Federation University of Australia
School of Science, Engineering and Information Technology

A New Perceptual Dissimilarity Measure for Image Retrieval and Clustering

Hamid Shojanazeri

Thesis submitted in partial fulfillment of the requirements
for the Doctor of Philosophy of Sciences degree

Under the supervision of
Prof. Guojun LU
Assoc. Prof. Shyh Wei TENG
Dr. Dengsheng ZHANG
Prof. Kai Ming TING

December 2018

Abstract

Image retrieval and clustering are two important tools for analysing and organising images. Dissimilarity measure is central to both image retrieval and clustering. The performance of image retrieval and clustering algorithms depends on the effectiveness of the dissimilarity measure.

'Minkowski' distance, or more specifically, 'Euclidean' distance, is the most widely used dissimilarity measure in image retrieval and clustering. Euclidean distance depends only on the geometric position of two data instances in the feature space and completely ignores the data distribution. However, data distribution has an effect on human perception. The argument that two data instances in a dense area are more perceptually dissimilar than the same two instances in a sparser area, is proposed by psychologists. Based on this idea, a dissimilarity measure called, 'm_p', has been proposed to address Euclidean distance's limitation of ignoring the data distribution.

Here, m_p relies on data distribution to calculate the dissimilarity between two instances. As prescribed in m_p, higher data mass between two data instances implies higher dissimilarity, and vice versa. m_p relies only on data distribution and completely ignores the geometric distance in its calculations.

In the aggregation of dissimilarities between two instances over all the dimensions in feature space, both Euclidean distance and m_p give same priority to all the dimensions. This may result in a situation that the final dissimilarity between two data instances is determined by a few dimensions of feature vectors with relatively much higher values. As a result, the dissimilarity derived may not align well with human perception.

The need to address the limitations of Minkowski distance measures, along with the importance of a dissimilarity measure that considers both geometric distance and the perceptual effect of data distribution in measuring dissimilarity between images motivated this thesis. It studies the perfor-

mance of m_p for image retrieval. It investigates a new dissimilarity measure that combines both Euclidean distance and data distribution. In addition to these, it studies the performance of such a dissimilarity measure for image retrieval and clustering.

Our performance study of m_p for image retrieval shows that relying only on data distribution to measure the dissimilarity results in some situations, where the m_p's measurement is contrary to human perception. This thesis introduces a new dissimilarity measure called, *perceptual dissimilarity measure* (PDM). PDM considers the perceptual effect of data distribution in combination with Euclidean distance. PDM has two variants, PDM1 and PDM2. PDM1 focuses on improving m_p by weighting it using Euclidean distance in situations where m_p may not retrieve accurate results. PDM2 considers the effect of data distribution on the perceived dissimilarity measured by Euclidean distance. PDM2 proposes a weighting system for Euclidean distance using a logarithmic transform of data mass.

The proposed PDM variants have been used as alternatives to Euclidean distance and m_p to improve the accuracy in image retrieval. Our results show that PDM2 has consistently performed the best, compared to Euclidean distance, m_p and PDM1. PDM1's performance was not consistent, although it has performed better than m_p in all the experiments, but it could not outperform Euclidean distance in some cases.

Following the promising results of PDM2 in image retrieval, we have studied its performance for image clustering. k-means is the most widely used clustering algorithm in scientific and industrial applications. k-medoids is the closest clustering algorithm to k-means. Unlike k-means which works only with Euclidean distance, k-medoids gives the option to choose the arbitrary dissimilarity measure. We have used Euclidean distance, m_p and PDM2 as the dissimilarity measure in k-medoids and compared the results with k-means. Our clustering results show that PDM2 has perfromed overally the best. This confirms our retrieval results and identifies PDM2 as a suitable dissimilarity measure for image retrieval and clustering.

Publications on Ph.D. Research

Hamid Shojanazeri, Shyh Wei Teng, Dengsheng Zhang, Guojun Lu, A Hybrid Data Dependent Dissimilarity Measure for Image Retrieval, in Digital Image Computing: Techniques and Applications (DICTA), 2017.

Hamid Shojanazeri, Shyh Wei Teng, Dengsheng Zhang, Sunil Aryal, Guojun Lu, A Novel Perceptual Dissimilarity Measure for Image Retrieval, in Image and Vision Computing New Zealand (IVCNZ), 2018.

Hamid Shojanazeri, Sunil Aryal, Shyh Wei Teng, Dengsheng Zhang, Guojun Lu, Image clustering using a similarity measure incorporating human perception, in Image and Vision Computing New Zealand (IVCNZ), 2018.

Acknowledgements

I would like to express my deepest appreciation to my advisors, Professor Guojun Lu and Associate Professor Shyh Wei Teng not only for their academic guidance but also for all the kind support they have given me during these years. Having the chance to work with my astute advisers, helped me a lot to develop a better vision on problems and proper way of addressing them. I found Professor Guojun Lu as a kind hearted father figure who has an extensive knowledge and expertise in his area coupled with extremely professional manner. Associate Professor Shyh Wei Teng has gone beyond the role of an academic adviser in all his very kind dedication and effort in all the stages of this work.

I can not thank my lovely wife enough who was extremely supportive during all these years, without her encouragement, sacrifices and patience this PhD definitely would not have happened. A special thanks must go to her family who really motivated, encouraged and supported us throughout this period as well.

I am also very grateful to Doctor Sunil Aryal who was a great support for me in enhancing my knowledge about this work throughout these years.

A special thanks goes to my associate advisers: Doctor Dengsheng Zhang and Professor Kai Ming Ting for their very kind support and guidance throughout this work.

A very special gratitude goes to all in the School of Science, Engineering and Information Technology of Federation University Australia for helping and providing the funding for this work.

Hamid Shojanazeri was supported by an Australian Government Research Training Program (RTP) Fee-Offset Scholarship through Federation University Australia.

This research was partially supported by Australian Research Council Discovery Projects scheme: DP130100024

Contents

List of Figures

List of Tables

1 Introduction

In this information age, a tremendous amount of data is collected everyday from different sources and stored in databases. This data has no value unless they are effectively analysed to extract useful information and knowledge. Information retrieval and clustering are two essential tools for letting every organisation have the ability to analyse the data collected. Information retrieval is the activity of obtaining information from a given database that is relevant to the information needs of a user. Clustering is the grouping of data instances into multiple sets, such that the instances in the same set are more similar to each other than to those in other sets.

Images are an important part of available information and are becoming an integral part of human communications. Organisations use them as a part of their daily activities in various sectors including business, medicine, education, and entertainment. The creation, storage, manipulation, and transmission of images have become less costly and more efficient. People can even capture, store, transmit and print images using their mobile telephones. Consequently, the quantity of images and their users are growing rapidly. For example, Google, Facebook and Instagram deal with millions of images. This presents a big challenge for those who design and implement retrieval and clustering systems.

Evaluation of dissimilarity between two instances has a key role in many different techniques and algorithms introduced for information retrieval and clustering. For example, in content-based information retrieval, the task is to rank instances in a given database based on their similarities to a given query instance [6]. Whereas in clustering, the task is to group data into clusters based on their similarity. One of the main challenges relating to the indexing or grouping of similar images, is the measurement of their dissimilarity for organisation and retrieval purposes.

Similarity measure is a numerical value to show how alike two objects are and often falls between 0 (no similarity) to 1 (complete similarity). On

the other hand, a dissimilarity measure is a numerical value that shows how different two objects are, and can have the value from 0 (objects are similar) to ∞ (objects are different).

1.1 Dissimilarity measures commonly used in image retrieval and clustering

In datasets, an entity is represented as a data instance defined by a fixed number of selected features or properties. Let M be a collection of N data instances where each instance x is represented as d-dimensional feature vector, $x = \{x_1, x_2, ..., x_d\}$, and x_i be the i^{th} dimension of the vector. Let $D(x, y)$ be the measure of dissimilarity of two instances x and y.

The analysis of image dissimilarity in many areas has been dominated by geometric models [6, 117]. These models represent objects as points in certain coordinate space, such that the observed dissimilarity between two data instances, $D(x, y)$, correspond to distances between the respective points. The higher the distance between x and y, the greater dissimilarity between them. Minkowski distance (also known as l_p where $p > 0$) is the most widely- used distance measure [6].

The Minkowski distance of x and y is calculated by aggregating their geometric distances in every dimension. Euclidean distance (also known as $l_2 - norm$) is a popular choice of distance measure in image retrieval and [6, 135] clustering [110].

1.2 Motivation

Although Minkowski distance, specifically Euclidean distance, has performed well in many application they have limitations which we will discuss in the following section. To address these limitations a data dependent dissimilarity measure, m_p has been proposed [5]. Overcoming the limitations of Minkowski distance, along with the importance of a dissimilarity measure that consider both geometric distance and the perceptual effect of data distribution motivated this thesis.

1.2.1 Limitations of Minkowski distances

Minkowski distance (e.g. Euclidean distance) has two major limitations. It has been observed that the performance of geometric distance measures is not consistent in different datasets with varied data distributions [6]. There is an argument that the variation in performance of geometric distances may occur because they only rely on the geometric position of two instances in the feature space and completely ignore the data distribution. Psychologists have expressed their concerns about geometric distance in measuring the dissimilarity between two instances [6, 62, 117]. They have argued that data distribution has an influence on the judged similarity between two data instances. Karmasul [62] introduced a similarity model which suggests two instances in a relatively dense region would be less similar than the same two instances located in a less dense region. For example consider evaluating the similarity between two red apples in two different contexts, where the two red apples are among green apples as shown in Figure 1.1 and among red apples as shown in Figure 1.2.

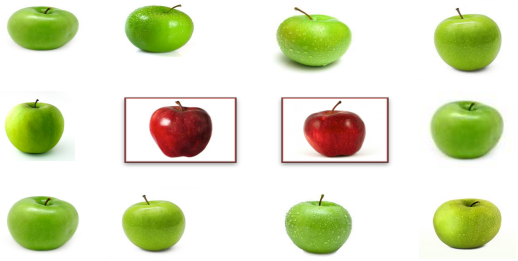


Figure 1.1: Two red apples in a group of green apples look more similar.

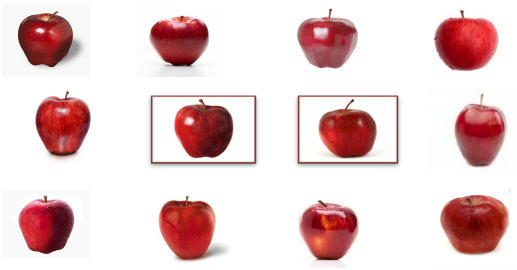


Figure 1.2: Two red apples in a group of other red apples look more dissimilar compared to Figure 1.1.

The two red apples among green apples are perceptually more similar than the same two red apples among other red apples. The same two red

apples that have the same geometric distance in a specified feature space among other red apples look perceptually more dissimilar only because there are more instances of the same kind in that context. This highlights the important role of data distribution in evaluation of dissimilarity.

The other limitation with the Minkowski distance measures is that it gives the equal weight to all dimensions when combining distances. This may result in a situation where the dissimilarity between two instances is determined by a few dominant dimensions. There is a tendency that the small number of the dimensions of feature vectors with relatively much higher values dominate the overall dissimilarity value. As a result, the dissimilarity derived might not align well with how humans perceived dissimilarity.

1.2.2 Mass-based dissimilarity

To address the discussed limitation of Minkowski distance measures with ignoring the data distribution, a data dependent dissimilarity measure has been proposed [5]. This measure is called m_p and it focuses on the data distribution of the dataset instead of simply measuring the geometric distance between two instances. The measure, m_p is developed based on a dissimilarity model proposed by Krumhausl [62] and a psychological argument suggesting that two instances in a sparse region are perceptually more similar than in a dense region. In this measure, the dissimilarity between two instances, $\mathbf{x}$ and $\mathbf{y}$, is measured by considering data distribution.

1.3 Thesis aims

Motivated by the need to address the limitations of Minkowski distance measures, along with the importance of a dissimilarity measure that considers both geometric distance and the perceptual effect of data distribution in evaluating the similarity between images, this thesis aims to:

- Investigate suitability of m_p dissimilarity measure for image retrieval.
- Develop a new dissimilarity measure that incorporates the geometric distance and data distribution between two images in the feature space.
- Study the performance of the proposed dissimilarity measure for image retrieval.

- Study the performance of the proposed dissimilarity measure for clustering of images.

1.4 Thesis Contribution

This thesis makes the following main contributions:

1. It studies the suitability of using m_p as a dissimilarity measure for image retrieval.

2. It introduces a new dissimilarity measure called Perceptual Dissimilarity Measure (PDM), for images. The proposed measure determines the dissimilarity between two images in the feature space based on Euclidean distance in a specified feature space and the density of data in the region covering the two instances. Two variants of the new dissimilarity measure has been proposed: $PDM1$ and $PDM2$. The two variants of the proposed dissimilarity measure have the following characteristics:

   - They incorporate the perceptual effect of data distribution on the human similarity judgment with geometric distance.
   - The proposed weighting system in both variants moderates the potential undesired effect of a few dimensions with much higher values in calculating the total dissimilarity with Euclidean distance and m_p.

3. It studies the effect of considering region density along with geometric distance in measuring the dissimilarity between images in image retrieval. It analyses the relationship of the proposed perceptual dissimilarity measure with Euclidean distance, m_p and compares their performances.

4. It studies the effect of considering the perceptual effect of region density along with geometric distance in clustering of images. It analyses the relationship of the proposed perceptual dissimilarity measure with Euclidean distance and m_p in clustering images. It compares the performances of PDM with Euclidean distance and m_p in K-medoids. It also compares their results with k-means that uses Euclidean distance as the dissimilarity measure.

The results of our first contribution and PDM1 stated in the second contribution are published in:

- Hamid Shojanazeri, Shyh Wei Teng, Dengsheng Zhang, Guojun Lu, A Hybrid Data Dependent Dissimilarity Measure for Image Retrieval, in Digital Image Computing: Techniques and Applications (DICTA), 2017.

The results of the PDM2 stated in the second contribution, along with third contribution are published in:

- Hamid Shojanazeri, Shyh Wei Teng, Dengsheng Zhang, Sunil Aryal, guojun Lu, A Novel Perceptual Dissimilarity Measure for Image Retrieval, in Image and Vision Computing New Zealand (IVCNZ), 2018.

The results of our fourth contribution are also published in:

- Hamid Shojanazeri, Sunil Aryal, Shyh Wei Teng, Dengsheng Zhang, guojun Lu, Image clustering using a similarity measure incorporating human perception, in Image and Vision Computing New Zealand (IVCNZ), 2018.

2 Related Works

This chapter presents a critical review of the literature relevant to the research presented in this thesis. This includes image retrieval, clustering and dissimilarity measures. The focus of this thesis is on developing a new dissimilarity measure for image retrieval and clustering. Images are represented using extracted features. Feature extraction is a common component in image retrieval and clustering. More specifically, this chapter presents the image retrieval framework and methods proposed in this area along with image feature extraction methods. Also, we review the clustering methods proposed in the literature and the common dissimilarity measures used in image retrieval and clustering.

2.1 Image retrieval

The development of the Internet, different image capturing devices, such as digital cameras, smart phones and cheaper storages, has led to a rapid increase in the number of available images and users. Efficient image retrieval tools are required in various domains, including remote sensing, education, fashion, crime prevention and medicine. As a result, many general purpose image retrieval methods have been developed that can be categorised into two groups: text-based and content-based [72].

The text-based method was first introduced in 1970s. In this method, the images are manually annotated by text descriptors, which are then used by a *database management system* (DBMS) to perform image retrieval. There are two main limitations with this method. First, a considerable level of human labour is required for manual annotation. Second, is the annotation inaccuracy which is due to the subjectivity of human perception [32, 72, 102]. To overcome the above limitations in a text-based image retrieval system, content-based image retrieval (CBIR) was introduced during the early 1980s . In CBIR, images are represented by their visual con-

tents, such as colour, texture and shapes. They are then ranked based on their similarity to the query [72]. Therefore effective feature extraction methods and dissimilarity measures are the essential components of this method. Some of the early stage commercial and prototype developments in CBIR systems are QBIC [15] , Photobook [95], Virage [46], Netra [78] and SIMPLIcity [121]. In recent years ,image retrieval has become a part of Google search engine and many other softwares, which have been developed to manage the photo libraries based on their contents.

Humans tend to use high-level features (concepts), such as keywords and text descriptors, to interpret images and measure their similarity. The features automatically extracted using computer vision methods are mostly low-level features e.g. colour, texture, shape and spatial layout [72]. There is a gap between the extracted low-level features and high-level concepts used by humans. The gap between the high-level concepts in a user's mind and, low-level features extracted from images, effect the performance of CBIR systems [142].

Three types of query types have been mainly discussed in CBIR in [32, 72]. Type 1: Retrieval based on low-level features such as colour, texture and shape. A query from this type is query by example, 'find pictures like this'. Type 2: Objects retrieval of a given type identified by extracted features. The example of this query type is, 'searching an object inside an image such as a human in an image of a beach'. Type 3: Retrieval by concept presented in the image, this involves a significant amount of high-level logic about the purpose of the objects or scenes in an image. The example of this type of query is 'named events', of pictures with emotional or religious significance. Considering the query as, 'searching pictures of a festival', this includes images of people with happy facial expressions in addition to many objects that can be related to a ceremony. Therefore, the link between different objects should conclude a concept. The gap between low-level features and high-level concepts are reflected in queries of Type 2 and 3 queries.

In Type 1 query retrieval users are usually required to submit an example image as a query. However, in a case where the user does not have an example image, semantic image retrieval, as in Type 2 and 3, is more convenient for the user. In Type 2 and 3, the user can use keywords as the query, unlike Type 1, which requires the user to have an example image.

The survey in [72] reviews the methods that aim to bridge the semantic gap between low-level features and high-level concepts and mainly categorises them into: using object ontology, using machine learning tools and

using relevance feedback (RF) in retrieval method.

The Type 3 query is complicated as it targets multiple objects and the link between them. The query of Type 2 usually involves image segmentation [72]. Image segmentation, partitions an image into non-overlapping regions, each of which is homogeneous in one or more features and maximal in terms of this homogeneity. Type 1 query involves using low-level features from entire image for retrieval purposes. There are two fundamental components in image retrieval systems: (1) image feature extraction, (2) dissimilarity measure. A schematic flow of an image retrieval framework is illustrated in Figure 2.1.

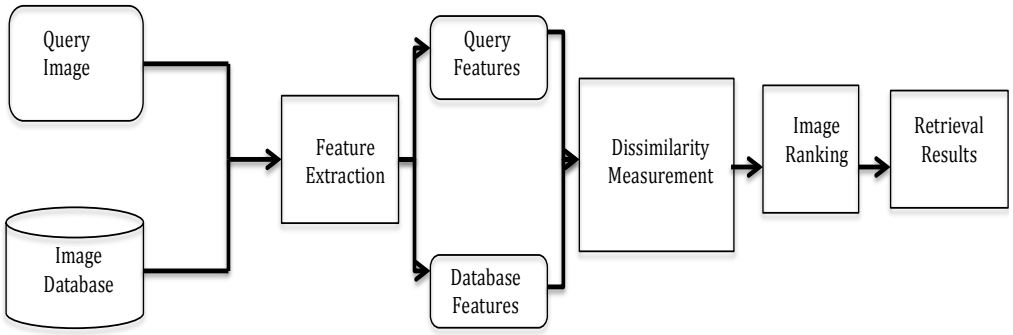


Figure 2.1: Image retrieval framework

In the following section we will review different components of an image retrieval framework with the focus on feature extraction and dissimilarity measurement.

2.2 Image Features

An image retrieval method is likely to be more effective if it can extract distinct visual information from images. Some of the methods for feature extraction aim to extract the visual properties from the whole image, while some other methods first detect distinct key areas (points), then extract information from the neighbourhood of detected areas. A robust representation of an image can be achieved using the features that are invariant to image transformations such as rotation, scale, translation and affine deformations. In the following, we will review various feature extraction methods.

2.2.1 Colour features

Colour is one of the most important features, and they are defined by colour spaces, such as RGB, LUV, HSV, HMMD [134]. A histogram is a helpful tool in colour image analysis [68]. The colour histogram describes the colour distribution of an image. It quantises a colour space into different bins and counts the number of pixels belonging to each colour bin. They are invariant to the rotation and translation of image content. Many researchers have used colour histograms and developed variations of it to represent the images [47, 54, 68, 93, 111]. Colour histograms do not capture the spatial information. Colour moments are one of the simplest features. They are used in many retrieval methods [34, 35, 42, 56, 98, 118, 128]. The common moments are mean, standard deviation and skewness. Usually they are calculated for each colour channel (component) separately, therefore resulting in nine features that form the feature vector. These features are useful when they are calculated for region or object.

The *colour coherence vector* (CCV) incorporates spatial information into the basic colour histogram. The *colour structure descriptor* (CSD) is also a histogram-based descriptor. The CSD histogram is created by moving a structuring element (e.g., square) throughout the image. Bin i of the histogram indicates how many times the structuring element contains at least one pixel with colour. If the window is of size 1 pixel, the CSD is an ordinary histogram. The *dominant colour descriptor* (DCD) is also a variation of histogram which selects a small number of colours from the highest bins of a histogram [34]. The number of colours (bins) selected as DCD depends on the threshold of bin height. MPEG-7 recommends that 1-8 colours are sufficient to represent a region. Unlike the traditional histogram, the selected colours in DCD are adapted to the region instead of being fixed in the colour space. Thus, the colour representation with DCD is more accurate and compact than the conventional histogram. However, the similarity or distance calculation of two DCDs needs many-to-many matching.

2.2.2 Texture features

Another important feature that is widely used in image retrieval (due to its discriminative power) is texture. Texture extraction methods have been broadly classified into spatial and spectral methods in [134]. In spatial method, texture features are extracted by computing the pixel statistics or finding the local pixel structures in the original image domain. The spatial texture feature extraction methods can be further classified as structural,

statistical and model based. Structural methods describe textures using a set of texture primitives (texon or texture elements) and their placement rules [99, 112]. Textons are organised into a string descriptor, and syntactical pattern recognition methods are used to find the similarity between two descriptors. Local binary pattern (LBP) is one of the powerful texture features in this category [123]; a special case of texture spectrum proposed in 1990 [90, 122].

Statistical texture features characterises texture as a measure of low-level statistics of grey level images. The common spatial domain for statistical features are moments [74, 99], Tamura texture features [50, 112, 130] and features derived from grey level co-occurrence matrix (GLCM) [74, 92]. Statistical features are compact and robust because they are derived from great support. However, they are not sufficient to describe the large variety of textures.

In model based methods, texture is interpreted using stochastic (random) or generative models. Model parameters characterise the underlying texture property of the image. Examples of popular texture models are; Markov random field (MRF) [17, 24, 71, 74, 81, 118], simultaneous auto-regressive (SAR) model, and fractal dimension (FD) [17, 24]. As these models involve optimisation, they are usually computationally expensive.

In spectral texture, feature extraction methods, an image is transformed into frequency domain and then the feature is calculated from the transformed image [67] . The common spectral methods include *Fourier transform* (FT), *discrete cosine transform* (DCT) [77], wavelet [34, 92], and Gabor filters [80, 112, 137, 140]. FT and DCT are very fast to compute but are not scale and rotation invariant. Wavelet is both efficient and robust, however it only captures horizontal and vertical features. Among them all, Gabor features are most robust because they captures image features in multi-orientations and multi-scale. Recently, researches on multi-resolution analysis have shown that curvelet features have significant advantages over Gabor features and wavelet features, because curvelet features are more effective in capturing curvilinear properties, like lines and edges [140].

2.2.3 Shape features

Shape is known to be an important cue for human to identify and recognise real world objects. Shape features have been employed for image retrieval in many applications. Shape feature extraction methods can be broadly classified into two major groups: contour-based and region-based

methods [136]. Contour-based methods calculate shape features only from the boundary of the shape, while region-based methods extract features from the entire region. Because contour-based methods use only a portion of the region, they are more sensitive to noise than region-based methods, as small changes in the shape significantly affect the shape contour. Therefore, image retrieval methods usually employ region-based shape features. A number of commonly used simple region shape descriptors are; area, moments, circularity and eccentricity. The area-based descriptor is used in a number of works [31, 82, 115, 128, 130]. Circularity and moments are used in [31, 128, 130]. Circularity measures the ratio of area to boundary. Eccentricity is the ratio of the length of the major axis to that of minor axis [82]. Individual simple shape descriptors are not robust. Therefore, they are normally combined to create a more effective shape descriptor.

