
```

2.4 Implementation of Universal Lossy Source Coder

In this section, we set our focus on universal lossy source coding by extension of the multidimensional incremental parsing code suggested in Section 2.3. Using an approximate pattern matching algorithm, the encoder can search such matches that have more symbols than those by the absolute pattern matching, and it retains fewer dictionary entries during the coding process. As a result, it may require fewer coding iterations as well as lower computing resources such as the number of bits, the amount of memory, and the amount of computation. However, it inevitably involves a distortion between the original symbols and the approximate symbols.

Design of a universal lossy source coder for a multidimensional source can be achieved by redefining the schemes from the incremental parsing code. First, with a given distortion bound ε, we redefine the approximate pattern matching as follows:

$$\mathbf{H} = \{j \mid \rho\left(\mathbb{D}_j, \mathbf{X}(\Delta; [\mathbb{D}_j])\right) < \varepsilon,\ 0 \le j < |\mathbb{D}|,\ \varepsilon \in \mathbb{R}_+\}\,, \tag{14}$$

where ρ is a distortion function. We now propose two classes of distortion functions for approximate pattern matching, the local average distortion and the local minimax distortion. In order to construct a general framework, we employ a threshold τ for each source symbol. τ can be understood to represent a certain perceptual significance of the corresponding symbol. If we design a compression algorithm for such signals that are to be consumed by humans, τ might be the perceptual threshold. Given two sets of patches, the reference patch $\mathbf{X}$ and the approximate patch $\hat{\mathbf{X}}$, the local average distortion is

$$\rho_a\left(\mathbf{X}, \hat{\mathbf{X}}\right) = \frac{1}{|\mathbf{X}|}\left(\sum_{\vec{x}\in[\mathbf{X}]}\left|\frac{\max\{0,\, |\mathbf{X}(\vec{x}) - \hat{\mathbf{X}}(\vec{x})| - \tau(\vec{x})\}}{\tau(\vec{x})}\right|^p\right)^{1/p}\,, \tag{15}$$

where p is a real number. If the distance between the two symbols, $|\mathbf{X}(\vec{x}) - \hat{\mathbf{X}}(\vec{x})|$, is lower than the threshold $\tau(\vec{x})$, it is considered that $\hat{\mathbf{X}}(\vec{x})$ is not distorted, even though $|\mathbf{X}(\vec{x}) - \hat{\mathbf{X}}(\vec{x})| > 0$. The average of each distortion for each symbol is then computed in a similar fashion to a weighted L_p norm. The set of indices whose dictionary patches have ε_a-bounded distortions at the k^{th} coding epoch is

$$\mathbf{H}_a = \{j \mid \rho_a\left(\mathbb{D}_j, \mathbf{X}(\Delta_k; [\mathbb{D}_j])\right) \le \varepsilon_a,\ 0 \le j < |\mathbb{D}|,\ \varepsilon_a \in \mathbb{R}_+\}\,. \tag{16}$$

Using the same threshold $\tau(\vec{x})$, the local minimax distortion is defined as

$$\rho_m\left(\mathbf{X}, \hat{\mathbf{X}}\right) = \max_{\vec{x}\in[\mathbf{X}]}\left\{\max\left(0, \frac{|\mathbf{X}(\vec{x}) - \hat{\mathbf{X}}(\vec{x})| - \tau(\vec{x})}{\tau(\vec{x})}\right)\right\}. \tag{17}$$

The set of indices by ε_m-bounded distortion at the k^{th} coding epoch is

$$\mathbf{H}_m = \{j \mid \rho_m\left(\mathbb{D}_j, \mathbf{X}(\Delta_k; [\mathbb{D}_j])\right) \le \varepsilon_m,\ 0 \le j < |\mathbb{D}|,\ \varepsilon_m \in \mathbb{R}_+\}\,, \tag{18}$$

where ε_m is typically set to 0.

At each coding epoch, the encoder constructs the set of indices with respect to the distortion criterion; then it measures the decimation level of each candidate patch. Finally, it selects the match of the index $k_{\max}$ that gives the highest level of decimation by (9). Since the overall algorithm is similar to Algorithm 2, the details are omitted.

2.5 Experimental Results

In this section, we implement both lossless and lossy image compression algorithms to evaluate the performance of the universal source coder described in Section 2.3 and Section 2.4. For both image compression experiments, we use gray images with 256 quantization levels (i.e., $|\mathcal{A}| = 256$). For a better compression efficiency, we make two modifications: the dictionary initially contains 256 entries covering all the pixel values, and the encoder does not transmit any information about the augmentative symbols. These variations can be viewed as those similar to LZW. Since a decoder eventually receives all the pixel information, it can reconstruct the source symbols in a lossless manner with memory manipulations.

2.5.1 Lossless Image Compression

In this experiment, we compare, in terms of the coding performance, a two-dimensional implementation of the MDIP algorithm with the LZW algorithm, which is a one-dimensional variable-to-fixed length code based on pattern matching. In order to facilitate the use of LZW, the given image is vectorized into a long one-dimensional vector through concatenation of the columns. Although results are not provided here, we have tried other scanning methods such as row-wise reading and the Peano-Hilbert curve [47]. We could not observe any statistical correlation between the scanning method and the resultant images in terms of coding efficiency, which is also reported in [3, Section 4], [2]. At each coding epoch, the anchor point Δ moves in a similar manner to the raster scan. If the symbol at the next anchor point is already encoded, those symbols are skipped so that $\mathbf{F}(\Delta) = 1$ at any time.

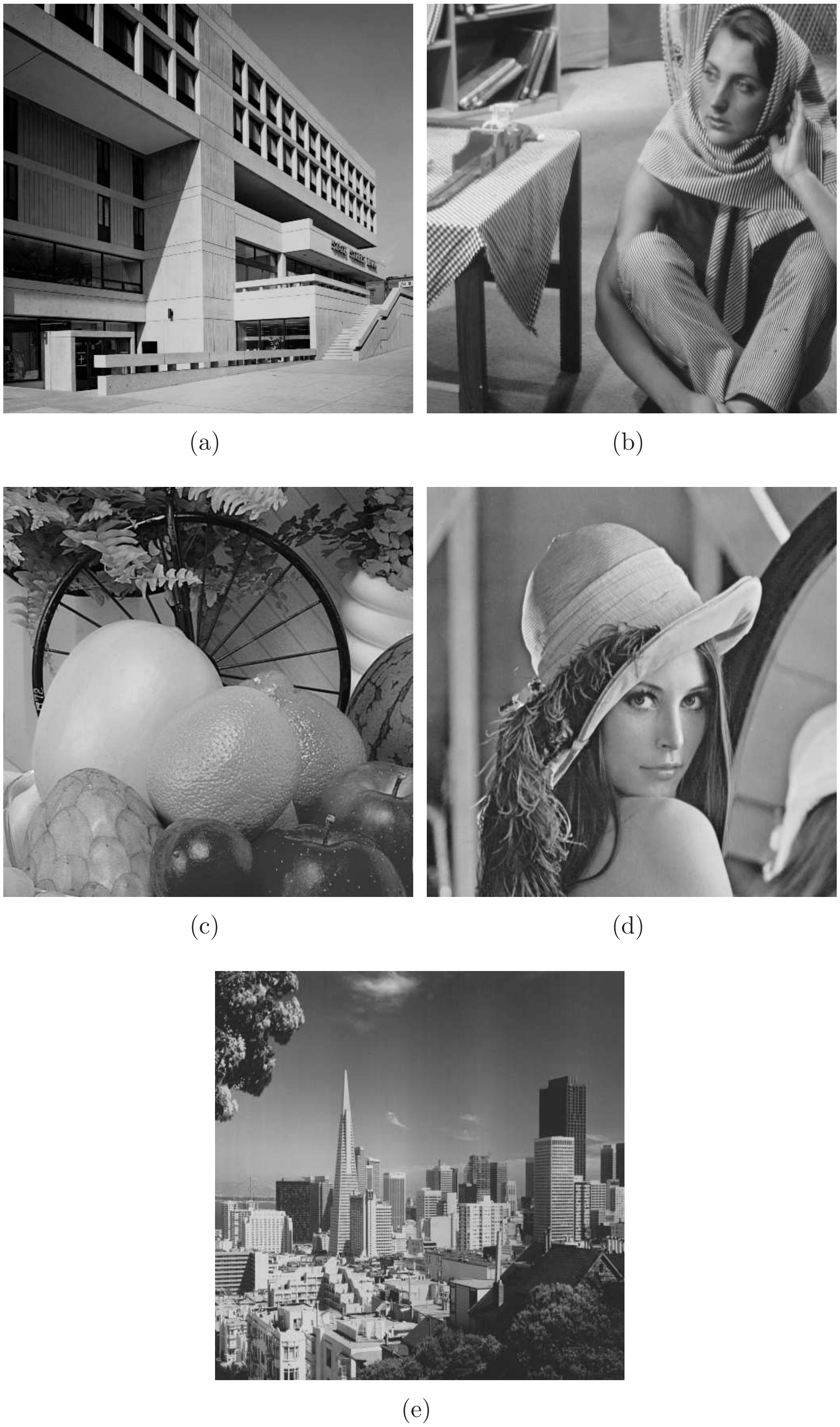
We have tested a number of images, but results of five of them are provided here. The test images are shown in Figure 7. Table 2 shows the comparison of the two algorithms. The dimensions and the entropies of the images are first given. The average bitrates $\mathcal{L}(\mathbf{X})/n$ and $\mathcal{L}(\mathbf{X})/|\mathbf{X}|$, the number of distinct patches Γ, the number of dictionary entries $|\mathbb{D}|$, and the number of referred dictionary entries $|\mathbb{D}|^*$ for both coding algorithms are provided.

We observe that, for both cases, the average bitrates of the MDIP are higher than those of the LZW. This is mainly because the MDIP appends its dictionary with twice as many entries at each coding epoch. For the image "Bank," the dictionary of the MDIP contains approximately 52% more entries than that of the LZW. However, another comparison of $|\mathbb{D}^*|$ attracts our attention because the higher number of referred entries implies that the MDIP scheme can efficiently capture the structure of the source regardless of its redundancy. Figure 8 shows the average bitrates in the encoding process of both coding algorithms.

Table 2: The lossless compression results of the MDIP are compared to those of the LZW with a columwise linearization method. $H(\mathcal{X})$ denotes the one-dimensional entropy rate of a given image. Γ denotes the number of distinct patches. $|\mathbb{D}|^*$ is the number of referred dictionary entries.

Images		Bank	Barbara	Bike	Lena
Dimension		512^2	512^2	512^2	512^2
$H(\mathcal{X})$ (bits/symbol)		7.66	7.63	7.61	7.45
LZW	$\mathcal{L}(\mathbf{X})/n$	6.63	7.45	7.07	6.57
	Γ	109788	122416	116572	108877
	$\lvert\mathbb{D}\rvert$	110042	122671	116728	109131
	$\lvert\mathbb{D}\rvert^*$	31547	31944	30956	31441
MDIP	$\mathcal{L}(\mathbf{X})/\lvert\mathbf{X}\rvert$	6.88	7.90	7.27	7.02
	Γ	109436	123759	116752	111371
	$\lvert\mathbb{D}\rvert$	167683	178202	172013	170276
	$\lvert\mathbb{D}\rvert^*$	34022	34878	33784	33603

If the number of symbols is relatively small, the average bitrates for the MDIP are considerably lower than for the LZW, which implies that the inherent ability of the MDIP for source characterization outperforms that of the LZW.

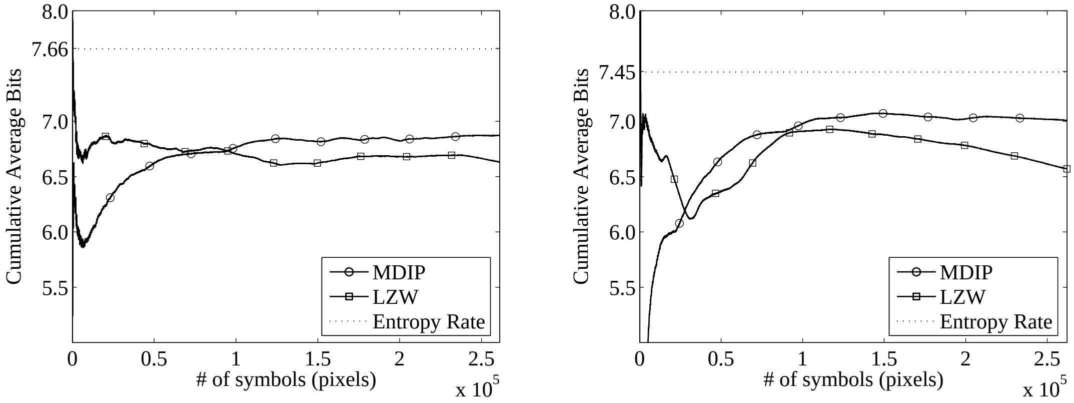


(a) (b)

(c) (d)

(e)

Figure 7: Standard images used in the experiments. Quantization levels are 256. (a), (b), (c), and (d) are used for the experiments of lossless compression. (a), (c), (d), and (e) are for those of lossy compression. (a) Bank, (b) Barbara, (c) Bike, (d) Lena, (e) San Francisco.



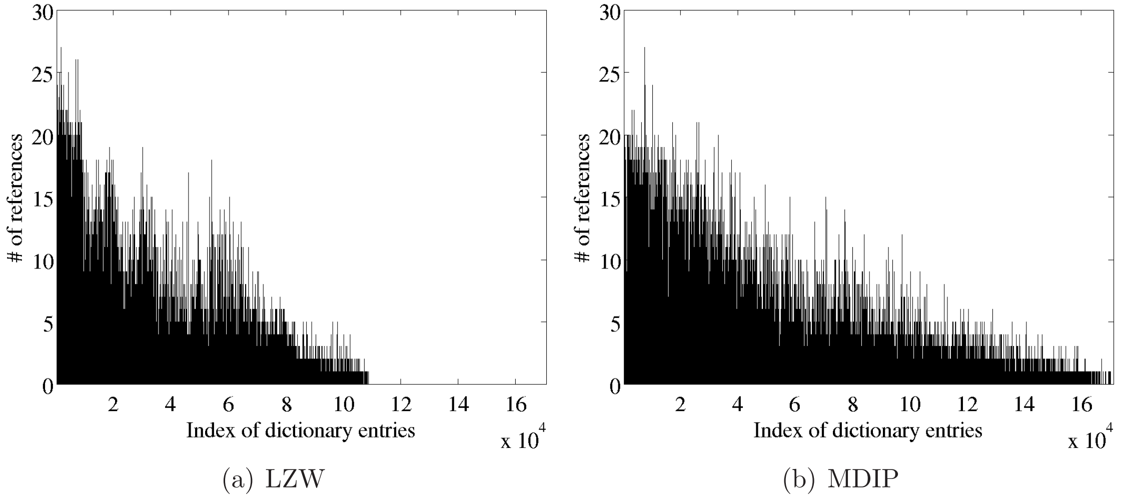
(a) Bank (b) Lena

Figure 8: Average bitrates as the encoding proceeds. (a) is of the image "Bank," (b) is of the image "Lena."

Figure 9 depicts the histogram of the reference rates of the encoder outputs to the dictionary entries. Since the reference rate for the source alphabets is too high, the first 256 entries are excluded in these figures. We notice that the two encoder outputs are obviously not equiprobable; the empirical distribution of the outputs of the LZW are 0.474 and 0.526 for 0's and 1's, respectively, and those of the MDIP are 0.475 and 0.525. If one wants to achieve a higher coding rate, employing additional lossless coding schemes can be considered.

2.5.2 Lossy Image Compression

Based on the framework of the universal lossy source coder described in Section 2.4, we now design image compression algorithms with the two distortion functions in order to evaluate the performance of the MDIP. Regarding the local average distortion (15), p is set to 2 so that it becomes a weighted L_2 norm. By changing ε_a, the compression algorithm upperbounds the distortion of each approximate pattern matching. For the local minimax distortion (17), ε_m is set to 0 as typical. It is worth noting that many lossy universal source coders, such as [18,24,42], compute the distortion over symbols for corresponding pattern matching by averaging local distortions.



(a) LZW

(b) MDIP

Figure 9: Reference rate of the dictionary entries for the LZW and the MDIP. Note that the first 256 entries of each dictionary are excluded. Test image is "Lena." (a) is by the LZW and (b) is by the MDIP.

It is known that the average distortion is not a good model in the sense of human visual perception because human eyes are more sensitive to salient image regions [62]. Thus, the comparison of the compression algorithms guided by the two different distortion functions in various distortion measures is meaningful only in the sense of image compression. Regarding the noise threshold $\tau(\vec{x})$, we introduce the just noticeable distortion (JND) model proposed by Chou *et al.* [13], in which they consider three human visual sensitivity models: the base sensitivity, the luminance sensitivity, and the texture sensitivity. For both distortion metrics, any distortion below the JND at each pixel location is considered invisible. In order to compare the reconstructed images with the original, we introduce two types of image fidelity measures: the mean structural similarity (MSSIM) [82] and the peak-signal-to-noise ratio (PSNR). Because of the manipulation of the structural information, the MSSIM is known to be effective for measuring suprathreshold compression distortions. The range of the MSSIM value is from 0.0 to 1.0, where 0.0 corresponds to a total loss of all structural similarity and 1.0 to having a noise transparent image. The PSNR is traditionally derived from the mean-squared error (MSE) as

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{(2^q - 1)^2}{\mathrm{MSE}} \right), \tag{19}$$

where $\mathrm{MSE} = \frac{1}{n} \sum_i^n (x_i - \hat{x}_i)^2$ and q is the number of quantization bits for pixel. Similar to the setup for lossless image compression, we set $|\mathcal{A}| = 256$ and the anchor point Δ sweeps a given image from top-left to bottom-right in a similar manner to the raster scan. In this experiment, we use four test images, "Bank," "Bike," "Lena," and "San Francisco." [1] For performance comparison, we also implemented lossy extensions of the LZW with the similar distortion measures as in the MDIP.

Figure 10 shows the comparison of the five compression algorithms: two from the MDIP, two from the LZW, and the 2DPMC. The top three figures, which are of PSNR, imply that the MDIP-A encodes a given image with minimum signal distortion at the same bitrate. However, as the bottom three figures show, the MDIP-M gives minimum perceptual distortion although its PSNR is lower than that of MDIP-A. Figures 11-14 show the images by the MDIP-M, MDIP-A, and 2DPMC. From the images by the MDIP-A, one can observe block artifacts on flat regions, while the corresponding PSNRs are still higher than those by the MDIP-M. In all, these results indicate that the MDIP algorithms make better use of the source statistics that are captured into the dictionary.   Figure 15 illustrates the estimated bitrates computed during the encoding procedure at the target fidelity. The estimated bitrate of the MDIP-M decreases rapidly as in the initial coding stage and does not show any sharp change so that the eventual bitrate is almost 1.5 bpp lower than the 2DPMC.

[1] Observe that the "Lena" image used for performance analysis in [1] is not consistent with the image used in this dissertation. This incongruity resulted in different coding performances from those provided in the dissertation. The images in this dissertation and in [1] can be retrieved from http://www.ece.rice.edu/∼wakin/images/ and http://www.cs.purdue.edu/homes/spa/Compression/2D-PMC.html, respectively.

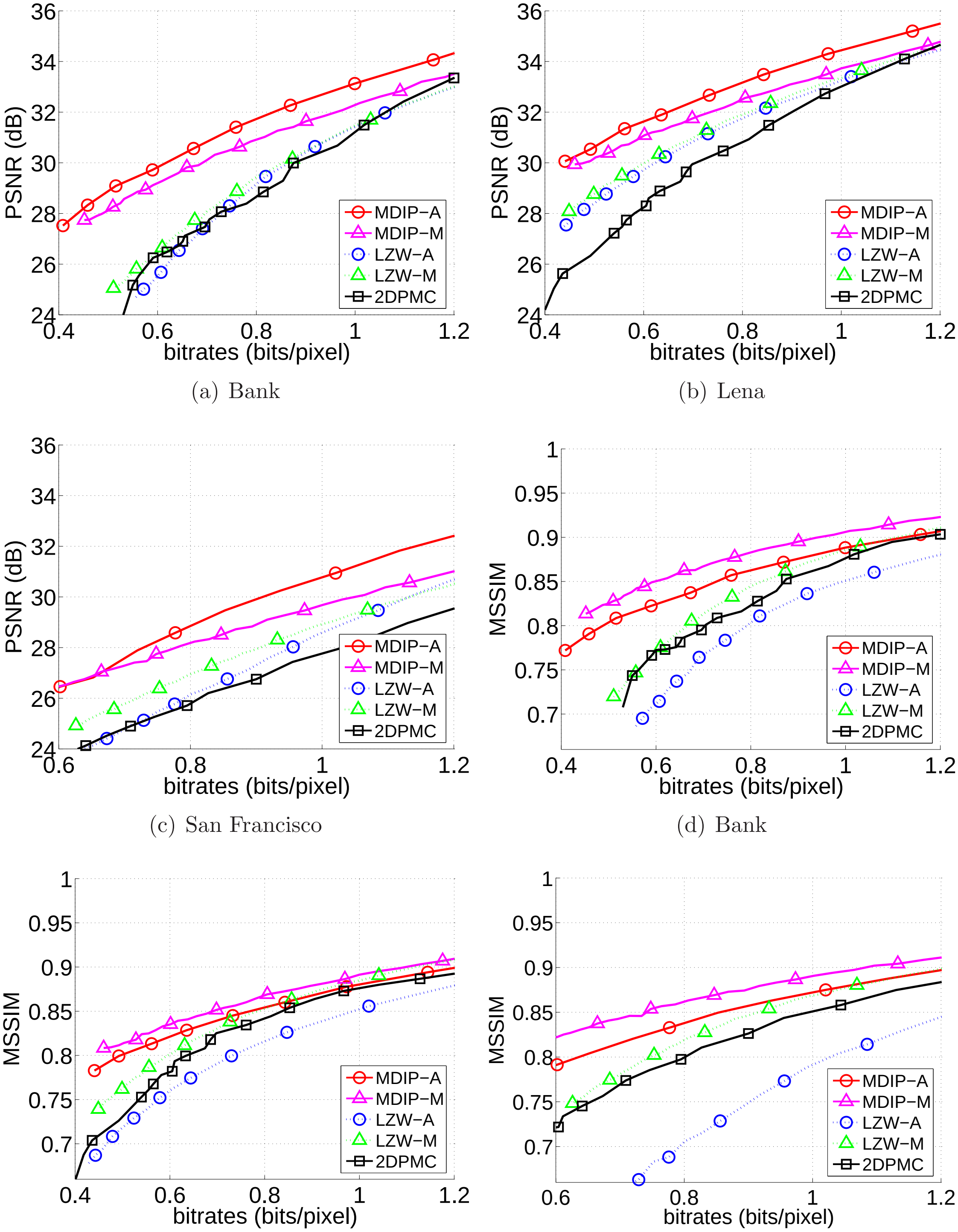
2.6 Discussions

In this chapter, we present a framework for multidimensional incremental parsing, which is a generalization of the Lempel-Ziv incremental parsing algorithm. We present three essential component schemes in the design of multidimensional incremental parsing for universal lossless source coding; namely, the hierarchical structure of multidimensional source coding, the dictionary augmentation, and the maximum decimation matching.

To evaluate the performance of the proposed algorithm for lossless compression, we compared it with an existing universal source coding algorithm, the LZW, which works for one-dimensional discrete sequences. The result shows that the coding efficiency of the image compression algorithm based on the MDIP is behind the LZW. Nevertheless, it is observed that the MDIP scheme can efficiently capture the structure of the source.

By giving two types of the distortion functions, the local average distortion and the local minimax distortion (corresponding to a weighted L_2 norm and the L_∞ norm), a framework for lossy extension of the MDIP algorithm is proposed. In lossy image compression experiments, it is shown that the proposed MDIP outperforms a state-of-the-art image compression algorithm based on two-dimensional pattern matching, 2DPMC in terms of signal distortion and perceptual fidelity. In almost all cases, the MDIP with the local average distortion criterion gives the highest PSNR among the five lossy image compression algorithms. The images by MDIP with the local minimax distortion criterion, nevertheless, have the best perceptual fidelity among all, which are evaluated by the object image fidelity metric MSSIM.

Although the experiments provided here are mainly on image compression, the fundamental framework of MDIP can also lead to the design of higher dimensional universal lossless/lossy source coding algorithms. To the best of our knowledge, no general framework of the Lempel-Ziv incremental parsing for a higher dimensional source has been reported so far. Although the image compression results shown in this chapter are not comparable to the state-of-the-art image compression algorithms, such as JPEG or JPEG2000, the properties of the MDIP addressed in this chapter point to worthwhile considerations not only in data compression but also in data modeling and feature extraction.