In order to retrieve an image from a large database, descriptors need to be invariant to rotation, scale and translation. Image moments as region based shape features are introduced as good candidates that meet these criteria. However, geometric image moments suffer from information redundancy and high computational complexity. Zernike moments introduced by [113] using orthogonal basis function claimed to address the problem with information redundancy and computational complexity of geometric moments.

2.2.4 Local features

Local features have greatly improved image representation and recognition performance [73]. The purpose of local features is to provide a robust representation of the local structures that are essential to images and invariant to many image transformations, such as scaling, rotation, translation and affine deformation. Local features have to be stable, repeatable and distinctive, so that images can be matched through the measurements of local features. The extraction process of local features can be divided into four phases: key-points detection, region refinement around each key-point, region content normalisation and descriptor computation. Key-points detection can be classified into two main methods. One is dense sampling that divides an image into dense grids and set key-points on the crosses of the grids. It then extracts overlapped regions around key-points. This method aims to provide a full coverage of the whole image. The second method is sparse interest point detection. This method aims to detect a set of key-points that can be reliably localised under varying imaging conditions, viewpoint changes, object transformation and also in

the presence of noise [73].

There have been several key-points detection methods proposed in the past. The Hessian detector searches for image locations with strong derivatives in two orthogonal directions [10]. The popular Harris/Förstner detector was explicitly designed for geometric stability [41, 48]. A detector for blob-like features that searches for scale space extrema of a scale-normalised, Laplacian of Gaussian (LoG) proposed in [70]. It is shown in [76] that the scale-space Laplacian can be approximated by a difference of Gaussian (DoG). The Harris-Laplacian operator was proposed to increase the discriminative power compared to the Laplacian and DoG operators [83, 84]. The discussed methods up to here extract local features that are invariant to translation and scale changes. The other important challenge in many applications is to find features that can be reliably extracted under large viewpoint changes [73]. Both the Harris-Laplace and Hessian-Laplace detectors can be extended to yield affine covariant regions. A different method for finding affine covariant regions has been proposed by Matas et al, which starts from a segmentation perspective [9].

After key-points detection and scale invariant region refinement, the next step is normalisation for orientation invariance. This is typically done by rotating the region by the angle of its dominant orientation. Once regions of interest have been extracted, their content needs to be represented. The Scale Invariant Feature Transform (SIFT) introduced by [76] has been proven to have generally good performance with any kind of region detector. The SIFT descriptor encodes the image information in a localized set of gradient orientation histograms to achieve robustness to illumination variations and small positional shifts. For each (orientation-normalised) scale invariant region, image gradients are sampled into a regular grid, and are then entered into a larger 4×4 grid of local gradient orientation histograms. Computation efficiency became a concerned while the local feature detector and descriptor became more popular. The Speeded-Up Robust Features (SURF) method [9] has been designed as an efficient alternative to SIFT.

For specific object recognition, interest point detector in conjunction with patch descriptor is quite effective due to its repeatability and distinctiveness. While for generic object recognition, such a sparse set is often insufficient. Instead, dense sampling of local features is suggested for category level recognition due to its full coverage of an image [89]. A combination of dense sampling and key-points detection, to form dense interest points proposed in [116]. Dense interest points are repeatable and cover

the whole image.

The feature extraction using dense local descriptors is composed of the following three steps: (i) extraction of local image features (e.g.SIFT descriptors), (ii) feature quantisation to build visual dictionary, (iii) encoding of the quantised local features in an image descriptor (e.g. a histogram of the quantised local features).

The procedure of obtaining image level features from local descriptors is shown in Figure 2.2. To make use of local features for the encoding phase, it is required to partition the local descriptor space into informative regions. This step is called *feature quantisation*. These regions are called visual words and a collection of visual words is called a *visual vocabulary /dictionary*. Clustering is used for feature quantisation to produce the visual words. K-means is the most widely used clustering algorithm in this step for feature quantisation. The next step involves the encoding of quantised local features. The baseline method in for this phase is bag of words (BOW), introduced in [25, 66, 107]. It computes a histogram of visual words (quantised local features), and basically counts the frequency of the occurrence of each visual word in the image. The effectiveness of feature extraction has considerable effect on the performance of image retrieval.

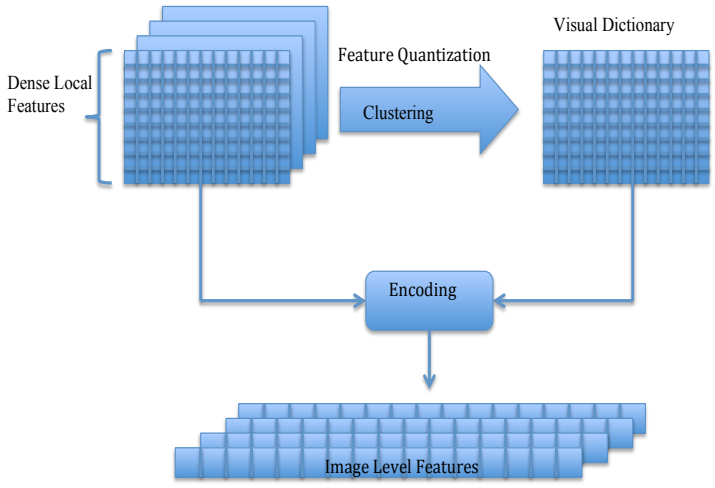


Figure 2.2: Dense local features extraction procedure

2.3 Dissimilarity measures in image retrieval

Once images are represented by their extracted features, a user can search for similar images in the database. Dissimilarity measurement is the second component in CBIR in which the query image is compared with other database images. Evaluation of similarity between the query image and the database images is performed by calculating the difference between the query feature vector and the database feature vectors by using dissimilarity measures. In the following sections we present two categories of dissimilarity measures: geometric distances and non-geometric dissimilarity measures.

2.3.1 Geometric distances

Geometric distances are dissimilarity measures which are based on geometric models. They mainly rely on the geometric position of data instances in the feature space and are the most widely used measures in image retrieval. In the following section we review some of the most commonly used ones in image retrieval. The comparative study in [135] evaluated the performance of six common dissimilarity measures in image retrieval. The measures compared in [135] are as follows.

Minkowski distance: Minkowski $(l_p - norm)$ distance is defined as:

$$l_p(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{2.1}$$

where if $p = 1$, l_1 is city block distance, and if $p = 2$, l_2 is Euclidean distance (ED).

Cosine distance: Cosine distance is a dissimilarity measure that measures the cosine of the angle between two vectors using the inner product of the vectors. This distance measure focuses on the direction of two vectors rather than the length of them and it is defined as:

$$d_{cos}(\mathbf{x}, \mathbf{y}) = cos\,(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^{d} x_i yi}{\sqrt{\sum_{i=1}^{d} x_i^2} \sqrt{\sum_{i=1}^{d} y_i^2}} \tag{2.2}$$

X^2 **Statistics**: Chi square or X^2 statistics is used to investigate if the distri-

bution of features is different from each other. It is defined as:

$$d_{x^2}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} \frac{x_i - E_i}{E_i} \tag{2.3}$$

where $E_i = \frac{x_i + yi}{2}$ [135].

Histogram intersection: The histogram intersection proposed by to measure the similarity between colour histograms of two images. It has the ability of partial matching where sizes of features vectors do not match. For example the size of feature vector of the query image is less than the size of the database image feature vectors [135]. It is defined as:

$$d_{hist}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d-1} \frac{min(x_i - y_i)}{|\mathbf{x}|} \tag{2.4}$$

Quadratic distance: Unlike the other mentioned distances that only consider similarity between each dimension, and do not make use of information across dimensions, quadratic distance is proposed to consider the similarity across dimensions.

$$d_{quad}(\mathbf{x}, \mathbf{y}) = \left[(\mathbf{x} - \mathbf{y})^t C (\mathbf{x} - \mathbf{y}) \right]^{\frac{1}{2}} \tag{2.5}$$

where $C = [c_{ij}]$ is $N \times N$ matrix of similarity coefficient between dimensions i and j. $C = [c_{ij}]$ is given by $c_{ij} = \frac{1 - d_{ij}}{d_{max}}$ where $d_{ij} = [x_i - y_j]$

Mahalanobis distance: Mahalanobis distance is special form of quadratic form distances where the correlation between features has been considered by using covariance matrix. It is defined as :

$$d_{mah} = \left[(\mathbf{x} - \mathbf{y}) \Sigma^{-1} (\mathbf{x} - \mathbf{y}) \right]^{\frac{1}{2}} \tag{2.6}$$

in a case if features x_i are statistically independent and have different variances it can be defined as:

$$d_{mah}(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{N-1} \frac{(x_i - y_i)^2}{\sigma_i^2} \tag{2.7}$$

which is the weighted l_2 distance. This distance gives more weight to dimension with smaller variance and gives less weight to dimension with larger variance [135].

The results of the experiments for image retrieval using shape features in [135] show that ED, along with city block and x^2 distance, had the best performance. In [79], authors compared the performance of sum of squared of absolute differences, sum of absolute difference, maximum value, Canberra, city block, Minkowski (p=3) and ED in image retrieval. In the following section we mention the definition of these distances that are not defined in the above distances.

Sum of absolute difference: The sum of absolute difference (SAD) is proposed by [100] distance metric and frequently used for as dissimilarity measure in CBIR [100]. This distance calculates the the sum of the differences of the absolute values of the two feature vectors. This distance metric is defined as:

$$d_{SAD}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} |x_i| - |y_i| \tag{2.8}$$

Sum of the squared absolute difference: This metric is a special case of SAD where the dissimilarity is calculated based on the sum of the squared differences of absolute values of the two feature vectors. This distance metric is defined as:

$$d_{SSAD}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} |x_i|^2 - |y_i|^2 \tag{2.9}$$

The squaring highlights the big differences.

Maximum distance: This maximum distance metric also known as Chebyshev distance, is used to get the largest value of the absolute differences of corresponding dimensions in feature vectors [101] and defined as:

$$d_{max}(\mathbf{x}, \mathbf{y}) = max(|x_i| - |y_i|) \tag{2.10}$$

The distance value is the maximum of the difference of the features of the pair of images, which shows the maximum dissimilarity of the two images.

Canberra: To normalise the large distance values from city block distance Canberra distance has been proposed which is the weighted City block L_1 distance. It is defined as:

$$d_{Canberra}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} \frac{|x_i - y_i|}{|x_i| + |y_i|} \tag{2.11}$$

The results of the study [79] on the mentioned distances show that ED performed the best and above all other distance measures in this study.

2.3.2 Non-geometric dissimilarity measures

The other category of dissimilarity measures are the non-geometric ones that do not rely on the geometric distance between of two data instances. These dissimilarity measures are not commonly used in image retrieval. In the following section we describe one of the recently developed measures that focuses on data distribution.

Mass-based dissimilarity : Recently, to address the limitation of Minkowski's distance with ignoring the data distribution, a mass-based dissimilarity has been proposed by [6]. This dissimilarity measure is called m_p and is based on the data distribution. The name of mass-based dissimilarity comes from the fact that m_p calculates the dissimilarity based on the data mass between two instances. The main idea behind this measure is the perceptual effect of data distribution as described in Chapter 1, Section 1.2.1. The number of data (data mass) in the region covering two instances has been used as a proxy of data distribution. In the next chapter, we will describe this measure in greater detail and study the suitability of this dissimilarity measure for image retrieval.

2.4 Clustering

Clustering is the unsupervised classification of data (observations, objects and feature vectors) into groups or clusters [53]. The objective of clustering is to group the data into clusters in such a way that maximises the similarity between data instances within a same cluster (intra-clusters) and minimises the similarity between data instances from different clusters (inter-clusters). The need to organise the huge amount of unstructured data into meaningful groups, categories, partitions, or classes in different domains has made clustering a valuable tool in data analysis [53, 58]. Clustering has been used to analyse and find hidden patterns in data for different applications such as pattern recognition, machine learning [20, 36, 37], statistics, biology, sociology [7, 85, 96, 104] and information retrieval [21, 30, 58, 94]. More specifically, it has been used for image clustering [86, 106], image retrieval [19, 103, 132] and image segmentation [4, 45, 57, 63, 87]. Clustering procedure consists of feature extraction, definition of dissimilarity

measure and clustering or grouping [53]. The image feature extraction has been discussed in Section 2.2. Later in this chapter, the dissimilarity measures used in clustering methods will be reviewed. First in the following section, we will provide an overview of different clustering methods.

Before the overview of clustering methods, we provide the notations and definitions as follows. A data instance is represented by a feature vector as $\mathbf{X}_i = \{x_{i1}, x_{i2}, ..., x_{id}\}$ where d is the dimension of the feature space. A C_j is a group of data instances assigned to a cluster and expressed by : $C_j = \left\{\mathbf{X}_1^j, \mathbf{X}_2^j, ..., \mathbf{X}_{nj}^j\right\}$, where nj is the number of data instances in cluster C_j.

2.4.1 Clustering methods

Clustering output can be classified as hard or soft (fuzzy) assignment. In hard clustering, each data instance belongs only to one cluster, while in fuzzy, each data instance belongs to each cluster with a certain probability [53]. Clustering methods are mainly classified into two main categories, "Hierarchical" and "Partitioning" algorithms [52, 53]. In hierarchical clustering the output is a nested series of partitions and it has two main categories: agglomerative or divisive. In agglomerative methods, the algorithm considers each data instance as a distinct (singletone) cluster and successively merges them based on a stopping criterion. Divisive algorithms, on the contrary, consider the entire dataset as a single cluster and start splitting it into other clusters until meeting a stopping criterion. Partitioning algorithms, partition the dataset into a number of clusters and optimise (usually locally) a clustering criterion [52].

2.4.1.1 Hierarchical clustering

Hierarchical clustering methods have mostly been used in biology, sociology and behavioural sciences. Divisive algorithms are not very popular and limited to few [58]. The most popular agglomerative algorithms are single-link [108], complete-link [29, 61], average-link and Ward's method [85, 124]. The main attribute that characterises the variation of hierarchical clustering algorithms is their criterion in merging two clusters, that is, the method of evaluation of similarity between clusters [53, 58].

The single-link clustering is defined by considering the distance between a pair of clusters as the minimum distance between all pairs of data in-

stances. Minimum distance between pairs of clusters is the criterion for combining them. Despite of simplicity of single-link for implementation, it has not achieved good results compared to some other agglomerative clustering methods [58].

The complete-link [29, 61] is a variation of single-link where the distance between a pair of clusters is defined as the maximum distance between all pairs of data instances [53]. The average-link is another variation of single-link which use the average distance instead of maximum or minimum distance. Producing clusters of outliers has been cited as the drawback of this method. The complete-link algorithm has the tendency of producing compact clusters, while in single-link clusters, they are usually elongated [53, 58].

In Ward's [124] method, the criterion to merge two clusters is that it minimises the sum of squared errors within the cluster. In this algorithm two clusters will join if the resulted cluster minimises the total square error within that cluster using ED between centroids. Although this method is recognised to achieve high accuracy among hierarchical methods, but there is a concern that it has the tendency of producing spherical clusters. This can impose some limitations on its performance in different applications [53, 58].

Generally the procedure in hierarchical methods can be summarised as the following steps [53, 58]:

- Calculating the dissimilarity matrix between all pairs of data instances in the dataset, considering each data instance as a cluster.
- Merging each cluster with the most similar cluster based on the criterion of similarity in the algorithm and updating the dissimilarity matrix to reflect the merge operation.
- Stopping in case of meeting stopping criterion or repeat the previous step.

Some of stopping criteria proposed in the literature reviewed in [58] are as follows: number of iteration when it reaches to n-1, predefining K number of clusters, all data instances are in one cluster, defining a threshold for the average dissimilarity within a cluster, maximum distance between data instances in clusters reaches a threshold, and when relative similarity within cluster reaches a threshold.

Hierarchical methods are more versatile than partitioning methods, however they are more expensive in terms of time and computation complex-

ity [53, 58]. Also, in the majority of these methods, finding a center or representative for clusters is not addressed.

2.4.1.2 Partitioning clustering

Partitioning clustering methods output a single partition of the dataset (consists of multiple clusters), instead of the dendrogaram in hierarchical methods. Unlike hierarchical methods that build the clusters gradually, partitioning methods learn the clusters directly [12]. Partitioning methods build clusters either by relocating the data instances between initial defined subsets or by identifying highly populated areas as clusters. This leads to two classes of methods, namely, relocating partitioning and density-based partitioning methods.

Relocating partitioning clustering: These methods optimise an objective function by relocating the data instances between clusters. A further categorisation for these methods is probabilistic methods , k-means and k-medoids [12]. In the following we review these methods.

Probabilistic methods: In these methods the underlying assumption is that the data instances to be clustered are drawn from a mixture model of several distributions. The main assumption is: the area around the mean of the model that the data instances are drawn from constitutes a cluster. Hence, the cluster is associated with the some parameters from the distribution such as mean and variance [12, 53]. The probability (likelihood) of the assignment of a data instance to a distribution can be estimated that it leads to a soft assignment (each data instance belongs to a distribution with a certain likelihood). Expectation Maximisation (EM) as a well-known method in this category and has two steps: (i) perform the soft assignment (ii) find the approximation to a distribution given the soft assignment. This leads to finding a mixture model parameters that maximise the log-likelihood. The process continues until the log-likelihood convergence is achieved. There are other methods to find a better local optimum which are comprehensively reviewed in [12].

k-means :As there are a large number of possible ways to partition a dataset into k number of clusters, finding a best way is impossible. Hence, partitioning methods usually optimise a criterion function which is either defined locally (within a cluster) or globally (for the whole dataset) [52, 53, 58]. Sum of squared error (SSE) is the most intuitive and widely used criterion (objective) function in partitioning methods. The SSE for k clusters

of a dataset, **X** is defined as:

$$SSE(\mathbf{X}, k) \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| X_i^{(j)} - C_j \right\| \tag{2.12}$$

where C_j is the centroid of the j^{th} cluster and X_i is a data instance in **X**. Partitioning clustering methods has the advantage to determine the number of clusters that is expected as the output. Generally, partitioning methods using SSE is consists of the following steps [53]:

- Selecting the initial partitions of the dataset with specified number of clusters and centers.
- Assigning each data instance to a closest cluster center and recomputing the cluster center based on the new cluster.
- Updating the representative of the clusters.
- Repeating the two previous steps until it converges and cluster membership becomes stable.

k-means [49] is the most common and well-known of partitioning methods that has been widely used in scientific and industrial applications [12]. It has been proposed using the idea known as Forgy's method [58, 126]. The name comes from the partitioning of the dataset to k number of clusters which are represented by the mean (average) of data instances in each cluster called centroid. k-means makes use of SSE as objective function. Despite the popularity of k-means, there are some concerns about it. The results strongly depend on the initialisation step where centroids are selected randomly, as there can be a huge difference between local and global optimum, the right choice of k, the sensitivity of k-means to outliers and the fact that it just supports the numerical attributes (features). Also, the choice of SSE as the objective function limits k-means to work only with ED. Many studies have proposed different methods to address these limitations, a comprehensive review of k-means extensions can be found in [12, 52]. We will discuss the details of k-means method and its dependency to ED in Chapter 5. K-means has been frequently used in image analysis.

The study [106] used 'colour moment' and Block Truncation Coding (BTC) as the features to represent the images from Corel dataset. Authors have proposed to use k-means algorithm to cluster the dataset images,and their

results show an acceptable performance. The other study in [86], exploited two different clustering methods to group the images represented by colour features. In this study, authors first have applied hierarchal clustering, then, the k output clusters of this stage have been used as the input for k-means clustering. Finally the clustered images have been used for the purpose of image retrieval to improve the accuracy of top ranked images.

We have discussed the image segmentation as a component in some image retrieval when dealing with objects in image in Section 2.1. The study in [63], proposed a variation of k-means clustering where initial centers are selected using a density estimation method. The proposed clustering method has been used for image segmentation based on colour features. Also, k-means has been used as a step before applying the improved watershed segmentation method in [87]. Using k-means in [57, 87] has been proposed as a primary segmentation to address the limitations of watershed segmentation in over segmentation.

k-medoids :Unlike k-means, where clusters are represented by the mean of data instances assigned to a cluster, k-medoids represent a cluster using one of its assigned data instances. The data instance that represents a cluster in k-medoids is simply called, 'medoid'. k-medoids is not limited to numerical attributes. The objective function considered in this method is minimising the sum of dissimilarities in a cluster. As the objective function is not minimising SSE as considered in k-means, the choice of dissimilarity measure is not limited to ED. The use of medoids instead of centroids also alleviates the sensitivity to outliers as peripheral cluster data instances do not affect the updating of medoids[12]. After selection of medoids, clusters are defined as subsets of data instances close to respective medoids.

PAM (Partitioning Around Medoids) is the earliest method implemented the k-medoids idea, followed by its extension for large applications, CLARA (Clustering LARge Applications) [59]. In PAM, first medoids are selected randomly. An iterative process updates the medoids by considering each of the data instances in a cluster as medoids until the objective function is satisfied. CLARA relies on sampling to handle the large datasets, and it uses PAM to cluster a sample drawn from the dataset to identify the medoids. It uses the identified medoids for the assignment of the data instances which are not present in the sample. The process of sampling will iterate for a predefined number of times to alleviate the bias effect of sampling [12, 59]. CLARANS (Clustering Large Applications Based Upon Randomized Search) is another extension of PAM which uses a graph in context of clustering of spatial databases [12, 88]. Details of k-meodids

implementation will be discussed further in Chapter 5.

Density-based clustering :Density-based clustering is based on the idea that a set of data points in Euclidean space can be split to a set of its connected components. This implements the concept of connectivity for partitioning a set of data instances in the feature space. A cluster is defined as a connected dense set of data instances/points, the cluster can grow in the direction that density leads. This enables density-based methods to discover arbitrary shapes of clusters and embed a natural resistance against outliers [12]. One of the drawbacks with density-based methods is their weakness in handling clusters with different densities. The density-based methods require metric space and this naturally makes them suitable for spatial database clustering. Important concepts in these methods are density, connectivity and boundary which can be measures in term of local distribution of neighbours [12].

The main representative method in this category is DBSCAN (Density Based Spatial Clustering of Applications with Noise) which uses two parameters of ϵ and $Minpoints$ to the define the following concepts used in this method:

- An ϵ- neighbourhood of x which is the $N_\epsilon = \{x, y \in \mathbf{X}, d(x, y) \leq \epsilon\}$,
- A core point (a point with a neighborhood consisting of more than $MinPts$ points), points in the neighbourhood of a core point are directly-reachable points.
- A density-reachable point of a core point x is a point in the ϵ- neighbourhood of one of a directly-reachable point of x.
- A density-connectivity translates to two points, x,y, being density-reachable from each other.

So, based on these definitions, all the data points in a dataset are categorised into: directly or density reachable from a core point and outliers. The points which are not related to any core points are considered as outliers. The non-core points in a cluster represents the boundary of that cluster. DBSCAN suffers from the limitation of detecting meaningful clusters in datasets with varying densities. Ordering points to identify the clustering structure (OPTICS) is proposed in [3] based on the idea of DBSCAN and to address the mentioned limitation. In OPTICS, first data points in the dataset are ordered such that points which are spatially closest become neighbours. In addition to this, a special distance is defined for each data point that represents the required density for a cluster.

Recently, Ting et al (2016) [114], proposed a new method based on the mass estimation to address the limitation in DBSCAN with varying density. Unlike DBSCAN which was based on ϵ-neighbourhood, in this method a μ-neighbourhood is defined that considers the data mass (number of data points in the area covering the target points). In this method core point is defined based on the mass around it, instead of the *Minpoints* in a ϵ distance of a neighbourhood of that point. The mass around a point should reach to a certain threshold where makes this method not relying on the a ϵ distance. Relying on ϵ distance to define the core points in DBSCAN raises its limitation in identifying the clusters with varying densities.

2.5 Dissimilarity measures in clustering

The discussed clustering methods make it obvious that dissimilarity is a fundamental concept in definition of a cluster. A dissimilarity measure is essential in almost all of the clustering methods to evaluate the similarity of two data instances in the feature space [53]. Due to the variety of feature types, scale, and the aspect of similarity that an expert needs to investigate from the data, the choice of dissimilarity measure has a great importance and affect on the output of clustering. In the following section, we will review the most common dissimilarity measures in clustering methods.