(a) Bank

(b) Lena

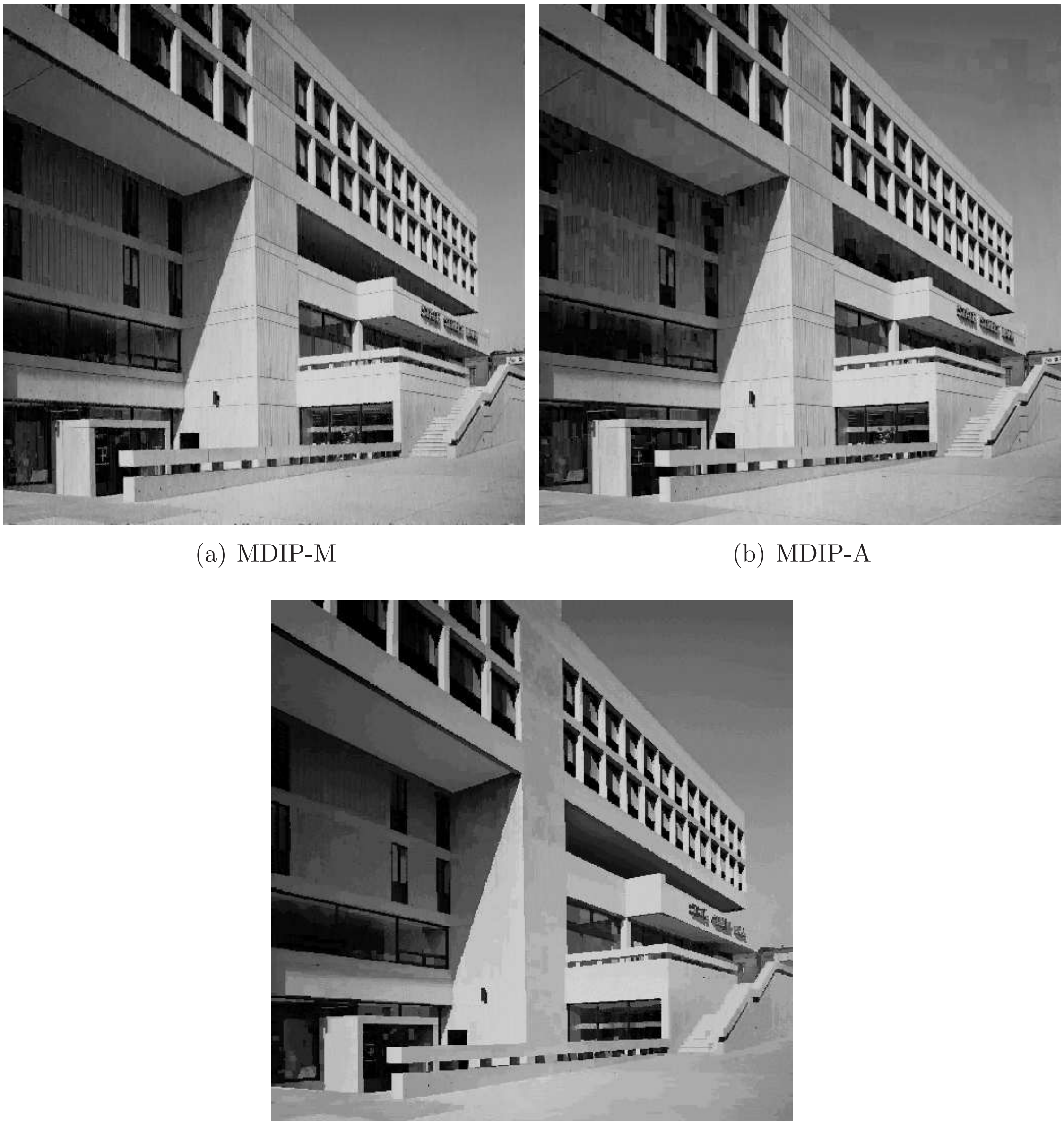
(c) San Francisco

(d) Bank

(e) Lena

(f) San Francisco

Figure 10: Comparison of compression performances of five algorithms. MDIP-A, MDIP-M, LZW-A, and LZW-M correspond to the MDIP with the local average and the local minimax distortion, and the LZW with the local average and the local minimax distortion, respectively. (a) and (d) are the results of "Bank," (b) and (e) are of "Lena," (c) and (f) are of "San Francisco."

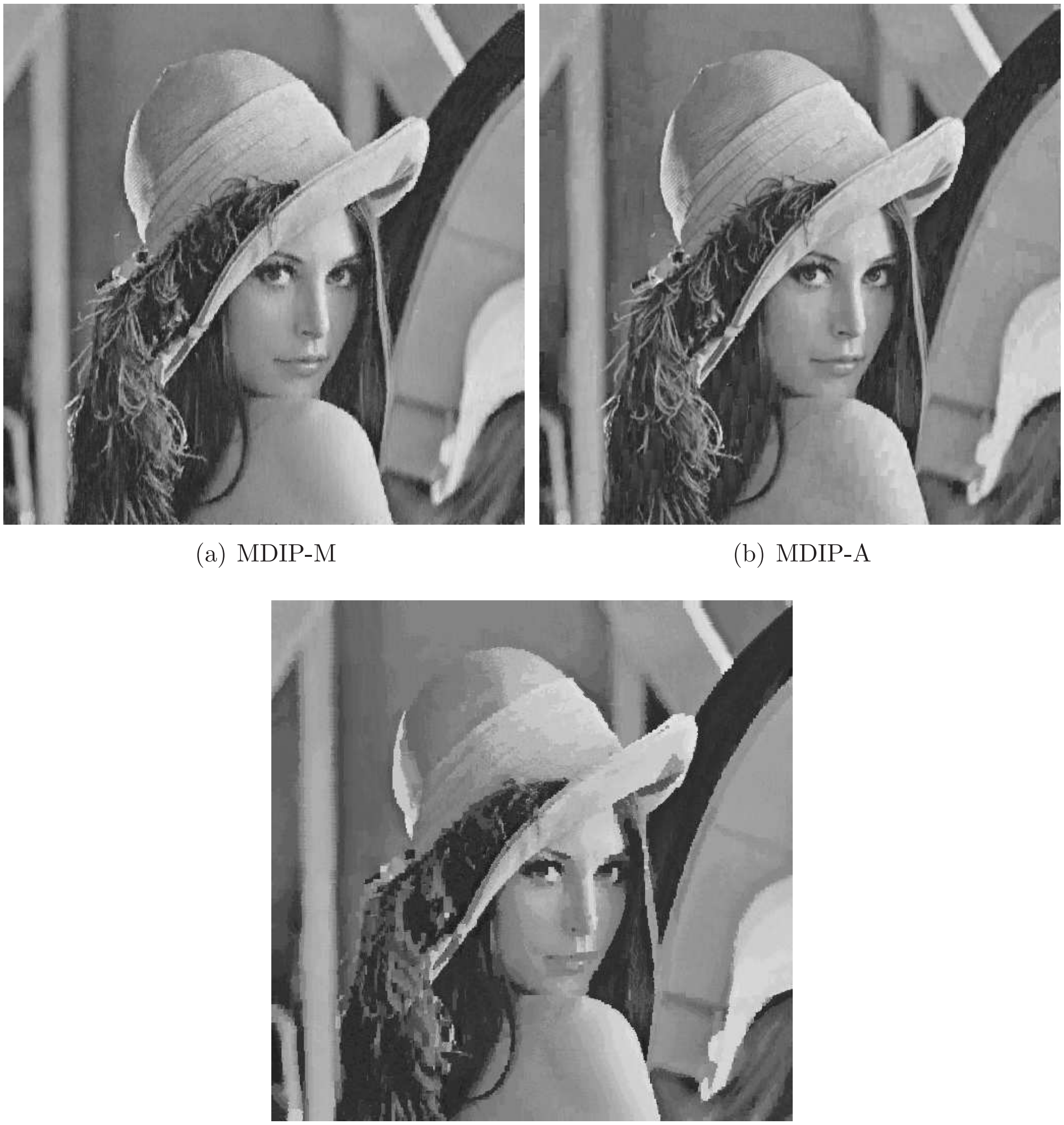


(a) MDIP-M

(b) MDIP-A

(c) 2DPMC

Figure 11: (a) is the result of minimax distortion at 0.60 bpp with PSNR=29.27dB and MSSIM=0.8516. (b) is of the average distortion at 0.60 bpp with PSNR=29.96dB and MSSIM=0.8248. (c) is of the 2DPMC at 0.60 bpp with PSNR=26.32dB and MSSIM=0.7725.

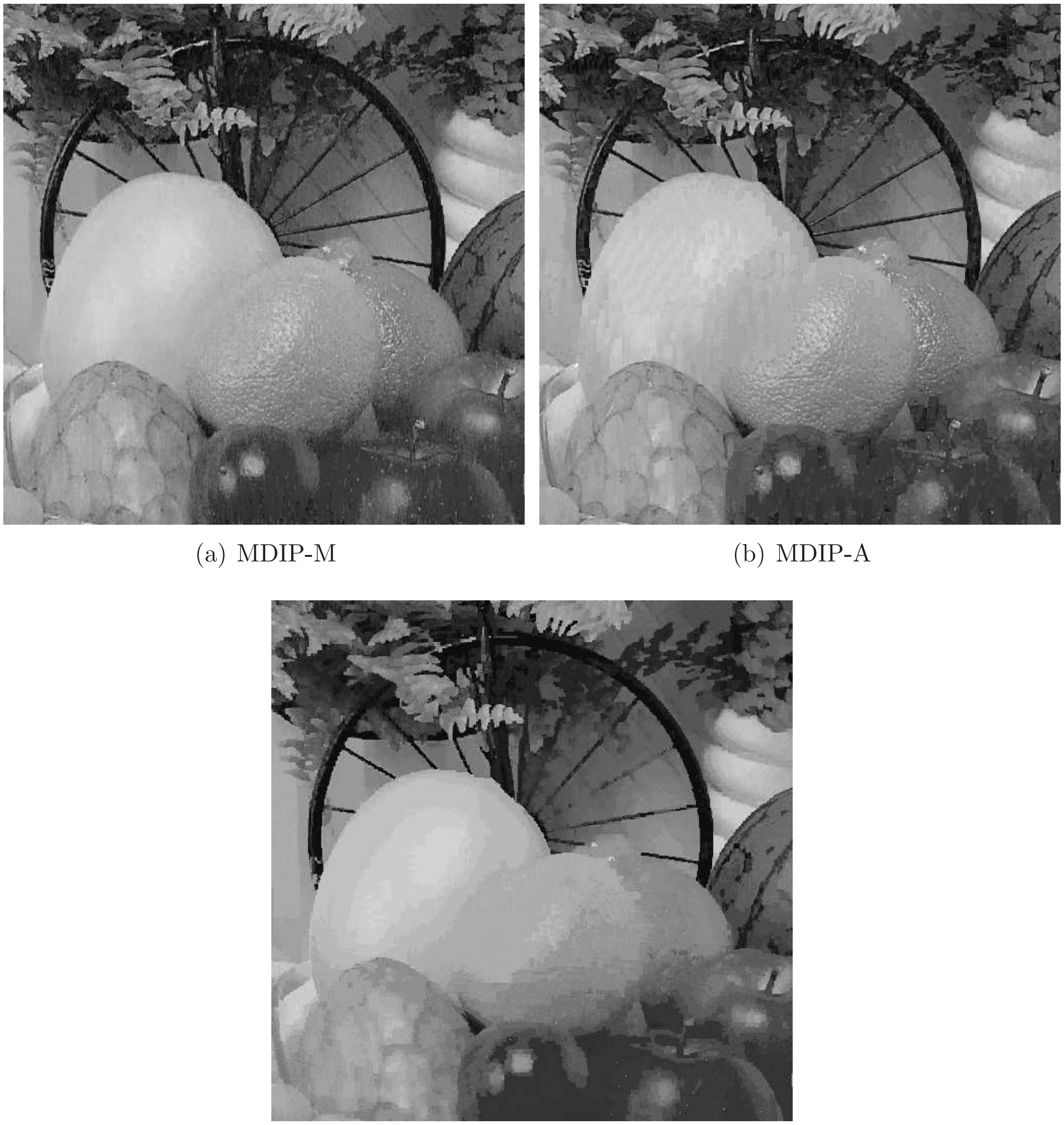


(a) MDIP-M

(b) MDIP-A

(c) 2DPMC

Figure 12: (a) is the result of minimax distortion at 0.49 bpp with PSNR=30.07dB and MSSIM=0.8115. (b) is the result image of the average distortion at 0.49 bpp PSNR=30.54dB and MSSIM=0.7993. (c) is of the 2DPMC at 0.49 bpp with PSNR=26.33dB and MSSIM=0.7261.

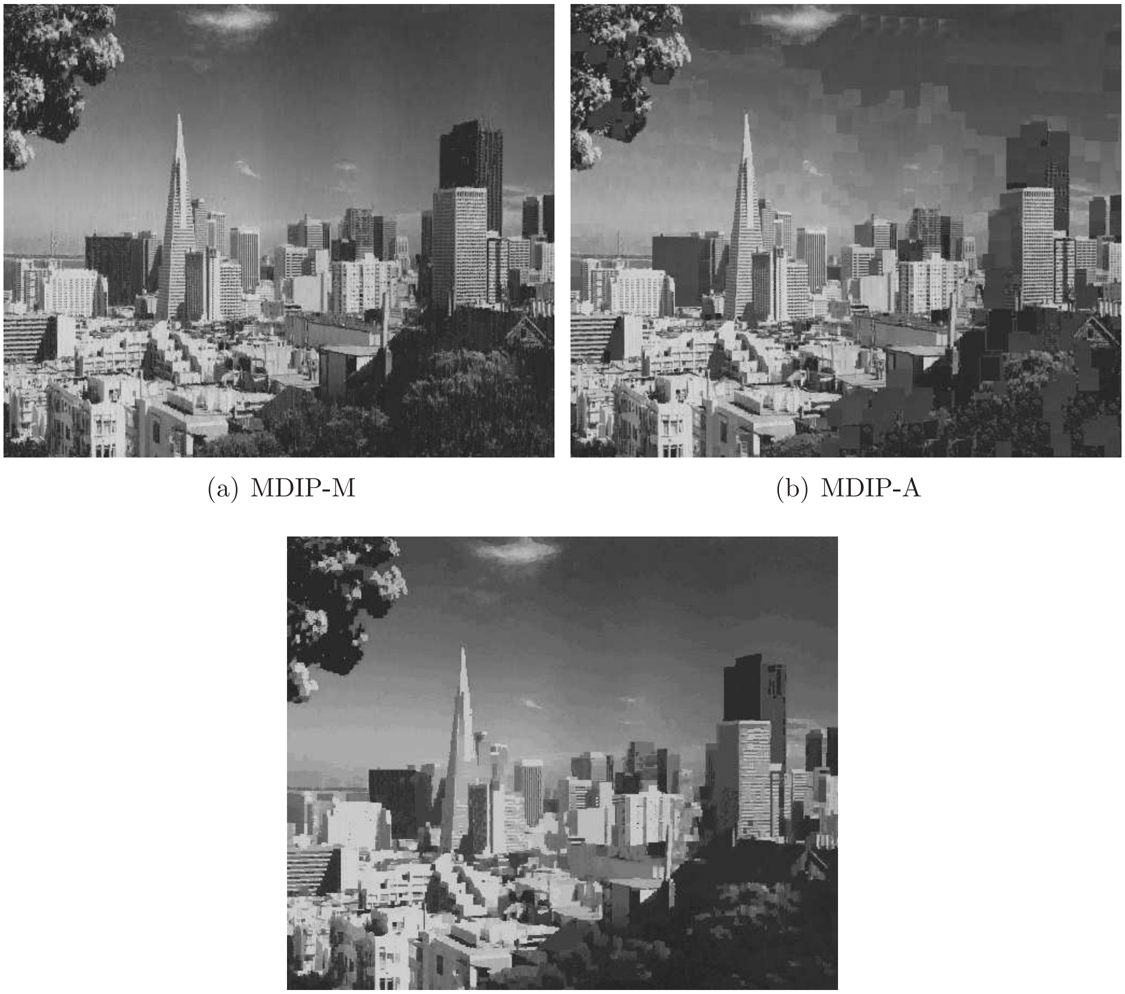


(a) MDIP-M

(b) MDIP-A

(c) 2DPMC

Figure 13: (a) is the result of minimax distortion at 0.57 bpp with PSNR=27.18dB and MSSIM=0.7325. (b) is of the average distortion at 0.57 bpp with PSNR=27.26dB and MSSIM=0.7150. (c) is of the 2DPMC at 0.57 bpp with PSNR=25.84dB and MSSIM=0.6682



(a) MDIP-M

(b) MDIP-A

(c) 2DPMC

Figure 14: (a) is the result of minimax distortion at 0.60 bpp with PSNR=26.52dB and MSSIM=0.8207. (b) is of the average distortion at 0.60 bpp with PSNR=26.46dB and MSSIM=0.7915. (c) is of the 2DPMC at 0.60 bpp with PSNR=23.55dB and MSSIM=0.7218.

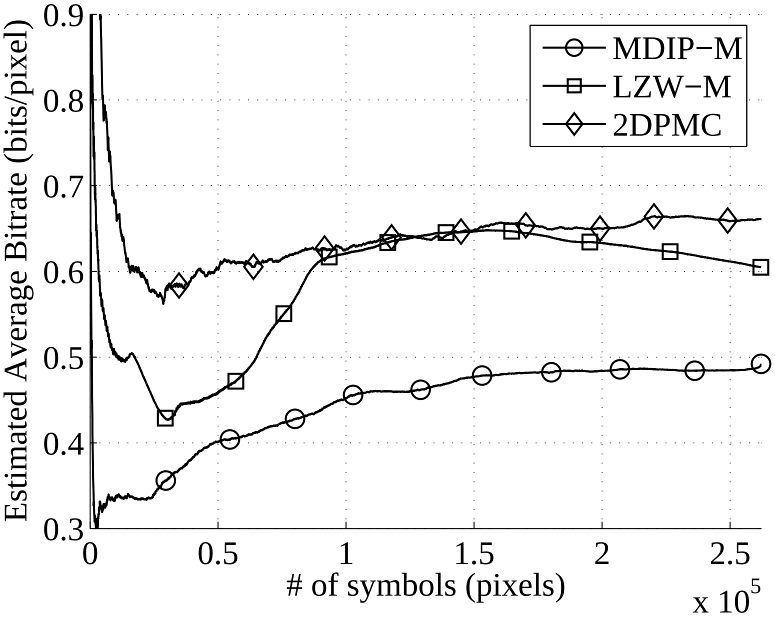


Figure 15: Estimated bitrates as the compression algorithms proceed. The test image is "Lena." The target PSNR is 30dB.

CHAPTER III

IPSILON: INCREMENTAL PARSING FOR SEMANTIC INDEXING OF LATENT CONCEPTS

3.1 Introduction

The success of linguistic information retrieval has inspired the adaptation of linguistic processing techniques for visual information analysis followed by CBIR systems, as discussed in Section 1.0.1. The theory of linguistics teaches us the existence of a hierarchical structure in linguistic expressions, from letter to word root, and on to word and sentences. By applying syntax and semantics beyond words, one can further recognize the grammatical relationship among words and the meaning of a sequence of words. This layered view of a spoken language is useful as it allows effective analysis and automated processing for humans and machines. Thus, it is interesting to ask if a similar hierarchy of representation of visual information does exist. A class of techniques that have a similar nature to the linguistic parsing is found in universal source coding, i.e. the Lempel-Ziv incremental parsing scheme (LZ78) [97]. The LZ78 is designed for compressing a given one-dimensional source sequence in an optimal way. The fundamental operation of the scheme is to parse a given sequence into a number of phrases that contain the same amount of information while constructing a dictionary with previously registered patterns of symbols. Since the parsing scheme reconstructs the given source with the minimum number of occurrences of dictionary entries, it is known that the coding algorithm is asymptotically optimal for a stationary ergodic source.

Although LZ78 has been successfully employed in many data compression applications, due to the dimensional constraint, it cannot be readily used in multidimensional data compression. This previously motivated us to design a multidimensional incremental parsing scheme for universal source coding in the previous chapter. Instead of using any particular scanning pattern, it parses a given multidimensional source into a number of multidimensional patches and assigns the same number of bits to the codewords in the dictionary. It has been experimentally verified that it asymptotically captures the source statistics into the dictionary and outperforms other existing universal lossy data compression algorithms.

In this chapter, based on the parsing scheme, we propose a query-by-example CBIR framework that uses the dictionary entries generated in the coding procedure as features of the given image and applies them to the LSA paradigm. In order to compare the effectiveness of the use of the dictionaries by incremental parsing (IP), we implemented a benchmark system that uses a visual dictionary trained by VQ [27]. To see the effect of the perceptual distortion bound on visual information analysis, we also tried three different perceptual distortion thresholds in the proposed retrieval system. The performance of these systems, in terms of retrieval precision, is compared with that of one of the recent systems, the SIMPLIcity proposed by Wang *et al.* [48, 81]. By the analysis of the latent semantic dimensions, we experimentally justify the relevance of the proposed image retrieval framework.

The rest of this chapter is organized as follows. In the next section, we briefly review incremental parsing algorithms for universal source coding. A typical setup of latent semantic analysis is discussed in Section 3.3. We provide detailed implementations of the proposed and the benchmark systems in Section 3.4. A performance comparison of the suggested implementations with the SIMPLIcity system is presented in Section 3.5. A discussion and the final remarks are found in Section 3.6.

3.2 Incremental parsing

In the proposed research, the query image and the images in a database are represented as a number of variable-size patches by a multidimensional incremental parsing algorithm. Then, the occurrence pattern of these parsed visual patches are fed into a semantic analysis framework. Before proceeding to a detailed implementation of visual dictionary generation and image retrieval system, we briefly review two incremental parsing algorithms in this section.

3.2.1 Lempel-Ziv Incremental Parsing

In 1977 and 1978, Lempel and Ziv consecutively developed two universal source coding algorithms, called the Lempel-Ziv sliding window (LZ77) [96] and the Lempel-Ziv incremental parsing (LZ78) [97]. These two algorithms have been widely employed in many data compression applications because of the fact that without any prior knowledge of the statistical distribution of the given source, the algorithms asymptotically achieve a source rate approaching the entropy of the source. As mentioned above, the LZ78 is of our interest because it implicitly embeds source statistics into the dictionary.

An essential operation of the LZ78 algorithm is to parse the given source sequence into a number of distinct phrases and to construct a dictionary containing the previously registered patterns of symbols. The algorithm has three main parsing steps: pattern matching, dictionary augmentation, and codeword assignment. At each coding epoch, the algorithm searches the longest dictionary entry that is exactly matched with the source symbols at the corresponding coding point. Then, it augments the dictionary with the last parsed phrase appended with the next source symbol. Next, it transmits the match index and the new symbol to a decoder with $\lceil \log_2 \Gamma \rceil + \lceil \log_2 |\mathcal{A}| \rceil$ bits, where $\lceil x \rceil$ denotes the least integer not smaller than x, $|\mathcal{A}|$ denotes the cardinality of a finite alphabet $\mathcal{A}$, and Γ corresponds to the number of dictionary entries and the number of distinct phrases. Thus, the total length of the code for the given source $\mathbf{X}$ is

$$\mathcal{L}(\mathbf{X}) = \Gamma \cdot (\lceil \log_2 \Gamma \rceil + \lceil \log_2 |\mathcal{A}| \rceil). \tag{20}$$

It is known that for a stationary ergodic source of length n, the average number of bits per symbol $\mathcal{L}(\mathbf{X})/n$ approaches the entropy of the given source $H(\mathbf{X})$ as $n \to \infty$.

3.2.2 Multidimensional incremental parsing

Although the LZ78 scheme has been implemented in many data compression applications and extensively studied in many research sectors, it still has a fundamental limitation: the coding algorithm applies only to one-dimensional source sequences. There hass been a plethora of research on multidimensional universal source coding based on generating a one-dimensional source sequence from a given multidimensional source with the aid of a scanning scheme. However, it is not clear that any of the linearization methods is suitable for analyzing the local property of a given source. Thus, in Chapter 3 we devised a multidimensional incremental parsing scheme for universal source coding Instead of using a scanning method, the proposed scheme parses the source into variable size patches and achieves outstanding performance compared with existing algorithms. The algorithm can be implemented for lossless or lossy compression depending on the selection of a pattern matching function. Here we are interested in a two-dimensional lossy compression scheme. As the counterparts of the three essential procedures in LZ78, the parsing algorithm consists of three essential component schemes: maximum decimation matching, hierarchical structure of multidimensional source coding, and dictionary augmentation. Here we focus on the maximum decimation matching rather than all three. A detailed description on the three component schemes can be found in Chapter 3.

As LZ78 does, the multidimensional incremental parsing scheme starts with an empty dictionary. At each coding epoch, it first searches the dictionary for the maximum decimation patch that matches the two-dimensional source at the current coding point according to a given distortion criterion, and then augments the dictionary with two new entries for two-dimensional source coding. Let $\mathbf{X}$ be a two-dimensional vector field taking values from a three-dimensional finite vector, each element of which represents each color component, i.e. red, green, and blue (RGB). $\hat{\mathbf{X}}$ denotes an approximation of $\mathbf{X}$. The design of a lossy color image compression can be achieved by the following minimax distortion function:

$$\rho_m\left(\mathbf{X},\hat{\mathbf{X}}\right)=\max_{\vec{x}\in\mathbf{X}}\left\{\max_{c\in\{\mathrm{r,g,b}\}}\left(0,\frac{|\mathbf{X}_c(\vec{x})-\hat{\mathbf{X}}_c(\vec{x})|-\mathcal{T}_c(\vec{x})}{\mathcal{T}_c(\vec{x})}\right)\right\}, \tag{21}$$

where $\mathcal{T}_c(\vec{x})$ is the threshold of the corresponding color component at the location $\vec{x}\in\mathbb{Z}^2$. Regarding $\mathcal{T}_c(\vec{x})$, we use the color just-noticeable distortion (JND) model proposed by Yang *et al.* [89], denoted by a 3-tuple $\mathcal{T}_{\mathrm{jnd}}=(\mathcal{T}_Y,\mathcal{T}_{Cb},\mathcal{T}_{Cr})$, in which the visibility thresholds of the two chrominance components in YCbCr domain are computed in addition to the baseline JND model for the luminance component. Since the images that we are dealing with are represented in 8-bit RGB color component format, the JND model needs to be transformed from YCbCr to RGB by the following conversion matrix:

$$\begin{bmatrix}\mathcal{T}_r\\ \mathcal{T}_g\\ \mathcal{T}_b\end{bmatrix}=\begin{bmatrix}1.1689 & 0 & 1.6023\\ 1.1689 & 0.3933 & 0.8162\\ 1.1689 & 2.0251 & 0\end{bmatrix}\begin{bmatrix}\mathcal{T}_Y\\ \mathcal{T}_{Cb}\\ \mathcal{T}_{Cr}\end{bmatrix}. \tag{22}$$

Since we deal with visibility threshold values, not the pixel values, we modified the YCbCr-to-RGB pixel conversion matrix, introduced in the ITU-R BT.601-4 [35], to this threshold conversion matrix.

The JND values derived here represent the threshold levels of noise visibility below which a human will not be able to perceive the noise. To achieve a suprathreshold color image compression, we introduce a minimally-noticeable distortion

(MND) model as follows:

$$\mathcal{T}_{\text{mnd}} = \mathcal{T}_{\text{jnd}} \times \theta_{\text{jnd}}, \tag{23}$$

which is simply multiplying every element of JND by a constant scale factor $\theta_{\text{jnd}} > 1.0$.