2.5.1 Geometric distances

In this section, we review some geometric distance choices in clustering. A distance that measures dissimilarity between two instances and denoted by $d\left(\mathbf{X}_i, \mathbf{X}_j\right)$ should satisfy the metric properties of:

- $d\left(\mathbf{X}_i, \mathbf{X}_i\right) = 0$
- $d\left(\mathbf{X}_i, \mathbf{X}_j\right) = d\left(\mathbf{X}_j, \mathbf{X}_i\right)$
- $d\left(\mathbf{X}_i, \mathbf{X}_j\right) = 0$ if and only if $\mathbf{X}_i = \mathbf{X}_j$
- $d\left(\mathbf{X}_i, \mathbf{X}_j\right) \leq d\left(\mathbf{X}_i, \mathbf{X}_m\right) + d\left(\mathbf{X}_m, \mathbf{X}_j\right)$

Popular distance measures used in clustering have been reviewed in [12, 53, 58]. The most well-known dissimilarity measure used in clustering is ED as shown in equation 2.1 where $p = 2$. ED is intuitive and has been

used commonly to measure the dissimilarity of objects in two or three dimensional space. It has been shown that it works well with compact or isolated clusters. Some of the concerns about this measure are as follows. It has the tendency that dimensions of feature vectors with relatively large values dominate the others [53]. The performance of ED varies considerably depending on the application in hand [6]. Additionally, ED relies only on the geometric position of two data instances in the feature space and completely ignores the effect of data distribution in its dissimilarity calculation [6, 62]. Despite of these concerns, ED has been widely used in clustering and it is the main distance measure used in k-means method and its extensions, which are popular clustering methods[52].

To address the concern about the adverse effect of features correlation on distance measure, Mahalanobis distance has been proposed as we discussed in Section 2.3. The definition of Mahalanobis distance has been shown in equation 2.6. Manhattan distance, as defined in equation 2.1 where $p = 1$, has been reported as a distance metric more robust to outliers in clustering [58].

2.5.2 Non-geometric dissimilarity measures

There are few dissimilarity measures in the literature that consider the neighbourhood of data instance in the feature space in calculating the dissimilarity between two data instances [53]. Mutual neighbour distance (MND) has been proposed in [44] and defined as:

$$MND\left(\mathbf{X}_i, \mathbf{X}_j\right) = NN\left(\mathbf{X}_i, \mathbf{X}_j\right) + NN\left(\mathbf{X}_j, \mathbf{X}_i\right) \tag{2.13}$$

where $NN\left(\mathbf{X}_i, \mathbf{X}_j\right)$ is the number of neighbours $\mathbf{X}_j$ with respect to $\mathbf{X}_i$. The higher number of neighbours between two instances shows the higher dissimilarity. MND has been used for clustering [43], however it does not satisfy the metric properties of a distance [53, 139].

The other dissimilarity proposed in this group is *shared nearest neighbour* (SSN) [55]. To estimate the SSN between two data instances, the N nearest neighbours of each of them is calculated. The greater number of shared nearest neighbours shows a higher confidence on their similarity. In [55], a shared nearest neighbor graph is constructed from the similarity matrix as follows. A link is created between a pair of points, p and q, if and only if p and q have each other in their N nearest neighbours lists. This process is called *k-nearest neighbor sparsification*. The weights of the links between two points in the SNN graph can either be simply the number of near

neighbours that the two points share, or one can use a weighted version that takes the ordering of the nearest neighbours into account. This dissimilarity measure has been used in the clustering method presented in [33].

This measures is not metric (does not satisfy the triangle inequality). Relying only on the neighbouring instances implies that it is possible to make any two arbitrary patterns similar only by encoding them with sufficiently large features. Consequently, any arbitrary patterns are equally similar, unless we use some additional domain information [53]. This imposes a potential limitation for these methods. In addition to this, these measures are not suitable to be used in algorithms such as k-means that has been used widely in both scientific and industrial applications.

2.5.3 Summary

In this chapter we have reviewed the works closely related to the subject of this thesis. Image retrieval framework has been reviewed along with their most important components, feature extraction and common dissimilarity measures used in these methods. We have discussed clustering, its different methods and common dissimilarity measures used in clustering. Through the study of literature, we have identified that despite of limitations associated with ED, as discussed in the Introduction chapter, it has been one of the most widely used distance measures, for both image retrieval and clustering. In addition to this, k-means is one of the most popular clustering methods for numeric features, which also used ED as its distance measure. Data distribution as considered in m_p has a perceptual effect on human judgment of similarity. In the next chapter, we will study the suitability of mass-based dissimilarity measure proposed in [5] for image retrieval. Also, we will investigate a novel dissimilarity measure that addresses limitations of ED by incorporating the data distribution.

3 A Novel Perceptual Dissimilarity Measure

As we discussed in previous chapters, dissimilarity measure is central to image retrieval for evaluation of similarity between the query and other data base images in the feature space. Also it is fundamental in data/image cluster definition for minimising the intra cluster similarity and maximising the inter cluster similarity. In the Introduction chapter, we have briefly mentioned the limitations of geometric distances, specifically ED. Also, we have explained our motivation for developing a new dissimilarity measure, to overcome the limitations of ED and utilising the potentials of mass-based dissimilarity. Through our study of the literature, we have identified that ED is the most common distance measure that has been most widely used in image retrieval and clustering methods. In this chapter, we first discuss the ED and its limitations in greater detail. Then we study the suitability of m_p [5, 6] for image retrieval. Section 3.4 presents our main contribution; developing a novel perceptual dissimilarity measure that combines the perceptual effect of data distribution and ED. The last section summarises this chapter.

3.1 Euclidean distance

The distance between two points in the the Euclidean space is the length of the line segment directly connecting them and is known as *Euclidean distance* (ED). We will use ED and distance interchangeably here after in this thesis. Distance between two points of x and y (one dimension) is the absolute value of their difference [64]:

$$ED = |x - y| = \sqrt{(x - y)^2} \tag{3.1}$$

Generalisation of ED for multiple dimensions is as follows: consider x and y as two d-dimensional vectors, then the ED between them will be [64]:

$$ED(x, y) = |x - y| = \left(\sum_{i=1}^{d} (x_i - y_i)^2 \right)^{\frac{1}{2}} \tag{3.2}$$

ED satisfies the metric principles:

- The minimum value for the distance between two data instances is zero when they are identical.
- The distance between two data instances when they are different is a positive number.
- The distance is symmetric which means distance between data instance x and y is the same as distance between y and x.
- The triangle inequality: considering three data instances, the distance between a pair of instances is smaller than the sum of distance between the other two pairs of them.

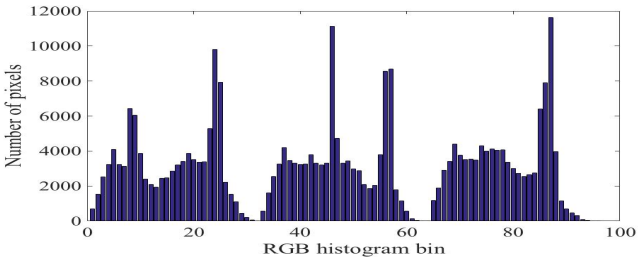
Equation 3.2 is the aggregation of the distance between two data instances in each dimension to calculate the total distance. The ED calculations completely depend on the geometric position of each data instance in the feature space and the distribution of neighbouring data has no impact on that. However, psychologists argue that human perception of dissimilarity is affected by the distribution of the data in the neighbourhood around the two data instances [62]. This effect is completely ignored in distance calculation of ED. An example of the perceptual effect of data distribution is presented in Chapter 1, Section 1.2.1.

The other limitation with ED is that it gives the equal weight to all dimensions when combining distances. This may result in a situation where the dissimilarity between two instances is determined by a few dominant dimensions. There is a tendency that the small number of the dimensions of feature vectors with relatively much higher values dominate the overall dissimilarity value. As a result, the dissimilarity derived might not align well with how humans perceived dissimilarity.

Figure 3.1 shows the colour histograms for two beach images and one image from building class of Corel dataset. ED finds the smaller distance between the beach image in Figure 3.1 (a) and the building image in Figure 3.1 (b) compared to the second beach image in Figure 3.1 (c). This

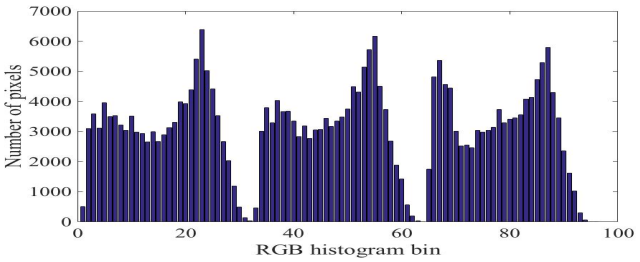
dissimilarity is not aligned with human perception. As we can see, the difference between the largest values in the colour histogram of the beach image in Figure 3.1 (d) and the building image in Figure 3.1 (e) is about 5000. However, this difference between the histogram of the two beach images in Figure 3.1 (d and f) is more than 10,000. ED gives equal weight to all dimensions when combining distances. As a result, a few dimensions with relatively much higher values contribute substantially to the total distance calculated, whereas the contribution from the remaining dimensions might be negligible to impact the final distance value. For example in Figure 3.1 (a), the colours of the sand (around histogram bins 28, 59 and 90), on the beach dominate the distance calculated, whereas the detailed colours, such as the skin colours of people (around histogram bins 21, 43 and 74), though perceptually important when comparing similarity between these images, are not dominant enough to impact the final distance measurement.

(a) Beach



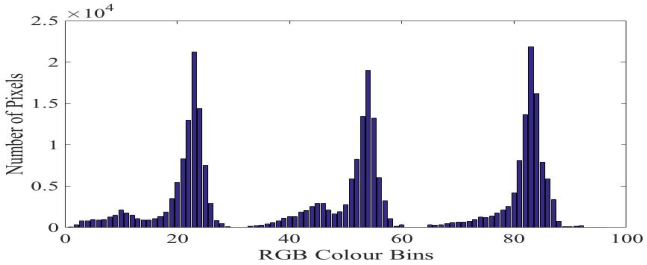
(d) RGB histogram of beach

(b) Building



(e) RGB histogram of building

(c) Beach



(f) HSV histogram of beach

Figure 3.1: Two images from beach and an image from building class of Corel dataset along with respected RGB colour histograms

These issues demonstrate ED may not be most suitable to measure perceptual similarity between images in some situations.

3.2 Mass-based dissimilarity (m_p)

To address the limitation of ED in ignoring the data distribution, a data dependent dissimilarity measure has been proposed by Aryal et al [5, 6] called m_p. The idea of m_p is based on the perceptual effect of data distribution when human judge similarity. In the Introduction chapter, we

mentioned the example of two red apples in two different contexts, where two red apples are among green apples and are among red apples. In other words, two data instances in a relatively denser region of the feature space are perceived to be less similar than the two objects of equal distance but located in a sparse region.

.

The measure m_p uses the neighborhood data to make decisions about the similarity of two data instances. It considers the data distribution in a region that covers the two instances. m_p works as follows: in each dimension i, it defines a region $R_i(x, y)$ which encloses x and y, and computes data mass in $R_i(x, y)$ as the measure of dissimilarity of x and y instead of their distance. Data mass is the number instances falling in $R_i(x, y)$. m_p is defined as:

$$m_p(x, y) = \left(\frac{1}{d} \sum_{i=1}^{d} \left(\frac{|R_i(x, y)|}{N} \right)^p \right)^{\frac{1}{p}} \tag{3.3}$$

where

- $|R_i(x, y)|$ is the data mass in region of $R_i(x, y)$,
- N is the total number of instances in the dataset,
- $R_i(x, y) = [min(x_i, y_i) - \sigma, max(x_i, y_i) + \sigma]$,
- σ is a small number and $\sigma \geq 0$.

Although m_p employs the same power mean formulation as l_p, the core calculation is based on mass rather than distance. It signifies the degree of dissimilarity: the higher the measure, the more dissimilar the two instances are; just like l_p.

Calculation of m_p is expensive as it requires a range search in each dimension, so to address this problem a new implementation has been proposed in [6]. In this new implementation, a histogram is used and the real values in each dimension i are divided into m bins. The number of instances in each bin is computed in a preprocessing step, and then data mass between two points can be computed using the number of bins between them. An illustration of defining $R_i(x, y)$ using bin implementation is shown in Figure 3.2.

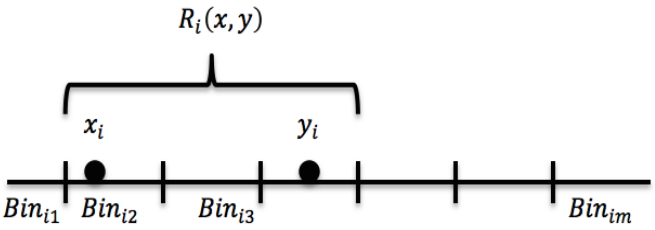


Figure 3.2: Illustration of m_p, data mass calculation between two data points x and y.

3.3 Evaluating suitability of m_p for image retrieval

Since m_p was introduced as an alternative to ED, which is the most widely used measure in image retrieval, it is necessary to investigate the performance of a dissimilarity measure that relies only on data distribution in this context. In this section, we evaluate the suitability of m_p for image retrieval and compare its performance with ED using three image datasets. In the following section we first introduce the benchmark datasets used for our experiments. Then we present the feature extraction and evaluation metrics, followed by experimental results and discussion.

3.3.1 Benchmark datasets

To evaluate the performance of m_p for image retrieval, we have selected three different image datasets. These datasets have their own characteristics in terms of image concepts and ground truth. In the following we introduce these three datasets.

eBay [119]: This dataset is a collection of 528 images of objects which are categorised into 11 classes based on their colour. Images in the red class are shown in Figure 3.3 as a sample of eBay dataset. As we can see in each class, different objects with the same colour are collected. The ground truth (class label) is the primary colour in images and objects of the target colour are segmented in dataset images. Figure 3.4 provides a sample of images from eBay dataset along with their mask images that the primary colour objects are segmented in them.

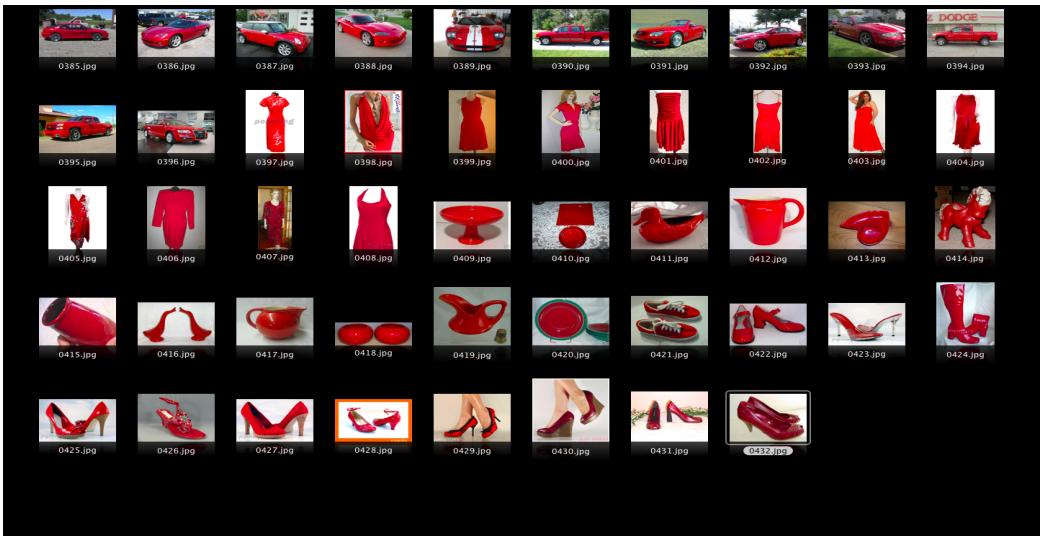


Figure 3.3: Images in red class of eBay dataset

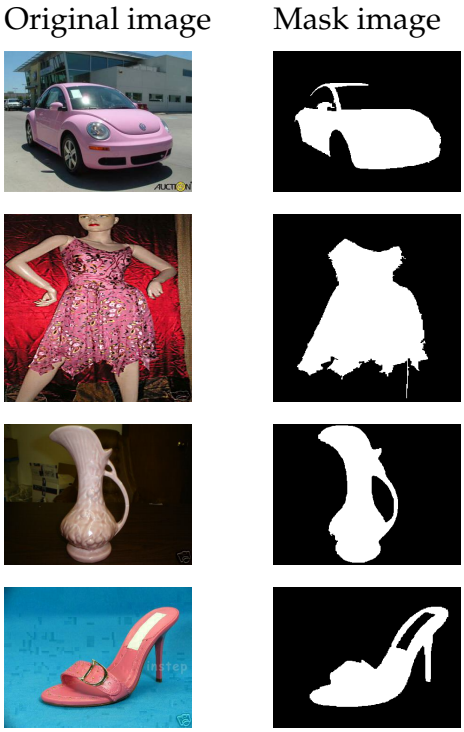


Figure 3.4: Sample of images from pink class in eBay dataset and their mask images

Texture [65] : It has 1,000 images categorised in 25 classes and each class has 40 images. Each class represents a different texture, such as wood, wallpaper, water and brick. Figure 3.5 shows a sample of this dataset from the brick class.

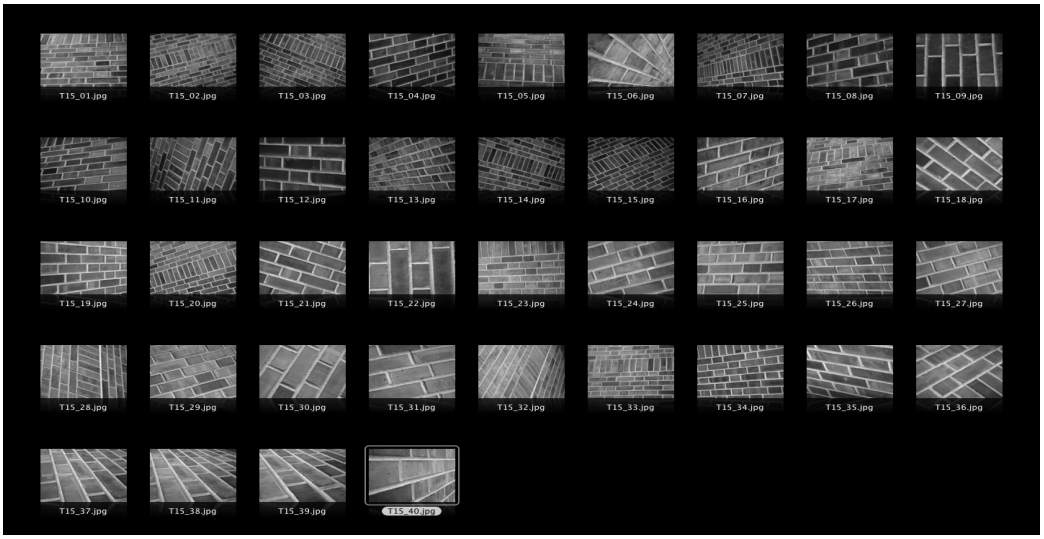


Figure 3.5: Images in brick class of Texture dataset

Corel [69]: This dataset is a collection of 1,000 images categorised into 10 classes. The images are a mixture of objects and natural scenes. Some example of classes in this dataset are beach, mountain, flower and bus. Figure 3.6 shows a sample of images in the beach class.

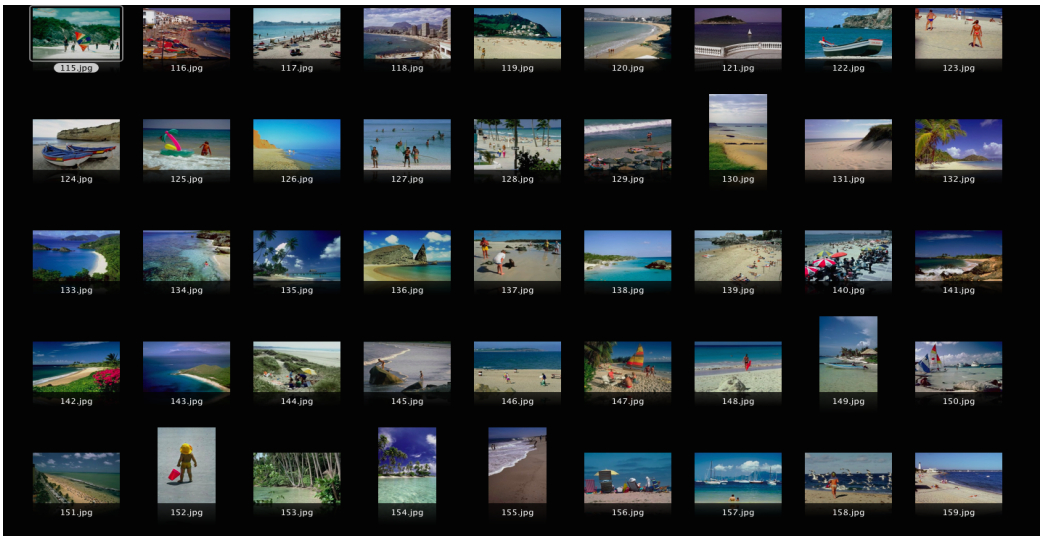


Figure 3.6: Images in beach class of Corel dataset

3.3.2 Feature extraction

We can see that the three benchmark datasets present different types of images. To represent these images from these datasets, we need to extract appropriate features that matche the characteristics of each dataset. We will use common features which are matched for the datasets. In the following section we describe the feature extraction process for each of these datasets.

3.3.2.1 HSV colour histogram

We have mentioned that the eBay dataset has the ground truth of colour, so colour features are our choice to represent the images in this dataset. Our primary purpose for choosing eBay dataset is the intuitive nature of colour as compared to more complex image features. Colour histograms represent the distribution of colour in an image. They are rotation invariant and have been used in image retrieval studies [134]. A colour histogram of an image is produced first by quantising of the colours (components of colour space) in the image into a number of bins, and counting the number of image pixels in each bin.

We now explain the colour space used for histogram extraction in this work. A colour space is the specification of a coordinate system and subspace within that system where a single point represents a distinct colour

value. This representation is used for image analysis like extraction of colour histograms. Each colour space has its own merits and demerits depending on the application and hardware specification where it is going to be used. Hue Saturation Value (HSV) is one of the popular colour spaces.

A three dimensional representation of the HSV colour space is a hexacone, where the central vertical axis represents the Intensity (value). Hue is defined as an angle in the range $[0, 2\pi]$ relative to the red axis with red at angle 0, green at $2\pi/3$, blue at $4\pi/3$ and red again at 2π. Saturation is the depth or purity of the colour and is measured as a radial distance from the central axis with value between 0 at the center to 1 at the outer surface. For V=0, as one moves higher along the Intensity axis, one goes from Black to White through various shades of grey. On the other hand, for a given Intensity and Hue, if the Saturation is changed from 0 to 1, the perceived colour changes from a shade of grey to the most pure form of the colour represented by its Hue. Looking from a different angle, any colour in the HSV space can be transformed to a shade of grey by sufficiently lowering the Saturation. The value of Intensity determines the particular grey shade to which this transformation converges. When Saturation is near 0, all pixels, even with different Hues, look alike and as we increase the Saturation towards 1, they tend to get separated and are visually perceived as the true colours represented by their Hues.

We have chosen to use colour histogram in HSV colour system to represent eBay dataset. We have divided each dimension of HSV colour system to 30 bins and extracted the histogram, so the final histogram feature has 90 dimensions.

3.3.2.2 Local Binary Patterns

Local Binary Patterns (LBP) [90, 122] is a simple yet very efficient texture operator, which labels the pixels of an image by thresholding the neighbourhood of each pixel and considers the result as a binary number. Due to its discriminative power and computational simplicity, LBP texture operator has become a popular approach in image retrieval and classification. LBP features are extracted from an image as follows.

- Divide an image into square blocks e.g 16×16.
- For each pixel in the block, compare each pixel to its 8 neighbours.
- Neighbours with value greater than the centre's pixel, will be assigned as 1 and 0 otherwise. This way each pixel is described by 8

binary digits.

- Compute the histogram over each block counting the frequency of occurring values.
- Concatenate the histogram of each block to obtain the image level features.

We have extracted the LBP features from Texture dataset that resulted in feature vectors of 236 dimensions.

3.3.2.3 SIFT bag of words (BOW)

The Corel dataset has been categorised in classes such as beach, building and animals. Through the literature [11, 16, 133] SIFT BOW has been used to represent the Corel dataset. As discussed in Chapter 2, Section 2.2.4, SIFT features proposed by David Lowe [75] in 1999 is one of the popular local features being used in image retrieval [120, 141]. They are scale invariant and share some properties with the neuron responses of an inferior temporal cortex in primate vision [75]. To obtain the image level features using dense SIFT, we need to go through the encoding process. BOW is the baseline of the encoding methods, which has been used successfully in image retrieval [18, 105, 120, 129, 141]. The procedure of extracting SIFT features and BOW has been discussed in Chapter 2, Section 2.2.4. We will just briefly describe the steps as follows:

- Extracting SIFT features from images
- Quantising features using the k-means clustering, we quantise the extracted features to N number of visual words.
- Building the BOW histogram by finding the frequency of occurrence of each visual word in the image.

The number of visual words is usually determined based on the experiments on the dataset. The study in [11] has used 50 visual words and the authors in [16] have used 100 visual words for SIFT BOW to represent Corel dataset. Also, the experiments in [133] on the size of visual words ranged between 50 to 250 of Corel dataset and show a comparable performance for the choice of 100 visual words. The other study [14] compared the performance of SIFT BOW with different dictionary sizes from 100-2,100 for a dataset called "OT" which is very similar to Corel dataset.

Their results show visual words of 100 had similar performance with 500, 1700. Also, it has performed slightly better than using 1900 and 2100 visual words. Based on our study of the literature and computational complexity trade off, we have chosen to use the SIFT BOW with 100 visual words in this work to represent the images in Corel dataset.

3.3.3 Evaluation metrics

This section presents the evaluation metrics for image retrieval that we will use in this work. Precision- Recall curves are common metrics to evaluate the performance of an algorithm in image retrieval [28]. In image retrieval, the retrieved image would be considered as relevant or non-relevant based on the ground truth provided in the dataset. A retrieved image would be considered as relevant if it is from the same class with the query, however, if it is from a different class with the query, it would be known as non-relevant. Based on these definitions precision and recall are defined as follows:

$$Precision = \frac{\#\left(relevant\ images\right)}{\#\left(retrieved\ images\right)} \tag{3.4}$$

$$Recall = \frac{\#\left(relevant\ retrieved\ images\right)}{\#\left(relevant\ images\right)} \tag{3.5}$$

Precision is the fraction of retrieved images that are relevant and Recall is the fraction of relevant images that are retrieved [22]. We can also define them as :

$$P = \frac{TP}{(TP + FP)} \tag{3.6}$$

$$P = \frac{TP}{(TP + FN)} \tag{3.7}$$

where a true positive (TP) is a retrieved image which is relevant. False positive (FP) is a retrieved image which is non-relevant. False negative (FN) is a relevant image which is not retrieved. We will use P-R curve to evaluate the performance of ED, m_p and PDM for image retrieval in this work.

3.3.4 HSV colour histogram weighting

This section presents our proposed weighting system for extracting the HSV colour histograms that let each of HSV components contribute proportionally based on their importance for humans perception. HSV colour space has been described in Section 3.3.2.1, as the selected colour space to extract colour histograms for the images of the eBay dataset. As suggested in [109, 141], Hue has more importance in distinguishing the perceived colour and two other components have equal importance in feature representation using colour histogram. To balance the contribution from each of the HSV colour system components in the feature space, we propose a weighting that assigns higher weight to Hue and equal weights to Saturation and Value components. To apply the proposed weights for HSV colour histogram, we will modify equations 3.2 and 3.9 for calculation of ED and m_p as follows:

$$ED(x,y) = \left(\sum_{i=1}^{d} HSVW_i \times (x_i - y_i)^2 \right)^{\frac{1}{2}} \tag{3.8}$$

$$m_p(x,y) = \left(\frac{1}{d} \sum_{i=1}^{d} HSVW_i \times \left(\frac{|R_i(x,y)|}{N} \right)^p \right)^{\frac{1}{p}} \tag{3.9}$$

where $HSVW_i$ is the respective weight of each of the HSV components for each dimension. For example, consider the HSV colour histogram as having 90 dimensions (bins) in total, 30 bins for each of the HSV components and the weights of 50%, 25% and 25% for HSV components. In this case $HSVW_i$ for H component (first 30 bins) would be 50% and 25% for the S and V components (second and third sets of 30 bins).

To determine the appropriate weight for different HSV components, we have performed image retrieval experiments on the eBay dataset using ED and m_p as dissimilarity measures. We have extracted colour histograms with 30 bins for each HSV component. Figures 3.7 and 3.8 show the retrieval results of eBay dataset using ED and m_p as dissimilarity metrics. We have experimented with different sets of weights ranging from $(50\%, 25\%, 25\%)$ to $(80\%, 10\%, 10\%)$. The weights of 60% for Hue, 20% for Saturation and 20% for Value components have shown the optimum performance. So we have chosen this set of weights for our experiments in this work.

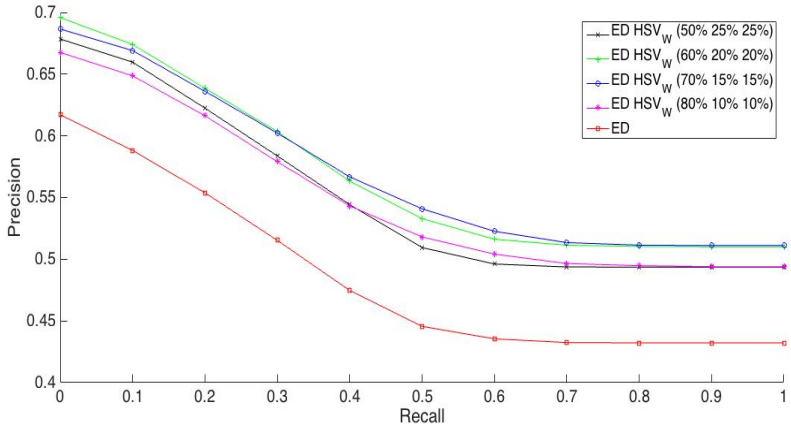


Figure 3.7: Image retrieval results on eBay dataset using HSV colour histogram with different sets of weights on ED.

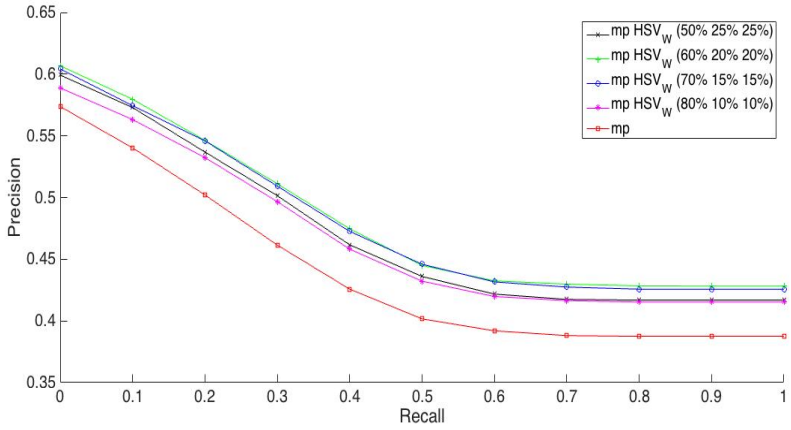


Figure 3.8: Image retrieval results on eBay dataset using HSV colour histogram with different sets of weights on m_p.

3.3.5 Experimental result for evaluating the performance on m_p

In this section, we present the retrieval results of three benchmark datasets. Figures 3.9- 3.11 show the retrieval results using precision-recall curves. As we can see, m_p does not have a consistent performance. Retrieval results from Texture dataset in Figure 3.10, show m_p outperforms ED, while retrieval results from the two other datasets in Figures 3.9 and 3.11 show

ED performed better. m_p has been proposed to address the limitation of ED, relying only on geometric position of two instances in the feature space and ignoring the data distribution. m_p has the strength of using perceptual effect of data distribution in measuring the dissimilarity and thus was expected to perform better than ED. In the next section we will discuss the performance of m_p.

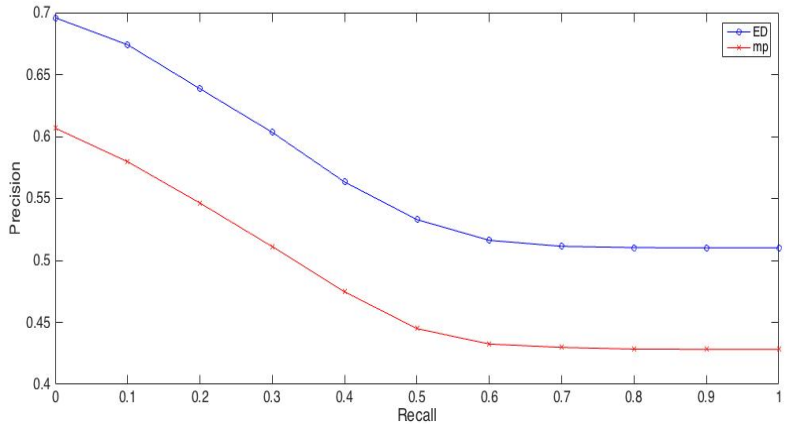


Figure 3.9: Image retrieval results of eBay dataset using ED and m_p as dissimilarity measures.

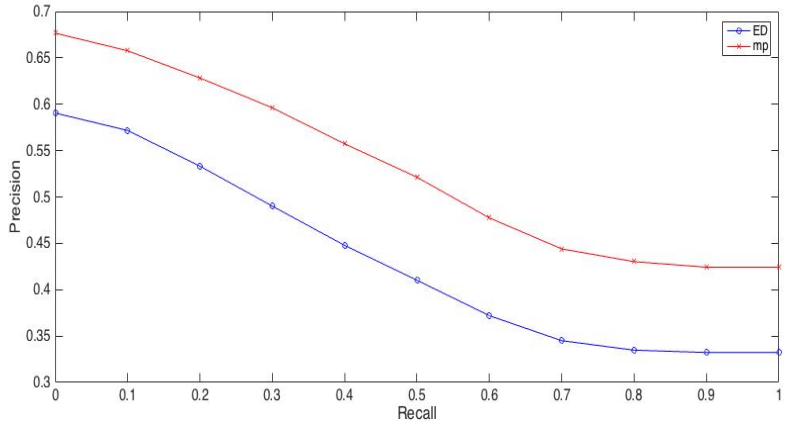


Figure 3.10: Image retrieval results of Texture dataset using ED and m_p as dissimilarity measures.

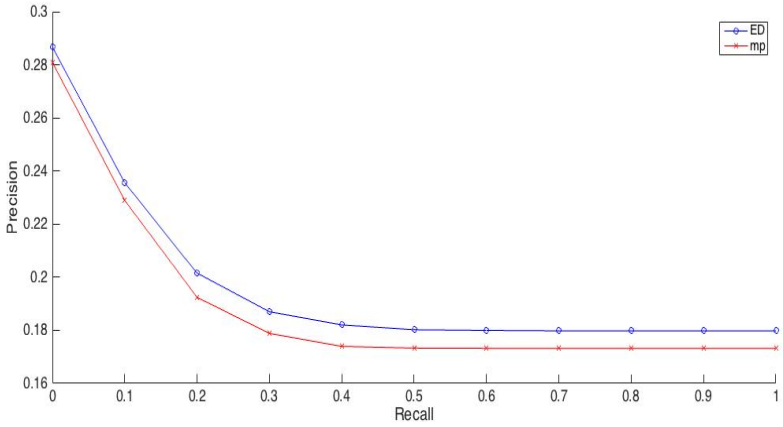


Figure 3.11: Image retrieval results of Corel dataset using ED and m_p as dissimilarity measures.

3.3.6 Discussion on m_p's performance

In this section, we discuss the limitations of m_p that resulted in its inconsistent retrieval performance for different datasets. m_p is a data-dependent dissimilarity measure and relies on data distribution. Data distribution has an effect on the perceived similarity as considered in m_p. However, the ED between two instances should not be ignored, as it intuitively corresponds to the defined dissimilarity in the real three-dimensional world, specifically when the magnitude of vectors in feature space matters. m_p calculates the dissimilarity between two instances solely based on data distribution in the region covering the two instances. Consider the following example. We have two pairs of data instances: the instances of the first pair are perceptually similar but they are located in a dense region while the second pair are perceptually dissimilar but are located in a sparse region. m_p will consider the second pair to be more similar than the first pair, contrary to the perceptual similarity. This example shows that m_p alone is not suitable to measure perceptual dissimilarity between images.

An effective dissimilarity measure should be accurately mimic the human's judgment of dissimilarity. m_p considers the lower data mass between two instances as lower dissimilarity and vice versa. ED measures the difference between magnitudes of two features using their geometric position in the feature space. m_p and ED have their own strengths. As both of them measure the dissimilarity from different aspects that partially complies with human judgment of dissimilarity, they should not be

in extreme conflict. In some situations, m_p will find two instances similar due to low data mass between them, while they have a large distance. In other situations, m_p may find two instances dissimilar based on high data mass between them while they are perceptually similar and have small distance. In such situations, m_p may not retrieve accurate results.

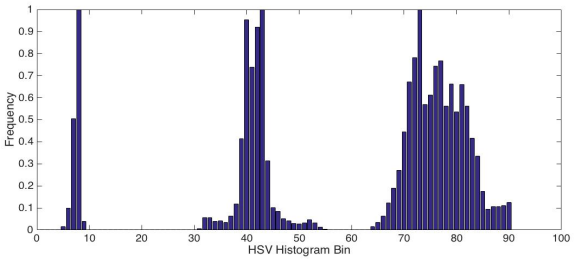
To illustrate this situation, we investigate m_p through a visual example. We discuss m_p's limitation through a few dimensions of the feature space. As m_p aggregates the dissimilarity over all dimensions of feature vectors. considering such a limitation in multiple dimensions affects its performance in calculation of total dissimilarity. Consider the retrieval performance of m_p for eBay dataset, Figure 3.9, where m_p did not work better than ED. In the following we choose a visual example from eBay dataset to explain how the discussed situation can adversely affect the dissimilarity measured by m_p.

The example in Figure 3.12, shows two similar images (green images) and one non-similar image (white shoe) along with their feature vectors. m_p finds the white shoe more similar to the green shoe compared to the green pot based on the high data mass between them. This is contrary to human judgment of dissimilarity. In the following, we first explain the scenario that we use to analyse the m_p's limitation is explained. After this, an example is presented in accordance with this scenario.

The following scenario is used to discuss m_p's limitation. m_p combines the data masses over all dimensions to calculate the overall dissimilarity. As discussed both m_p and ED measure the dissimilarity that is partially complies with human perception of dissimilarity. Therefore, these two should complement each other and should not be in extreme conflict. We will show that the distance between feature values of two perceptually similar images in a specific dimension is very low, however, the data distribution in that dimension results in a high data mass between them. This will result in high dissimilarity measured by m_p which is very different with the dissimilarity measured by ED. On the other hand, we will show that distance between feature values of two non-similar images (green and white shoe) in a specific dimension is relatively high while due to data distribution in that dimension of feature space the data mass is very low. This results in very low dissimilarity measured by m_p, which is very different from the measurement of ED. The details of this example are as follows.

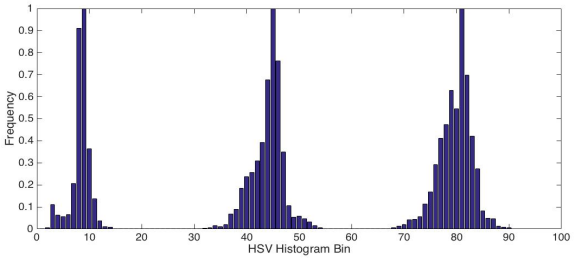
Figures 3.12 (d, e and f) show the colour histograms of the selected images from eBay dataset, in which ground truth is based on colour and images of the same colour are considered as relevant. To calculate the dissimi-

(a) Green shoe



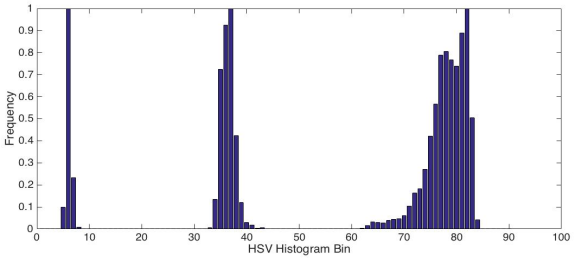
(d) HSV Histogram of Green Shoe

(b) Green pot



(e) HSV Histogram of Green pot

(c) White shoe



(f) HSV Histogram of White shoe

Figure 3.12: Sample of images from eBay dataset and their colour histograms

larity, m_p uses the data mass between two instances in each dimensions then aggregates them. In this example feature vectors are HSV colour histograms with 90 dimensions and their values ranges between 0 and 1, the total number of images (feature vectors) being 528 in the dataset.

We select one dimension of feature space and show how features are distributed in that dimension. Figure 3.13 shows the feature distribution in Dimension 6 of the feature space, which has very dense and sparse regions. This figure shows how many of the 528 data instances in the dataset have the value falling in each of the intervals of (0-0.1, 0.1-0.2,...0.9-1). As shown in Figure 3.13, the number of the instances that has values between 0-0.1 is much higher compared to the rest of the intervals. So, if the values of two instances in the feature space fall between 0 and 0.1, the data mass between them will be very high compare to falling in other intervals.

In Figure 3.12 values of HSV colour histograms for the green shoe, green pot, and white shoe in Dimension 6 are 0.1,0.03 and 1. Therefore, feature values of two similar images, green shoe and green pot fall in the interval of 0-0.1, which is a very dense region and the data mass between them is 400. However, the feature values of the green pot and the white shoe falls in a sparse region where the data mass is only 144.

The distance between the green shoe and the green pot is 0.07, which is much smaller compared to 0.9 between the green and white shoe. In this situation, m_p will find the green shoe more similar to the white shoe in Dimension 6 due to lower data mass between them compared to green pot. Having very dense and sparse regions in this data distribution resulted in the situation that dissimilarity measured by m_p is in conflict with ED. Considering this situation in multiple dimensions results in a low dissimilarity mass measured by m_p between two perceptually dissimilar images compared to two perceptually similar images. This is in conflict with human perception of dissimilarity.

Also, similar to ED, m_p gives equal weight to all dimensions when aggregating the data masses. This may result in a situation where the dissimilarity between two instances is determined by a few dimensions. There is a tendency that the small number of the dimensions of feature vectors with relatively much higher data masses dominate the overall dissimilarity value. As a result the measured dissimilarity may not align well with humans perception. We name this limitation of ED and m_p, the effect of dominant dimensions, and refer to it in the rest of this thesis. In the next chapter, we will provide more visual examples and analysis from all three datasets.

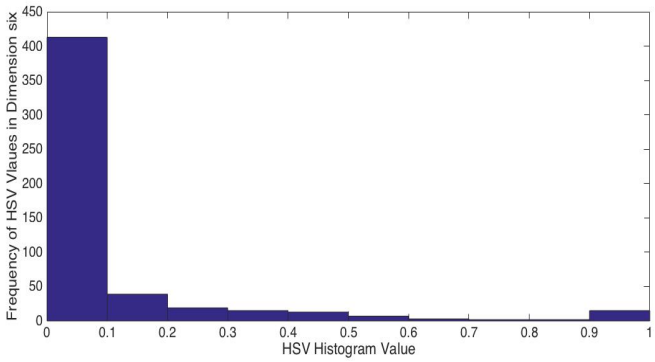


Figure 3.13: A sample of a skewed distribution from features in Dimension 6 of eBay dataset.

3.4 Image dissimilarity measure incorporating human perception

In this section we propose a new dissimilarity measure to address the limitations of m_p and ED. Our proposed dissimilarity measure will incorporate the perceptual effect of region density and ED. The data mass between two instances is used as a proxy for region density. We propose two variants for our new dissimilarity measure as described in the following sections. In the first variant we focus on improving m_p and overcoming its limitations by using ED. m_p has its limitations due to ignoring the distance, and we propose a weighting using ED to improve its performance. The second variant focuses on the ED's limitation of not considering any impact of data distribution on perceived dissimilarity. To address this limitation, we propose to incorporate the effect of data distribution by weighting the ED in each dimension using the data mass.

As mentioned previously, both ED and m_p gives equal weight to all dimensions when combining distances or data masses. This may result in a situation where the dissimilarity between two instances is determined by a few dimension. There is a tendency that the small number of the dimensions of feature vectors with relatively much higher distance or data masses dominate the overall dissimilarity value. As a result the measured dissimilarity by ED and m_p may not align well with human perception. The proposed weighting in both variants of the proposed dissimilarity measure helps to moderate this effect.

The tendency that a few dimensions with very much higher data masses

dominate the overall dissimilarity value measured by m_p, and it may not align with humans perception, will be moderated with assigning small distances as the weight in the first variant. In the second variant, to address the tendency that a few dimension with very much higher distances dominate the overall distance value measured by ED, where it may not align with humans perception, the log of data mass has been considered as the weight. Therefore, in dimensions that the distances have relatively much higher values but they are located in a sparser region, assigning low data masses as the weight for these dimensions help to moderate this effect. In the next two sections, the proposed weighting for both variants of our proposed dissimilarity measure is discussed in detail.

3.4.1 Perceptual dissimilarity measure 1 (PDM1)

The first variant of the new dissimilarity measure we are proposing is called Perceptual Dissimilarity Measure 1 (PDM1). It uses the ED as a weight for mass-based dissimilarity where mass-based dissimilarity may fail to retrieve accurate results. Generally, when we calculate the dissimilarity between two instances using m_p one of the following four cases may occur.

- Case 1: m_p is small (data instances are in a sparse region) and ED is small also;
- Case 2: m_p is small (data instances in a sparse region), but ED is large,
- Case 3: m_p is large (data instances in a dense region) and ED is large also,
- Case 4: m_p is large (data instances in a dense region), but ED is small.

In Cases 1 and 3, m_p and ED are not in conflict. However, in Cases 2 and 4, their measurements are opposite of each other and using m_p alone may not be effective. Both m_p and ED measure the dissimilarity from a different aspect that partially complies with humans perception. Therefore, they should not be in extreme conflict, but complement each other. In cases 2 and 4, ED has small dissimilarity between two instances while m_p finds them highly dissimilar. The m_p's measurement of dissimilarity in these cases may not align well with how humans perceive dissimilarity.

As the definition of the four abovementioned cases are based on sparse/dense region and small/large distance, we need to define them. A threshold is defined in each dimension to identify sparse/dense region and small/large

distance. The threshold is the mid-point of minimum and maximum values of data mass and distance values between a query and all other data instances in the dataset.

To address the discussed limitations of Cases 2 and 4, we will use the distance between two instances to weight the data mass in dimensions that have the situations described in these cases. The weighted m_p is defined as:

$$Wm_p\left(x,y\right) = \frac{1}{d}\sum_{i=1}^{d} abs\left(W_i \frac{\left|R_i\left(x,y\right)\right|^p}{N}\right)^{\frac{1}{p}} \tag{3.10}$$

PDM1 is defined as conventional m_p in Cases 1 and 3, and weighted m_p in Cases 2 and 4 as follows:

$$PDM1 = \begin{cases} Cases1\&3 \qquad m_p(x,y) = \left(\frac{1}{d}\sum_{i=1}^{d}\left(\frac{|R_i(x,y)|}{N}\right)^p\right)^{\frac{1}{p}} \\ \\ Cases2\&4 \quad Wm_p\left(x,y\right) = \frac{1}{d}\sum_{i=1}^{d} abs\left(W_i\frac{|R_i(x,y)|^p}{N}\right)^{\frac{1}{p}} \end{cases} \tag{3.11}$$

In Case 2 where data mass is low but distance is large, we set W_i to the distance between two instances, $W_i = abs\left(x_i - y_i\right)$. This way $W_i > 0$ assigns a higher weight to the data mass, to moderate the very low dissimilarity resulted from low data mass.

In Case 4, where data mass is high between two points but distance is small, we set the W_i as the normalised distance, $W_i = \frac{abs(x_i - y_i)}{max_{m\in D} abs(x_i - m_i)}$, between two instances. $max_{m\in D}$ is the maximum distance between the query instance and all the data instances in the dataset in that dimension. This way W_i assigns a lower weight to the high data mass. This moderates the high dissimilarity resulted from high data mass. The low weight from distance also moderates the effect of dominant dimensions with relatively much higher data masses (in some dimensions) in overall dissimilarity calculation of m_p.

In PDM1, m_p is the basis of dissimilarity calculations and ED has been used as the weight in Cases 2 and 4. In dissimilarity calculation, Cases 2 and 4 may happen in some dimensions of feature vectors, PDM1 using the weighted m_p improves the situation in these dimensions.

3.4.2 Perceptual dissimilarity measure 2 (PDM2)

In this section, we propose the second variant of our new dissimilarity measure that combines the ED and data distribution. In this approach, to consider the effect of data distribution on the final perceived dissimilarity, we will moderate the ED between two instances using the density of the region. In the following, we first describe this variant of our new dissimilarity measure, called Perceptual Dissimilarity Measure 2 (PDM2), and then define the nature of region density effect on the final perceived dissimilarity. In the last part, we discuss how region density should be used as a weight for ED.