Since the patches are of variable size, they do not always fit the image like regular tiles, and there may be overlap with previous patches. As mentioned above, the algorithm finds the match that covers a maximal previously uncoded area, which is referred to as a *maximum decimation matching*. Note that the decimation level of a match is defined as the number of symbols that can be encoded with the given match. We first define two complementary sets $\mathbf{E}$ and $\mathbf{E}^c$. If the source symbol at location $\vec{x}$, denoted by $\mathbf{X}(\vec{x})$, is already coded, then $\vec{x}$ is in $\mathbf{E}$; otherwise, it is in $\mathbf{E}^c$. Let us define the decimation field $\mathbf{F}(\vec{x})$ as follows:

$$\mathbf{F}(\vec{x}) = \begin{cases} 0, & \vec{x} \in \mathbf{E} \\ 1, & \vec{x} \in \mathbf{E}^c. \end{cases} \tag{24}$$

Also, we define the decimation field area for an area vector $\vec{a} \in \mathbb{Z}^2$ as follows:

$$\mathbf{F}(\vec{x};\vec{a}) = \{F_{\bar{x}_1,\bar{x}_2} : x_i \le \bar{x}_i < x_i + a_i,\ i = 1,2\}. \tag{25}$$

Given a dictionary $\mathbb{D}$, we define two operations $|\cdot|$ and $[\cdot]$. $|\mathbb{D}|$ denotes the number of elements of $\mathbb{D}$, and $[\mathbb{D}_j]$ refers to a vector whose element represents the number of pixels of the j^{th} patch along each axis. The set of indices by ε_m-bounded distortion at the current coding epoch is

$$\mathbf{H}_m = \left\{ j \mid \rho_m\left(\mathbb{D}_j, \mathbf{X}(\Delta;[\mathbb{D}_j])\right) \le \varepsilon_m,\ 0 \le j < |\mathbb{D}|,\ \varepsilon_m \in \mathbb{R}_+ \right\}, \tag{26}$$

where Δ corresponds to the current coding location. ε_m is typically set to 0. At each coding epoch, the encoder constructs the set of indices with respect to the distortion criterion; then it measures the decimation level of each candidate patch. Finally, it selects the match of the index $k_{\max}$ that gives the highest level of decimation by

$$k_{\max} = \underset{k \in \mathbf{H}_m}{\operatorname{argmax}} \left\{ \sum_{\vec{x} \in \mathbf{F}(\Delta;[\mathbb{D}_k])} \mathbf{F}(\vec{x}) \right\}. \tag{27}$$

At each coding epoch, once the maximum decimation match is found, the algorithm appends two new entries to the dictionary, each of which is obtained by appending pixels along the horizontal and the vertical axes. For example, if the match found is of size $n_1 \times n_2$, the new entries then are of $(n_1 + 1) \times n_2$ and $n_1 \times (n_2 + 1)$ in size. Note that for an m-dimensional incremental parsing, m augmentative patches of m-1 dimensions generated along all the m axes are appended into the dictionary at each coding epoch.

Figure 16 shows the four largest patches that were used when an image is compressed using the proposed incremental parsing algorithm. The arrows point to the locations of the patches in the image. The patches the algorithm generates can be used to form an intermediate representation, which in combination with linguistic processing techniques, can be used to build a visual dictionary that will facilitate the association of semantic information from images.

3.3 Latent Semantic Analysis

Document retrieval has been extensively studied for decades and widely implemented into many commercial products. Recent advances are primarily due to the successful approaches to two fundamental problems in linguistic information processing: *synonymy* and *polysemy*. To deal with these two, there have been a number of research on representations of documents. One of the successful techniques is LSA [19], which summarizes a given document by a number of hidden concepts, rather than by conventional term counts. The main idea behind LSA is that *a bag of words* preserves enough relevant information for semantic retrieval.

LSA first defines a mapping of words and documents onto a semantic vector space. Suppose that we have a collection of N documents $D = \{d_1, \cdots, d_N\}$ and a lexicon with M words $W = \{w_1, \cdots, w_M\}$. Based on the vector space model [71], a document is represented as an M-dimensional vector.

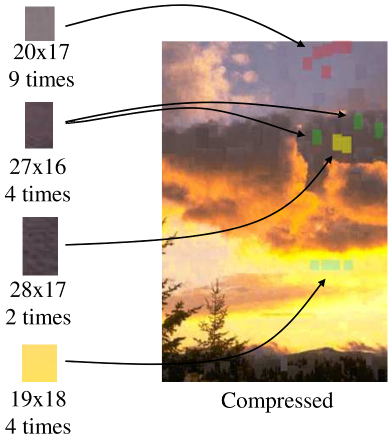


Figure 16: Example of the two-dimensional incremental parsing for image compression.

Collated together, they form an $M \times N$ co-occurrence matrix $L = \{l_{i,j}\}$, where $l_{i,j}$ denotes the number of occurrences of word w_i in document d_j. Bellegarda *et al.* reported in [4] that by taking inter- and intra-document normalization into account, the co-occurrence matrix can also be represented in word-counts weighted by a term-frequency normalized-entropy. The normalized entropy of i^{th} word ϵ_i is

$$\epsilon_i = -\frac{1}{\log N} \sum_{j=1}^{N} \frac{l_{i,j}}{t_i} \log \frac{l_{i,j}}{t_i}, \tag{28}$$

where $t_i = \sum_{j=1}^{N} l_{i,j}$ is the total number of times the i^{th} word occurs in D. The value ϵ_i indicates the singularity of the i^{th} word. We construct a co-occurrence matrix $L_n = \{\tilde{l}_{i,j}\}$ that is normalized by document size and word entropy as follows:

$$\tilde{l}_{i,j} = (1 - \epsilon_i) \frac{l_{i,j}}{n_j}, \tag{29}$$

where n_j is the total number of words present in the j^{th} document.

For document retrieval, a given query (document) is projected onto the word space to find the closest documents to the query. Instead of mapping onto a full-rank space, LSA projects the documents onto a subspace of reduced dimensionality, called *latent semantic space*. LSA uses the standard singular value decomposition (SVD) [28] to decompose the co-occurrence matrix into $L_n = U \Sigma V^T$, where U and V are unitary matrices, and the superscript T denotes matrix transpose. Then, taking the K largest singular values in Σ, L can be approximated as $\hat{L}_{n,K} = \hat{U}\hat{\Sigma}_K\hat{U}^T$. $\hat{L}_{n,K}$ captures the K most salient ensembles of the words. Given a vector $\mathbf{q}$ that represents a query in the word-space, one can map the corresponding query vector onto the reduced space by

$$\hat{\mathbf{q}} = \hat{\Sigma}_K^{-1} U^T \mathbf{q}. \tag{30}$$

The similarity between the j^{th} document in the corpus and the query vector $\mathbf{q}$ can be computed as

$$s(\hat{\mathbf{d}}_j, \hat{\mathbf{q}}) = \frac{\hat{\mathbf{d}}_j^T\ \hat{\mathbf{q}}}{||\hat{\mathbf{d}}_j||\ ||\hat{\mathbf{q}}||}, \tag{31}$$

where $\hat{\mathbf{d}}_j$ is the vector corresponding to the j^{th} document in the K-dimensional reduced space. A value of $s(\hat{\mathbf{d}}_j, \hat{\mathbf{q}}) = 1.0$ means the two documents are semantically equivalent, while for $s(\hat{\mathbf{d}}_j, \hat{\mathbf{q}}) < 1.0$ decreasing values denote decreasing similarity between the documents.

To apply the LSA paradigm to image retrieval, the co-occurrence matrix is generated with images from an image database and patches from a visual dictionary as illustrated in Figure 17. The similarity between images are computed by (31). In consideration of the effect of dimensionality reduction, instead of choosing one particular value for K, we will study the effect of different Ks on the performance of image retrieval systems.

3.4 Implementation of Image Retrieval Systems

To evaluate the performance of the proposed image retrieval framework compared with that of existing ones, we implemented four image retrieval systems; three use the parsed representation based on incremental parsing with different perceptual distortion thresholds, and one uses the conventional vector quantization for visual pattern analysis.

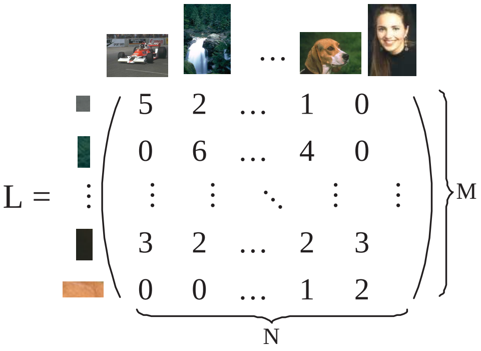


Figure 17: An illustrative example of a co-occurrence matrix of images and patches.

The four image retrieval systems are designed upon the same database, and the same query images are used for performance evaluation. A detailed description of the implementations and the database is given in this section.

3.4.1 Image Database

We implement the image retrieval systems using 20,000 images obtained from the Corel Stock Photo Library, stored in JPEG format with size 384×256 or 256×384. The boundary regions of all the images are cut off since some of them have black boundaries. Note that although a majority of researchers use the database, the Corel database has been heavily criticized because of the bias and the quality of the database [58, 61]. For example, the ontology used in labeling the images of the database is not precisely defined, and several images of the same scene are taken with a small angle change. Thus, there are sensible doubts about the use of the database in the evaluation of an image retrieval or an annotation system. One way to mitigate this criticism is to define a new set of ontology and associate each image with one of them or more.

In many image retrieval systems or in many pattern recognition applications, it is assumed that without exception every image is categorized into only one group. We argue here that each image can be a realization from multiple visual sources. Hence, rather than categorizing images into classes, which are non-overlapping universal groups, we classify them according to their visual concepts, which are overlapping groups. Thus, it is allowed that one image may contain more than one concept or may not belong to any. From the image database, we identify 15 visual concepts, examples of which are presented in Figure 18. Among the 20,000 images, there are 9,039 images with one concept, 323 images with two, and 12 with three. Table 3 presents the number of images in each visual concept. All the remaining 10,199 images are not associated with any visual concept listed in the table. The remaining images are of paintings, apes, fowls, antiques, etc. Note that paintings and portraits are not identified with such concepts as "human" or "snow." The relevance of the number of visual concepts and the image semantics depends on the reader's point of view. If those are associated with any of the listed concepts, the performance will be improved more than those shown in the results of the current system. From the image corpus, we randomly chose 600 image queries, each of which contains only one visual concept. Table 3 also presents the number of query images in each visual concept.

3.4.2 IPSILON systems

As discussed in Section 3.3, to analyze a given text document corpus under the LSA framework, the co-occurrence matrix is constructed by counting the word occurrences in each text document. For an image corpus, before constructing the co-occurrence matrix, we first have to generate a visual dictionary with which all the images from the corpus can be reconstructed. One ideal way is described as follows: one manually constructs a dictionary with any symbols of any size. Once the occurrence patterns of all the images are analyzed with a given distortion criterion, such dictionary entries that are not used in the reconstruction are pruned off.

Visual concept	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of images	821	101	257	212	1343	587	908	227	1054	1145	1655	211	365	107	515
# of queries	36	28	40	42	56	36	55	33	40	27	62	16	53	40	36

Table 3: Number of images and queries in each visual concept.

Since the number of dictionary entries in this case is colossal, we consider a sequential way of dictionary generation. Among many sequential generation schemes, one naive way is to encode all or some of the images separately with the incremental parsing scheme and merge all the dictionary entries into one. However, since there may exist duplicate entries across the dictionaries, after the compression step, all the dictionary entries are required to be compared with each other by a similarity measure, and the duplicated ones are pruned off. Assuming that the parsing algorithm generates n_e entries separately over the N images, the complexity is of the order $O(n_e^2 N^2)$. Thus, we consider an efficient approach with moderate complexity. At each compression iteration, a given k^{th} image is encoded with the dictionary $\mathbb{D}_{k-1}$ generated at the previous iteration. By encoding all the N images, one can generate the dictionary with redundant entries. To make the dictionary more compact, one may prune those dictionary entries that are not used in reconstructing the N images. An alternative heuristic for a more efficient construction is that for every n_p image during the encoding procedure, those dictionary entries that are not used for encoding the previous n_p images are pruned, and the dictionary with reduced entries is fed back to the encoding step. This heuristic leads the dictionary to bear the larger patches for more frequent visual patterns and vice versa. A comprehensive schematic of the visual dictionary generation process is depicted in Figure 19.

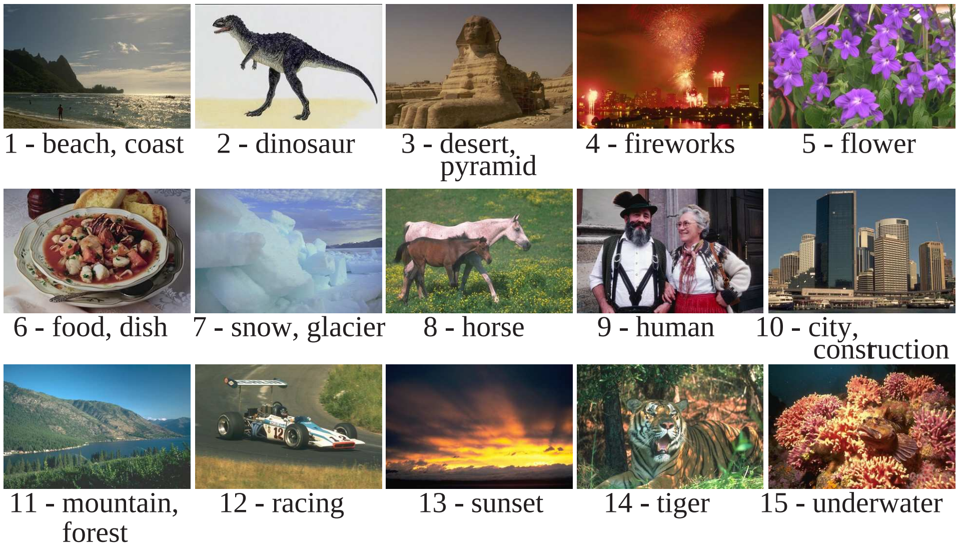


Figure 18: Visual concepts identified from the Corel Photo Stock Library.

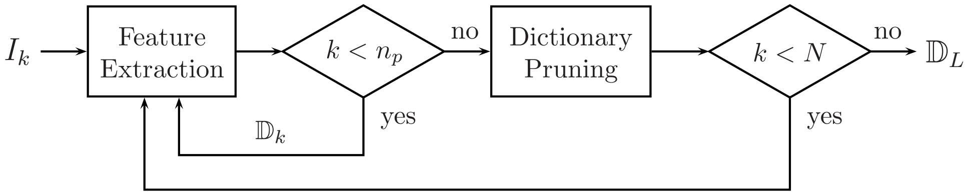


Figure 19: Schematic overview of visual dictionary generation by the incremental parsing. I_k is the k^{th} image in the database, n_p is the number of images for the task of pruning the unused entries, and N is the total number of images. $\mathbb{D}_L$ is the visual dictionary.

In this experiment, we randomly chose 1,200 images for the generation of the visual dictionary, without considering the number of visual concepts of each image. In the implementation, n_p is set to 100. As discussed in Section 3.2.2, we use an MND model for a supra-threshold image compression. By selecting three different values, θ_{jnd}=1.6, 2.0, and 2.5, we constructed three different visual dictionaries. Figure 20 depicts the number of entries during the generation of visual dictionary. At every iteration of the generation process, the dictionary is appended with new entries. At every n_p image, the unused entries in the dictionary are pruned so that the number of the entries drops significantly. After approximately 400 images, the number of entries oscillate without any major change.

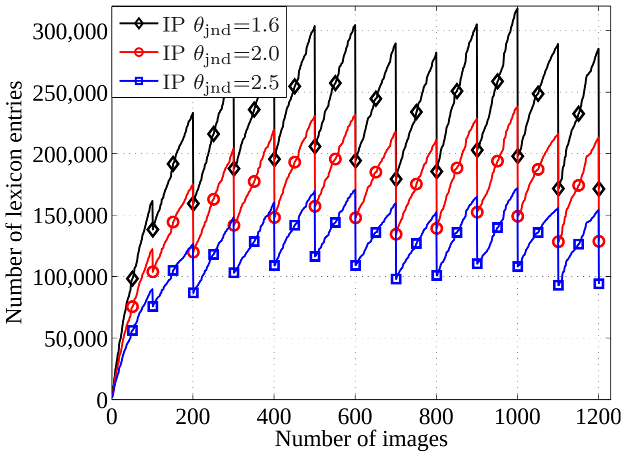


Figure 20: Number of entries in the process of the generation of the visual dictionary.

The number of entries in the dictionaries are 171,329, 128,673, and 94,142 for θ_{jnd}=1.6, 2.0, and 2.5, respectively. Once each visual dictionary is generated, three corresponding co-occurrence matrices are constructed by analyzing the occurrence patterns of the visual patches. Their dimensions are 171329×20000, 128673×20000, and 94142×20000. The densities of the matrices are 0.0208, 0.0207, and 0.0213 for θ_{jnd}=1.6, 2.0, and 2.5, respectively. In the LSA framework described in Section 3.3, the co-occurrence matrix is projected onto a lower dimensional space and the similarities between images are computed on the space by (31).

3.4.3 Image retrieval with vector quantization

One of the contributions of this research lies on the source representation by the aforementioned variable size patches for visual information analysis. Different from the proposed technique, many of the existing visual information analysis techniques in image retrieval systems extract features from regular blocks of fixed sizes. To compare the proposed technique with the conventional approach, we design an image retrieval system that constructs its visual dictionary by a fixed-block representation trained by VQ as a reasonable benchmark system. Similar to the proposed system shown in Section 3.4.2, we first construct a visual dictionary, then generate a co-occurrence matrix for the given image corpus, and retrieve images for a given query image.

In this VQ-based system, each color image represented in RGB color components is partitioned into 8×8 blocks. The dimension of the VQ codebook is therefore 8×8×3=192. Because the average number of pixels of the visual dictionary formed by the incremental parsing algorithm with θ_{jnd}=1.6 is approximately 63, a similar square block size, here 8×8, is chosen. Also, the number of codewords in the desired codebook is the same as the number of visual dictionary entries at θ_{jnd}=1.6, which is 171,329.

Since we have 20,000 images and each image is represented with 1536 data points, we are required to train the 192-dimensional quantizer with over 3 million points, of which the computation requirement is tremendous. Thus, rather than generating data points by all the images and all the data points, we limit the number of images to 10,000. In addition, from each image we train each individual codebook, which contain 128 codewords. Now, we train the 192-dimensional quantizer with 1,280,000 training data points. Since the training process still experiences the curse of dimensionality, we manually separated the training data points into four groups and trained four separate codebooks. To train these quantizers, we use the Linde-Buzo-Gray (LBG) algorithm [27, 50], which guarantees that the distortion from one iteration to the next does not increase. It is worth noting that in the training process we observed a number of empty cells, which do not contain any data point. To correct this problem, we place these codewords in the vicinity of the highest population codeword as is commonly done in VQ training. Once the four codebooks are generated separately, they are merged into one codebook that we use as a visual dictionary.

Table 4 shows average peak-signal-to-noise ratio (PSNR) values for reconstructed images both for the incremental parsing and VQ.

	Red	Green	Blue
IP θ_{jnd}=1.6	24.26	24.50	22.04
IP θ_{jnd}=2.0	23.00	23.37	21.17
IP θ_{jnd}=2.5	21.62	22.08	20.09
VQ	26.60	26.88	26.85

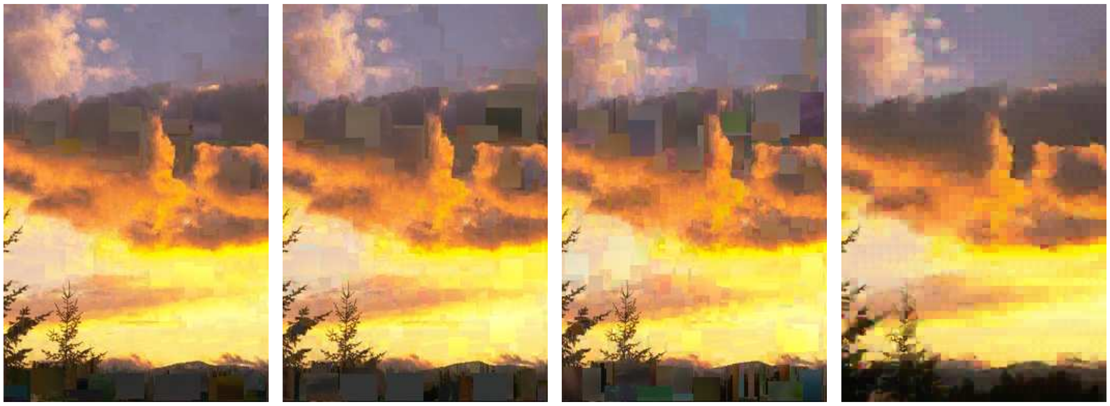
Table 4: Average PSNRs in each color channel for all the images.

The PSNR is traditionally derived from the mean-squared error (MSE) as

$$\text{PSNR} = 10\log_{10}\left(\frac{(2^q-1)^2}{\text{MSE}}\right), \tag{32}$$

where $\text{MSE} = \frac{1}{n}\sum_i^n (x_i - \hat{x}_i)^2$ and q is the number of quantization bits for each pixel. In the table, the PSNR values for VQ are higher than those for the incremental parsing due to the nature of VQ that minimizes the average distortion. Examples of the reconstructed images by the techniques are provided in Figure 21. Although the number of codewords in the dictionary formed by the VQ is the same as the number of entries in the dictionary by incremental parsing with θ_{jnd}=1.6, the reconstructed images by the VQ gives higher fidelity than the one by incremental parsing.

Similar to the image retrieval systems implemented in Section 3.4.2, the co-occurrence matrix is generated with the visual dictionary and projected onto a reduced space, and finally, the similarity between images is computed. The image retrieval system with VQ is essentially the same as that with IPSILON except the following two points. First, the incremental parsing minimizes the number of visual patches used in the reconstruction of a given image under a prescribed distortion bound, while VQ minimizes the average distortion of the reconstructed image. Second, the visual patches in the incremental parsing is of variable size, while the shape in VQ is fixed. These two factors are the key to their differential performances.



(a) IP θ_{jnd}=1.6 (b) IP θ_{jnd}=2.0 (c) IP θ_{jnd}=2.5 (d) VQ

Figure 21: Examples of reconstructed images. Average PSNRs across the color components are (a) 24.09 dB, (b) 23.01 dB, (c) 21.83 dB, and (d) 27.31 dB.

3.5 Experimental Results

We demonstrate the performance of the IPSILON systems and the benchmark system using a database of 20,000 images of natural scenes and compare with that of one of the recent image retrieval systems, the SIMPLIcity. The SIMPLIcity algorithm first segments a given image into a few regions based on the k-means algorithm. Then by a semantic classification method, the image is categorized into one of four groups (graph versus photograph and textured versus non-textured) so that the retrieval system limits the search range in the entire database. The similarity between images is computed by an integrated region matching (IRM). For a detailed description of SIMPLIcity, the reader is referred to [81].

3.5.1 Retrieval Precision Evaluation

A common practice in information retrieval for performance evaluation is to use precision/recall tests. Since the number of retrieved images that are to be rendered on a user interface of the system is limited, it is pragmatic to focus on a few most relevant images without examining the entire retrieved image group. Let m denote the number of visual concepts, which is 15 in this experiment. n_i and q_i denote the number of images and queries in the i^{th} visual concept, as shown in Table 3. Let $s_{i,j}$ be the number of relevant images for the j^{th} query in the i^{th} concept, $j = 1, \cdots, q_i$. r is the number of most relevant images that we are interested in. In this experiment, r is in $\{20, 40, 100\}$. The r-most relevant precision for the j^{th} query in the i^{th} concept is computed as

$$\frac{s_{i,j}}{r}, \tag{33}$$

and the average precision for all the queries in the i^{th} concept is

$$\frac{\sum\limits_{j=1}^{q_i} s_{i,j}}{r \cdot q_i}. \tag{34}$$

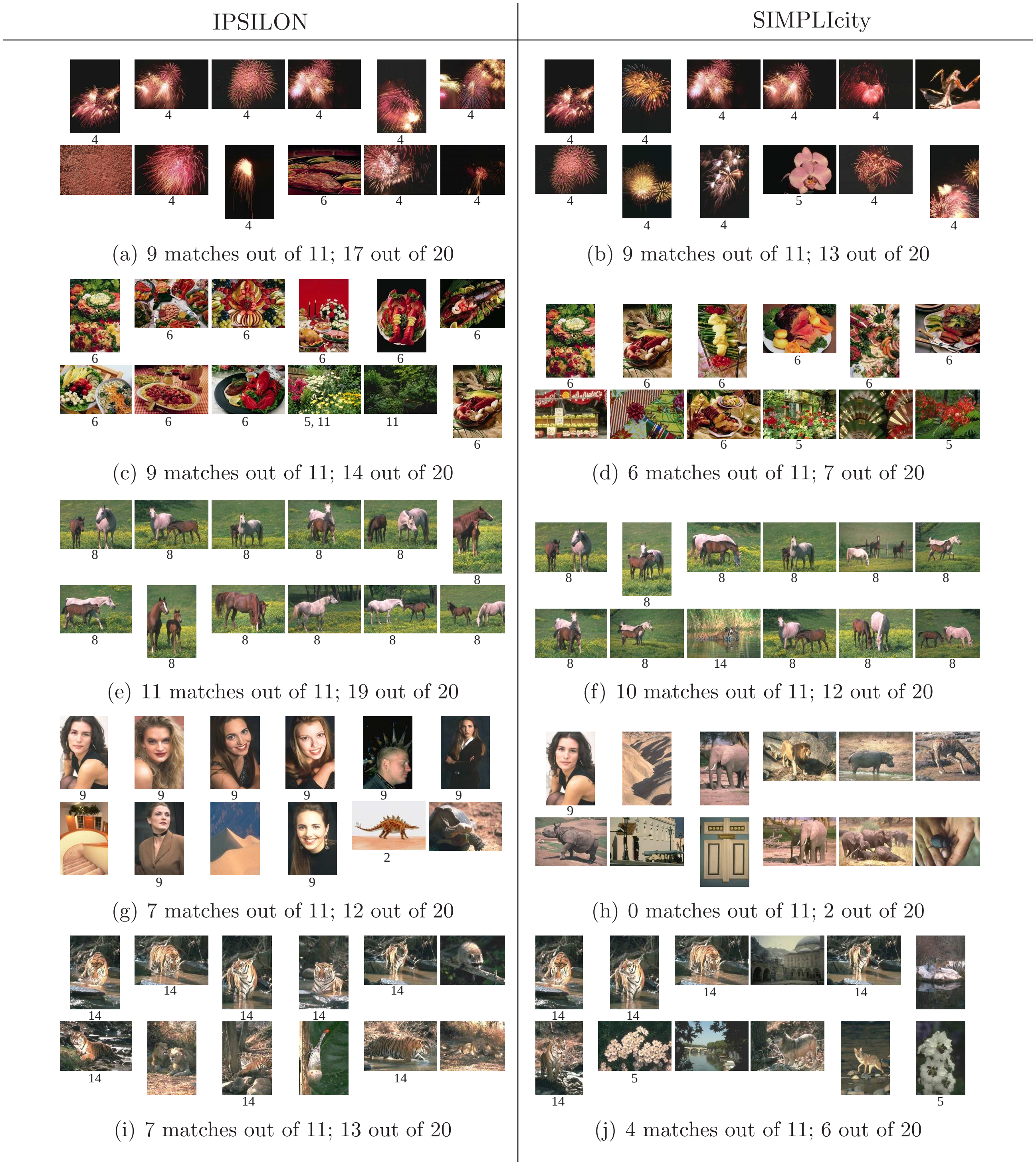
By taking the nonuniform prior shown in Table 3, we define the weight w_i for the i^{th} concept as

$$w_i = \frac{m \cdot n_i}{\sum\limits_{i=1}^{m} n_i}. \tag{35}$$

Finally, the total average precision for all the queries is

$$\frac{\sum\limits_{i=1}^{m} w_i \sum\limits_{j=1}^{q_i} s_{ij}}{r \cdot \sum\limits_{i=1}^{m} q_i} \tag{36}$$

To evaluate the performance of all five systems, all 600 query images were employed. We provide a few retrieved examples for illustration in Figure 22 because of space limitations. Figures 22 (e) and (f) compare two different characteristics of the two systems. For the same query image of “brown and white horses on a grass field,” we observe that the SIMPLIcity tries to retrieval such images that contain the concepts of “brown and white horses” while IPSILON focuses more on finding the images of “horses on a grass field.” Similar examples are shown in Figures 22 (a) and (b).