The effect of region density is explained as two instances located in a dense region looking more dissimilar compared to locating them in a sparse region. Considering this, when dissimilarity between two instances is measured using ED, we need to proportionally weight it based on the density of the region. We propose our PDM2 as:

$$PDM2\left(x,y\right)=\left(\sum_{i=1}^{d}abs\left(x_i-y_i\right)\times T\left(\left|R_i\left(x,y\right)\right|\right)^p\right)^{\frac{1}{p}} \tag{3.12}$$

where $T\left(\left|R_i\left(x,y\right)\right|\right)$ is the transformation of data mass between x and y in Dimension i of the feature vector, data mass has been used as a proxy of region density. If ED is measured in a denser region, then a higher weight will be assigned to that, while in a sparser region this weight is lower. So, the perceived dissimilarity would be different depending on the density of the region that distances are measured.

The weighting in PDM2 helps to moderate the effect of dominant dimensions in ED calculation. If ED between two instances in a dimension is large but they are located in a sparse region, the low weight of data mass moderates the effect of dominant distance in this dimension.

3.4.2.1 Interaction Effect of Region Density on Euclidean distance

As we can see in equation 3.12, to weight the measured distance, the transformation of data mass has been multiplied into that. In this section, we explain why the effect of region density on the final perceived dissimilarity is multiplicative.

The nature of data distribution effect on final perceived dissimilarity is interactive. The interaction effect is said to exist when the effect of inde-

pendent variable on a dependent variable differs depending on the value of a third variable, called a 'moderator' variable. This effect is not additive, but a multiplicative effect [8, 23, 39, 51]. We define the effect of region density on perceived dissimilarity as an interaction effect, as follows. In our case, dependent variable is the final perceptual dissimilarity, which depends on the ED between two data points. The ED between two data points is an independent variable and data mass (region density) is the moderator variable. The effect of ED on final perceptual dissimilarity will differ depending on the region density. So, an interaction effect exists between ED and region density, which is multiplicative.

Also, multiplication is more robust to outlying values with significantly large/ small ranges than addition, which can be dominated by those outlying values. We can see this varying value ranges in data masses as they can range from the minimum of two (each defined region between two data instances at least covers those two points) to a maximum which is the number of data instances in the dataset.

3.4.2.2 Log Transform of region density

So far, we have discussed the multiplicative effect of the density of a region on the perceived dissimilarity, and proposed to use it as a weight for ED. In our proposed dissimilarity, the ED will be multiplied with data mass. The values of data mass, $|R_i(x, y)|$ in a dataset can range between 2 and N (number of instances in the dataset). The minimum data mass is two as the defined region between two instances enclose the two instances, and in case there is no data point between them, the region includes at least two points. The maximum could be the region that covers all the data points in the dataset, N. So data mass may have a very varied range. As shown in equation 3.12, we consider the effect of region density in each dimension of feature vector, and finally aggregate the dissimilarity in all the dimensions. In this aggregation process, data masses with very large values, which represent denser regions, will dominate the final dissimilarity. In this case, the effect of sparse regions/ low data mass, will not contribute to the final dissimilarity calculation. To consider the region density effectively in our new dissimilarity, we have to use a transformation of data mass that balances the influence of very high and /or very low data mass in some dimensions in the overall dissimilarity.

Logarithmic transformation is an established method to deal with highly skewed data distributions [60, 91, 125]. We will use Logarithmic trans-

formation and Log transformation interchangeably from here after in this thesis. The Log transformation changes a highly skewed data to a distribution closer to normal and draws out the small numbers. As we mentioned we aim to use data mass as weight for ED and using Logarithmic transform balances the contribution of very high and/or very low data mass in some dimensions in the overall dissimilarity. Thus, we used log transformation of $|R_i(x,y)|$ i.e.,$T(|R_i(x,y)|) = Log(|R_i(x,y)|)$. Figure 3.15, shows the Logarithmic transform of data masses presented in Figure 3.14. We can see that not only is the distribution less skewed, but also, the dimensions with low data masses are drawn out in the distribution. Therefore, Logarithmic transformation can serve for both of our purposes in rescaling data masses and also giving all dimensions a balanced contribution in final dissimilarity estimation as a weight.

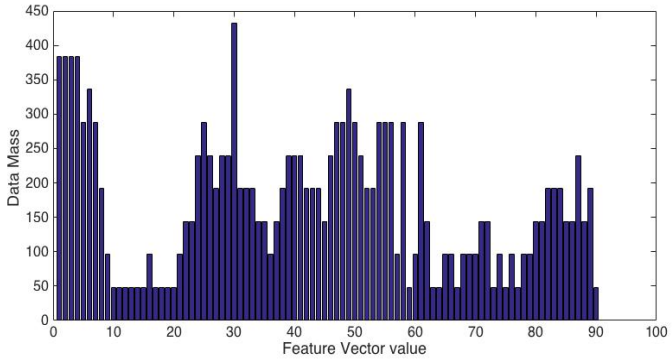


Figure 3.14: Data masses between two feature vectors.

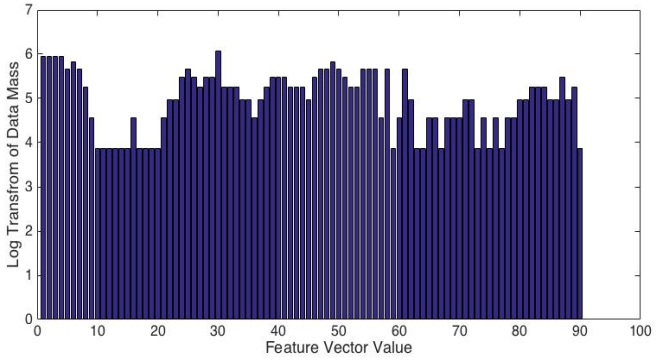


Figure 3.15: Log of data masses between two feature vectors.

3.5 Summary

In this chapter, we have discussed the characteristics of ED, which has been widely used in image retrieval and we highlighted its limitations in detail. Also, we have evaluated the suitability of the mass-dissimilarity measure, m_p [5, 6] for image retrieval and discussed the potential situations where m_p may have undesirable results. As we have explained, both ED and m_p are two dissimilarity measures that contribute to human judgment of similarity from different aspects. Therefore, to overcome their limitations and make use of their strengths, we have proposed a new dissimilarity measure that incorporates both ED and perceptual effect of data distribution. The proposed perceptual dissimilarity measure has two variants, PDM1 and PDM2.

In PDM1 the focus is on improving m_p by proposing a weighting system using ED. PDM2 incorporates the perceptual effect of region density and ED. It weights the ED between two instances in each dimension by the data mass between them. PDM1 and PDM2 are more robust dissimilarity measures compared to ED and m_p alone as they consider the strength of both and address their limitations. In the following chapters we will study the performance of PDM for image retrieval and clustering.

4 Performance Study of Perceptual Dissimilarity Measure for Image Retrieval

In previous chapters, we have discussed the importance of image retrieval in the modern communication era and the essential role of an effective dissimilarity measure in their performance. Based on our study of the literature, we have identified that ED is the dissimilarity measure that is most commonly used in image retrieval. Also, we have discussed the recently proposed mass-based dissimilarity, m_p, to address the limitation of ED with ignoring data distribution. In Chapter 3, we have analysed the characteristics of both ED and m_p to overcome their limitations and have proposed a new dissimilarity measure. The new dissimilarity measure combines data distribution and ED. The proposed perceptual dissimilarity measure, PDM, has two variants: PDM1 and PDM2.

PDM1 focuses on improving m_p, by weighting the data mass with ED where data mass may not work well. PDM2 focuses on improving ED by using data mass as the weight to incorporate perceptual effect of data distribution. In this chapter we aim to improve the image retrieval accuracy using our proposed dissimilarity measures. We study the performance of PDM for image retrieval.

In the following, we begin by explaining our experimental set up, followed by description of using PDM for image retrieval. Then, we will present the empirical results. Section 4.4, compares the performance of PDM1 and m_p, followed by the comparison of PDM2 and ED in Section4.5. The final section will summarise this chapter.

4.1 Experimental setup

To perform image retrieval experiments, we have used the same setup used in Chapter 3, Section 3.3. Three image datasets: eBay, Texture and Corel datasets are used as our benchmark datasets. We have used HSV colour histograms to represent the eBay dataset, LBP for Texture dataset and SIFT BOW for Corel dataset.

In Chapter 3, Section 3.3.4 a weighting system has been introduced for HSV colour histograms. To apply this weighting for PDM1 and PDM2, their equations 3.10 and 3.12 will change to :

$$PDM1(x,y) = \left(\sum_{i=1}^{d} HSVW_i \times PDM1(x,y)^p \right)^{\frac{1}{p}} \tag{4.1}$$

$$PDM2\,(x,y) = \left(\sum_{i=1}^{d} HSVW_i \times (|x_i, y_i| \times T\,(|R_i\,(x,y)|))^p \right)^{\frac{1}{p}} \tag{4.2}$$

p is set to 2 for all the equations in our experiments. We have evaluated the results using precision-recall curves.

4.2 Performance study of the PDM for image retrieval

In this section we will discuss the use of our proposed dissimilarity in image retrieval. In image retrieval, where one image is presented as the query to the system and all the images in the dataset will be ranked and retrieved based on the similarity to the query. In the following section we first discuss the procedure to use PDM as the dissimilarity measure in image retrieval.

4.2.1 How PDM works in image retrieval

This section describes the procedure of using PDM in image retrieval. In Section 3.4.1, we have discussed the four cases that generally occur in calculating the dissimilarity using data distribution as considered in m_p,

where in Cases 2 and 4 m_p may not retrieve accurate results. We have proposed PDM1 to address the m_p's limitations with these two cases by using ED as the weight.

To evaluate the dissimilarity between two images, PDM1 works as follows. Having dataset and query images represented by their feature vectors, first in each dimension of the feature space, PDM1 calculates the ED and data mass between query and all the images in the dataset in all dimensions. Threshold is then defined which is the mid-point between the minimum and maximum of ED/data mass between query and all dataset images. Then, PDM1 checks the distance and data mass between them against the threshold. If both distance and data mass are below or above the threshold, PDM1 will use conventional m_p as the dissimilarity measure. Otherwise, PDM1 will use weighted m_p as defined in equation 3.10 to calculate the dissimilarity between the two images. Finally, it aggregates the dissimilarities in each dimension of feature space to calculate the overall dissimilarity.

PDM2 considers the effect of region density on the perceived dissimilarity by weighting ED between two images in all the dimensions. Unlike PDM1, PDM2 does not define any threshold. PDM2 is calculated using equation 3.12.

4.3 Empirical results

This section presents the empirical results of image retrieval. First, we present the overall retrieval results for each dataset using ED, m_p, PDM1 and PDM2. This will be followed by presenting visual examples from the respected datasets to provide more insight on the performance of the mentioned dissimilarity measures.

4.3.1 Retrieval results of eBay dataset

Figure 4.1 shows the retrieval results of eBay dataset using equations 3.8-3.9 and 4.1- 4.2 to consider our proposed weights for the HSV colour histogram. Figure 4.1, shows that PDM2 performs the best overall for image retrieval. ED performs the second best, followed by PDM1 and then m_p. As visual examples, Figures 4.2-4.3 show the top 10 retrievals for two query images from eBay dataset using PDM2, PDM1, ED and m_p. Non-relevant retrieved images are marked as N-R.

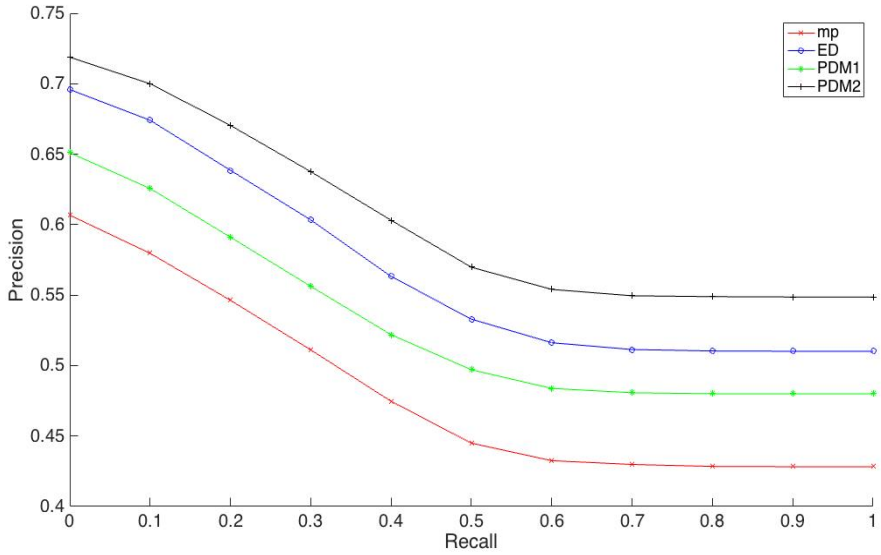
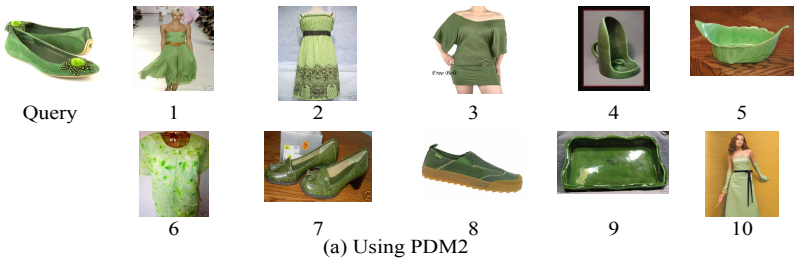


Figure 4.1: Image retrieval results of eBay dataset using ED, m_p, PDM1 and PDM2 as dissimilarity measures.



(a) Using PDM2

(b) Using ED

(c) Using $\boldsymbol{m_p}$

(d) Using PDM1

Figure 4.2: Top 10 retrieval for Query 1 from eBay dataset

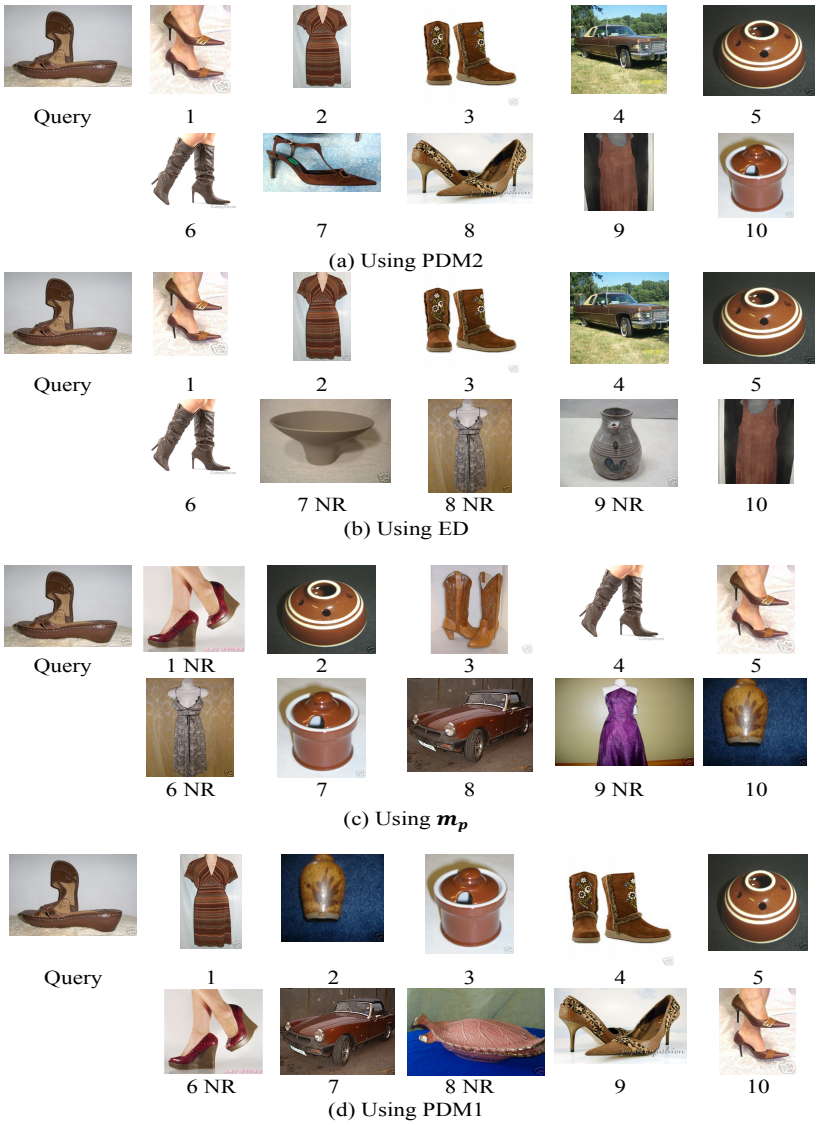


Figure 4.3: Top 10 retrieval for Query 2 from eBay dataset

4.3.2 Retrieval results of Texture dataset

Figure 4.4 shows the overall retrieval results of Texture dataset. It shows that PDM2 has the best performance and PDM1 performs the second best higher than m_p and ED in the Texture dataset. As visual examples, Figures 4.5- 4.6 show the top 10 retrievals for two query images from Texture dataset using the four mentioned dissimilarity measures.

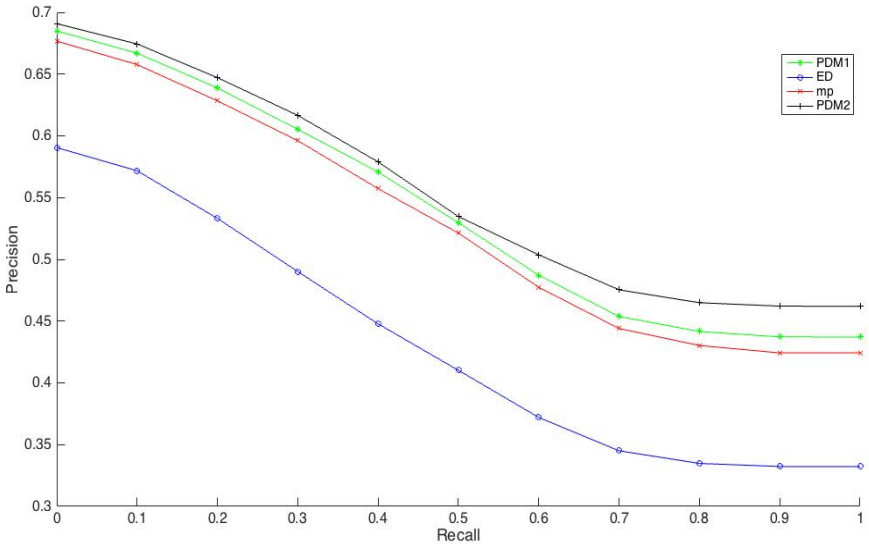


Figure 4.4: Image retrieval results of Texture dataset using ED, m_p, PDM1 and PDM2 as dissimilarity measures.

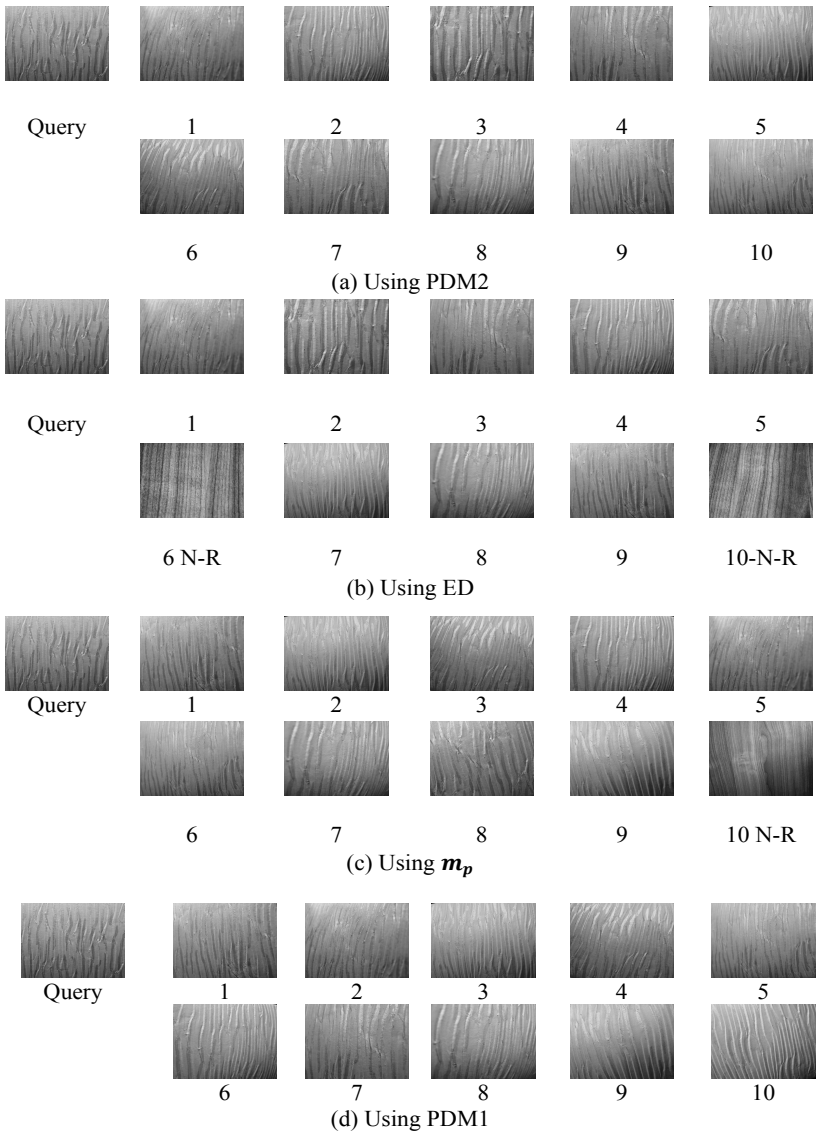


Figure 4.5: Top 10 retrieval for Query 1 from Texture dataset

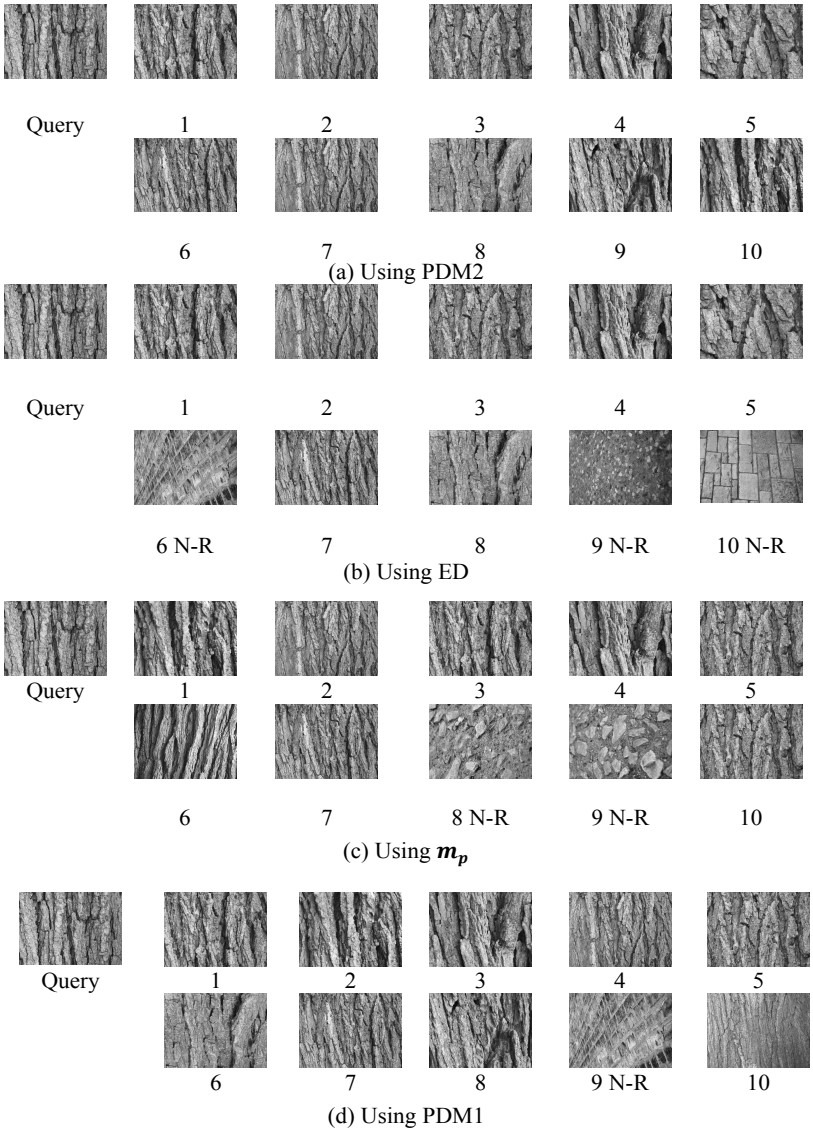


Figure 4.6: Top 10 retrieval for Query 2 from Texture dataset

4.3.3 Retrieval results of Corel dataset

In a similar trend for Corel dataset, Figure 4.7 shows that PDM2 has the best performance along with PDM1 higher than ED and m_p. As visual examples, Figures 4.8- 4.9 show the top 10 retrievals for two query images from Corel dataset using above mentioned dissimilarity measures.