(a) 9 matches out of 11; 17 out of 20

(b) 9 matches out of 11; 13 out of 20

(c) 9 matches out of 11; 14 out of 20

(d) 6 matches out of 11; 7 out of 20

(e) 11 matches out of 11; 19 out of 20

(f) 10 matches out of 11; 12 out of 20

(g) 7 matches out of 11; 12 out of 20

(h) 0 matches out of 11; 2 out of 20

(i) 7 matches out of 11; 13 out of 20

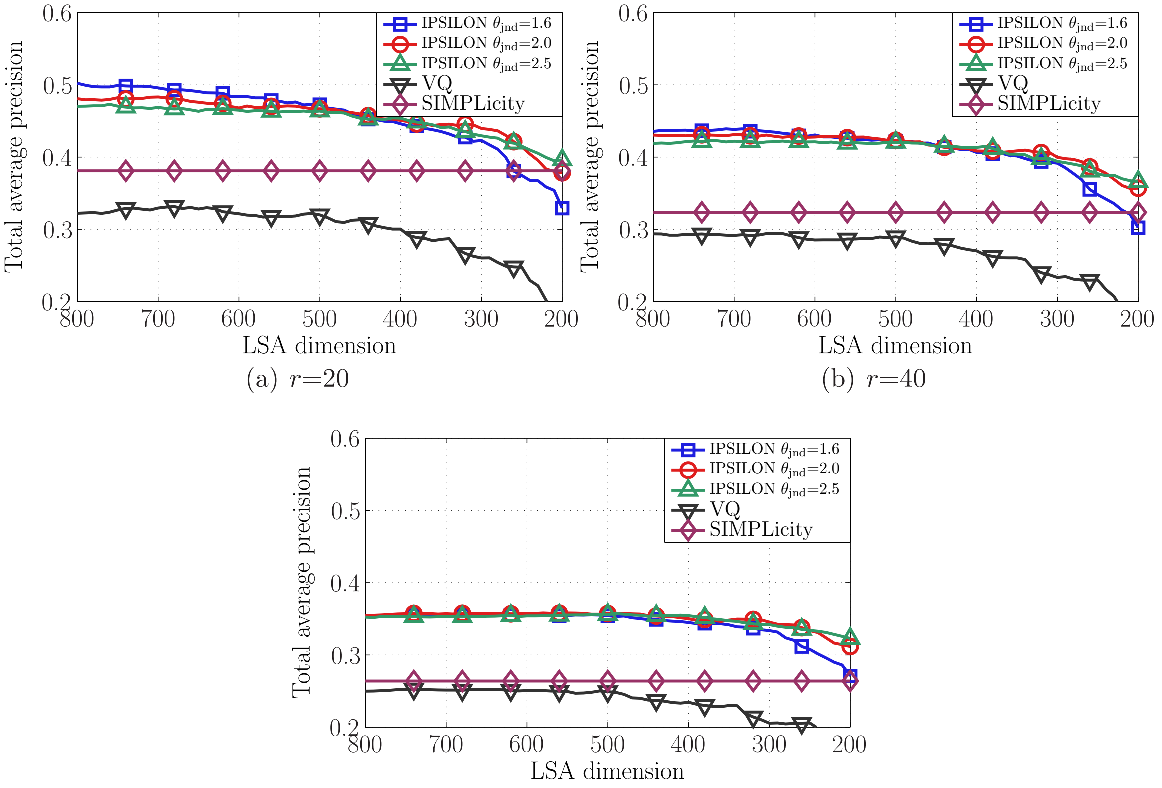
(j) 4 matches out of 11; 6 out of 20

Figure 22: Comparison of IPSILON and SIMPLIcity. For IPSILON, θ_{jnd}=1.6 and K=800 are chosen. The upper-left image of each set of images is the query image. Each number below each image is the index of visual concept.

For the query of "pink fireworks on dark sky," we understand that the SIMPLIcity attempts to retrieve the images of "pink object on dark background," while IPSILON aims at "pink scattering." We agree that the above subjective understanding depends on the reader's perspective. Nevertheless, it is reasonable to observe that the SIMPLIcity depends more on shape and color of an image than IPSILON does since the SIMPLIcity algorithm is based on a segmentation technique that results in a region-specific semantic matching. On the other hand, the patches that underlie IPSILON are morphologically lower level representations than regions or objects. The patches seem to lead to reasonable image representation for visual semantic analysis. From the overall subjective performance evaluation, the proposed IPSILON system efficiently captures the enlisted semantic concepts and provides more precise results as compared with the SIMPLIcity system.

We now evaluate the total average precisions of the IPSILON systems and the benchmark system across a different number of reduced space dimensions. Figure 23 provides the total average precision of all five systems for the same queries and the same image database. It is obvious that the total average precisions of the IPSILON systems are significantly higher than those of the benchmark system and the SIMPLIcity system. In addition, from this comparison, we carefully argue that the fixed-block representation, which underlies the VQ-based retrieval system, may not be effective for visual information analysis. We will see further supporting evidence later in this section. As shown in Figure 23 (c), even when r=100, the total average precisions of the IPSILON systems are much higher than those of the other two by as much as 0.1. For the three rs, the precisions notably decrease at $K < 500$, which means approximately 500 latent concepts are captured from the image corpus by LSA.

Figure 24 compares the average precisions of the five retrieval systems. For all 15 visual concepts, the retrieval precisions of the IPSILON systems are significantly higher than those of the two other systems. Except for the visual concepts "beach-coast' and "tiger," the precisions of the proposed systems are over 0.30. For a few visual concepts, i.e., "flower," "horse," "racing," and "sunset," the average precisions of the IPSILON systems are over 0.2 higher than the other two at r=20.



(a) r=20

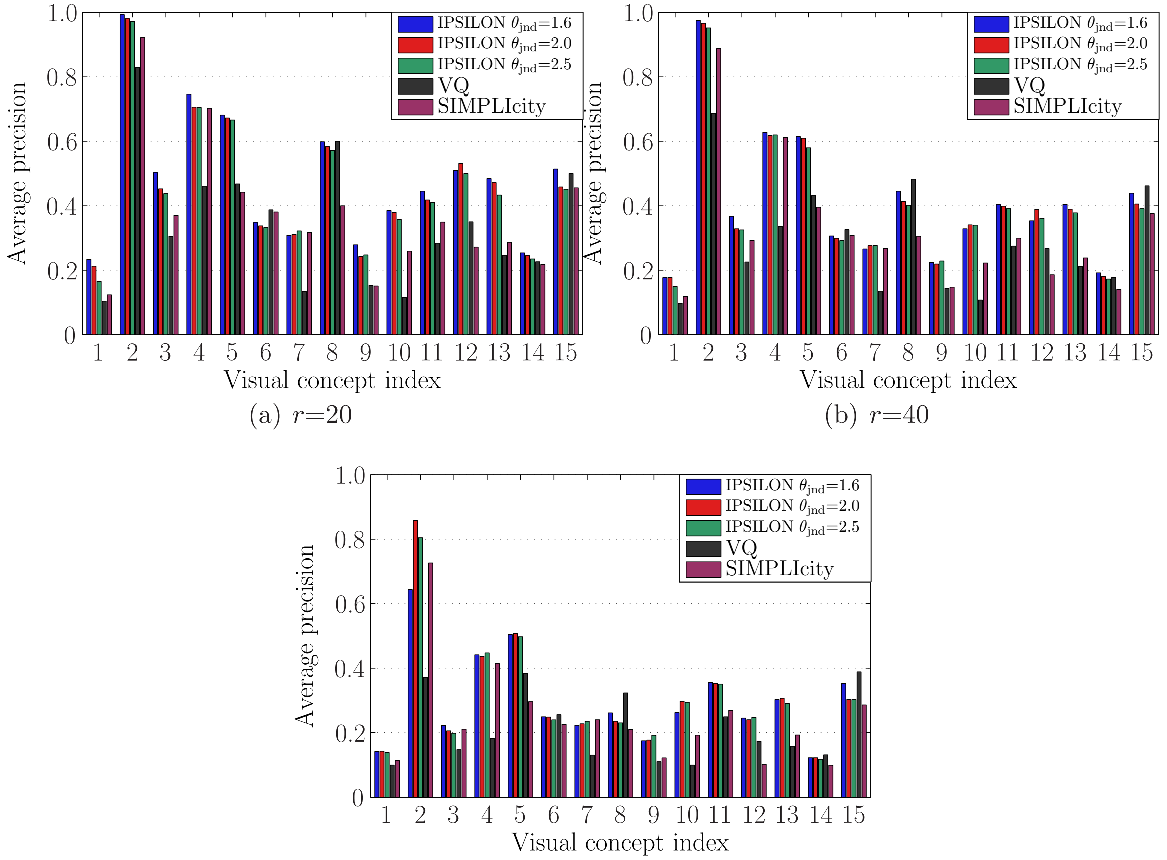
(b) r=40

(c) r=100

Figure 23: Comparison of the five image retrieval systems over the LSA dimensions.

Table 5 provides a numerical comparison of the total average precisions. Obviously, different perceptual distortion in visual pattern matching does not have serious effects on the retrieval precision, although allowing looser perceptual thresholds in image compression result in poor reconstruction fidelity as shown in Figure 21.

From Figures 23 and 24 and Table 5, we observe that the parsed representation organized by incremental parsing outperforms the fixed-block representation trained by VQ in visual information analysis for image retrieval. For an in-depth justification, we analyze the LSA dimensions formed by the two different techniques. Figure 25 provides two-dimensional illustrations of latent semantic dimensions of images and visual words. For all four plots, x-axis and y-axis correspond to the first and the second LSA dimensions.



(a) r=20

(b) r=40

(c) r=100

Figure 24: Comparison of average precisions of the five image retrieval systems for each visual concept. r=20 and K=800.

The two dimensions in each plot correspond to the two orthogonal axes that capture the first and the second largest eigenvalues of the given co-occurrence matrix. Note that only those data points with large magnitude are specified in this figure. In Figure 25 (a), the superimposed images on the corresponding locations have three primary visual concepts, e.g., “underwater,” “blue sky,” and “bright sky.” It is important to note that the two similar semantic concepts, “blue sky” and “bright sky,” are well differentiated so that these two concepts are not confused in the proposed retrieval systems.

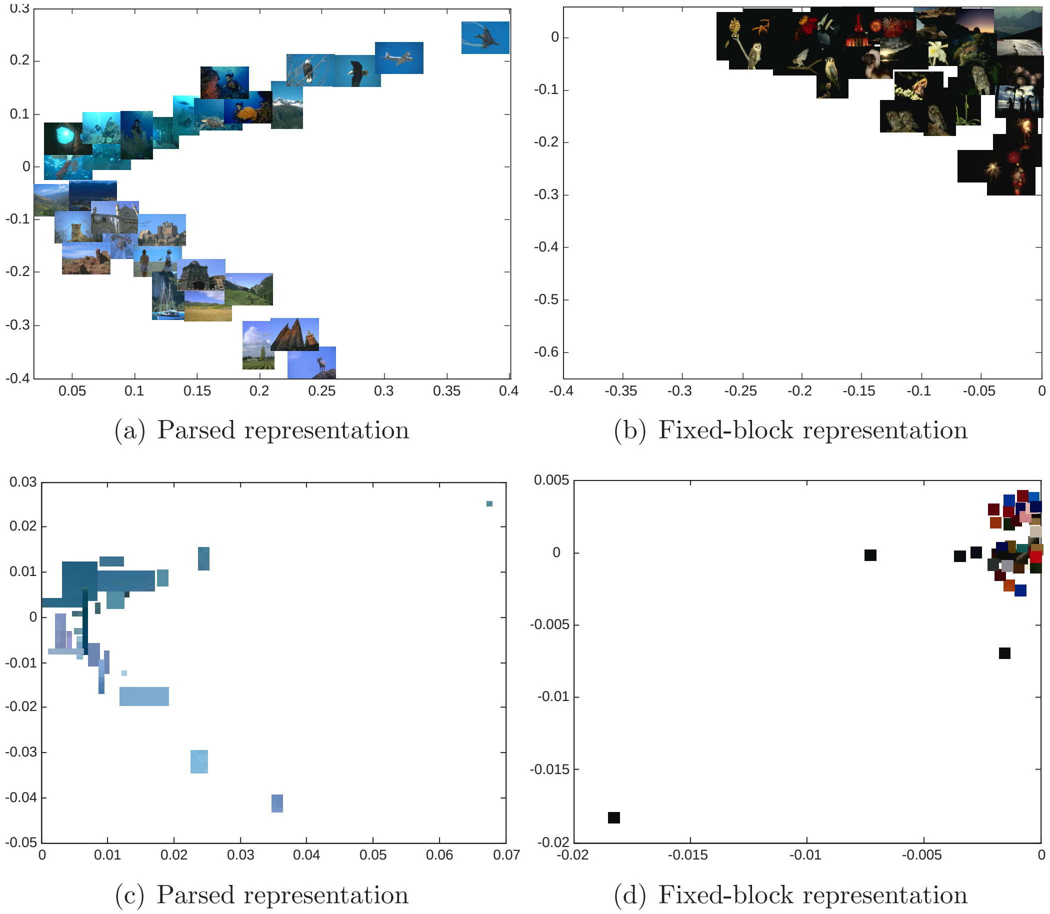
	IPSILON θ_{jnd}=1.6	IPSILON θ_{jnd}=2.0	IPSILON θ_{jnd}=2.5	VQ	SIMPLIcity
r=20	0.502	0.480	0.470	0.322	0.381
r=40	0.435	0.431	0.419	0.294	0.323
r=100	0.353	0.355	0.352	0.250	0.264

Table 5: Total average precisions of the image retrieval systems. K=800.

On the other hand, it is shown in Figure 25 (b) that the fixed-block representation primarily considers black regions dominant in the first two LSA dimensions; most of the superimposed images in the figure are those with large black areas from various semantic concepts such as “orchid,” “fireworks,” “birds,” “flowers,” “beach,” and even “sunset.” Figure 25 (c) shows two primary axes along which two sets of visual patches are superimposed. These two sets of patches are well differentiated and used to represent the three visual concepts shown in Figure 25 (a). However, as shown in Figure 25 (d), most of the image blocks trained by VQ are clustered around the origin, and it is not clear if those blocks capture semantically meaningful visual information.

3.6 *Discussions*

In this dissertation chapter, we have proposed a new representation of visual information generated by a universal source coding technique and applied it to the design of IPSILON, a CBIR system. With a multidimensional incremental parsing technique, as a multidimensional extension of the Lempel-Ziv incremental parsing, a given image is compressed and the accompanying dictionary is generated. It is previously verified that the statistics of a given source is implicitly embedded in the dictionary. With the incremental parsing technique, a given image is decomposed into a number of patches of variable size. This patch can be thought of as a morphological interface between elementary pixels and a higher level representation than the conventional fixed-block representation.



(a) Parsed representation

(b) Fixed-block representation

(c) Parsed representation

(d) Fixed-block representation

Figure 25: Two-dimensional illustrations of latent semantic dimensions of images and patches.

The proposed image retrieval framework uses the dictionary entries generated in the coding procedure as features of the given image and employs the LSA paradigm for deriving the semantic relationship among images. We have designed three image retrieval systems with different perceptual distortion thresholds. For an objective evaluation of the performance of the IPSILON systems, we compare them with those of two other systems: a benchmark system that uses fixed-block representations of visual information trained by VQ and the SIMPLIcity system based on an image segmentation technique. These five systems are tested with 20,000 images of natural scenes and 600 query images. The experimental results show that the proposed parsed representation efficiently captures the visual semantics appeared in the image corpus and that the image retrieval systems based on the parsed representation outperform the other systems in terms of retrieval precision.

However, the proposed framework has methodological limitations:

1. LSA basically assumes that the lexicon enumerates all the words appeared in a corpus. If an image contains unseen visual patterns to the visual dictionary, the image may not be appropriately analyzed by the current framework and, in turn, may not be correctly retrieved.

2. As shown in the experimental results, the proposed framework is not robust to pixel variational distortions. This is mainly due to the pixel-wise distortion criterion.

3. Currently, a full search technique is employed for generating the occurrence vectors of images, which is computationally demanding.

4. A fundamental limitation of a query-by-example CBIR system is that it can only deal with imagery queries. Frequently, users of CBIR systems may experience difficulty in accessing those images that reflect users' query in mind.

To overcome these limitations, one may consider the following approaches in future work:

1. By employing numerical or statistical detection techniques, a visual dictionary could be adaptively updated for unseen images. Also, rather than the current pixel-wise distortion criterion, one may consider first- and second-order statistics of pixel values for visual pattern matching.

2. Organizing the visual dictionary into a tree-structured dictionary along some components, e.g., dominant colors, perceptual salience, will reduce computational complexity for dictionary search.

3. As extending the current framework, one may consider probabilistic modeling of a given corpus for dealing with heterogeneous queries, such as hand-drawing, keywords, and even spoken query.

CHAPTER IV

ASPECT MODELING OF PARSED REPRESENTATIONS FOR IMAGE RETRIEVAL

4.1 Introduction

The IPSILON systems proposed in the previous chapter are content-based retrieval systems, which accept imagery queries. As discussed in Chapter 1, users of CBIR systems may experience difficulty in accessing relevant images that reflect users' query in mind. The LSA paradigm on which the IPSILON systems are based cannot readily allow projecting heterogeneous queries onto the vector space model. In addition, the LSA technique has considerable limitations. The fundamental assumption of LSA is that words and documents form a joint Gaussian distribution. In a context of count data, it is known that Poisson or negative binomial distribution is more appropriate for term counts [52, Section 15.4.3]. Another drawback of LSA is that since the approximate co-occurrence matrix is also a Gaussian distribution, it may contain negative entries for occurrence counts, which is obviously an unsuitable approximation for term counts. To overcome the limitations of LSA, which underlies the IPSILON systems, we study probabilistic linguistic processing techniques in this chapter. Probabilistic linguistic analysis provides great potential for a flexible framework of information retrieval in many aspects. First, heterogeneous queries as well as imagery queries can be mapped into a probabilistic source model by supervised/unsupervised learning algorithms. Second, one can achieve a generative model of a given corpus and train the model with either observed or unobserved documents. Third, in turn, since a given document is modeled as a probability distribution of a set of hidden parameters, the document can be easily analyzed in an understandable fashion.

Different probabilistic document models have been proposed in literature [5,9,30]. Their models arise from a similar assumption: Humans usually compare text documents based on topic similarities, not based on word similarities. The topics of a text document are not explicitly obtained but can be estimated from a document corpus, and they represent a document in a compact epitome. The basic assumption of this type of document modeling is that a document is a mixture of hidden variables, commonly referred to as *latent aspects*. The latent aspects are drawn from multinomial distributions over words of each text corpus. It has been shown that latent aspects learned from a corpus well follow document topics identified by humans [5, 30]. Among the probabilistic document models, we focus on probabilistic latent semantic analysis (PLSA) [30] in this chapter because the other two [5, 9] are variational models of PLSA except for the probability model of documents.

In visual information analysis applications, e.g., image retrieval and annotation, a considerable number of techniques that take advantage of the aspect modeling by PLSA have been reported in literature. Most of them extract visual features from fixed-block partition [10, 23, 56] or image segments [91]. As discussed in Chapter 1, many natural objects are not tessellations of fixed shape blocks and image segmentation is yet far from mimicking the way a human identifies real world objects. However, it is unclear that such representations of visual information are the best practice in higher level processing of images. To tackle this problem, we developed a multidimensional incremental parsing scheme proposed in Chapter 2 and applied it to the design of image retrieval systems in Chapter 3. In this chapter, based on the parsing scheme, we propose a query-by-example probabilistic CBIR framework which uses the dictionary entries generated in the coding procedure as features of the given image. Once the co-occurrence matrix of a given image corpus is generated, we train the aspect model by PLSA and design image retrieval systems, called AMPARS (Aspect Modeling of PArsed RepreSentation). To compare the effectiveness of the use of the dictionaries by incremental parsing (IP) under PLSA framework, we implemented a benchmark retrieval system that uses a visual dictionary trained by VQ [27]. Also, the performance of these systems, in terms of retrieval precision, is compared with two systems: the IPSILON system proposed in Chapter 1 and the SIMPLIcity proposed by Wang *et al.* [48,81]. By analyzing the aspect model parameters, we experimentally justify the relevance of the parsed representation in image retrieval framework.

One of the notable differences of AMPARS systems compared with the previous systems is the pattern matching scheme. IPSILON systems use a minimax distortion function (21) for evaluating the perceptual similarity of visual patches. By the minimax distortion function, the pattern matching scheme searches the dictionary entries whose perceptual distortions are bounded within a just-noticeable distortion. As shown in Chapter 2, this pattern matching shows high efficiency in perceptually based lossy data compression. However, a desirable attribute of pattern matching in image retrieval systems is to minimize semantic discrepancy and not to minimize perceptual distortion. One alternative way beyond the pixel-wise perceptual distortion measure is to compare the first- and second-order statistics of given patches. One of the recent measures in this regard is the structural similarity measure (SSIM) proposed by Wang *et al.* [82]. It has been experimentally verified in [8] that SSIM is robust to various types of image perturbations that are not sensitive to human eyes, e.g., geometric transformation, contrast variation, brightness variation and so on. Since SSIM is originally designed for comparison of gray-scale images, we propose a variation of SSIM for the measurement of color image patches.

The rest of this chapter is organized as follows: in the next section we briefly review probabilistic latent semantic analysis. In Section 4.3, we study a class of structural similarity index measures and propose a new measure for color image similarity. We provide detailed implementations of the proposed and the benchmark image retrieval systems in 4.4. The performance comparison of the four systems is presented in

Section 4.5. A discussion and final remarks are found in Section 4.6

4.2 Probabilistic Latent Semantic Analysis

Suppose that we have a collection of N documents $D = \{d_1, \cdots, d_N\}$ and a lexicon with M words $W = \{w_1, \cdots, w_M\}$. Based on the vector space model [71], a document is represented as an M-dimensional vector. From the observation of a given corpus, we can compute the empirical distribution $p(w, d)$ that corresponds to the term-document co-occurrence in LSA. Basically, PLSA estimates the term-document joint distribution $P(w, d)$ that minimizes the Kullback-Leibler (KL) divergence with respect to the empirical distribution $p(w, d)$ subject to K latent aspects, as opposed to the L_2 norm minimization performed by LSA. In aspect modeling, each document is a mixture of latent aspects $z_k \in Z = \{z_1, \cdots, z_K\}$.

PLSA model has two independence assumptions. First, observation pairs (w_i, d_j) are generated independently. Second, the pairs of random variables (w_i, d_j) are conditionally independent given the hidden aspect z_k, i.e.,

$$P(w_i, d_j|z_k) = P(w_i|z_k)P(d_j|z_k). \tag{37}$$

The joint distribution of the observation pairs is the marginalization over the K latent aspects z_k as follows:

$$\begin{aligned} P(w_i, d_j) &= P(d_j)P(w_i|d_j) & (38) \\ &= P(d_j) \sum_{z_k \in \mathcal{Z}} P(w_i|z_k)P(z_k|d_j) & (39) \end{aligned}$$

Figure 26 illustrates a PLSA graphical model and the conditional independence assumptions of the PLSA model shown in (39). A document d_j is first selected with the probability $P(d_j)$, and an aspect z_k is selected from the conditional probability $P(z|d_i)$. According to the conditional probability $P(w|z_k)$, each word w_i is selected.

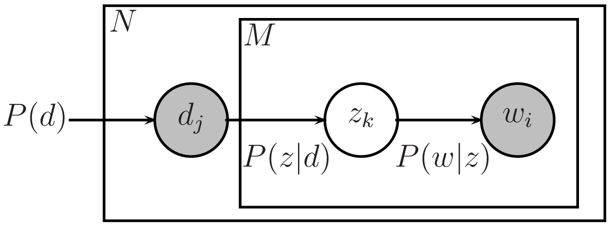


Figure 26: Graphical model for PLSA for a corpus with N documents and an M-word lexicon. Shaded nodes are observed. Square boxes denotes that they are replicated the number of times indicated in the top-left corner.

4.2.1 Learning Model Parameters

The estimation of the conditional probability distributions $P(w_i|z_k)$ and $P(z_k|d_j)$ can be resolved by applying the expectation-maximization (EM) technique [20], which maximizes the likelihood function of the observed data

$$\mathcal{L} = \prod_{j=1}^{N} \prod_{i=1}^{M} P(d_i) \sum_{k=1}^{K} P(z_k|d_j) P(w_i|z_k)^{p(w_i,d_j)}, \tag{40}$$

or equivalently the log-likelihood function

$$\log \mathcal{L} = \sum_{j=1}^{N} \sum_{i=1}^{M} p(w_i, d_j) \log P(w_i, d_j). \tag{41}$$

This is equivalent to the minimization of the cross entropy of the empirical distribution $p(w, d)$ and the term-document joint-distribution $P(w, d)$

$$H(p(w,d), P(w,d)) = -\sum_{j=1}^{N} \sum_{i=1}^{M} p(w_i, d_j) \log P(w_i, d_j), \tag{42}$$

which is also equivalent to minimizing the Kullback-Leibler (KL) divergence of $p(w, d)$ and $P(w, d)$. The EM algorithm alternates in two steps: an expectation (E) step and a maximization (M) step. In an expectation step, the conditional probability distribution of the latent aspect z_k, given the observation pair (w_i, d_j), is computed based on the previous estimates of the parameters:

$$P(z_k|w_i, d_j) = \frac{P(w_i|z_k) P(z_k|d_j)}{\sum_{k'=1}^{K} P(w_i|z_{k'}) P(z_{k'}|d_j)}. \tag{43}$$

In a maximization step, the parameters of the multinomial distribution $P(w_i|z_k)$ and $P(z_k|d_i)$ are replaced with the updated conditional probability distribution $P(z_k|w_i, d_j)$:

$$P(w_i|z_k) = \frac{\sum_{j=1}^{N} p(w_i, d_j)P(z_k|w_i, d_j)}{\sum_{i=1}^{M} \sum_{j=1}^{N} p(w_i, d_j)P(z_k|w_i, d_j)}, \tag{44}$$

$$P(z_k|d_j) = \frac{\sum_{i=1}^{M} p(w_i, d_j)P(z_k|w_i, d_j)}{p(d_j)}. \tag{45}$$

The initial values for $P(w_i|z_k)$ and $P(z_k|d_j)$ can be set to either uniform or random distributions. The variability of solutions obtained from various initial conditions is observed small in [30]. The output of the EM algorithm under PLSA is the two multinomial distributions $P(w|z)$ and $P(z|d)$, from which the joint distribution $P(w, d)$ is estimated.

4.2.2 Overfitting Control

The PLSA algorithm described above focuses only on maximizing the likelihood function to fit a model to a given corpus. However, the likelihood cannot be a measure of the quality for unseen test data. Thus, one has to control the tradeoff between the predictive performance on the training data and on unseen data by a regularization term. In [30], Tempered Expectation-Maximization (TEM) is introduced to deal with this problem. In a standard EM, only the E step is replaced with

$$P_\beta(z_k|d_j, w_i) = \frac{P(z_k)[P(w_i|z_k)P(d_j|z_k)]^\beta}{\sum_{z'_k=1}^{K} P(z')[P(w_i|z'_k)P(d_j|z'_k)]^\beta}, \tag{46}$$

where β is a control parameter that scales the likelihood function. In the TEM algorithm, β is first set to 1.0. If the performance of the likelihood function improves, β does not change in the TEM. Otherwise, β is updated as $\beta{=}\eta\beta_{\text{old}}$ with $\eta < 1.0$.

4.2.3 Learning a New Document

When an unseen document d_{new} is given, the conditional probability $P(z|d_{\text{new}})$ can be estimated by the *folding-in* method proposed in [30]. By fixing the previously learned models $P(z)$ and $P(w|z)$ in the EM algorithm, one can estimate the $P(z|d_{\text{new}})$ which maximizes the likelihood of the document d_{new} with respect to the $P(z)$ and $P(w|z)$.

4.3 Structural Similarity Index

The most commonly used objective quality measure is the mean squared error (MSE), commonly expressed as peak-signal-to-noise ratio (PSNR), which is known to be inadequate as a measure of perceptual distortion. A number of perceptual measures have also been proposed [22, 62]. These measures have relied upon certain explicit low-level models of human perception that account for sensitivity to subband noise as a function of spatial frequency, local luminance, and the contrast or effect of texture masking. Another recently proposed class of quality measures, known as the structural similarity (SSIM) [82, 85], is not based on explicit models of the HVS or measurements of noise sensitivities, but instead, account for higher-level functionalities of the HVS, and in particular, make use of the fact that it can extract structural information (in the form of relative spatial covariance) from the viewing field. An important property of SSIM metrics is that they allow imperceptible point-by-point distortions, such as spatial and intensity shifts and contrast and scale changes, and only respond to significant structural changes. Thus, they are expected to be more effective at measuring suprathreshold compression distortions, which affect the structure of an image.

4.3.1 Review of Structural Similarity Measures

There are several SSIM implementations, both in the image domain and the wavelet domain. The basic SSIM metric presented in [82], combines three terms: a luminance term, a contrast term, and a structure term. Given two images or partial images $\mathbf{x}$ and $\mathbf{y}$, the *luminance* component is defined as

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \tag{47}$$

where μ_x and μ_y are the means of the two images. The *contrast* comparison term is defined as

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \tag{48}$$

where σ_x^2 and σ_y^2 are the variance of the two images. The *structure* term is defined as

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}, \tag{49}$$

where σ_{xy} is the covariance between the two images. These three terms are combined to give a composite measure of structural similarity:

$$\mathrm{SSIM}(\mathbf{x}, \mathbf{y}) = l(\mathbf{x}, \mathbf{y})^{\alpha} c(\mathbf{x}, \mathbf{y})^{\beta} s(\mathbf{x}, \mathbf{y})^{\gamma}, \tag{50}$$

where α, β, and γ are positive weights. Following the parameter settings in [82], $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$, we get a specific implementation of SSIM quality metric

$$\mathrm{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x \mu_y + C_1)\,(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)\,(\sigma_x^2 + \sigma_y^2 + C_2)}. \tag{51}$$

The basic SSIM index proposed in [82] is a real number in the range $[-1, 1]$. A more general form of this metric can be found in [82]. The spatial domain SSIM has been shown to provide good quality prediction across a variety of artifacts, but is highly sensitive to spatial translation.

The complex wavelet domain implementation (CWSSIM) [84] allows imperceptible spatial translations as well as small rotations and scaling changes. The CWSSIM of a given subband is given by

$$S_c(\mathbf{c}_x, \mathbf{c}_y) = \frac{2\left|\sum\limits_i c_{x,i} c_{y,i}^*\right| + C}{\sum\limits_i |c_{x,i}|^2 + \sum\limits_i |c_{y,i}|^2 + C}, \tag{52}$$

where $\mathbf{c}_x$ and $\mathbf{c}_y$ are the wavelet coefficients corresponding to two images or image patches, $\mathbf{c}^*$ denotes the complex conjugate of $\mathbf{c}$, and C is a small positive constant.

Note that the mean of the wavelet coefficients (except the base-band) is zero due to the bandpass property of the wavelet transform. The overall metric value is computed as the mean of the CWSSIM subband indexes. The main idea of this quality metric is that the relative phase patterns of the wavelet coefficients contain the structural information of the local image features and the image distortion generates a nonhomogeneous perturbation to phase components. A variation of this metric, the weighted CWSSIM (WCWSSIM), was proposed by Brooks *et al.* [7, 8], whereby the subband indexes are weighted based on the human contrast sensitivity function. The subband weights are derived by the normalized contrast sensitivity over a spatial frequency range as follows:

$$w_s = \frac{\displaystyle\int_0^{u_m} C(u) H_s(u)\, du}{\displaystyle\int_0^{u_m} H_s(u)\, du}. \tag{53}$$

Here, u_m is the maximum spatial frequency at given viewing parameters, $C(u)$ is the frequency response of the CSF, and $H_s(u)$ is the wavelet subbands. The WCWSSIM incorporates an explicit models of subband sensitivity to noise, and thus provides a link to the perceptual metrics described above.

Although the above SSIM indexes primarily focus on comparing the structural information from the images, it has not been used in texture comparison applications since the original SSIM is too constrained to capture the perceptual similarity of two textures. Zhao *et al.* in [93] replaced the structure term with structural texture terms that are sensitive to local textual statistics. The first-order autocovariance in the horizontal direction is defined as

$$\rho_x(0,1) = E\{(x_{i,j} - \mu_x)(x_{i,j+1} - \mu_x)\}/\sigma_x^2 \tag{54}$$

The autocovariance in the vertical direction is defined in a similar fashion. The texture term in the horizontal direction is formulated as

$$c_{0,1}(\mathbf{x}, \mathbf{y}) = 1 - 0.5(|\rho_x(0,1) - \rho_y(0,1)|)^p, \tag{55}$$

which gives a real number in the range [0,1]. In [93], p is set to 1. These horizontal and vertical texture terms are combined with the luminance and the contrast terms in the original SSIM as follows:

$$\mathrm{STSIM}(\mathbf{x},\mathbf{y}) = l(\mathbf{x},\mathbf{y})^{\frac{1}{4}} c(\mathbf{x},\mathbf{y})^{\frac{1}{4}} c_{0,1}(\mathbf{x},\mathbf{y})^{\frac{1}{4}} c_{1,0}(\mathbf{x},\mathbf{y})^{\frac{1}{4}}. \tag{56}$$

In [93], they experimentally showed that the proposed STSIM performs well for texture retrieval applications.

The structural similarity measures mentioned above only account for measuring the similarity of gray-level images. An attempt for measuring the structural similarity of color images and video sequences were made by Wang *et al.* [83]. Although they aim at evaluating the overall fidelity of two video sequences, a part of their method can be used for measuring the structural similarity of two color images. For j^{th} sampling window in i^{th} video frame, the structural similarity is measured on YCbCr domain as follows:

$$\mathrm{SSIM}_{ij} = w_Y \mathrm{SSIM}_{ij}^{Y} + w_{Cb} \mathrm{SSIM}_{ij}^{Cb} + w_{Cr} \mathrm{SSIM}_{ij}^{Cr} \tag{57}$$

where the weights are fixed to be $w_Y = 0.8$, $w_{Cb} = 0.1$, and $w_{Cr} = 0.1$, respectively. The structural similarity of i^{th} video frames are

$$Q_i = \frac{\sum_{j=1}^{R_s} w_{ij} SSIM_{ij}}{\sum_{j=1}^{R_s} w_{ij}}, \tag{58}$$

where R_s denotes the number of sampling windows per video frame and $w_i j$ is the weighting value given to the j^{th} sampling window in the i^{th} frame. The overall quality of the entire video sequence is given by

$$Q = \frac{\sum_{i=1}^{F} W_i Q_i}{\sum_{i=1}^{F} W_i}, \tag{59}$$

where F is the number of frames and W_i is the weighting value assigned to the i^{th} frame.

4.3.2 Color Structural Texture Similarity Measure

Remember that we are interested in an information retrieval framework that takes advantage of lossy universal source coding schemes. In IPSILON systems proposed in Chapter 3, as a lossy source coding scheme, we implemented a lossy two-dimensional incremental parsing scheme with the minimax distortion function (21). As mentioned in Section 4.1, the distortion function for a universal coding in visual information analysis, not perceptual image compression, aims at measuring semantic discrepancy rather than perceptual distortion between patches. In this regard, structural similarity measures are of our interest due to the following advantages. First, since they are based on higher-orders statistics, not based on the pixel-wise distortion, they are robust to geometric distortions, such as rotation, shift, scaling, and so on. Second, the *luminance* term of the measure alleviates the effect of luminance or brightness change. However, there has been no structural similarity measure that takes into account color images or image patches in image retrieval applications. Thus, we here propose a color structural texture similarity measure (CTSIM) for a matching criterion of the universal source coding.

Let $\mathbf{x}$ and $\mathbf{y}$ be two images or image patches to be compared represented in three color components, i.e., red, green, and blue (RGB). When each image patch is represented in YCbCr domain as $\mathbf{x} = \{\mathbf{x}_Y, \mathbf{x}_{Cb}, \mathbf{x}_{Cr}\}$ and $\mathbf{y} = \{\mathbf{y}_Y, \mathbf{y}_{Cb}, \mathbf{y}_{Cr}\}$ we observed rich textures of the luminance component. Thus, we propose to use the STSIM for Y component and the original SSIM for Cb and Cr components as follows:

$$\mathrm{CSTSIM}(\mathbf{x}, \mathbf{y}) = w_Y \mathrm{STSIM}(\mathbf{x}_Y, \mathbf{y}_Y) + w_{Cb} \mathrm{SSIM}(\mathbf{x}_{Cb}, \mathbf{y}_{Cb}) + w_{Cr} \mathrm{SSIM}(\mathbf{x}_{Cr}, \mathbf{y}_{Cr}), \quad (60)$$

where w_Y, w_{Cb}, and w_{Cr} are the weights for each component. In this implementation, we set w_Y=0.6, w_{Cb}=0.2, and w_{Cr}=0.2, respectively. Also, for SSIM, we follow the parameter settings shown in [82]: $C_1 = (K_1 L)^2$, $C_2 = (K_2 L)^2$, $L = 255$, $K_1 = 0.01$, and $K_2 = 0.03$. From our experiment, the proposed CSTSIM efficiently captures the textual information of a given patch and compares it with that of other patches. Also, the CSTSIM is robust to color mean shift, contrast variance of luminance component, and so on.

4.4 Implementation of Image Retrieval Systems

We have implemented two image retrieval systems for an objective evaluation of the proposed framework; one uses the parsed representation based on the incremental parsing and the other uses the conventional vector quantization for visual information analysis. Both of them are under the same aspect modeling paradigm. The performance of the systems are compared with that of one of the recent image retrieval systems, SIMPLIcity [48, 81], which is based on an image segmentation technique. In this section, a detailed description of the implementation of the two systems is provided.

4.4.1 Image database

We implemented the proposed and the benchmark image retrieval systems using 20,000 images obtained from the Corel Stock Photo Library, which is the same as the database used for the IPSILON systems in Chapter 3. Also, the specifications with regard to the definition of the visual concepts, the number of images, and the number of query images in each visual concept, are provided in Figure 18 and Table 3.

4.4.2 AMPARS system

The proposed image retrieval system parses the given images into a number of variable-size patches, which we refer to as a parsed representation, by a two-dimensional incremental parsing algorithm. Then the latent aspects of the occurrence pattern of the parsed representations are trained under the PLSA paradigm. The semantic similarity between two images is computed with the likelihood of the images on the concept space formed by the latent aspects. A detailed implementation of the proposed AMPARS systems is provided in the following section.

4.4.2.1 Two-dimensional Incremental Parsing

Among recent influential achievements in universal source coding, the most notable are the two coding algorithms proposed by Lempel and Ziv in 1977 and 1978, called the Lempel-Ziv sliding window (LZ77) [96] and the Lempel-Ziv incremental parsing (LZ78) [97]. Since then, they have been not only widely applied to many data compression applications but also extended to a variety of source coding algorithms, e.g., lossless coding schemes for nonstationary sources or multidimensional sources and lossy source coding algorithms. Their success is primarily due to the fact that without any prior knowledge of the statistical distribution of the given source, the algorithms asymptotically achieve a source rate approaching the entropy of the source. In particular, we are interested in the LZ78 because the statistics of the given source implicitly are embedded into its dictionary.

Although the LZ78 scheme has been successfully implemented in many data compression applications, it has a fundamental limitation: the coding algorithm pertains to only one-dimensional discrete source sequences. For multidimensional source sequences, a source scanning scheme is employed to generate a one-dimensional source sequence from the given sequence. This limitation motivated us to devise a multidimensional incremental parsing scheme for universal source coding. In Chapter 2, the incremental parsing scheme is implemented into lossy image compression algorithms with the two distortion functions: local average distortion and local minimax distortion. Also, for the pattern matching of the coding scheme, the scheme searches the dictionary for the maximum decimation patch. The experimental results show that the coding efficiency of the proposed scheme outperforms other existing pattern-matching based image compression algorithms. Also, when applied to image retrieval systems, the scheme parses the given visual information into a number of variable-size visual patches, and the proposed image retrieval systems shows significantly improved retrieval efficiency compared to those systems based on the conventional representations of visual information.

However, in spite of the improved performance of the image retrieval systems, it is yet to be confirmed that the incremental parsing scheme implemented is relevant for semantic matching of the given image patches. A desirable attribute of pattern matching in image retrieval systems is to minimize semantic discrepancy and not to minimize perceptual distortion. One alternative way beyond the pixel-wise perceptual distortion measure is to compare the first- and second-order statistics of given patches. Thus, we propose a different type of pattern matching criterion and implement an incremental parsing scheme for the applications of visual information analysis.

Let $\mathbf{X}$ be a two-dimensional vector field taking values from a set of three-dimensional finite vectors. Each element of the vector represents each color component, here red, green, blue (RGB), respectively. $\mathbf{X}(\vec{x})$ denotes the symbol vector at the location $\vec{x} \in \mathbb{Z}^2$. Also, we define a subset of $\mathbf{X}$ for an area vector $\vec{a} \in \mathbb{Z}^2$ as follows:

$$\mathbf{X}(\vec{x};\vec{a}) = \{\mathbf{X}(\bar{x}_1, \bar{x}_2) \,:\, x_i \leq \bar{x}_i \leq x_i + a_i,\, i = 1, 2\}. \tag{61}$$

Given a dictionary $\mathbb{D}$, we define two operations $|\cdot|$ and $[\cdot]$. $|\mathbb{D}|$ denotes the number of elements of $\mathbb{D}$, $[\mathbb{D}_j]$ refers to an area vector whose element represents the number of pixels of the j^{th} patch along each axis, and $|\mathbb{D}_j|$ corresponds to the number of symbols of the j^{th} patch. At the current pattern matching location, called an *anchor point*, denoted by Δ, the set of dictionary indices by ϵ_s-bounded similarity at Δ is

$$\mathbf{H}_s = \{j \mid \text{CSTSIM}_{\text{max}} - \text{CSTSIM}(\mathbb{D}_j,\, \mathbf{X}(\Delta; [\mathbb{D}_j])) \leq \epsilon_s,\; 0 \leq j \leq |\mathbb{D}|,\; \epsilon \in \mathbb{R}_+\}, \tag{62}$$

where $\text{CSTSIM}_{\text{max}}$ is the maximum value of CSTSIM, equivalent to 1.0, and ϵ_s denotes the bound of the structural similarity. In the proposed implementation, we manually set ϵ_s to 0.015. At each epoch, the parsing scheme constructs the set of indices following the similarity measure. Then, it selects the maximal match index $k_{\max}$ by

$$k_{\max} = \operatorname*{argmax}_{k \in \mathbf{H}_s} \{|\mathbb{D}_k|\} . \tag{63}$$

Once the maximal match is found, the scheme appends the dictionary with two new entries, each of which is obtained by appending pixels along the horizontal and the vertical axes. After the pattern matching and the dictionary augmentation are finished, Δ moves following a predetermined heuristic method. In the proposed implementation, a raster scanning order is employed for the movement of Δ. Note that, in Chapter 2, it is shown that either the raster scanning or other types of scanning order, e.g. column-wise scanning order, does not provide any considerable difference in terms of coding efficiency.

4.4.2.2 Generation of Visual Dictionary

In the aspect modeling of text document, a given text corpus is represented in its empirical distribution $p(w, d)$ by counting the number of words occurred in each document. For an image corpus, we train the aspect model from the empirical distribution of the image-patch observations by PLSA. To generate the empirical distribution, we first have to generate a visual dictionary with which all the images from the corpus can be reconstructed. We follow the heuristic shown in Section 3.4.2: at each coding iteration, a given image is encoded with the dictionary updated at the previous iteration. For every n_p image during the encoding procedure, the dictionary entries that are not used for encoding the previous n_p images are pruned, and the dictionary with reduced entries is fed back to the coding step.

In the proposed system, we set n_p to 100 and randomly chose 1,200 images for the generation of visual dictionary. Figure 27 plots the number of dictionary entries during the generation of the visual dictionary for the proposed incremental parsing scheme.

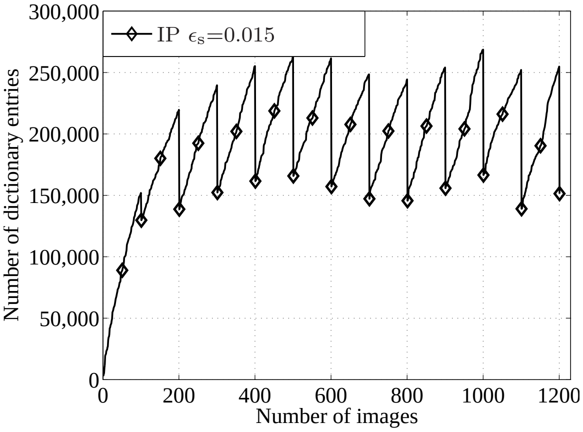


Figure 27: Number of entries in the process of the generation of visual dictionary.

As the coding scheme proceeds, the number of dictionary entries increases, and at every n_p image, the unused dictionary entries are pruned so that the number of entries decreases to a considerably lower number. It is observed that the number of entries oscillates without any major change. The number of entries in the resulting dictionary is 151,390 for $\epsilon_s = 0.015$.

4.4.2.3 Aspect Modeling by PLSA

Once the visual dictionary is generated, the occurrence patterns of the visual patches in the dictionary are analyzed for the 20,000 images in the database. Let $n(w, d)$ denote the co-occurrence count matrix of the image corpus for the given dictionary. Since the number of the occurrence of each document is heterogeneous, the distribution of occurrences within each image is smoothed out as follows:

$$p(w_i|d_j) = \frac{n(w_i, d_j)}{n(d_j)}, \tag{64}$$

where $n(d_j) = \sum_{i=1}^{M} n(w_i, d_j)$. By assuming a-priori distribution of the image uniform, i.e., $p(d) = 1/N$, we get $p(w, d) = p(w|d)p(d)$. In [4], by taking inter- and intra-document normalization into account, the co-occurrence count matrix can be weighted by a term-frequency normalized-entropy. The normalized entropy of i^{th} word ε_i is

$$\varepsilon_i = -\frac{1}{\log N} \sum_{j=1}^{N} \frac{n(w_i, d_j)}{n(w_i)} \log \frac{n(w_i, d_j)}{n(w_i)}, \tag{65}$$

where $n(w_i) = \sum_{j=1}^{N} n(w_i, d_j)$ is the total number of times the i^{th} visual patch occurs in the image corpus. The empirical distribution $p(w, d)$ is weighted by the normalized entropy ε_i as follows:

$$p_n(w_i, d_j) = (1 - \varepsilon_i)p(w_i, d_j). \tag{66}$$

For learning the latent aspects of a given image corpus by PLSA, the number of latent aspects, K, is manually set to 500. Before learning the PLSA model by EM, $P(z|d)$ and $P(w|z)$ are randomly initialized. The EM algorithm is executed until the log-likelihood function (41) converges. The condition of convergence for the EM algorithm is defined as

$$\log \frac{\mathcal{L}_{\text{new}}}{\mathcal{L}_{\text{old}}} < 0.05. \tag{67}$$

Once the latent aspects are modeled, the similarity between the query image and each image in the database is then computed. Generally, the similarity measure between documents under the PLSA paradigm is still an open problem. In other words, there is no measure that works for every information retrieval application. In one conventional approach, each document is represented as a vector from the origin on the latent space formed by the trained aspects. Then, the document similarity is measured by the cosine of the degree between the two vectors. One other approach is to use the Kullback-Leibler (KL) divergence [44] for the given conditional probabilities of the j^{th} document d_j and the query document d_q, i.e., $P(z|d_j)$ and $P(z|d_q)$, respectively. For two discrete random variables, p, and q, the KL divergence is defined as:

$$D_{\text{KL}}(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \tag{68}$$

The KL divergence is commonly referred to as the relative entropy or the KL distance. However, since the KL divergence is a non-commutative measure of the difference between two probability distributions, it is not symmetric and does not satisfy the triangle inequality; in turn, it is not a metric. The KL divergence is always non-negative and is zero if and only if $p = q$. Typically, p represents the true distribution of the random variable, and q denotes an approximation of p. It has two problems for practical distance measure. First, as mentioned previously, it is not a metric and does not satisfy $D(p \,||\, q) \neq D(q \,||\, p)$. Second, it gives ∞ when $p_i \neq 0$ and $q_i = 0$. To overcome the aforementioned problems, Lin proposed the Jensen-Shannon divergence [49]. It is defined as follows:

$$D_{\mathrm{JS}}(p \,||\, q) = D_{\mathrm{KL}}(p \,||\, m) + D_{\mathrm{KL}}(q \,||\, m), \tag{69}$$

where $m = (p + q)/2$. The Jensen-Shannon divergence is symmetric and satisfies $D_{\mathrm{JS}}(p \,||\, q) = D_{\mathrm{JS}}(q \,||\, p)$. Also, it does not give ∞ if either $p_i \neq 0$ or $q_i \neq 0$. In the proposed image retrieval system, the similarity between the j^{th} image and the query image is computed with the Jensen-Shannon divergence as

$$s(d_j,\, d_q) = D_{\mathrm{JS}}(P(z|d_j) \,||\, P(z|d_q)). \tag{70}$$

4.4.3 Fixed-block Representation under PLSA

One considerable difference between the proposed image retrieval system and the conventional systems is the source representation for visual information analysis. As mentioned previously, many of the existing image retrieval systems extract features from fixed-size image blocks or image segments. To compare the performance of the proposed system with that of the conventional approach, we design an image retrieval system based on the fixed-block image representations trained by vector quantization (VQ) as a benchmark system.