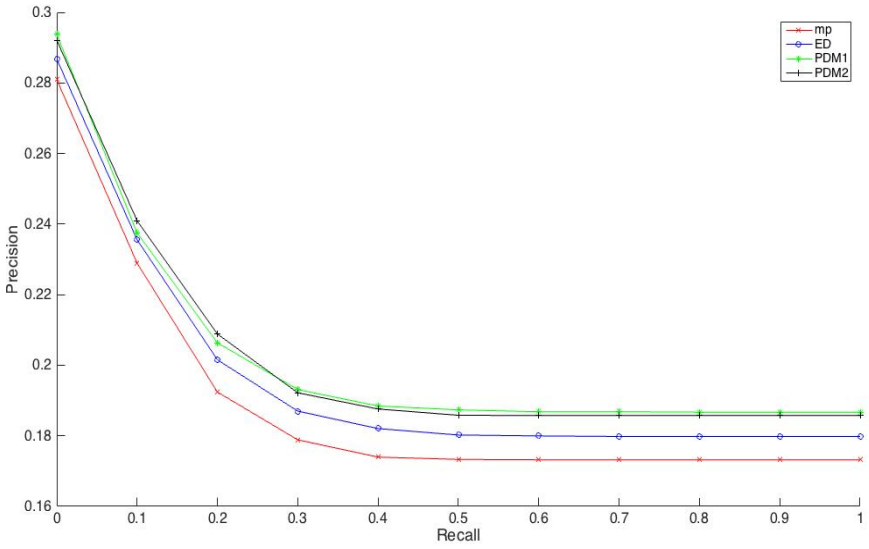
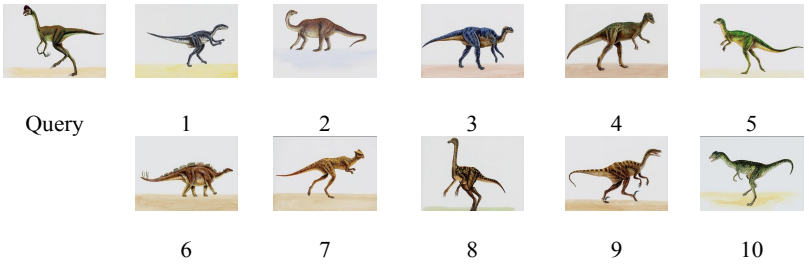
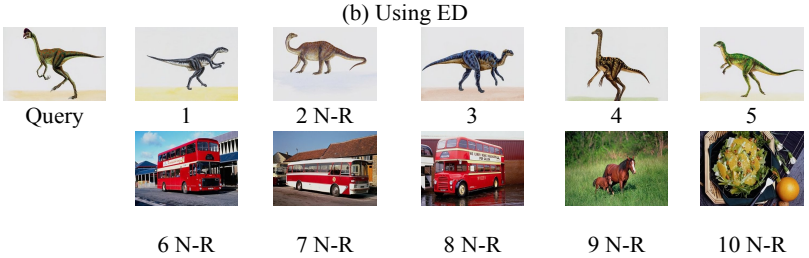


Figure 4.7: Image retrieval results of Corel dataset using ED, m_p, PDM1 and PDM2 as dissimilarity measures.

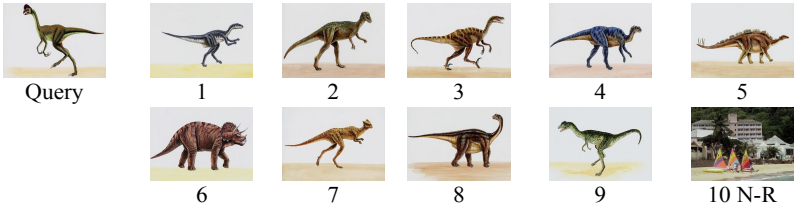


(a) Using PDM2

(b) Using ED



(c) Using $\boldsymbol{m_p}$



(d) Using PDM1

Figure 4.8: Top 10 retrieval for Query 1 from Corel dataset

Figure 4.9: Top 10 retrieval for Query 2 from Corel dataset

In the next three sections, we will provide a more in-depth comparison and discussion on the performance of PDM1 versus m_p, as well as the performance of PDM2 versus ED and PDM1 versus PDM2. We will also explain why PDM1 performs worse than ED in some cases.

4.4 Performance comparison between PDM1 and m_p

PDM1 is proposed to address the limitations of m_p. So in this section, we study the performance of PDM1 and m_p using visual examples of each dataset. The next three sections present our performance study of PDM1 and m_p for each dataset.

To analyse the performance of PDM1, we select some of the dimensions as examples to illustrate Cases 2 and 4 through data mass and distance in those dimensions. For example if in a dimension of feature vectors data mass was high but distance is small between two images (Case 4), we will then illustrate how PDM1 uses the proposed weighting to moderate the dissimilarity in that dimension and address the respected Case of 2 or 4. As the PDM1 combines the dissimilarity over all dimensions, considering this effect in multiple dimensions explains its performance.

In our visual examples, we show the components that PDM1 uses for dissimilarity calculations. PDM1 uses data mass between feature vectors of two images, along with their distance to calculate the dissimilarity. We present these components for all the dimensions of feature vectors.

4.4.1 Performance comparison between PDM1 and m_p for eBay dataset

This section presents our performance study of PDM1 and m_p for the eBay dataset. Figure 4.1 shows that PDM1 preforms better than m_p. PDM1 has improved the retrieval results by incorporating ED with region density where m_p may not be able to retrieve accurate results. We have used weighted m_p as in Equation 3.10, to address the limitations in Cases 2 and 4.

In the following examples, some dimensions of feature space are used to show Cases 2 and 4, where m_p retrieved a Non-relevant image. We show the effect of PDM1 with using ED as the weight for m_p in those dimensions that improves the situation. Considering this improvement in multiple dimension results in a retrieval of a relevant image.

In Figure 4.2 (c), m_p ranks a relevant image, which appears in the fifth rank of PDM1, lower than 10. It determines that the relevant image: the green pot, is more dissimilar to the query due to the high data mass between

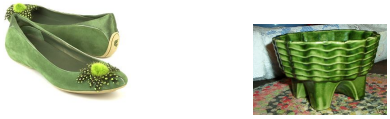
them compared to the white shoe which is in its fifth rank. Relying only on the data distribution of features results in this situation as explained in Chapter 3, Section 3.4.1.

As an example, it is shown in Figures 4.10 (c-d) that in Dimension 6 of features, the distance between the green shoe and the green pot is considerably smaller compared to the white shoe. However the data mass between the green shoe and the green pot is much higher compared to green and white shoes. So, in this situation two green images are found more dissimilar compared to the green and the white shoe images by m_p in that dimension. It is shown in Figure 4.10 (h) that PDM1 weights the data mass by distance in Dimension 6 (where the data mass between the green shoe and the green pot is above the threshold, while their distance is below the threshold). In PDM1, weighting the data mass by distance results in finding the green shoe and the green pot more similar in Dimension 6 compared to white shoe.

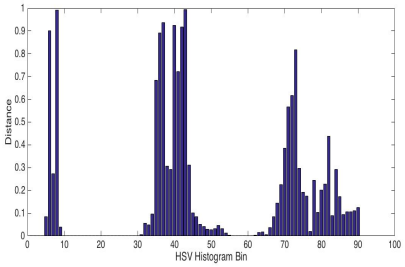
Also, PDM1 moderates the effect of dominant dimension in calculating the dissimilarity by m_p where a dimension with very high data mass could influence the total dissimilarity between the green shoe and the green pot. As in this example in Dimension 6 of feature vector, the high data mass between the green shoe and the relevant image (green pot) could dominate the lower data masses in other dimensions. PDM1 moderates this effect by assigning a lower weight to the high data mass in that dimension. PDM1 has calculated the dissimilarity of 1149.22 between the green shoe and the green pot, it ranked the green pot as fifth rank while the white shoe with dissimilarity of 1398.74 has been ranked much lower as 58th.

Figure 4.11 is another example where we can follow the same scenario in Dimension 3 of Figures 4.11 (c, e and g) and Dimension 80 of Figures 4.11 (d, f and h).

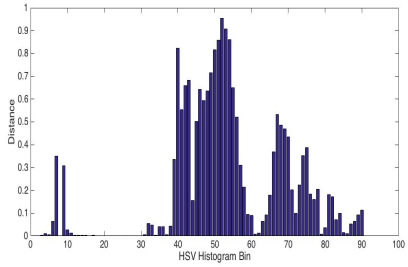
(a) Query (left) and Rank 5^{th} from m_p (right)



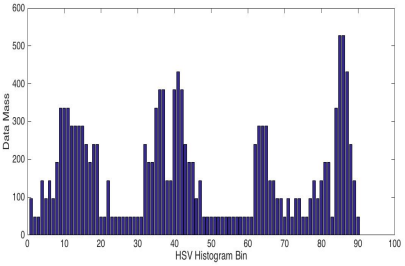
(b) Query (left) and Rank 5^{th} from PDM1 (right)



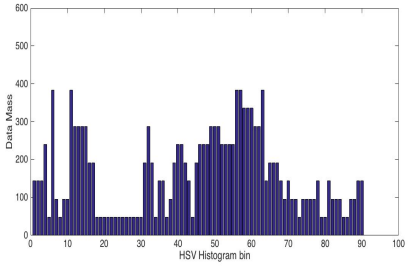
(c) Distance (ED) between images in (a)



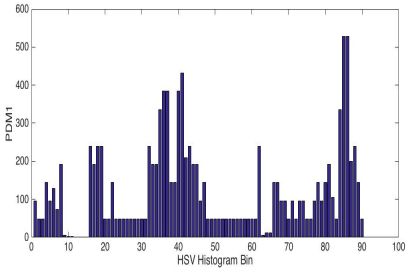
(d) Distance (ED) between images in (b)



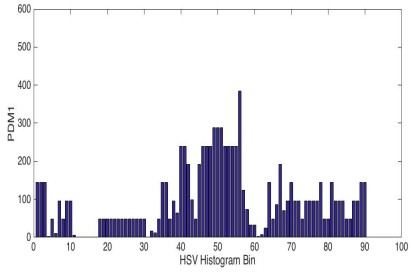
(e) Data mass between images in (a)



(f) Data mass between images in (b)

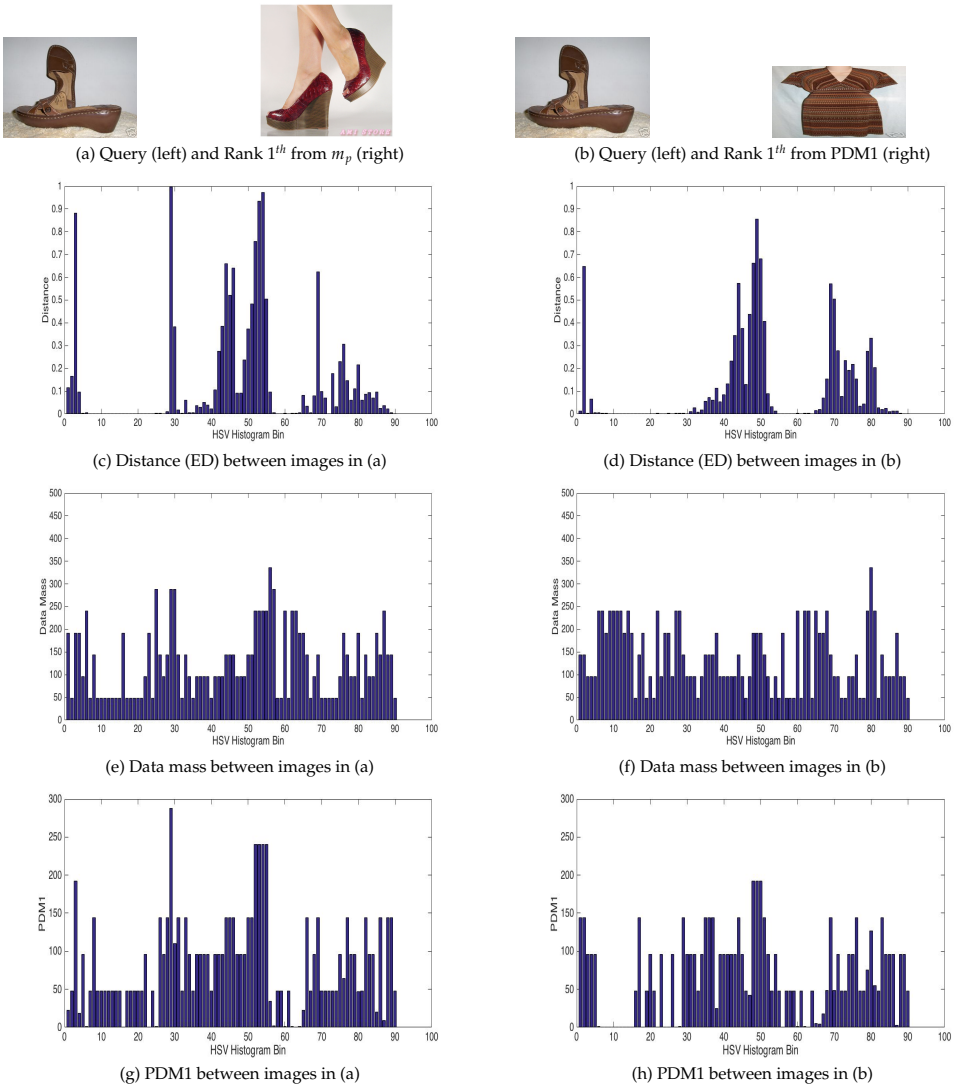


(g) PDM1 between images in (a)



(h) PDM1 between images in (b)

Figure 4.10: Comparison of PDM1 and m_p using visual examples of eBay dataset



(a) Query (left) and Rank 1^{th} from m_p (right)

(b) Query (left) and Rank 1^{th} from PDM1 (right)

(c) Distance (ED) between images in (a)

(d) Distance (ED) between images in (b)

(e) Data mass between images in (a)

(f) Data mass between images in (b)

(g) PDM1 between images in (a)

(h) PDM1 between images in (b)

Figure 4.11: Comparison of PDM1 and m_p using visual examples of eBay dataset

Using HSV colour histogram and eBay dataset, PDM1 has not performed better than ED.

The defined threshold may raise a potential limitation, as the points just below and above the mid point will be considered as low/high data mass or small/large distance. However, these border points are very close and their difference does not represent the actual difference between low and high data mass or small and large distance. PDM1 considers a border point just below the threshold as located in a low data mass area (sparse

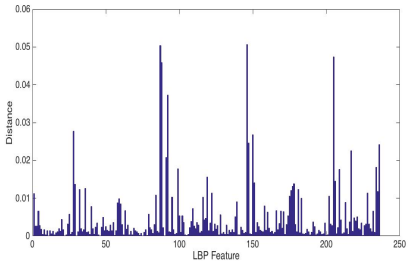
region) and the one just above the threshold as located in a dense area, however, they have very similar data masses. The same situation is for distance values.

4.4.2 Performance comparison between PDM1 and m_p for Texture dataset

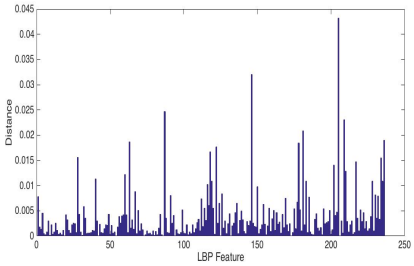
This section presents our performance study of PDM1 and m_p for Texture dataset using the visual examples.

(a) Query (left) and Rank 10^{th} from m_p (right)

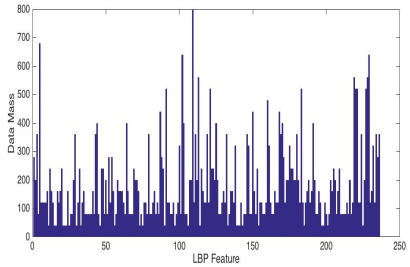
(b) Query (left) and Rank 10^{th} from PDM1 (right)



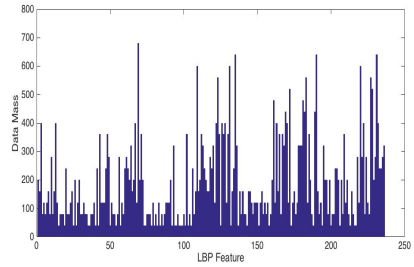
(c) Distance (ED) between images in (a)



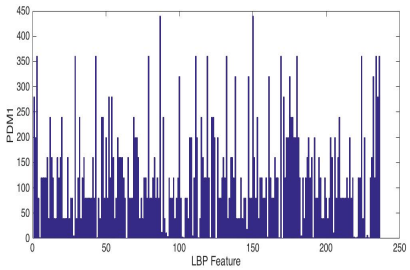
(d) Distance (ED) between images in (b)



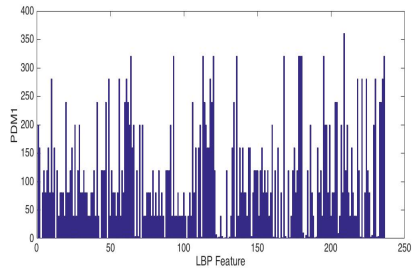
(e) Data mass between images in (a)



(f) Data mass between images in (b)



(g) PDM1 between images in (a)



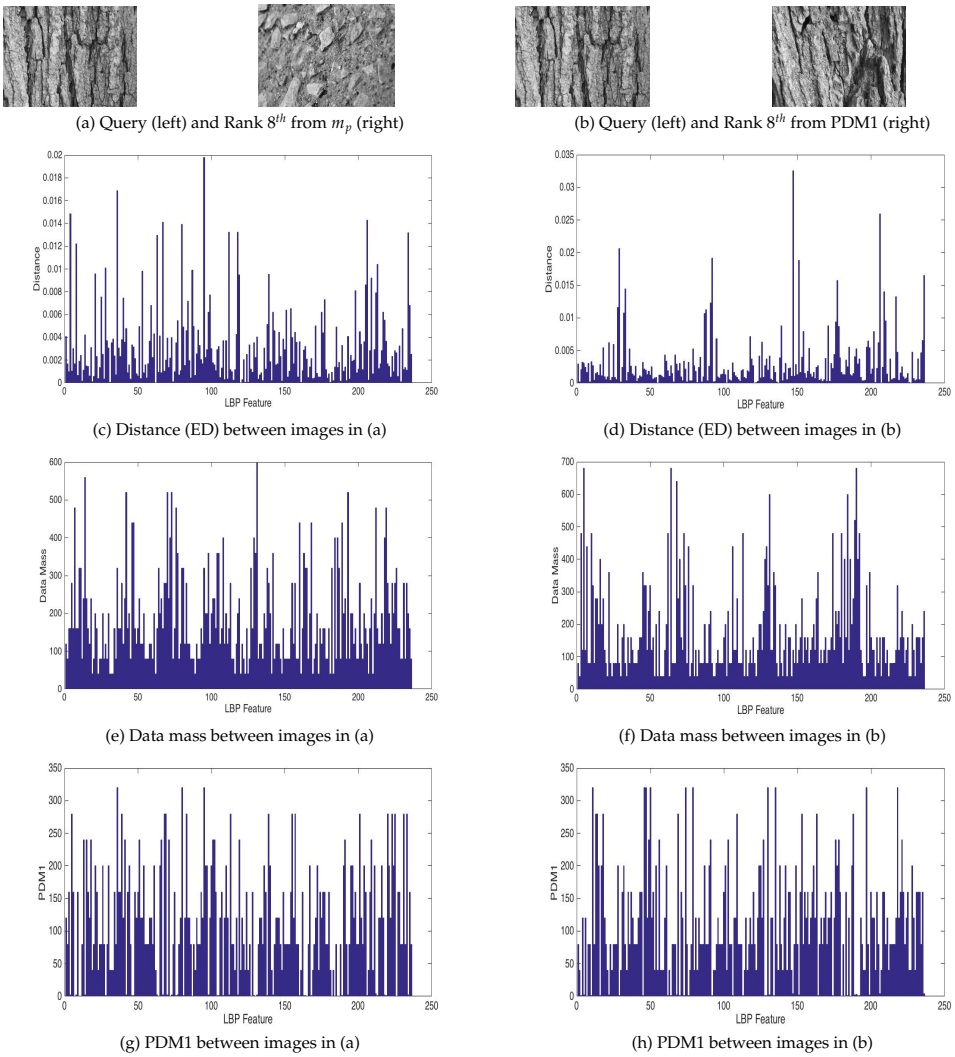
(h) PDM1 between images in (b)

Figure 4.12: Comparison of PDM1 and m_p using visual examples of Texture dataset

In Figure 4.4, m_p ranks a relevant image, which appears in the 10th rank of PDM1, lower than 10. It determines that the relevant image: the wallpaper, is more dissimilar to the query due to the high data mass between them compared to the wood which is in its 10th rank. As an example, it is shown in Figures 4.12 (d, f) that in Dimension 109 of features, the data mass between wallpaper and the wallpaper in PDM1's 10th rank is high, at 600, and above the defined threshold, while the distance between them is very small, 0.002 and below the threshold (Case 4). PDM1 will adjust the high data mass by weighting it accordingly with distance. It is shown in Figure 4.12 (h) that PDM1 weights the data mass by distance in Dimension 109. Using PDM1, the weighting of data mass by distance results in finding the two wallpapers more similar in Dimension 109 compared to the measurement from m_p. Considering this moderation in multiple dimensions ranks the non-relevant image: wood, lower than 10 in the retrievals of PDM1.

Additionally, we can see that PDM1 moderated the effect of dominant dimension in calculation of m_p. This effect is shown in the discussed example where data mass in Dimension 109 between two wallpapers in Figures 4.12 (d, f and h) was very high. This could results in a situation where other dimensions with lower data masses do not influence the overall dissimilarity and the two wallpaper images are found dissimilar by m_p. PDM1 moderates the effect of very high data mass by weighting it using the distance.

Following the same scenario, Figure 4.13 is another example where m_p may not retrieve an accurate result. We can see the above described scenario in Dimension 5 of Figures 4.13 (d, f and h) and Dimension 112 of Figures 4.13 (c, e and g).