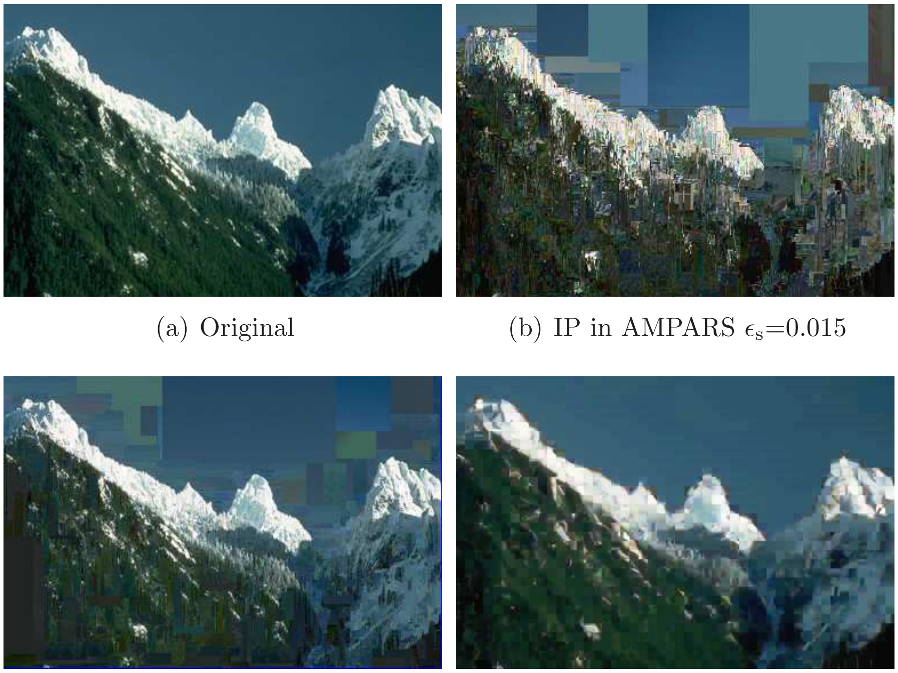
The benchmark system has the same components as the AMPARS system except that the visual dictionary is trained by VQ, and the co-occurrence of image partitions and images is generated accordingly. Then, the latent aspects of the block-image empirical distribution are trained by PLSA. Finally, the similarity between the two documents is computed by (70).

Similar to the image retrieval system with VQ shown in Chapter 3, each color image is partitioned into 8×8 blocks. Thus, the dimension of the VQ codebook is $8 \times 8 \times 3 = 192$. To train the quantizers, we use the Linde-Buzo-Gray (LBG) algorithm [27,50]. A detailed description of this training process is provided in Section 3.4.3.

Table 6 provides average peak-signal-to-noise ratio (PSNR) values for reconstructed images for three coding schemes: the incremental parsing used in IPSILON, the incremental parsing used in the proposed AMPARS system, and VQ. The PSNR values for VQ are higher than those for the incremental parsing schemes because of the nature of VQ that minimizes the average distortion. The PSNR values for the incremental parsing in AMPARS system are significantly lower than the other two. This is due to the fact that the structural similarity measure used in the parsing scheme normalizes the luminance and the contrast components so that the match found by the proposed CSTSIM may have different luminance from that of the reference patch. Figure 28 provides the examples of the reconstructed images by the three techniques. Compared to the original image, the reconstructed image by VQ has blur distortion but no salient perceptual discrepancy. In the image by the incremental parsing in IPSILON, we observe block artifacts on the “sky” and the “mountain” region of the image, but the image contains similar textual and color components on most of the regions. However, in the image by the incremental parsing in AMPARS, it is obvious that the color of the “mountain” is not the same as that in the other three images, and there are significant block-artifacts on the “sky” region. Nevertheless, we can still understand that the image is of “mountain,” “blue sky,” and “snow on the mountain.”

	Red	Green	Blue
IP in AMPARS ϵ_s=0.015	18.82	17.38	14.77
IP in IPSILON θ_{jnd}=1.6	24.26	24.50	22.04
VQ	26.60	26.88	26.85

Table 6: Comparison of average PSNRs of three coding schemes.



(a) Original

(b) IP in AMPARS ϵ_s=0.015

(c) IP in IPSILON θ_{jnd}=1.6

(d) VQ

Figure 28: Examples of reconstructed images. Average PSNRs across the color components are (b) 17.98 dB, (c) 23.84 dB, and (d) 24.86 dB.

4.5 Experimental Results

We present in this section the retrieval results and the performance evaluation of four image retrieval systems: the proposed AMPARS system, the IPSILON system, the benchmark system, and the SIMPLIcity. IPSILON and AMPARS systems use the parsed representations induced by the incremental parsing algorithms, while the benchmark system is based on the fixed-block representation of visual information trained by VQ. Aspect modeling learned by PLSA technique underlies the benchmark and the AMPARS systems, while the IPSILON systems analyzes the given image corpus by LSA. By the k-means clustering algorithm, the SIMPLIcity partitions a given image into a few regions, then the semantic similarity between the regions of the two given images is computed by an integrated region matching (IRM). For a detailed description of SIMPLIcity, the reader is referred to [48, 81].

Those systems are implemented with the same image databases and evaluated with the same query images. In many information retrieval applications, the performance of a retrieval system is commonly evaluated by precision/recall tests. Since the number of documents or images in recent databases is too large to compute the recall, we focus on a few most relevant images without examining the entire retrieved images. As provided in Chapter 1, the retrieval performance is measured by the total average precision. For m visual concepts, the average precision for the r-most relevant precision is

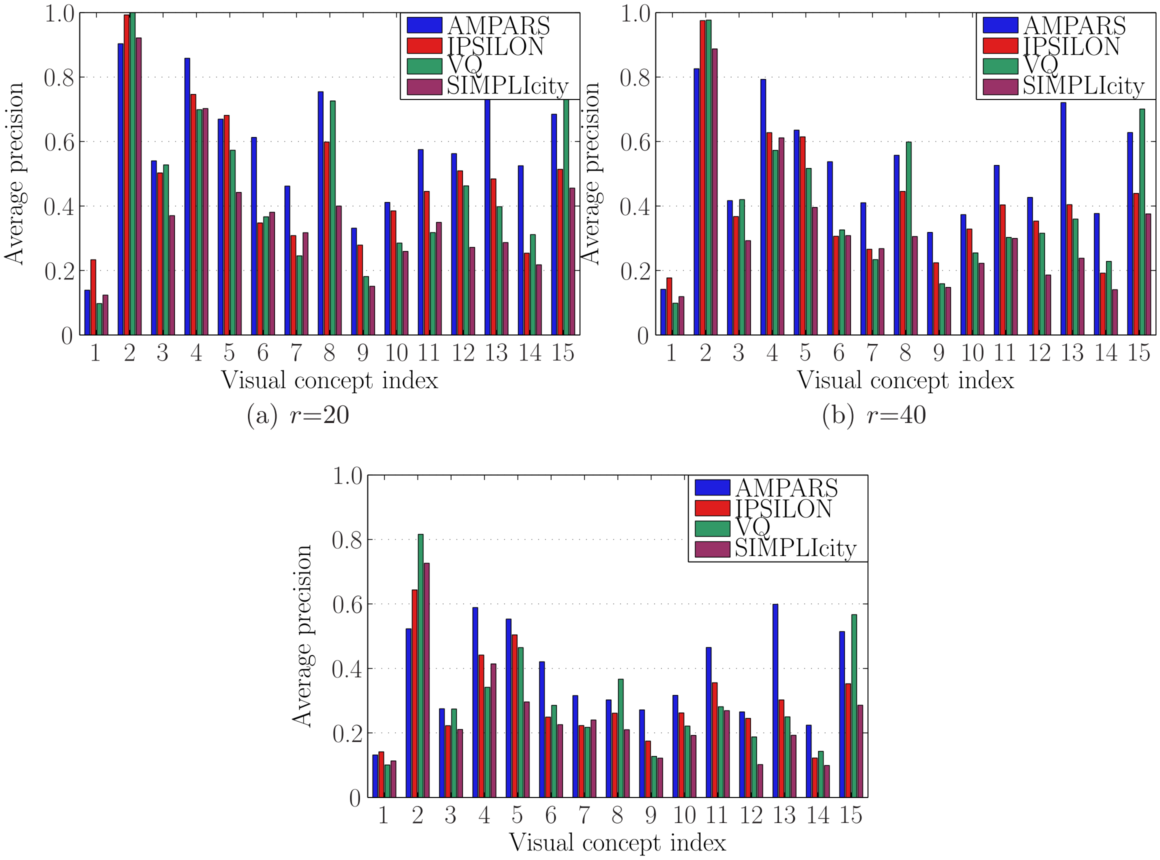
$$\frac{\sum\limits_{j=1}^{q_i} s_{i,j}}{r \cdot q_i}, \tag{71}$$

where q_i denotes the number of query images in the i^{th} visual concept and $s_{i,j}$ means the number of relevant images for the j^{th} query in the i^{th} concept. The total average precision for all the queries is defined as

$$\frac{\sum\limits_{i=1}^{m} w_i \sum\limits_{j=1}^{q_i} s_{i,j}}{r \cdot \sum\limits_{i=1}^{m} q_i}, \tag{72}$$

where $w_i = (m \cdot n_i)/(\sum_{i=1}^{m} n_i)$ and n_i refers to the number of images in the i^{th} visual concept.

Figure 29 compares the average precisions of the four retrieval systems. At r=20, for 11 visual concepts among 15, the average precisions of the proposed AMPARS system are significantly higher than those of the other three systems. Especially, for those concepts, “food-dish,” “snow-glacier,” “mountain-forest,” “sunset,” and “tiger,” the average precisions of the AMPARS system are over 0.1 higher than the other two.



(a) r=20

(b) r=40

(c) r=100

Figure 29: Comparison of average precisions of the four image retrieval systems for each visual concept.

Also, at r=40 and r=100, the average precisions of the AMPARS system are considerably higher, particularly for those concepts, "fireworks," "mountain-forest," and "sunset." Table 7 provides a numerical comparison of the total average precisions for the four retrieval systems. The total average precisions of the proposed AMPARS system are over 0.10 higher than the other three systems at all the rs, r=20, r=40, and r=100. When compared under the same aspect modeling paradigm, the AMPARS system outperforms the benchmark VQ system in terms of retrieval precision.

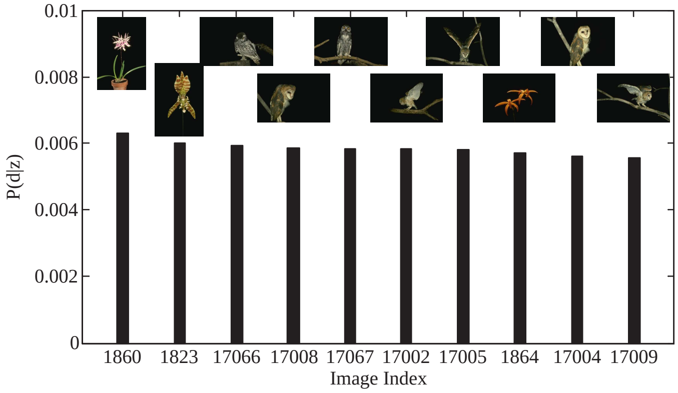
	AMPARS	IPSILON	VQ	SIMPLIcity
r=20	0.602	0.502	0.420	0.381
r=40	0.550	0.435	0.380	0.323
r=100	0.457	0.353	0.324	0.264

Table 7: Total average precisions of the four image retrieval systems.

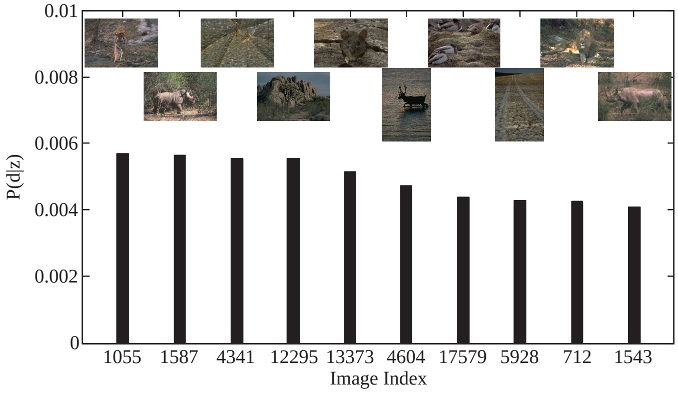
Note again, the only difference between the proposed and the benchmark system is that the proposed one uses the parsed representation by the incremental parsing that minimizes the number of visual patches used in the reconstruction of an given image, while the benchmark system uses the fixed-block representation by VQ that minimizes the average distortion of the reconstructed images.

Figures 30 and 31 illustrate the conditional probability distribution $P(d|z)$ for the two maximums of $P(z)$ in the modeling of the benchmark system and the proposed system. Due to the limitation of space, we only provide the distributions of the images of the 20 highest $P(d|z)$. Also, the corresponding images are superimposed on each plot. In Figure 30 of VQ, the latent aspects, which are the highest probabilities of $P(z)$, are at $P(z_k = 206)$ and $P(z_k = 133)$. These two latent aspects correspond to the two concepts, "black background" and "ground color." However, as seen from the images that have the highest probabilities of $P(d|z_k = 206)$ and $P(d|z_k = 133)$, we observe that the two latent aspects are not closely related to particular objects. On the other hand, as shown in Figure 31 pertaining to the incremental parsing, the two latent concepts reflect the two concepts: "green trees" and "desert." These two concepts are directly matched with the two visual concepts, "desert-pyramid" and "mountain-forest," among the 15 visual concepts that we identified from the image corpus.

Figures 32 and 33 provide the latent aspect decomposition and the probabilities of the words of the corresponding latent aspects for two images, "sunset" and "flower."



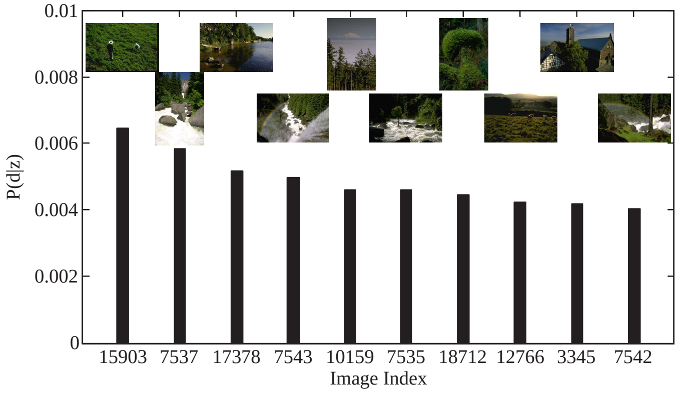
(a) z_k=206



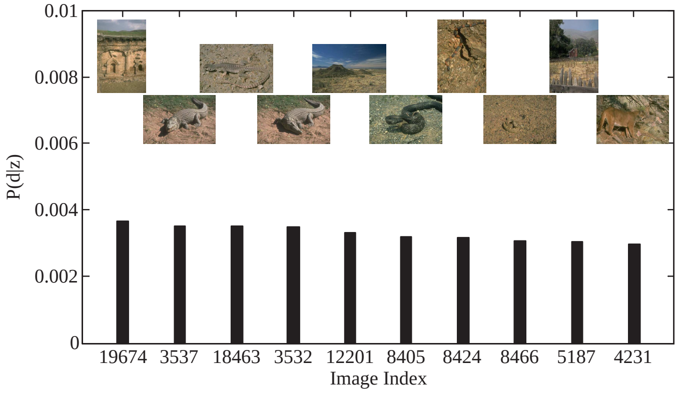
(b) z_k=133

Figure 30: Conditional probability distributions of $P(d|z)$ for the two maximums of $P(z)$ in the benchmark system based on the fixed-block representation trained by VQ.

In Figure 32, we observe that the image consists of three image regions: sun with yellow, sky with orange, and sea with deep blue and brown. In the parsed representation, the variable-size patches of the top three latent aspects are in orange, yellow, and dark brown color that correspond to the concepts, “sky,” “sun,” and “sea,” respectively. Conversely, in the fixed-block representations, the image blocks of the top three latent aspects are in brown and gray, which are mostly for the “sea” regions. Similarly, in Figure 33, the variable-size patches of the two latent aspects represent the “pink and red flowers,” and the patches of the other matches with the “green leaves.”



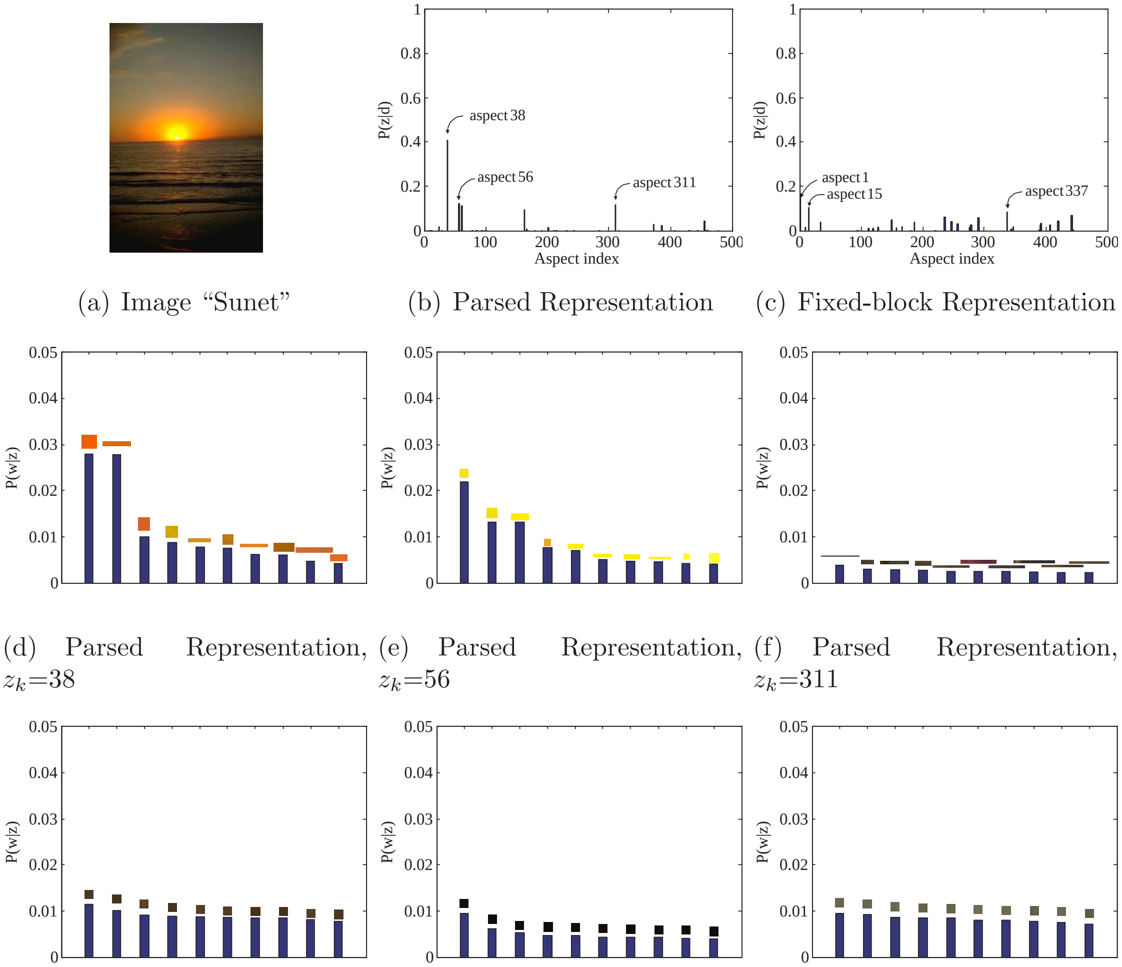
(a) z_k=253



(b) z_k=362

Figure 31: Conditional probability distributions of $P(d|z)$ for the two maximums of $P(z)$ in the proposed system based on the parsed representation generated by incremental parsing.

However, the fixed-block patches of the two latent aspects are for the "green leaves." This irrelevancy of the fixed-block representations is mainly due to the nature of VQ. Since the size of the image blocks is fixed, the occurrence patterns of blocks in VQ are dominated by the size of the regions. In turn, these comparisons justify that relaxing the constraint of the size of image blocks significantly affects the efficiency of visual information analysis.



(a) Image “Sunet”

(b) Parsed Representation

(c) Fixed-block Representation

(d) Parsed Representation, z_k=38

(e) Parsed Representation, z_k=56

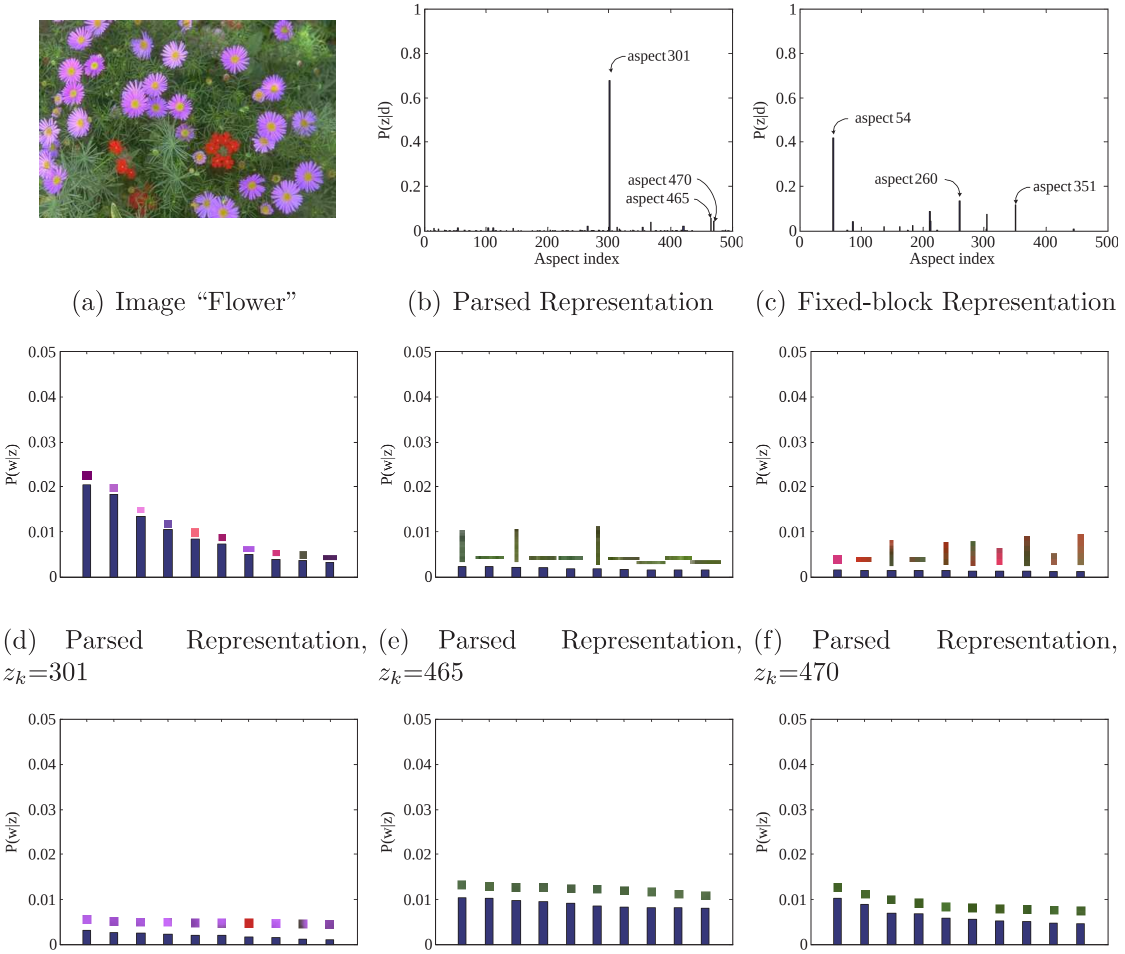
(f) Parsed Representation, z_k=311

(g) Fixed-block Representation, z_k=1

(h) Fixed-block Representation, z_k=15

(i) Fixed-block Representation, z_k=337

Figure 32: The probability distribution of latent aspects $P(z|d)$ for the given image “sunset” and the probability of words for three latent aspects of the highest probability.



(a) Image "Flower"

(b) Parsed Representation

(c) Fixed-block Representation

(d) Parsed Representation, z_k=301

(e) Parsed Representation, z_k=465

(f) Parsed Representation, z_k=470

(g) Fixed-block Representation, z_k=54

(h) Fixed-block Representation, z_k=260

(i) Fixed-block Representation, z_k=351

Figure 33: The probability distribution of latent aspects $P(z|d)$ for the given image "flower" and the probability of words for three latent aspects of the highest probability.

4.6 Discussions

In this chapter, we have proposed a probabilistic framework of content-based image retrieval based on the parsed representation and implemented AMPARS system. With a multidimensional incremental parsing technique, as an extension of the Lempel-Ziv incremental parsing, a given image is parsed into a number of patches of variable size. These patches can be thought of as a morphological interface between elementary pixels and a higher level representation beyond the conventional fixed-block representation. The incremental parsing technique is implemented with a new structural similarity measure that compares the first- and second-order statistics of image patches. By a dictionary generation heuristic approach, a visual dictionary for a given image corpus is generated. For an image corpus, we train the aspect model from the empirical distribution of the image-patch observations by PLSA. The semantic similarity between the given two images is computed by the Jensen-Shannon divergence of the two conditional probabilities of the latent aspects. We have implemented two image retrieval systems: one is the proposed AMPARS system, and the other is a benchmark system that uses fixed-block representations of visual information trained with VQ. The performance of these two systems is compared with two existing systems: IPSILON based on the incremental parsing with the minimax distortion under the LSA paradigm and the SIMPLIcity system based on an image segmentation technique. These four systems are tested with 20,000 images of natural scenes and 600 query images. The experimental results show that the proposed framework reasonably captures the visual semantics appeared in the image corpus and the AMPARS system, based on the proposed probabilistic analysis framework, outperforms other systems in terms of retrieval precision.

CHAPTER V

ROBUSTNESS EVALUATION OF IMAGE RETRIEVAL SYSTEMS

5.1 Introduction

In Chapters 3 and 4, we have proposed two types of image retrieval frameworks based on a universal source coding technique. Also, we have extensively compared the performance of the proposed image retrieval systems with that of existing systems in terms of retrieval precision for a given set of query images. However, the image that a user issues for pointing to the query in the user's mind may contain some distortions among the following three. First, the query issued may be in a heterogeneous shape although it is semantically homogeneous, we call this *visual synonymy*. Second, the query may be analyzed and understood as a different target from the one that the user desired, we call this *visual polysemy*. Third, the user issues incorrect or distorted visual queries, called *visual ambiguity*. Thus, to practically evaluate the performance of image retrieval systems, one has to take the noise robustness into account as well as retrieval precision.

In this chapter, we perform the robustness evaluation of the image retrieval systems described in Chapters 3 and 4. We set our focus on more specific image perturbations: scale variations, geometric variations, additive noise, and statistical pixel variation. In Section 5.2, we introduce nine different types of image perturbations and the detailed implementations of them. Section 5.3 provides the experimental results of the aforesaid image perturbations for the image retrieval systems in the previous two chapters. Section 5.4 discusses the results.

5.2 Simulation of Image Perturbations

In the performance evaluation of the image retrieval systems in the two previous chapters, each query image is a complete image as stored in the database. In many practical image retrieval systems, the query image may not be the same image as the desired one in the database. We thus performed a number of experiments on the evaluation of the robustness of the image retrieval systems. All of the 600 query images are distorted by nine different types of image perturbations. Therefore, we generated a total of 5400 query images. Figure 34 shows examples of the nine image perturbations, such as scale variation, rotation, sharpness variation, additive noise, cropping, brightness variation, and shifting. Detailed descriptions of the nine perturbations are as follows:

1. Scale variation: The spatial resolution of each image with size 384×256 or 256×384 is changed to 480×320 or 320×480 for up-sampling and to 288×192 or 192×288 for down-sampling using the bi-cubic interpolation filter [38].

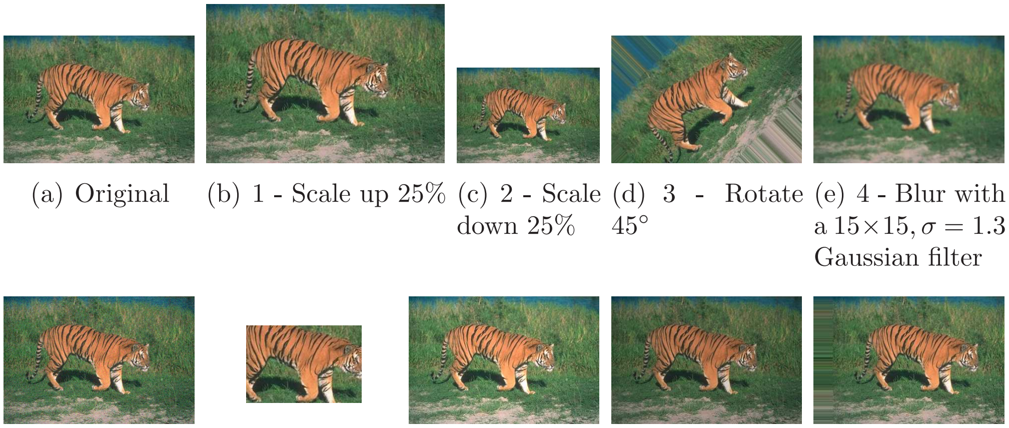
2. Rotation: Along the center of each image, the image is rotated by 45° in a counterclockwise direction. The bi-cubic interpolation filter is also used. Boundary pixels are extrapolated with the nearest pixels of the image.

3. Sharpness variation: Each image is blurred by convolving each color component with a rotationally symmetric two-dimensional Gaussian lowpass filter

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{73}$$

   where x and y are in $[-7, 7]$. σ is manually set to 1.3 at which a human can barely identify the blurred scene.

4. Additive noise: Each image is contaminated with a Gaussian noise with μ=0.0 and σ^2=0.002.



(a) Original (b) 1 - Scale up 25% (c) 2 - Scale down 25% (d) 3 - Rotate 45° (e) 4 - Blur with a 15×15, $\sigma = 1.3$ Gaussian filter

(f) 5 - Gaussian noise, $\mu = 0, \sigma^2 = 0.002$ (g) 6 - Cropping 50% (h) 7 - Brighten 15% (i) 8 - Darken 15% (j) 9 - Horizontal shifting by 10%

Figure 34: Examples of nine image perturbations for the robust evaluation of the image retrieval systems.

5. Cropping: The center portion of each image with size 192×128 or 128×192 is cropped. Each cropped image contains partial objects on the scene, but the semantics of the scene can still be recognized by a human.

6. Brightness variation: All the pixel values of each image are added or subtracted by $0.15 \times \mu_x$, where μ_x is the mean of the pixel values in each color channel.

7. Shifting: All the pixels of each image are horizontally shifted by 25 pixels for 384×256 images or 38 pixels for 256×384 images, respectively. As in the perturbation of rotation, the boundary pixels are extrapolated with the nearest pixels of the image.

For the evaluation of the IPSILON systems, the occurrence vector for each perturbed image is generated with the corresponding visual dictionary. Then, each vector is normalized and weighted by (28). Let us denote the perturbed query vector by $\mathbf{q}_p$. By (30), the vector is projected onto the K-reduced space as $\hat{\mathbf{q}}_p = \hat{\Sigma}_K^{-1} U^T \mathbf{q}_p$. Then, the similarity between the perturbed query vector $\hat{\mathbf{q}}_p$ and each image vector is computed by (31). Note that in this experiment, we focus only on the case of K=800.

For the evaluation of the AMPARS system, each perturbed query image $\mathbf{q}_p$ is represented as an empirical distribution $P(\mathbf{q}_p)$. Then, the conditional probability $P(z|\mathbf{q}_p)$ is estimated by the *folding-in* method proposed in [30]. In such a method, the likelihood of the query image with respect to the learned model $P(z)$ and $P(w|z)$ is maximized.

5.3 Experimental Results

In this section, we evaluate the noise robustness of the IPSILON and the AMPARS image retrieval systems by comparing them with existing image retrieval systems. We first generate 5400 query images, having 600 images for each image perturbation. With each set of 600 query images, we evaluate the average precision and the total average precision of the image retrieval systems. For both evaluations, we measure the retrieval precision for the top 20 most relevant images, i.e., r is set to 20.

5.3.1 IPSILON Systems

Figure 35 summarizes the results of the five retrieval systems: three are the IPSILON systems with different perceptual distortion thresholds, one is the benchmark the VQ system under the LSA paradigm, noted as "VQ+LSA," and the other is the SIMPLIcity system. Note that for the IPSILON and the VQ systems, the LSA dimension K is set to 800. From the figure, we observe that the IPSILON systems and the benchmark system are robust to geometric transformations, such as scale variation, rotation, cropping, and shifting. The poor performance of the proposed systems for the pixel value variations, e.g., brightness variation, sharpness variation, and additive noise, can be attributed to the pattern matching criterion because the prescribed distortion function (21) depends on the computation of pixel-by-pixel distortion not on the statistics of pixel values in the patch. On the other hand, the SIMPLIcity system is robust to most of the image perturbations except cropping. Since the SIMPLIcity system attempts to find the best correspondences of each image segment, the technique is fragile against partial segment matching. Table 8 compares the total average precisions of the five systems for the perturbed queries. Among the IPSILON systems, the system with θ_{jnd}=2.0 shows the best precision, but the precision differences among the IPSILON systems are not significant. For the six image perturbations (1, 2, 3, 4, 6, and 8), the total average precisions of IPSILON with θ_{jnd}=2.0 are significantly higher than those of the SIMPLIcity and the VQ systems. In all, the proposed technique is fairly robust to various image perturbations.

5.3.2 AMPARS Systems

In this section, the robustness performance of the AMPARS system is compared with that of three image retrieval systems: IPSILON with θ_{jnd}=1.6, the benchmark VQ, and the SIMPLIcity systems. To avoid confusion, "VQ+PLSA" denotes the benchmark VQ system under the PLSA paradigm. The LSA dimension K for IPSILON is set to 800. Figure 36 provides the results of the four retrieval systems. As noticed, the results for the IPSILON and SIMPLIcity systems are the same as those provided in Figure 35. The average precision of the proposed AMPARS system outperforms that of the other systems for seven types of perturbations (1, 2, 3, 4, 7, 8, and 9). In particular, the performance for brightness change has been significantly improved compared to the results of the IPSILON systems. This is primarily due to the characteristics of SSIM metrics; the luminance term of SSIM mitigates the effect of the brightness change. On the other hand, the two perturbations, blur (5) and Gaussian noise (6), notably exacerbate the average retrieval precision of the AMPARS system. These poor results are because the two perturbations, blur and Gaussian noise, seriously affect the structure term in the SSIM or the texture term in the STSIM. Brooks *et al.* in [8, Sec. IV-A] report their experimental results that local pixel variations, e.g., blur and additive noise, significantly degrade the structural similarity measures.

Perturbation Index	org	1	2	3	4	5
IPSILON $\theta_{\text{jnd}} = 1.6$	0.502	0.451	0.481	0.459	0.424	0.415
IPSILON $\theta_{\text{jnd}} = 2.0$	0.480	0.468	0.484	0.468	0.430	0.419
IPSILON $\theta_{\text{jnd}} = 2.5$	0.470	0.459	0.476	0.460	0.422	0.415
VQ+LSA	0.322	0.359	0.361	0.358	0.356	0.355
SIMPLIcity	0.381	0.401	0.408	0.389	0.348	0.416

Perturbation Index	6	7	8	9
IPSILON $\theta_{\text{jnd}} = 1.6$	0.422	0.425	0.419	0.428
IPSILON $\theta_{\text{jnd}} = 2.0$	0.425	0.428	0.423	0.432
IPSILON $\theta_{\text{jnd}} = 2.5$	0.421	0.425	0.421	0.429
VQ+LSA	0.349	0.341	0.332	0.335
SIMPLIcity	0.303	0.429	0.363	0.420

Table 8: Total average precision of IPSILON, the benchmark VQ, and the SIMPLIcity systems for nine types of image perturbations. r is set to 20.

Table 9 compares the total average precisions of the four systems for the perturbed queries. The total average precision of the proposed AMPARS system for the six image perturbations (1, 2, 3, 6, 7, 8, and 9) are substantially higher than the other systems. In particular, for the five perturbations (2, 3, 6, 7, 8, and 9), the precisions are over 1.00 higher than the other systems.

5.4 Discussions

In this chapter, we undertook the evaluation of the noise robustness of image retrieval systems. First, we studied three different types of visual distortions: *visual synonymy*, *visual polysemy*, and *visual ambiguity*. Also, the detailed implementations of the nine image perturbations that are of interest and the experimental procedure for the performance evaluation are provided.

For the robustness evaluation of the IPSILON systems, we compared the five image retrieval systems: three IPSILON systems with different perceptual thresholds, the benchmark VQ system under the same framework that underlies the IPSILON systems, and the SIMPLIcity system.

Perturbation Index	org	1	2	3	4	5
AMPARS	0.602	0.587	0.637	0.562	0.364	0.093
IPSILON	0.502	0.451	0.481	0.459	0.424	0.415
VQ+PLSA	0.420	0.521	0.512	0.491	0.480	0.509
SIMPLIcity	0.381	0.401	0.408	0.389	0.348	0.416

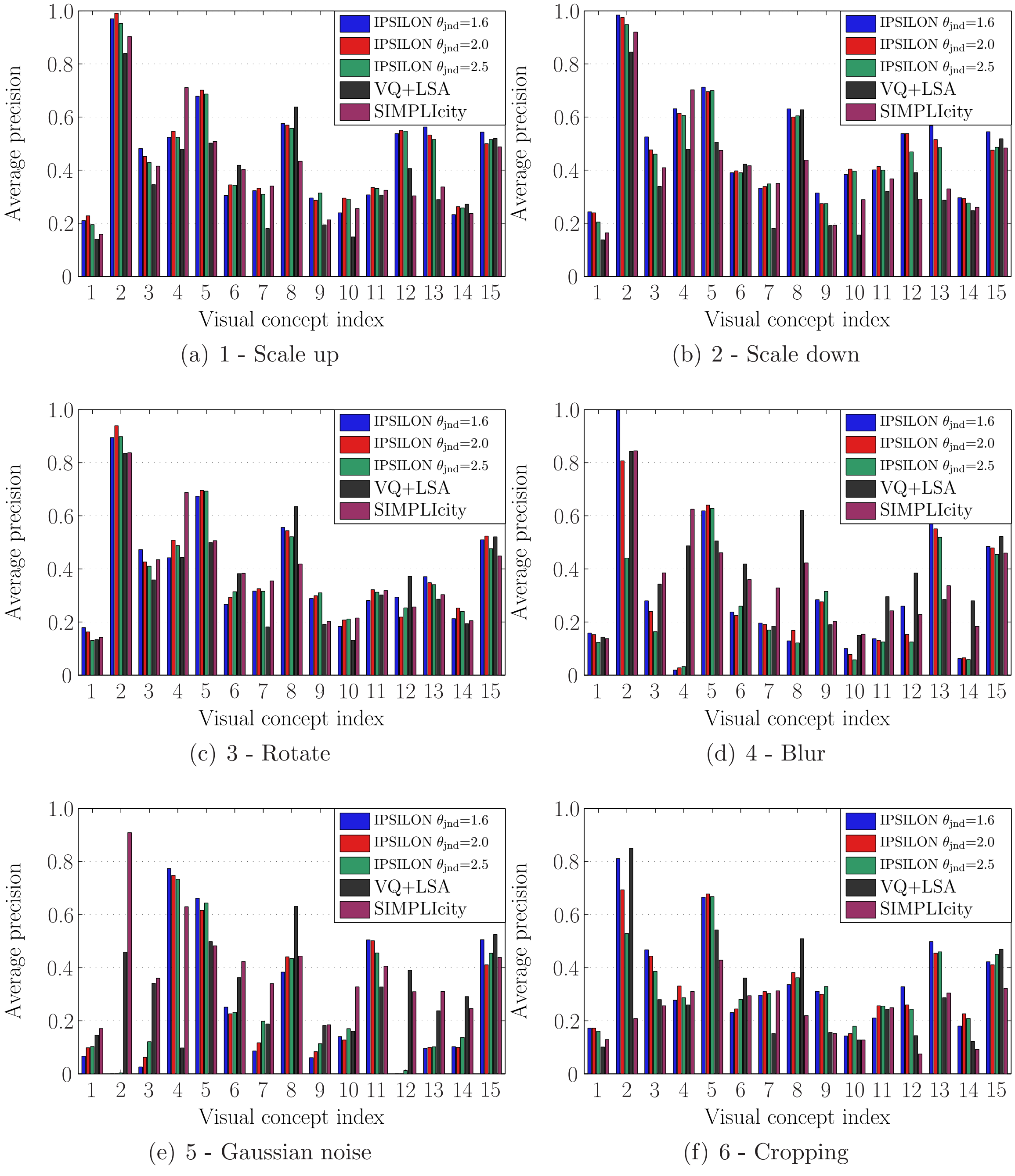
Perturbation Index	6	7	8	9
AMPARS	0.588	0.624	0.596	0.642
IPSILON	0.422	0.425	0.419	0.428
VQ+PLSA	0.451	0.461	0.440	0.521
SIMPLIcity	0.303	0.429	0.363	0.420

Table 9: Total average precision of AMPARS, IPSILON, the benchmark VQ, and the SIMPLIcity systems for nine types of image perturbations. r is set to 20.

The experimental results show the proposed IPSILON systems, especially the system with θ_{jnd}=2.0, are fairly robust to various image perturbations. It is observed that the proposed systems are comparatively less robust for the pixel value variations, e.g., brightness variation, sharpness variation, and additive noise. This weak performance can be attributed to the pattern matching criterion of the IPSILON systems, which is the minimax distortion function, because it depends on the computation of pixel-by-pixel distortion.

In the evaluation of the AMPARS system, we compared the four image retrieval systems: AMPARS, IPSILON with θ_{jnd}=1.6, the benchmark system based on VQ, and the SIMPLIcity system. From the robustness evaluation of the AMPARS system, we observed the improved retrieval performance of AMPARS compared with that of the IPSILON systems except two types of perturbations, blur and Gaussian noise. These two types of perturbations notably exacerbate the retrieval performance of the AMPARS system. These poor results can be attributed to the structural similarity measure, the pattern matching criterion of the AMPARS system.

Although the two proposed matching functions, the minimax distortion (21) and the CSTSIM (60), do not perform well for all the image perturbations introduced, we observed the performance improvement of the robustness evaluation from IPSILON to AMPARS by applying a different class of pattern matching criterion. A deeper study on the pattern matching criterion that guarantees a higher retrieval precision and an improved noise robustness is an open problem.



(a) 1 - Scale up

(b) 2 - Scale down

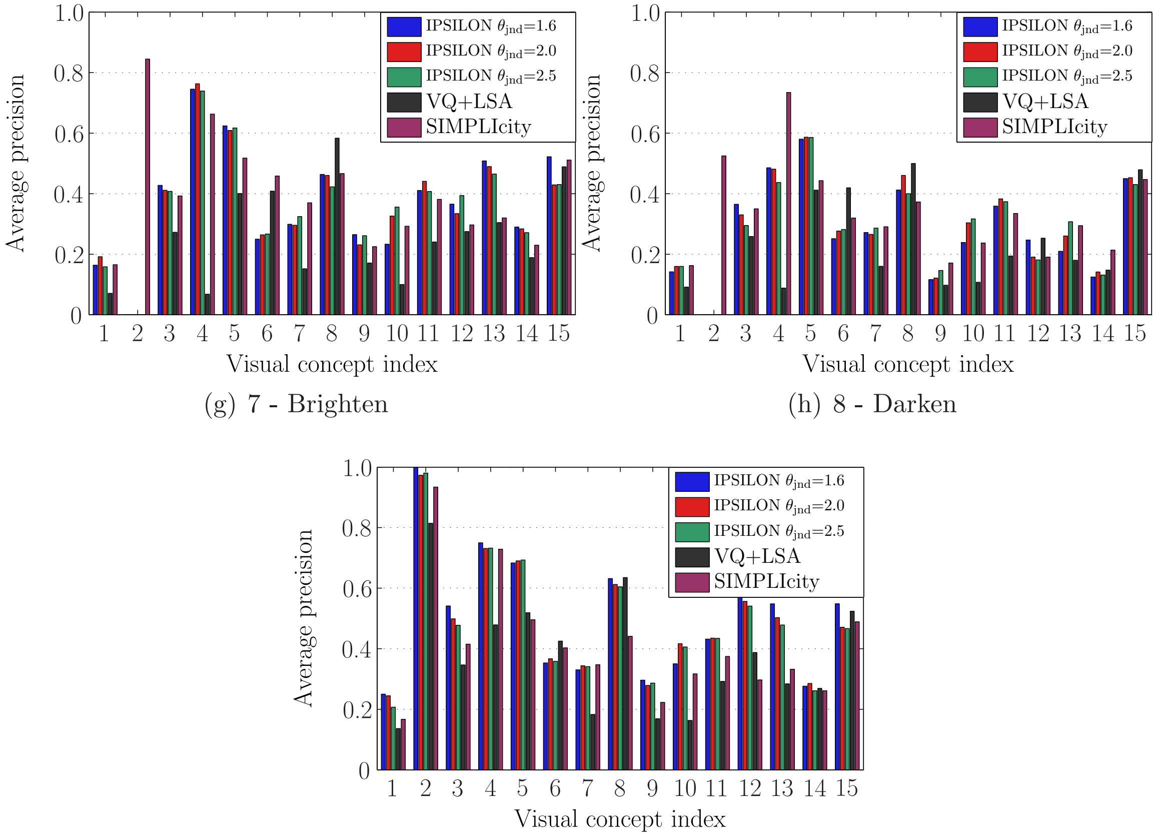
(c) 3 - Rotate

(d) 4 - Blur

(e) 5 - Gaussian noise

(f) 6 - Cropping

Figure 35: Comparison of the IPSILON systems with the benchmark VQ and the SIMPLIcity systems for nine types of image perturbations.

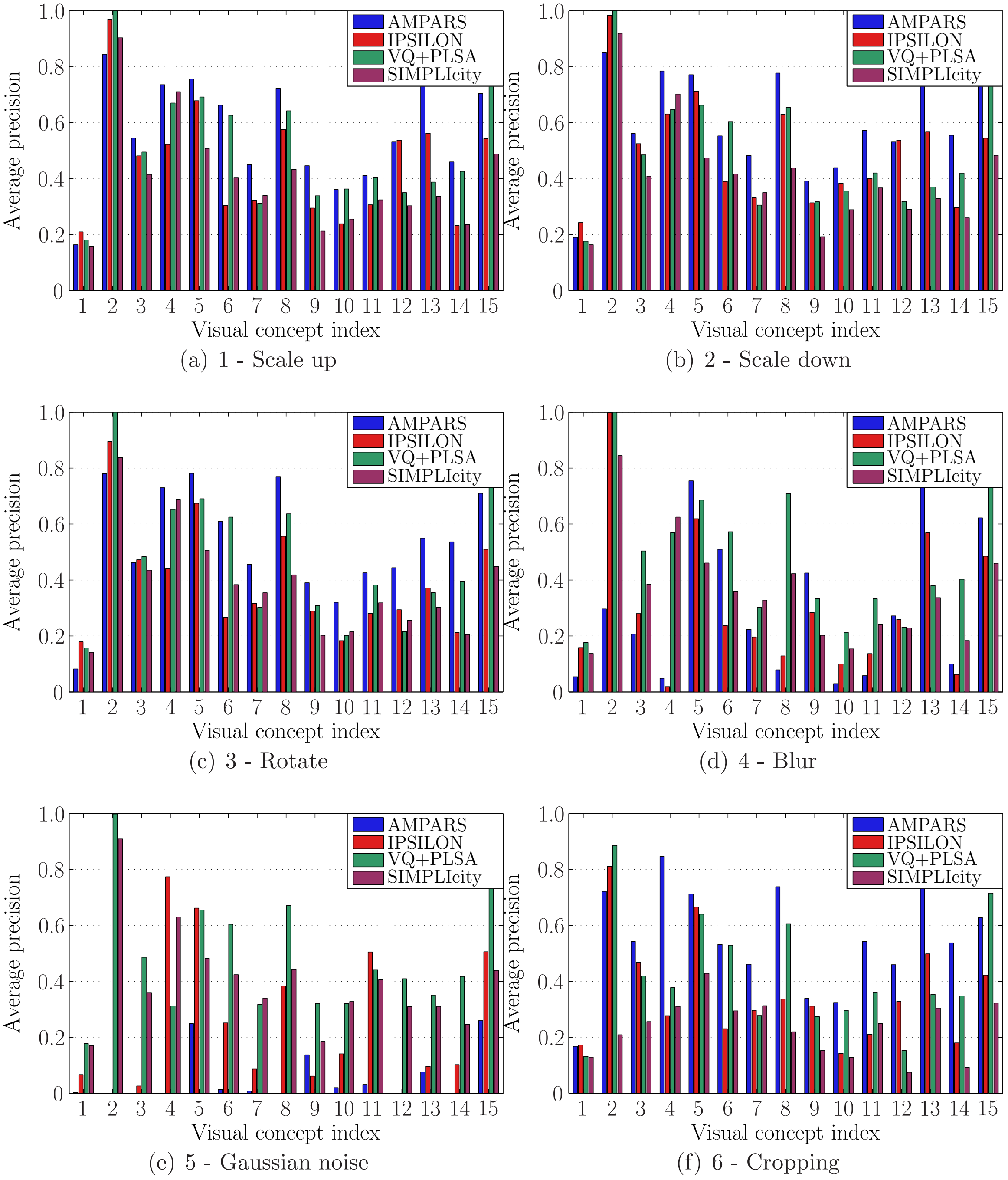


(g) 7 - Brighten

(h) 8 - Darken

(i) 9 - Horizontal Shift

Figure 35: *(Continued.)* Comparison of the IPSILON systems with the benchmark VQ and the SIMPLIcity systems for nine types of image perturbations.



(a) 1 - Scale up

(b) 2 - Scale down

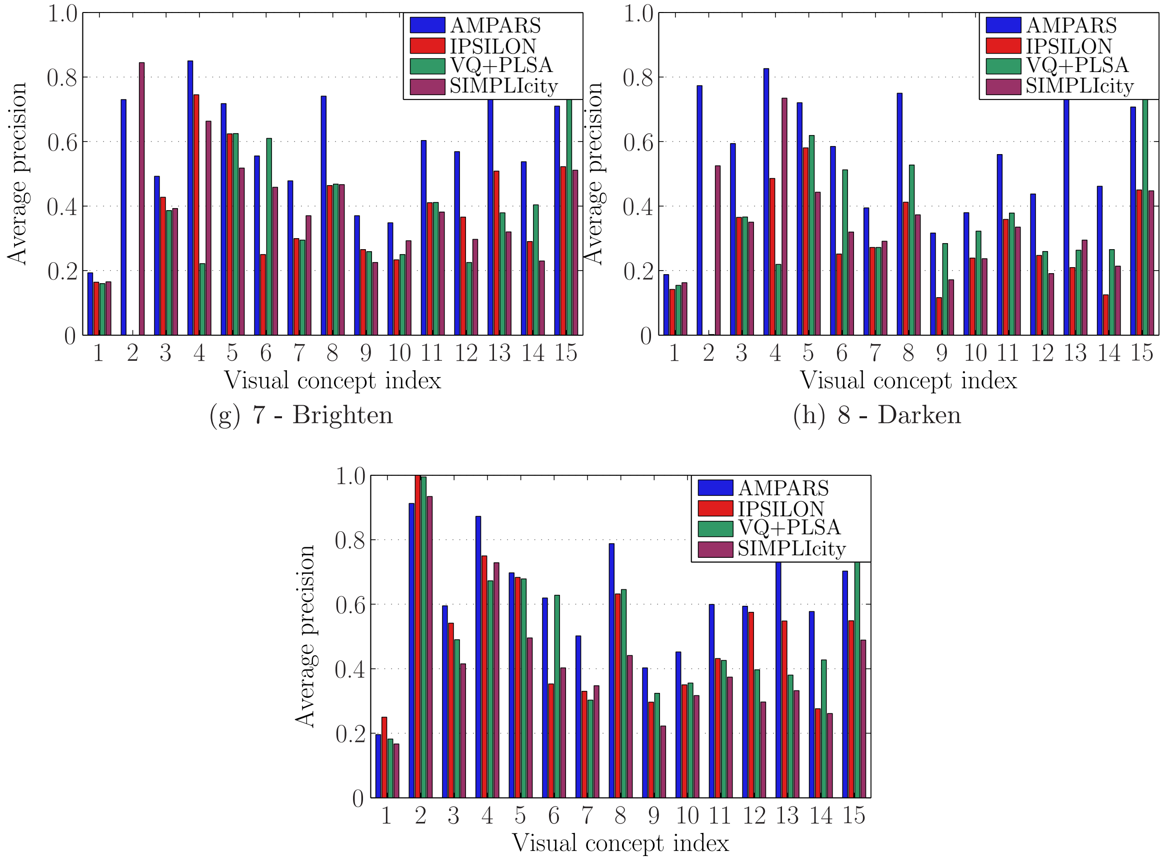
(c) 3 - Rotate

(d) 4 - Blur

(e) 5 - Gaussian noise

(f) 6 - Cropping

Figure 36: Comparison of the AMPARS system with IPSILON, the benchmark VQ, and the SIMPLIcity systems for nine types of image perturbations.