(a) Query (left) and Rank 8^{th} from m_p (right)

(b) Query (left) and Rank 8^{th} from PDM1 (right)

(c) Distance (ED) between images in (a)

(d) Distance (ED) between images in (b)

(e) Data mass between images in (a)

(f) Data mass between images in (b)

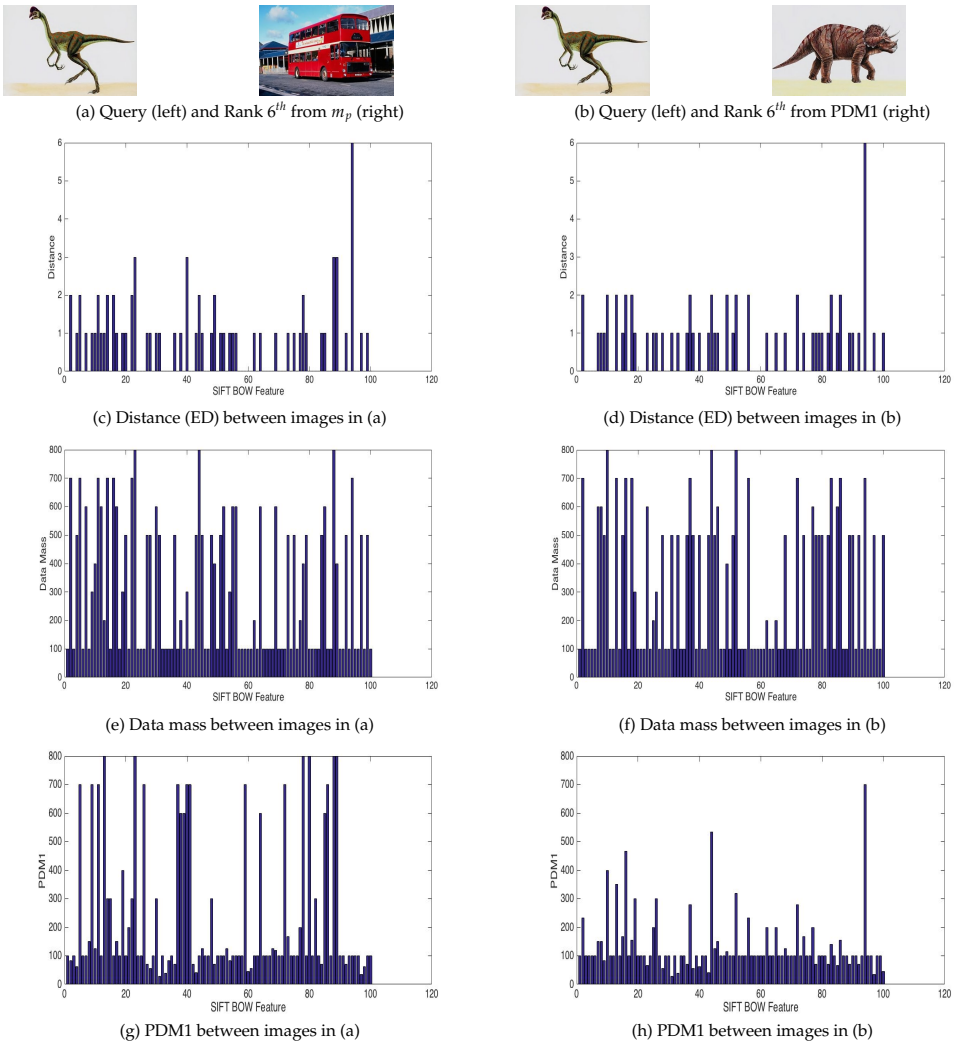
(g) PDM1 between images in (a)

(h) PDM1 between images in (b)

Figure 4.13: Comparison of PDM1 and m_p using visual examples of Texture dataset

4.4.3 Performance comparison between PDM1 and m_p for Corel dataset

This section presents our performance study of PDM1 and m_p for Corel dataset. Along with previous visual examples, in the following two examples from Corel dataset retrieval are presented in Figures 4.8 and 4.9.



(a) Query (left) and Rank 6^{th} from m_p (right)

(b) Query (left) and Rank 6^{th} from PDM1 (right)

(c) Distance (ED) between images in (a)

(d) Distance (ED) between images in (b)

(e) Data mass between images in (a)

(f) Data mass between images in (b)

(g) PDM1 between images in (a)

(h) PDM1 between images in (b)

Figure 4.14: Comparison of PDM1 and m_p using visual examples of Corel dataset

In Figure 4.8 (c), m_p ranks a non-relevant image in its sixth rank while the relevant image which appeared in the sixth rank of PDM1, is ranked lower than 10 in m_p's retrievals. m_p determines that the relevant image: the dinosaur, is more dissimilar to the query due to the high data mass between them compared to the bus which appeared in its sixth rank. The dissimilarity measurement of m_p is in conflict with human perception. As an example, it is shown in Figures 4.14 (d, f) that in Dimension 2 of features, data mass between the query and dinosaur in PDM1's sixth rank is high, at 700, and above the defined threshold which makes the relevant image more dissimilar to the query in this dimension. However, the dis-

tance between them is 2 (maximum distance in that dimension is 6), which is below the threshold. So PDM1 will adjust the high data mass by weighting it accordingly with distance. It is shown in Figure 4.14 (h) that PDM1 weights the data mass by distance in Dimension 2. Using PDM1, wighting the data mass by distance in multiple dimensions results in finding the two dinosaurs more similar compared to the bus in retrieval results.

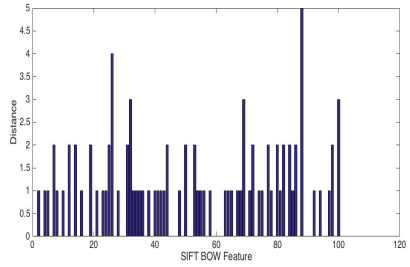
Also we can see in Figures 4.14 (c, e) in Dimension 85 of features, data mass between query and the bus is below the threshold, at 300, where the distance between them is as high as meeting the defined threshold. The low data mass results in situation where m_p found the non-relevant image similar to the query in this dimension. PDM1 weighted the data mass accordingly with distance and data mass changed to 600.

Following our discussion from previous section, we can see that the data masses in Dimension 2 of Figure 4.14 (f), could influence the total dissimilarity by m_p. This has been moderated in PDM1, Figure 4.14 (h) by weighting it using the distance.

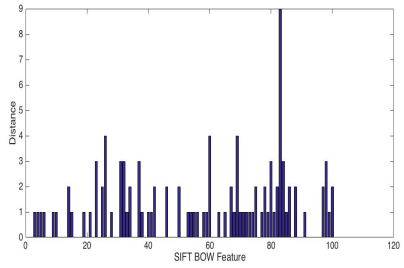
Figure 4.15 presents another example in Dimension 3 of Figures 4.15 (d, f and h) and Dimension 88 of Figures 4.15 (c, e and g).

(a) Query (left) and Rank 2^{th} from m_p (right)

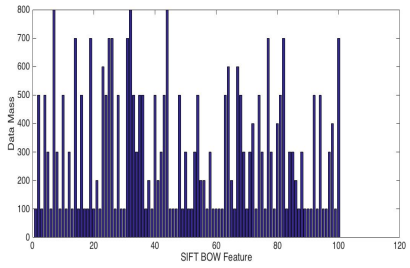
(b) Query (left) and Rank 2^{th} from PDM1 (right)



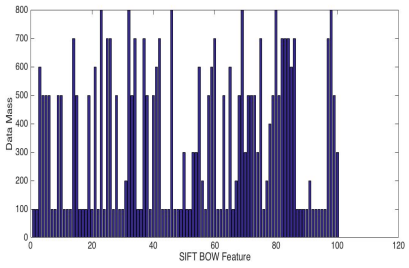
(c) Distance (ED) between images in (a)



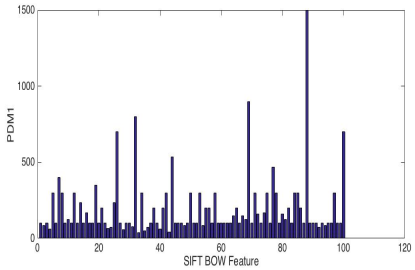
(d) Distance (ED) between images in (b)



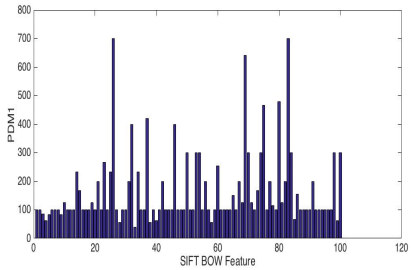
(e) Data mass between images in (a)



(f) Data mass between images in (b)



(g) PDM1 between images in (a)



(h) PDM1 between images in (b)

Figure 4.15: Comparison of PDM1 and m_p using visual examples of Corel dataset

4.5 Performance comparison between PDM2 and ED

PDM2 basis for calculation of dissimilarity is ED. PDM2 weights the ED between two images in each dimension of the feature space with the density of the region covering them. To study the effect of PDM2 on improvement of image retrieval accuracy we need to compare its performance with its base method, ED.

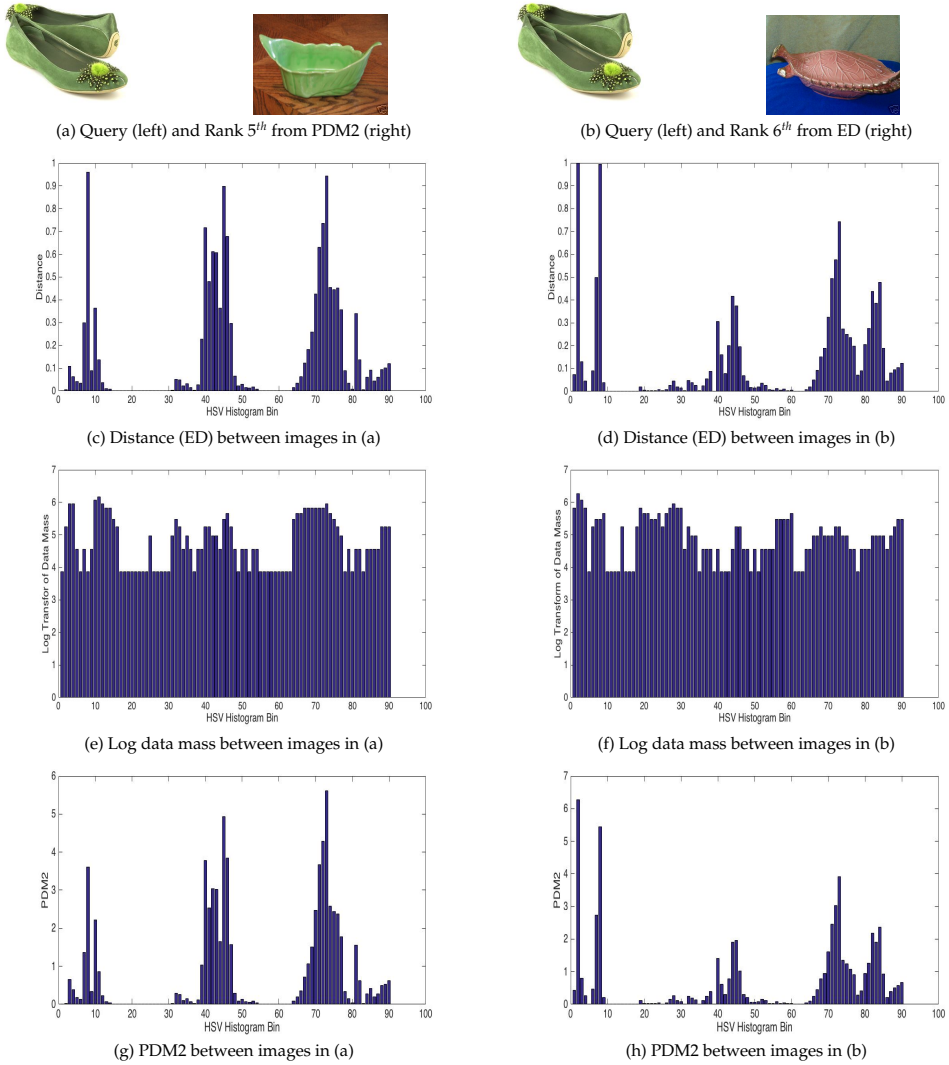
In this section we discuss the performance of ED and PDM2, through visual examples in Figures 4.2- 4.9. To discuss the performance of PDM2, we show the components that it uses for calculating dissimilarity between two instances, which are distance and Log transformation of data mass. The basis for calculation of PDM2 is ED, which is moderated by region density as the weight. We show PDM2 in each dimension, to illustrate how region density could moderate the ED between two images in the feature space.

4.5.1 Performance comparison between PDM2 and ED for eBay dataset

This section presents the performance study of PDM2 and ED for eBay dataset. As shown in Figures 4.2 and 4.3, PDM2 could retrieve all images from the same class with query compared to ED and m_p and PDM1. For example, in Figure 4.2 (a) and using PDM2 all the top 10 retrieved images are from the class of green, the same colour with the query, however in Figure 4.2 (b-d), ED and m_p and PDM1 retrieved images from other colours such as blue, pink and yellow. Figure 4.3 (a) also shows that top 10 retrieval using PDM2 are from the same class with the query which is brown colour, compared to Figures 4.3 (b-d) that have more retrievals from other colours such as, grey, red, purple and pink.

PDM2 improved retrieval results by considering both ED and the effect of region density in estimating the final dissimilarity between two images. Similar to the previous section, in the following examples we select a few dimensions from the feature vectors of presented images to show how relying only on distance could result in undesirable retrievals. Also, we show how using log data mass between two images as a proxy of region density to weight the distance could improve the situation in those dimensions. Finally, considering this effect in multiple dimensions resulted in more accurate retrievals.

In Figure 4.2 (b), ED has ranked the green pot as a relevant image in its seventh rank lower than a pink pot as a non-relevant image. This occurred as ED relies only on the distance between colour histogram of query and these two images and does not consider the data distribution. The green pot has been ranked higher in retrievals of PDM2 in Figure 4.2 (a) and the pink pot ranked lower than 10. Unlike ED, PDM2 considers if two instances are located in a dense/ sparse region, then their distance will be perceived differently.



(a) Query (left) and Rank 5^{th} from PDM2 (right)

(b) Query (left) and Rank 6^{th} from ED (right)

(c) Distance (ED) between images in (a)

(d) Distance (ED) between images in (b)

(e) Log data mass between images in (a)

(f) Log data mass between images in (b)

(g) PDM2 between images in (a)

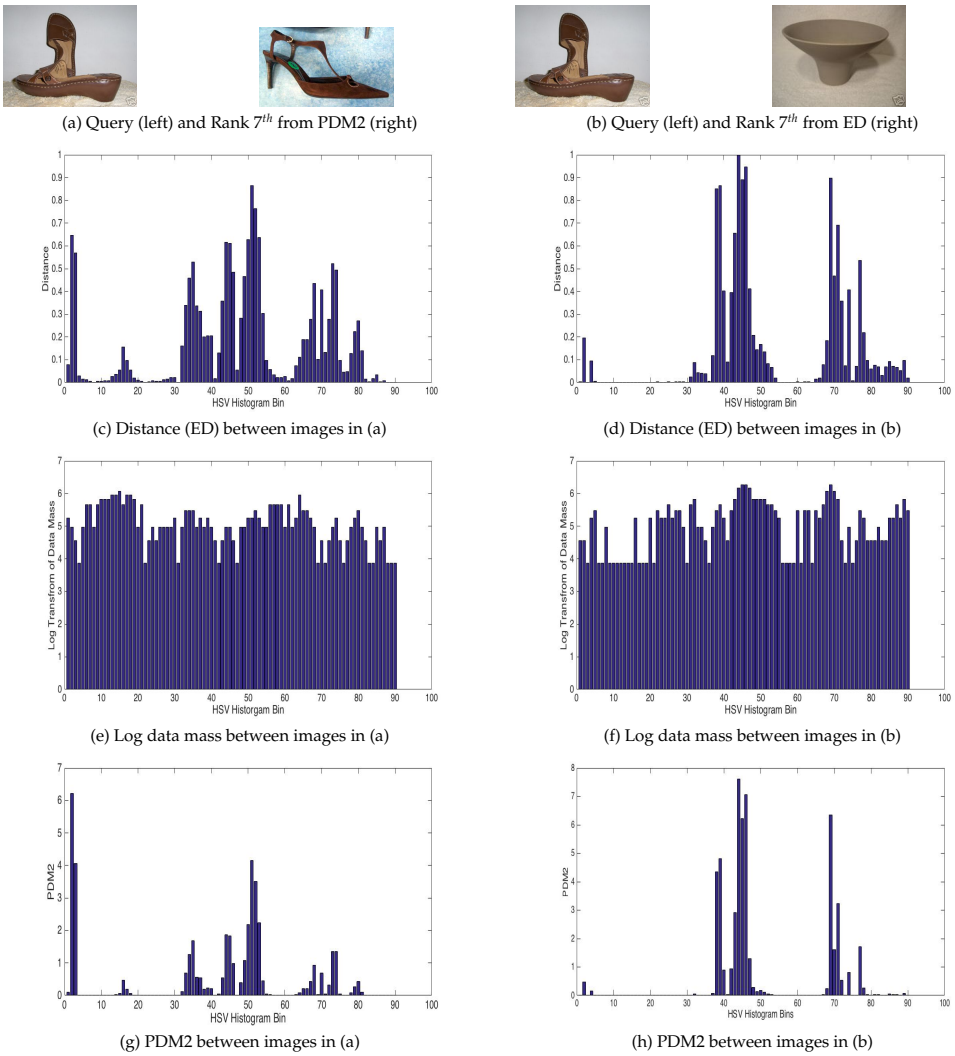
(h) PDM2 between images in (b)

Figure 4.16: Comparison of PDM2 and ED using visual examples of eBay dataset

Here we use examples in some dimensions of feature vectors to show how PDM2 is calculated and improves the retrieval results. Figures 4.16 (c-d) show that the distance calculated between the Dimension 8 of the query (green shoe), the green and the pink pots are very similar, at 0.96 and 0.99. However, in this dimension the distance between the query and the green pot has been measured where two points are located in a denser region as it is shown in Figure 4.16 (f), compared to the query and the pink pot, Figure 4.16 (e), where they are located in a sparser region. The Logarithmic transformation of data mass between the green shoe and the green pot in

Dimension 8 is 3.8, while it is 5.4 for the pink pot. Using the Logarithmic transformation of data mass as the proxy for region density assigns higher weight to the distance of the query and the pink pot, which is measured in a denser area and vice versa. PDM2 in Figures 4.16 (g-h) shows the weighted distances by the region density. The weighted distances in Dimension 8 between the green shoe, the green and pink pots are 3.6 and 5.34. Although they have similar distances, PDM2 weighted their distance and as a result pink pot in Dimension 8 is much more dissimilar to the query compared to the green pot.

As we mentioned in the previous chapter, the second limitation with ED is equal priority in aggregating the distances of all dimensions, which results in a situation where a few dimensions with large distances can dominate the others and influence the total dissimilarity. The weighting in PDM2 helped to moderate the effect of dominant dimensions in calculating the distance. For example in Dimension 8, the large distance between the green shoe and the green pot could influence the total distance. Considering this weighting in multiple dimensions moderate the distances and resulted in the dissimilarity of 7.3 between the green shoe and the green pot versus the dissimilarity of 7.8 between the green shoe and the pink pot. PDM2 ranked the pink pot lower than 10 in its retrievals. Figure 4.17 is another example that illustrates the same scenario.



(a) Query (left) and Rank 7^{th} from PDM2 (right)

(b) Query (left) and Rank 7^{th} from ED (right)

(c) Distance (ED) between images in (a)

(d) Distance (ED) between images in (b)

(e) Log data mass between images in (a)

(f) Log data mass between images in (b)

(g) PDM2 between images in (a)

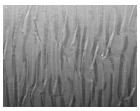
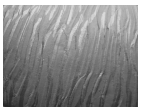
(h) PDM2 between images in (b)

Figure 4.17: Comparison of PDM2 and ED using visual examples of eBay dataset

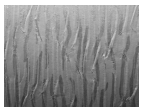
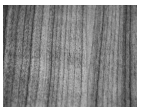
4.5.2 Performance comparison PDM2 and ED for Texture dataset

This section presents the performance study of PDM2 and ED for Texture dataset. Similar to the previous section, we compare the performance of PDM2 and ED using the visual examples of Texture dataset in Figures 4.5 and 4.6.

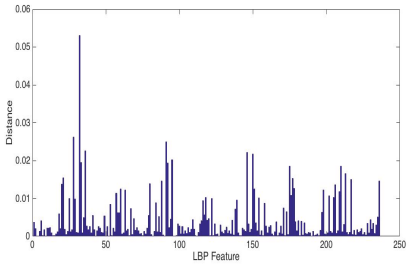
As shown in Figure 4.5, PDM2 and PDM1 as the dissimilarity measures

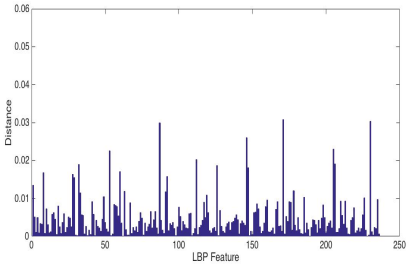
(a) Query (left) and Rank 6^{th} from PDM2 (right)

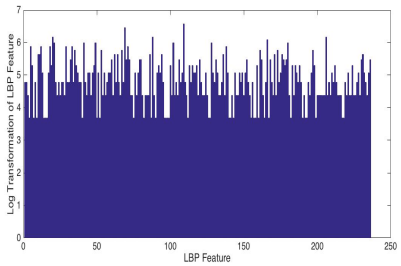
(b) Query (left) and Rank 6^{th} from ED (right)



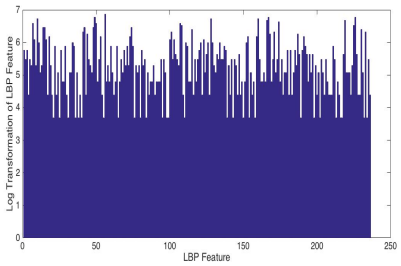
(c) Distance (ED) between images in (a)



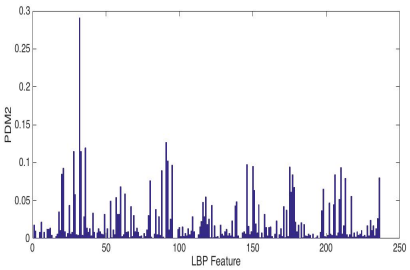
(d) Distance (ED) between images in (b)



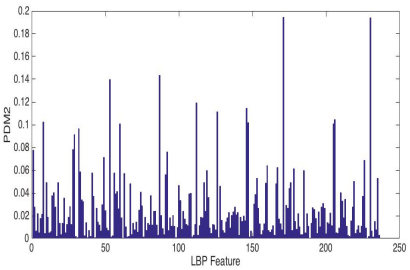
(e) Log data mass between images in (a)



(f) Log data mass between images in (b)



(g) PDM2 between images in (a)



(h) PDM2 between images in (b)

Figure 4.18: Comparison of PDM2 and ED using visual examples of Texture dataset

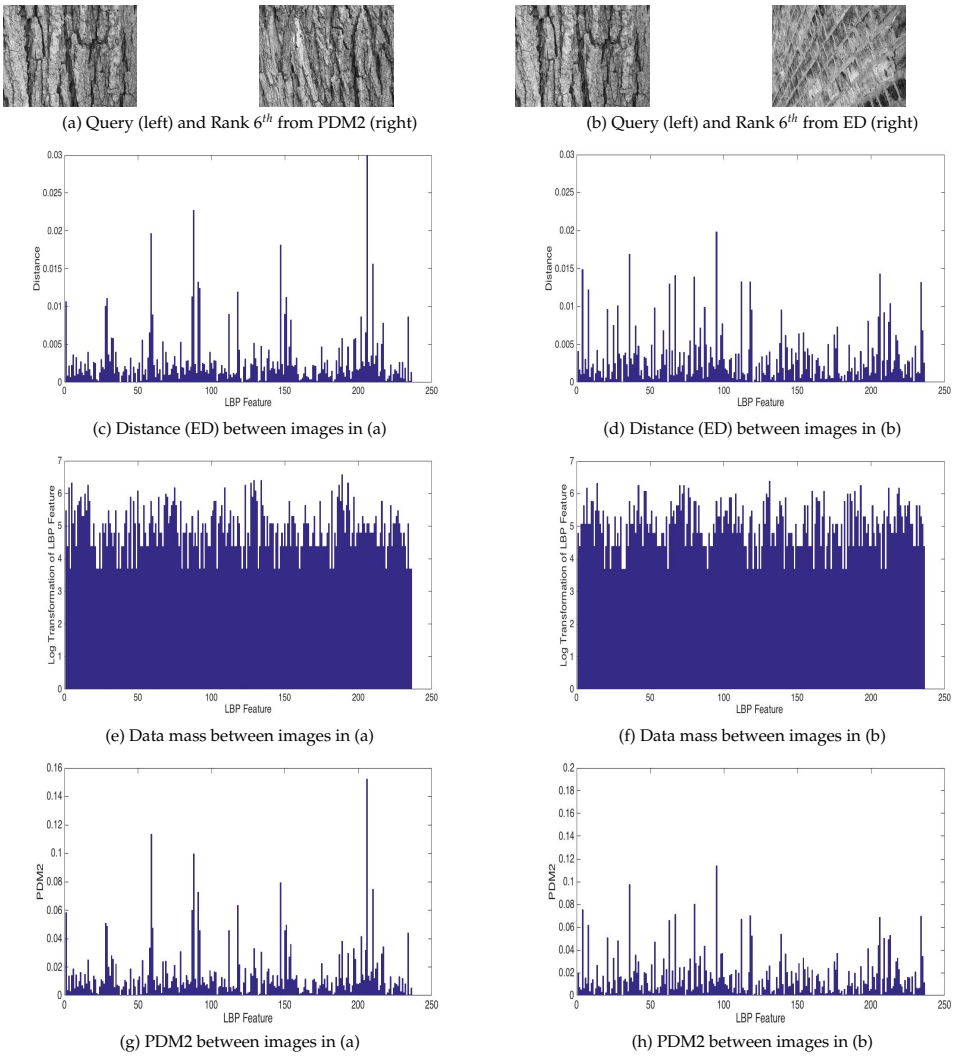
could retrieve all images from the same class with query compared to ED and m_p. In Figure 4.5 (a) using PDM2 all the top 10 retrieved images are from the class of wallpaper, however, in Figures 4.5 (b, d), ED and m_p retrieved images from another class: wood. In the following examples, we discuss how PDM2 could improve the retrievals compared to ED.