(g) 7 - Brighten

(h) 8 - Darken

(i) 9 - Horizontal Shift

Figure 36: *(Continued.)* Comparison of the AMPARS system with IPSILON, the benchmark VQ, and the SIMPLIcity systems for nine types of image perturbations.

CHAPTER VI

CONCLUSION AND FUTURE WORK

6.1 Contributions

The information retrieval frameworks that have been proposed so far have a fundamental limitation in source representation. A proper and efficient representation of a given multidimensional source has been received increasing attention from such areas as universal source coding, pattern recognition, and machine learning. In this dissertation, we have presented a new information retrieval framework based on a universal source coding technique, in particular incremental parsing algorithms. Given the motivation that the Lempel-Ziv incremental parsing algorithm achieves the optimal coding performance by efficiently parsing a given source sequence, we have applied the optimal parsing scheme to information retrieval problems by parsing given multidimensional sources. The original Lempel-Ziv incremental parsing pertains only to a one-dimensional source sequence. Thus, in Chapter 2, we have devised a multidimensional incremental parsing algorithm and studied the three component schemes. The algorithm was implemented into two-dimensional data compressions with two different distortion functions, and their performance was compared with that of existing image compression algorithms based on pattern matching. The proposed incremental parsing algorithm parses a given two-dimensional sequence into a number of variable size *patches*, each of which contains the same amount of bit information from an information theoretic standpoint. We call this methodology the parsed representation.

In Chapter 3, we have proposed a scheme for a visual dictionary generation with which we analyze the occurrence patterns of a given image corpus. Based on the vector space model, each image is represented as an occurrence vector. Once a patch-image co-occurrence matrix is formed by collecting all the vectors, it is projected onto a lower-dimensional space where the similarity between images is computed. We have implemented three image retrieval systems with different perceptual distortion thresholds and evaluated the performance of retrieval precision with two other image retrieval systems. We have designed an image retrieval system based on a fixed-block representation trained by VQ under the same LSA paradigm. The other is the SIMPLIcty system based on an image segmentation technique. The performance evaluation showed that the proposed IPSILON systems significantly outperform the other two image retrieval systems in terms of retrieval precision. We also studied the latent semantic dimensions of the parsed and the fixed-block representations and showed that the parsed representation induced by the incremental parsing algorithm is efficient in a semantic analysis of two-dimensional information.

Although the proposed IPSILON framework efficiently captures the visual semantics of given imagery information, the framework has several limitations against a flexible framework for information retrieval. Thus, in Chapter 4, we exploited a probabilistic framework for information retrieval and implemented an image retrieval system with a new type of pattern matching criterion. Though the probabilistic framework is implemented as a content-based image retrieval system, called AMPARS, along the same line as the IPSILON systems, the framework can be potentially extended to broader types of systems over the content-based systems. The pattern matching, which underlies the AMPARS system, is an extension of the structural similarity measure. The measure computes first- and second-order statistics of given patches to compute the four structural components: luminance, contrast, horizontal texture, and vertical texture. Although the proposed pattern matching criterion yields poor image reconstruction results, it is observed that the measure is efficient in the semantic comparison of visual patches. The experimental results provided that the proposed AMPARS systems showed superior performance compared with three image retrieval systems: the IPSILON, a benchmark system based on fixed-block representations, and the SIMPLIcity.

In Chapters 3 and 4, we have evaluated the image retrieval systems in terms of retrieval precision. However, the image that a user issues for pointing to the query in the user's mind may contain some distortions. Therefore, in Chapter 5, we have tested the noise robustness of the systems with perturbed queries. The query images that are used for the evaluation of the retrieval precision are distorted by nine different types of image perturbations. The experimental results show that the two proposed systems are fairly robust against the distorted query images. The IPSILON systems are particularly robust to geometric perturbations, while the AMPARS system is robust to most of the perturbations except sharpness variation and additive noise. We attributed the performance degradation to the characteristics of the proposed structural similarity measure.

The contributions of this dissertation can be summarized as the following list of journal publications:

- S. H. Bae and B.-H. Juang, "Multidimensional Incremental Parsing for Universal Source Coding," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 1837-1848, Oct., 2008

- S. H. Bae and B.-H. Juang, "IPSILON: Incremental Parsing for Semantic Indexing of Latent Concepts," submitted to *IEEE Transactions on Image Processing* 2008

- S. H. Bae, B.-H. Juang, and T. N. Pappas, "Aspect Modeling of Parsed Representations for Image Retrieval," submitted to *IEEE Transactions on Image Processing*

6.2 Avenues of Future Research

Significant progress on the problem of source representation toward visual information analysis has been made in this dissertation. Although not mentioned explicitly, the proposed source representation induced by the incremental parsing algorithm can be considered as an initial morphological interface between primitive source symbols, here pixels, to a higher level representation toward human semantics. A fundamental problem in the research of visual information analysis, e.g., image retrieval and image annotation, is the way to fill the gap between the multidimensional source and human semantics. We believe there exists a systematic view of these two ends, a visual representation hierarchy as shown in Figure 37. In the diagram, patches by the incremental parsing are the interface between image pixels and the higher level representations, e.g., objects and scenes. At each stage of the hierarchy, we can also add combinations of elements to obtain *super-regions* (that relate two or more segments or patches) and *super-objects* (that relate two or more objects). These are analogous to n-gram modeling in language processing [52]. However, deeper beneath and permeating this structure are the perceptual attributes of texture, color, and shape. We believe this hierarchy may be similar to the broad phonetic classes in a human language. A systematic study on the construction of this hierarchy will be an interesting research topic.

Another problem will be an extension of the aspect modeling technique, which is a generative model of a given document corpus for a given word lexicon. It estimates the term-document joint distribution $P(w, d)$ with respect to the empirical distribution $p(w, d)$. In the proposed image retrieval framework, documents and words correspond to images and patches, respectively. To go beyond the query-by-example paradigm, a new probabilistic model needs to be considered. One way is to devise a technique that can take *semantic keywords* into account by modeling a patch-image-keyword joint distribution $P(w, d, y)$, where $y \in Y = \{y_1, \cdots, y_s\}$ denotes a semantic

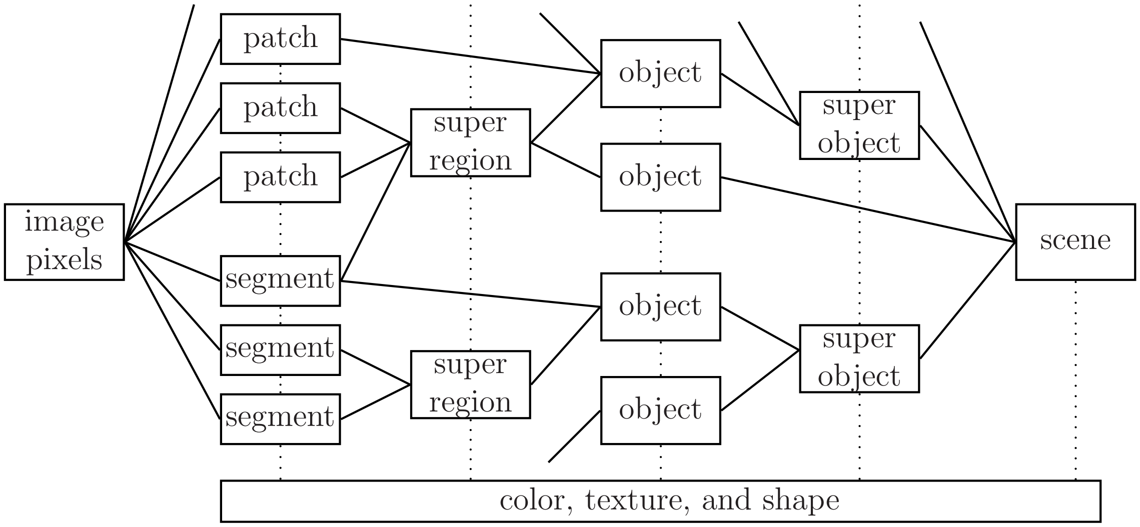


Figure 37: Block diagram of the visual representation hierarchy.

keyword.

We believe that this modeling can solve many problems, e.g., keyword-based information retrieval, image annotation, and so on.

To compare the proposed image retrieval systems, we evaluated the performance of the systems in terms of retrieval precision and noise robustness. However, we only dealt with a limited number of aspects of the robustness issues. The two types of distortions, visual synonymy and visual polysemy, are still crucial issues in robust evaluation. A more thorough and comprehensive strategy of robust evaluation will be an important step toward an objective and a practical evaluation of image retrieval systems.

REFERENCES

[1] Alzina, M., Szpankowski, W., and Grama, A., "2D-pattern matching image and video compression: Theory, algorithms, and experiments," *IEEE Trans. Image Processing*, vol. 11, pp. 318–331, Mar. 2002.

[2] Amir, A., Landau, G. M., and Sokol, D., "Inplace 2D matching in compressed image," *Journal of Algorithms*, vol. 49, pp. 240–261, Nov. 2003.

[3] Atallah, M., Genin, Y., and Szpankowski, W., "Pattern matching image compression: Algorithmic and empirical results," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 614–627, July 1999.

[4] Bellegards, J. R., Butzberger, J. W., Chow, Y.-L., Coccaro, N. B., and Naik, D., "A novel word clustering algorithm based on latent semantic analysis," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, (Atlanta, GA), pp. 172–175, May 1996.

[5] Blei, D., Bn, A., and Jordan, M., "Latent dirichlet allocation," *J. Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.

[6] Brittain, N. J. and El-Sakka, M. R., "Grayscale two-dimensional Lempel-Ziv encoding," in *Image Analysis and Recognition, 2nd ICIAR 2005*, (Toronto, Canada), pp. 328–334, Sept. 2005.

[7] Brooks, A. C. and Pappas, T. N., "Structural similarity quality metrics in a coding context: Exploring the space of realistic distortions," in *Proc. SPIE, Human Vision, Electronic Imaging XI* (Rogowitz, B. E., Pappas, T. N., and Daly, S. J., eds.), vol. 6057, (San Jose, CA), pp. 299–310, Jan. 2006.

[8] Brooks, A. C., Zhao, X., and Pappas, T. N., "Structural similarity quality metrics in a coding context: Exploring the space of realistic distortions," *IEEE Trans. Image Processing*, vol. 17, pp. 1261–1273, Aug. 2008.

[9] Buntine, W., "Variational extensions to EM and multinomial PCA," in *Proc. European Conf. Machine Learning*, (Helsinki, Finland), pp. 23–34, Aug. 2002.

[10] Carneiro, G., Chan, A. B., Moreno, P. J., and Vasconcelos, N., "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, pp. 394–410, Mar. 2007.

[11] Casstelli, V. and Bergman, L. D., "Digital imagery: Fundamentals," in *Image Databases* (Casstelli, V. and Bergman, L. D., eds.), pp. 1–10, New York, 2002.

[12] Chan, C. and Vetterli, M., “Lossy compression of individual signals based on string matching and one pass codebook design,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2491–2494, May 1995.

[13] Chou, C.-H. and Li, Y.-C., “A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 467–476, Dec. 1995.

[14] Cover, T. M. and Thomas, J. A., *Elements of Information Theory.* New York: Wiley, 1991.

[15] Daptardar, A. H. and Storer, J. A., “Reduced complexity content-based image retrieval using vector quantization,” in *Proceedings DCC 2006. Data Compression Conference*, pp. 342–351, Mar. 2006.

[16] Datta, R., Ge, W., Li, J., and Wang, J. Z., “Toward bridging the annotation-retrieval gap in image search,” *IEEE Trans. Multimedia*, vol. 14, pp. 24–35, July-Sept. 2007.

[17] Datta, R., Joshi, D., Li, J., and Wang, J. Z., “Image retrieval: Ideas, influences, and thrends of the new age,” *ACM Computing Surveys*, vol. 40, pp. 5:1–5:60, Apr. 2008.

[18] de Carvalho, M. B. and da Silva, E. A. B., “A universal multi-dimensional lossy compression algorithm,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 767–771, Oct. 1999.

[19] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Hershman, R., “Indexing by latent semantic analysis,” *J. Am. Soc. Inf. Sci.*, vol. 41, pp. 391–407, Sept. 1990.

[20] Dempster, A., Laird, N., and Rubin, D., “Maximum likelihod from incomplete data via the EM algorithm,” *J. Roy. Statist. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[21] Deselaers, T., *Features for Image Retrival.* PhD thesis, Aachen University of Technology, December 2003.

[22] Eckert, M. P. and Bradley, A. P., “Perceptual quality metrics applied to still image compression,” *Signal Processing*, vol. 70, pp. 177–200, Nov. 1998.

[23] Fei-Fei, L. and Perona, P., “A bayesian hierarchical model for learning natural scene categories,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 2, (San Diego, CA), pp. 524–531, June 2005.

[24] Finamore, W. A. and Leister, M. D. A., “Lossy Lempel-Ziv algorithm for large alphabet sources and applications to image compression,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 225–228, Sept. 1996.

[25] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P., "Query by image and video content: the QBIC system," *IEEE Trans. Comput.*, vol. 28, pp. 23–32, Sept. 1995.

[26] Frigui, H. and Caudill, J., "Unsupervised image segmentation and annotation for content-based image retrieval," in *Proc. IEEE. Conf. on Fuzzy Systems*, (Vancouver, Canada), pp. 72–77, July 2006.

[27] Gersho, A. and Gray, R. M., *Vector Quantization and Signal Compression*. Boston: Kluwer Academic Publishers, 1992.

[28] Golub, G. H. and Loan, C. F. V., *Matrix Computations*. The Johns Hopkins University Press, third ed., 1996.

[29] Grigorova, A., Natale, F. G. B. D., Dagli, C., and Huang, T. S., "Content-based image retrieval by feature adaptation and relevance feedback," *IEEE Trans. Multimedia*, vol. 9, pp. 1183–11192, Oct. 2007.

[30] Hofmann, T., "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, pp. 177–196, Jan. 2001.

[31] Huffman, D. A., "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, pp. 1098–1101, 1952.

[32] Idris, F. and Panchanathan, S., "Image and video indexing using vector quantization," *Mach. Vision. Appl.*, vol. 10, pp. 43–50, Nov. 1997.

[33] ISO/IEC 10918-1 and ITU-T Recommendation T.81, *Information Technology-JPEG-Digital Compression and Coding of Continuous-Tone Still Image-Part I: Requirements and Guidelines*, 1994.

[34] ISO/IEC 15444-1, *Information Technology-JPEG 2000-Image Coding System-Part I:Core Coding System*, 2000.

[35] ITU-R, Rec. BT.601-4, *Encoding parameters of digital television for studios.*

[36] ITU-T and ISO/IEC JTV 1, Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4) AVC, *Advanced video coding for generic audiovisual services*, 2003.

[37] Jeong, S., Won, C. S., and Gray, R. M., "Image retrieval using color histograms generated by Gauss mixture vector quantization," *Comput. Vis. Image Underst.*, vol. 94, pp. 44–66, Apr-Jun. 2004.

[38] Keys, R. G., "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 29, pp. 1153–1160, Dec. 1981.

[39] Kherfi, M. L. and Ziou, D., "Image collection organization and its application to indexing, browsing, summerization, and semantic retrieval," *IEEE Trans. Multimedia*, vol. 9, pp. 893–900, June 2007.

[40] Kieffer, J., "A survey of the theory of source coding with a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. IT-39, pp. 1473–1490, Sept. 1993.

[41] Ko, B. and Byun, H., "FRIP: A region-based image retrieval tool using automatic image segmentation and stepwise boolean and matching," *IEEE Trans. Multimedia*, vol. 7, pp. 105–113, Feb. 2005.

[42] Kontoyiannis, I., "Second-order analysis of lossless and lossy versions of Lempel-Ziv codes," in *31st Asilomar Conf. Signals, Systems and Computers*, vol. 2, pp. 1349–1353, Nov. 1997.

[43] Kontoyiannis, I., "An implementable lossy version of the Lempel-Ziv algorithm-Part I: Optimality for memoryless sources," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2293–2305, Nov. 1999.

[44] Kullback, S. and Leibler, R. A., "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, pp. 79–87, 1951.

[45] Lacoste, C., Lim, J.-H., Chevallet, J.-P., and Le, D. T. H., "Medical-image retrieval based on knowledge-assisted text and image indexing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, pp. 889–900, July 2007.

[46] Lempel, A. and Ziv, J., "On the complexity of finite sequences," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 75–81, Jan. 1976.

[47] Lempel, A. and Ziv, J., "Compression of two-dimensional data," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 2–8, Jan. 1986.

[48] Li, J., Wang, J. Z., and Wiederhold, G., "IRM: Integrated region matching for image retrieval," in *Proc. 8th ACM Int. Conf. Multimedia*, pp. 147–156, 2000.

[49] Lin, J., "Divergence measures based on the Shannon entropy," *IEEE Trans. Inform. Theory*, vol. 37, pp. 145–151, Jan. 1951.

[50] Linde, Y., Buzo, A., and Gray, R. M., "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95, Jan. 1980.

[51] Łuczak, T. and Szpankowski, W., "A suboptimal lossy data compression based on approximate pattern matching," *IEEE Trans. Inform. Theory*, vol. IT-43, pp. 1439–1451, Sept. 1997.

[52] Manning, C. D. and Schütze, H., *Foundations of Statistical Natural Language Processing.* Cambridge, Massachusetts: The MIT Press, 1999.

[53] Merriam-Webster, *The Merriam-Webster Dictionary.* Merriam-Webster, eleventh edition ed., July 2004.

[54] Miller, V. S. and Wegman, M. N., "Variations on a theme by Ziv and Lempel," in *Combinatorial Algorithms on Words* (Apostolico, A. and Galil, Z., eds.), pp. 131–140, Berlin, Germany: Springer-Verlag, 1985.

[55] Minka, T. P. and Picard, R. W., “Interactive learning using a society of models,” *Pattern Recognition*, vol. 30, pp. 565–581, Apr. 1997.

[56] Monay, F. and Gatica-Perez, D., “Modeling semantic aspects for cross-media image indexing,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, pp. 1802–1817, Oct. 2007.

[57] Morita, H. and Kobayashi, K., “An extension of LZW coding algorithm to source coding subject to a fidelity criterion,” in *4th Joint Swedish-Soviet International Workshop Information Theory*, (Gotland, Sweden), pp. 105–109, 1989.

[58] Müller, H., Marchand-Maillet, S., and Pun, T., “The truth about Corel - evaluation in image retrieval,” in *Proc. Int. Conf. Image and Video Retrieval (CIVR)*, (London, UK), pp. 38–49, July 2002.

[59] Naphade, M. R. and Huang, T. S., “Extracting semantics from audiovisual content: The final frontier in multimedia retrieval,” *IEEE Trans. Neural Networks*, vol. 13, pp. 793–810, July 2002.

[60] Natsev, A., Rastogi, R., and Shim, K., “WALRUS: A similarity retrieval algorithm for image databases,” *IEEE Trans. Knowledge Data Eng.*, vol. 15, pp. 1–16, Sept./Oct. 2003.

[61] Pappas, T. N., Chen, J., and Depalov, D., “Perceptually based techniques for image segmentation and semantic classification,” *IEEE Commun. Mag.*, vol. 45, pp. 44–51, Jan. 2007.

[62] Pappas, T. N., Safranek, R. J., and Chen, J., “Perceptual criteria for image quality evaluation,” in *Handbook of Image and Video Processing* (Bovik, A. C., ed.), pp. 939–959, Academic Press, second ed., 2005.

[63] Pennebaker, W. B. and Mitchell, J. L., *JPEG Still Image Data Compression Standard.* New York: Van Nostrand, 1992.

[64] Pentland, A., Picard, R. W., and Sclaroff, S., “Photobook: Tools for content-based manipulation of image databases,” *Int. J. Comput. Vis.*, vol. 2185, pp. 233–254, June 1996.

[65] Pigeon, S., “An optimizing lossy generalization of LZW,” in *Proceedings DCC 2001. Data Compression Conference*, p. 509, Mar. 2001.

[66] Qiu, G., “Color image indexing using BTC,” *IEEE Trans. Image Processing*, vol. 12, pp. 93–101, Jan. 2003.

[67] Rissanen, J., “Generalized Kraft inequality and arithmetic coding,” *IBM J. Res. Devel.*, vol. 20, pp. 198–203, 1976.

[68] Rissanen, J., “A universal data compression system,” *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 656–664, Sept. 1983.

[69] Rui, Y., Huang, T. S., and Chang, S.-F., "Image retrieval: Current techniques, promising directions, and open issues," *J. Vis. Commun. Image Represent.*, vol. 10, pp. 39–62, Mar. 1999.

[70] Rui, Y., Huang, T. S., Ortega, M., and Mehrotra, S., "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 644–655, Sept. 1998.

[71] Salton, G., Wong, A., and Yang, C. S., "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, pp. 613–620, Nov. 1975.

[72] Shannon, C. E., "A mathematical theory of communication," *Bell System Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.

[73] Sheinwald, D., Lempel, A., and Ziv, J., "Two-dimensional encoding by finite-state encoders," *IEEE Trans. Commun.*, vol. 38, pp. 341–347, Mar. 1990.

[74] Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R., "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 1349–1380, Dec. 2000.

[75] Smith, J. R. and Chang, S.-F., "VisualSEEk: A fully automated content-based image query system," in *Proc. ACM Multimedia*, (Boston, MA), pp. 87–98, Nov. 1996.

[76] Steinberg, Y. and Gutman, M., "An algorithm for source coding subject to a fidelity criterion, based on string matching," *IEEE Trans. Inform. Theory*, vol. IT-39, pp. 877–886, May 1993.

[77] Teng, S. W. and Lu, G., "Image indexing and retrieval based on vector quantization," *Pattern Recognition*, vol. 40, pp. 3299–3316, Nov. 2007.

[78] Theoharatos, C., Economou, G., Fotopoulos, S., and Laskaris, N. A., "Color-based image retrieval using vector quantization and multivariate graph matching," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, (Genova, Italy), pp. 537–530, Sept. 2005.

[79] Vailaya, A., Figueiredo, M. A. T., Jain, A. K., and Zhang, H.-J., "Image clssification for content-based indexing," *IEEE Trans. Image Processing*, vol. 10, pp. 117–130, Jan. 2001.

[80] Veltkamp, R. C. and Tanase, M., "Content-based image retrieval systems: A survey," Technical Report UU-CS-2000-34, Utrecht University, Oct. 2000.

[81] Wang, J. Z., Li, J., and Wiederhold, G., "SIMPLIcity: Semantics-sensitive integrated matching for picture libraries," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, pp. 947–963, Sept. 2001.

[82] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: From error visiblity to structural similarity," *IEEE Trans. Image Processing*, vol. 13, pp. 600–612, Apr. 2004.

[83] Wang, Z., Lu, L., and Bovik, A. C., "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, vol. 19, pp. 121–132, Feb. 2004.

[84] Wang, Z. and Simoncelli, E. P., "Translation insensitive image similarity in complex wavelet domain," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, (Philadelphia, PA), pp. 573–576, 2005.

[85] Wang, Z., Bovik, A. C., and Simoncelli, E. P., "Structural approaches to image quality assesment," in *Handbook of Image and Video Processing* (Bovik, A. C., ed.), pp. 961–974, Academic Press, second ed., 2005.

[86] Welch, T. A., "A technique for high-performance data compression," *IEEE Computer*, vol. 17, pp. 8–19, June 1984.

[87] Yang, E. H. and Kieffer, J., "Simple universal lossy data compression schemes derived from Lempel-Ziv algorithm," *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 239–245, Jan. 1996.

[88] Yang, E. H. and Kieffer, J., "On the performance of data compression algorithms based upon string matching," *IEEE Trans. Inform. Theory*, vol. IT-44, pp. 47–65, Jan. 1998.

[89] Yang, X. K., Ling, W. S., Lu, Z. K., Ong, E. P., and Yao, S. S., "Just noticeable distortion model and its applications in video coding," *Signal Process. Image Commun.*, vol. 20, pp. 662–680, Aug. 2005.

[90] Yeh, C.-H. and Kuo, C.-J., "Content-based image retrieval through compressed indices based on vector quantized images," *Optical Engineering*, vol. 45, pp. 43–50, Jan. 2006.

[91] Zhang, R. and Zhang, Z., "Effective image retrieval based on hidden concepts discovery in image database," *IEEE Trans. Image Processing*, vol. 16, pp. 562–572, Feb. 2007.

[92] Zhang, Z. and Wei, V., "An on-line universal lossy data compression algorithm via continuous codebook refinement. I. Basic results," *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 803–821, May. 1996.

[93] Zhao, X., Reyes, M. G., Pappas, T. N., and Neuhoff, D. L., "Structural texture similarity metrics for retrieval applications," in *Proc. IEEE Int. Conf. Image Processing*, (San Diego, CA), Oct. 2008.

[94] Zhou, X. S. and Huang, T. S., "Relevance feedback in image retrieval: A comprehensive review," *Multimedis Syst.*, vol. 8, no. 6, pp. 536–544, 2003.

[95] Ziv, J., "Coding theorem for individual sequences," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 405–412, July. 1978.

[96] Ziv, J. and Lempel, A., "A universal algorithm for sequential data compression," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 337–343, May. 1977.

[97] Ziv, J. and Lempel, A., "Compression of individual sequences via variable rate coding," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 530–536, Sept. 1978.

VITA

Soo Hyun Bae received the B.S. degree in electronics engineering from Yonsei University, Korea, in 1999, and worked for VirtualTek Corporation as a software development engineer before beginning graduate school at the Georgia Institute of Technology in the fall of 2003. He received the M.S. and the Ph.D. degree in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta, GA, under the supervision of Dr. Biing-Hwang Juang in 2005 and 2008, respectively. He spent the summer of 2004 as a research intern at Samsung Electronics, the summer of 2005 as a visiting research assistant in the Image and Video Processing Laboratory of Northwestern University, Evanston, IL, and the summer of 2007 as a research associate at NTT DoCoMo Communications Labs USA. His research interests include information retrieval, data compression, and perceptual model for image processing.