In Figure 4.5 (b), ED ranked the wood, a non-relevant image, in its sixth rank higher than a wallpaper (relevant image). This occurred as ED relies only on geometric position of query and these two images in the feature space and does not consider the data distribution. However PDM2 could retrieve a relevant image in its sixth rank in Figure 4.5 (a) and ranked the wood lower than 10.

Figures 4.18 (c, e) show that the distance between the query and the wallpaper in Dimension 91 is much higher, at 0.02, compared to many other dimensions. This makes these two images more dissimilar in this dimension measured by ED. However in this dimension of feature space these two instances are located in a sparser area where data mass is only 160. The Log of data mass between two wallpapers in Dimension 91 is 5.7 as it is shown in Figure 4.18 (e). Using the Log transformation of data mass as the proxy for region density, it assigned lower weight to the distance, which is measured in a sparser area and vice versa. PDM2 in Figures 4.18 (g-h) show the weighted distances, where ED is moderated by the region density.

In Figure 4.18 (d) we can see that in Dimension 107 the distance between the wallpaper and wood (non-relevant image) is as small as 0.006. However, the distance has been measured in a dense area where the data mass between the two points is 720. The Log of the data mass is 6.5 as shown in Figure 4.18 (f). PDM2 assigned a higher weight to the distance measured in denser area and resulted in moderated distance of 0.03. Considering this weighting in other dimensions that moderate the distances and resulted in smaller dissimilarity between the query and wallpaper compared to the wood, PDM2 ranked the wood lower than 10 in its retrievals.

Also, PDM2 could moderate effect of dominant dimension in calculation of ED through the discussed examples. As in Dimension 91. the large distance could influence on total dissimilarity measured by ED. This effect has been moderated in PDM2 using the relatively low data mass as the weight for the distance in this dimension.



(a) Query (left) and Rank 6^{th} from PDM2 (right)

(b) Query (left) and Rank 6^{th} from ED (right)

(c) Distance (ED) between images in (a)

(d) Distance (ED) between images in (b)

(e) Data mass between images in (a)

(f) Data mass between images in (b)

(g) PDM2 between images in (a)

(h) PDM2 between images in (b)

Figure 4.19: Comparison of PDM2 and ED using visual examples of Texture dataset

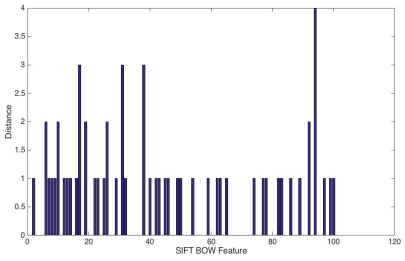
Figure 4.19 is another examples of a similar scenario where Figures (c, e and g) in Dimension 206 show the large distance between the query and a relevant image from the same class. However, they are located in a sparse area and PDM2 uses the region density to weight and moderate the large distance. Figures4.19 (d), in Dimension 131 show the small distance between the query and a non-relevant image, 0.0004, while being located in a dense area (Log data mass of 6.39), PDM2 could moderate the dissimilarity between the two images in this dimension.

4.5.3 Performance comparison between PDM2 and ED for Corel dataset

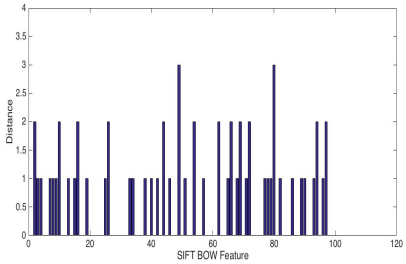
This section presents the performance study of PDM2 and ED for Corel dataset. In a similar trend the following examples are selected from Corel dataset as shown in Figures 4.8 and 4.9.

(a) Query (left) and Rank 8^{th} from PDM2 (right)

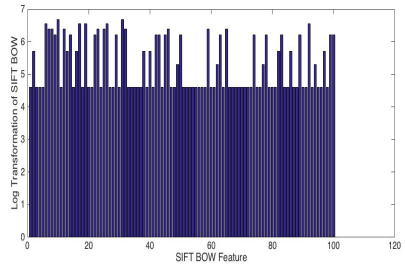
(b) Query (left) and Rank 8^{th} from ED (right)



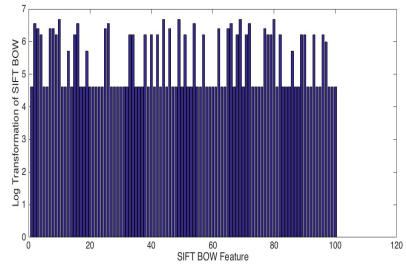
(c) Distance (ED) between images in (a)



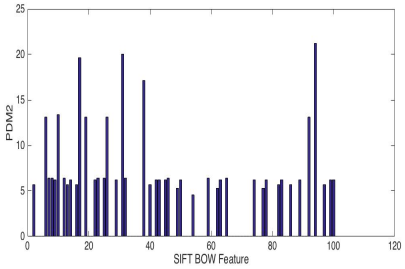
(d) Distance (ED) between images in (b)



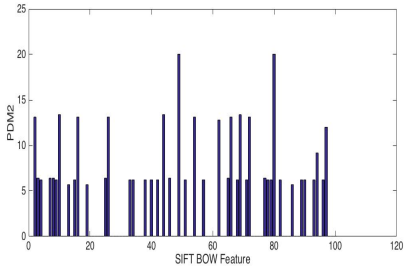
(e) Log data mass between images in (a)



(f) Log data mass between images in (b)



(g) PDM2 between images in (a)



(h) PDM2 between images in (b)

Figure 4.20: Comparison of PDM2 and ED using visual examples of Corel dataset

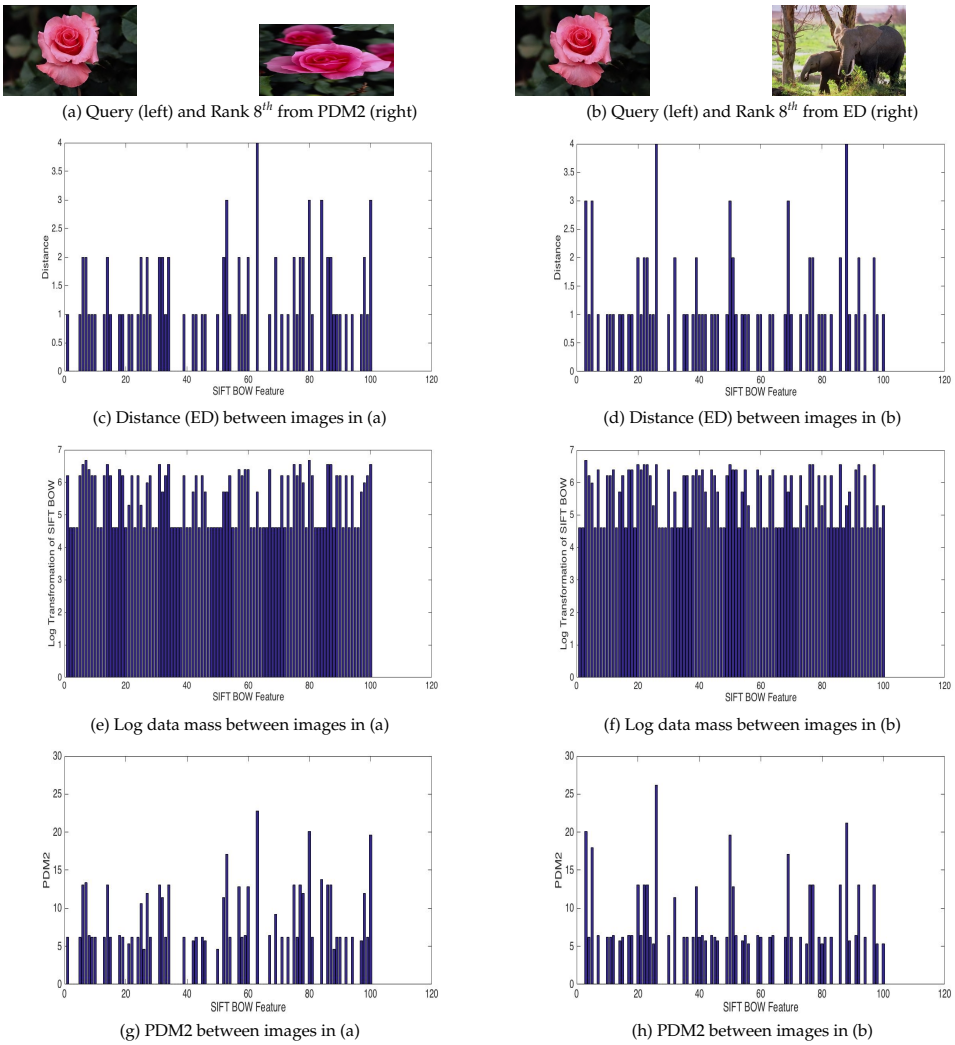
As shown in Figures 4.8 and 4.9, PDM2 performed better than ED, m_p and PDM1 in retrieving relevant images. In Figure 4.8 (a) using PDM2 all the top 10 retrieved images are from the class of dinosaur; however, in

Figure 4.8 (b-d), ED, m_p and PDM1 retrieved images from other classes. In the following examples, we discuss improvements made by PDM2, in the retrievals compared to ED.

In Figure 4.8 (b), ED ranked a beach, a non-relevant image in its eighth rank higher than a relevant image (dinosaur). However, PDM2 could retrieve a relevant image in its eighth rank in Figure 4.8 (a) and ranked the beach lower than 10.

Figures 4.20 (c, e) show that the distance between the query and the dinosaur in Dimension 41 is 4, which is relatively large. However, they are located in a sparser area with Log of data mass of 5.2. PDM2 in Figures 4.20 (g-h) shows the weighted distances, where PDM2 moderated the large distance using the low region density.

In Figure 4.20 (d) we can see that in Dimension 2 the distance between the query and the beach is as small as 1. However, the distance has been measured in a relatively denser area where the data mass between the two points is 600. The Log of the data mass is 6.39 as shown in Figure 4.20 (f). PDM2 assigned a higher weight to the distance measured in denser area and resulted in a moderated distance of 6.39. Applying this weighting in other dimensions that moderates the distances resulted in smaller dissimilarity between the query and the dinosaur compared to the beach.



(a) Query (left) and Rank 8^{th} from PDM2 (right)

(b) Query (left) and Rank 8^{th} from ED (right)

(c) Distance (ED) between images in (a)

(d) Distance (ED) between images in (b)

(e) Log data mass between images in (a)

(f) Log data mass between images in (b)

(g) PDM2 between images in (a)

(h) PDM2 between images in (b)

Figure 4.21: Comparison of PDM2 and ED using visual examples of Corel dataset

Figure 4.21 is another example that illustrates the same scenario. Figures 4.21 (c, e) show in Dimension 53 the large distance between two flowers, at 3, which makes the relevant image more dissimilar to the query. However, the distance has been measured in a sparser area where data mass is 300 and Log of data mass is 5.7. PDM2 assigns weight to the distance accordingly, as shown in Figure 4.21 (g). In another situation in Figures 4.21 (d, f) in Dimension 4 the distance is as low as 1, which shows small dissimilarity between a non-relevant image and query in this dimension. However, the distance has been measured in a dense region with data mass of 500, and Log of data mass is 6.2. So, PDM2 as shown in Figure 4.21 (h) weighted

the distance proportional to the density of the region to 6.2. This resulted in increasing the dissimilarity between the non-relevant image and query in this dimension.

The effect dominant dimension, in Dimension 53 of Figure 4.21 (c) having large distance, has been moderated by lower weights of region density in PDM2.

4.6 Performance comparison between PDM1 and PDM2

Throughout our image retrieval experiments, PDM2 has worked better than PDM1 and only in the Corel dataset PDM1 has similar perfromance. PDM1 focuses on improving m_p where in Cases 2 and 4, its measurements may not align with human perception. So apart from dimensions that these two cases occur in them, PDM1 uses conventional m_p to measure the dissimilarity in other dimensions. Despite of its good perfromance compared to m_p, there are two argument about PDM1: first, using a threshold in defining the two Cases 2 and 4 may raise a limitation, second, in dimensions that Cases 2 and 4 do not occur, it relies only on data distribution for dissimilarity calculations.

Defining a threshold as a mid-point of minimum and maximum of distances/ data masses may raise a limitation, where two border points just below/ above the theshold will be considered as small/ large distance or low/ high data mass. However, border points are very close and their actual values does not represent this difference. For example, suppose the threshold for data mass in a dimesion has been defined as 500. In this case the region with data mass of 499 is considered as sparse region while the region with data mass of 501, which has a very small difference with the sparse region, will be considered as a dense region. This may affect the perfromance of PDM1.

Although in Cases 1 and 3 the measurements of m_p are aligned with measurement of the ED but still PDM1 in those dimensions uses only the strength of data distribution. However, combining this with strength of ED which measure the dissimilarity from a different aspect could help to improve the PDM1's measurements.

On the other hand, PDM2 considers both the geometric distance and the data distribution in all the dimensions for its calculations. PDM2 considers

that ED between two instances is precieved differently depending on the density of the region that the two instances are located. PDM2 uses the log transfromation of data masses in all dimensions as the weight for ED. Combining the strength of geometric distance and perceptual effect of data distribution in all dimensions results in finding a dissimilarity which is closer to human perception. In PDM2 unlike PDM1, there is no need to define a threshold, and also, the combination of ED and data mass has been considered in all dimension not only where two Cases 2 and 4 occurs. This results in a consistently better perfromance of PDM2 comapared to PDM1.

4.7 Summary

In this chapter, we have studied the performance of our proposed dissimilarity measure for image retrieval. The proposed dissimilarity measure is based on the strengths of both ED and m_p and combines the ED and region density. We evaluated two variants of the proposed dissimilarity, PDM1 and PDM2 in our experiments. In PDM1, the basis for calculation of dissimilarity is m_p, which is weighted by ED in Cases 2 and 4 where it may fail to express their actual perceived dissimilarity. In the second variant, PDM2, ED is basis for calculation of dissimilarity and region density is used to moderate it in the final perceived dissimilarity.

The retrieval results of PDM1 and PDM2 are compared with ED and m_p alone. We provided a detail analysis of the strengths of our proposed dissimilarity that resulted in more accurate retrievals. Additionally, we have analysed the performance of PDM1 comapred to PDM2. Also, two variants of PDM have helped to moderate the effect of dominant dimension in dissimilarity calculation by both ED and m_p. Our results show that PDM2 consistently has achieved the best performance in all the image retrieval experiments.

5 Performance Study of Perceptual Dissimilarity Measure for Image Clustering

In Chapter 2, Section 2.4, we have discussed clustering as a useful tool that enables users to effectively extract useful information such as hidden patterns and organise a massive amount of data. Basically, the definition of a cluster is a group of similar data instances and the goal of clustering is to minimise the intra-cluster dissimilarity and maximise the inter-cluster dissimilarity. This highlights the fundamental role of a dissimilarity measure in clustering.

In Chapter 3, we have studied the characteristics of ED and m_p as two measures from geometric distance and mass-based dissimilarity families. We have proposed a new dissimilarity measure that combines ED and data distribution. The proposed dissimilarity measure, PDM, has two variants, PDM1 and PDM2. PDM1 focuses on addressing the limitations with m_p, where in two discussed Cases of 2 and 4, it may not work well. On the other hand, PDM2 focuses on improving ED by incorporating the region density as the weight for ED. PDM2 suggests that the distance between two data points will be perceived differently under the effect of region density.

In Chapter 4, we have studied the effect of both variants of PDM in image retrieval experiments. The results show that PDM2 consistently produces the best results compared to ED, m_p and PDM1. The central role of a dissimilarity measure in clustering and the promising results of PDM2 in image retrieval, motivated us to use PDM2 as an alternative to ED (which is widely used in clustering) to improve the accuracy of image clustering.

This chapter studies the effect of using PDM2 as the dissimilarity measure in improving the image clustering results. Also, we investigate the

performance of our proposed dissimilarity measure for a number of other standard clustering datasets as well. In the following section, first we introduce the benchmark datasets used in this work. The second section will discuss the two clustering methods: k-means and k-medoids in greater detail. The third section will explain the use of PDM2 in both mentioned clustering methods. Section 5.4 presents the clustering evaluation metrics used in this work. This will be followed by presenting the empirical results and analysis. Section 5.7 discusses the use of PDM2 as an alternative to ED in generating the codebook for SIFT BOW and its results for Corel dataset. Finally, we will summarise this chapter.

5.1 Benchmark datasets

In this section, we describe the benchmark datasets used in this work for the clustering purpose. We have selected 11 datasets of which three are the image datasets used in the previous chapter for image retrieval experiments. We will use the same set of features for the mentioned image datasets. The other eight datasets are all standard datasets for clustering and are publicly accessible on UCI machine learning repository [27]. The descriptions of the eight clustering datasets are provided as follows.

Iris [38]: Iris is a real life dataset found in pattern recognition literature. It includes three classes of 50 instances each where every class refers to a type of iris plant. Each iris plant has been represented with four features as, sepal length, sepal width, petal length and petal width.

Bank notes [26]: This is a real life dataset where data was extracted from images that were taken from genuine and forged banknote-like specimens. It has 1,375 instances categorised into two classes of 'forged' and 'genuine'. Wavelet transformation is used to extract features from images. This dataset is represented by five features: variance of Wavelet Transformed image, skewness of Wavelet Transformed image, curtosis of Wavelet Transformed image, entropy of image, and class.

wdbc [1]: This is a real life dataset about breast cancer. Features are extracted from digitised images of fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. This dataset has 569 data instances categorised into two classes malignant (M) and benign (B). Each data instances is represented using 32 attributes.

Wine [2]: This is a real life dataset where, data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three

different cultivators. It has 178 data instances categorised into three classes and each instance described with 13 attributes.

Seeds [118]: This is a real life dataset that presents three different varieties of wheat: Kama, Rosa and Canadian, 70 instances each. The soft X-ray technique has been used to visualise the kernels. Each instance of kernel is represented by seven attributes such as area, perimeter and compactness.

Column [13]: This is a biomedical dataset that provides three classes of patients, Normal (100 patients), Disk Hernia (60 patients) and Spondylolisthesis (150 patients), in total 310 patients. Each patient is represented by six biomedical attributes.

Breast cancer [127, 138]: This is another real life dataset about breast cancers that gathered information from 699 patients and classified them to malignant (M) and benign (B). Each patient is described using 10 biomedical attributes.

Dim[40] : It is a high dimensional dataset that has 1024 instances catergorised into 16 classes and each instance has been represented by 1,024 features.

Table 5.1 summarises the characteristics of mentioned datasets.

Table 5.1: Benchmark dataset characteristics

Dataset	# Instances	# Attributes	# Classes /k
Corel	1000	100	10
Texture	1000	236	25
eBay	528	90	11
Iris	150	4	3
Bank notes	1372	5	2
wdbc	569	32	2
Wine	178	13	3
Seed	210	7	3
Columns	310	6	3
Breast cancer	699	10	2
Dim	1024	1024	16

5.2 Clustering methods

As discussed in Section 2.4, clustering methods can be broadly categorised into hierarchal and partitioning methods and the latter has been used frequently in image clustering. Partitioning algorithms divide the data into several subsets and an iterative process relocates the data points between k clusters. A set of methods in this category starts with defining an objective function and the ultimate goal is to satisfy this function. Pair-wise distances or similarities can be used to compute iter- and intra-cluster relations. In iterative improvements, such pair-wise computations would be too expensive. To address this problem cluster representatives are introduced, where the distance is calculated between each data instance and representative of the cluster. Depending on how representatives are constructed, iterative optimisation partitioning algorithms are subdivided into k-medoids and k-means methods [12]. As distances are essential to these algorithms, we will focus on applying our proposed dissimilarity measure in k-medoids. In k-means, choice of distance is limited to ED, therefore, k-medoid is the most similar option to k-means that allows choosing an arbitrary dissimilarity measure. Hence, we will use our proposed dissimilarity measure in k-medoids.

We use the following definitions in this chapter for our clustering study. Given a dataset $M = \{\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \cdots, \mathbf{x_N}\}$ consists of data instances feature vectors and $k > 1$ clustering can be defined as mapping of $f : M \rightarrow \{1 \cdots k\}$ where each $\mathbf{x_i}$ is assigned to one cluster, C_j, $1 < j < k$.

5.2.1 k-means

The name is derived from representing each of the k clusters by the mean of its points, C_j, the socalled centroid. Considering the mean to represent each cluster, it does not support categorical data, while it does have very nice geometric and statistical properties for numerical data. The objective function in kmeans is to minimise the sum of square errors (SSE) in a cluster as discussed in Section 2.4.1.2. In fact, SSE is ED between data instances in a cluster and the assigned centroid. Therefore, k-means works best with ED. The classic k-means has the following steps:

1. Choose k centroids randomly
2. Calculate the distance of each point to all the centroids

3. Assign each point to the closest centroid
4. Update the centroid by calculating the mean of all the points in one cluster
5. Repeat steps 2-4 until it converges

Choosing the mean of the points in a cluster as the centroid minimises the sum of squared distances of all points to that centroid. In the following, it is shown that the mean of the data instances assigned as the centroid of a cluster is the only choice that satisfies the objective function. Considering the SSE of data instances to Centroid **a** we will have:

$$\begin{aligned} SSE(\mathbf{M}, a) = \sum_{i=1}^{N} (\mathbf{x}_i - a)^2 = \\ \sum_{i=1}^{N} \left(\mathbf{x}_i^2 - 2\mathbf{x}_i a + a^2\right) = \\ \sum_{i=1}^{N} \mathbf{x}_i^2 - \sum_{i=1}^{N} 2a\mathbf{x}_i + \sum_{i=1}^{N} a^2 = \\ \sum_{i=1}^{N} \mathbf{x}_i^2 - 2a \sum_{i=1}^{N} \mathbf{x}_i + \sum_{i=1}^{N} a^2 \end{aligned} \tag{5.1}$$

To minimise the above expression is to take the differentiate in respect to **a** which gives us:

$$S^{'}(a) = -2 \sum_{i=1}^{N} \mathbf{x}_i + 2na \tag{5.2}$$

This equation will be zero only if $a = \frac{\sum_{i=1}^{N} x_i}{N}$ which is the mean of the data instances. So we can see that mean of the data instances within a cluster is the only value that can minimize the sum of squared errors in a cluster.

5.2.2 k-medoids

In k-medoids, clusters are represented by one of its data points, the so-called medoid. Unlike k-means that attempts to minimise the sum of squared errors within a cluster, the objective function in k-medoids is to minimise the sum of dissimilarities within a cluster. Therefore, it allows us to choose an aribitary dissimilarity measure. Also, unlike k-means the

choice of medoid is determined by the location of predominant fraction of data, hence it is less sensitive to outliers.

In contrast to the k-means algorithm that randomly chooses k points as the centeroids, k-medoids chooses medoids from the actual data points. In the following section we presents the steps of PAM as the first and most widely used algorithm based on k-medoids.

1. Select k of the data points as the medoids randomly
2. Assign each data points to its closest medoid
3. Calculate the total dissimilarity between the medoid m and all non-medoid data points o
4. Swap the the m and o and recompute the total dissimilarity (sum of dissimilarities of data points to the assigned medoids)
5. Undo the swap if the total dissimilarity in the previous step has increased

In the following section, we will discuss the use of our proposed dissimilarity in clustering.

5.3 Clustering using PDM2

This section discusses the use of PDM2 in clustering. PDM2 has another component in addition to ED, in that it combines the region density and ED to measure dissimilarity between two data instances. We choose to use PDM2 as the dissimilarity measure in k-medoids.

PDM2 will be used in the second step of the algorithm to calculate the dissimilarity between the data instances and selected medoids. To use PDM2 we need to calculate two components, the data mass and the distance. The Log transformation of data mass as a proxy of the density of the defined region weights the the distance between the medoid and the data instance.

Later in this chapter, we will present the empirical results and analysis of clustering using our benchmark datasets. First, in the following section, we will present the clustering evolution metrics used in this work.

5.4 Evaluation metrics

To evaluate the clustering results, we use class labels (ground truths) of images. We use two metrics: Rand Index [131] and F-measure [97]. Rand Index (RI) is defined as:

$$RI = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.3}$$

where TP is the number of pair of images with the same class labels assigned to the same cluster (true positives), TN is the number of image pairs with different class labels assigned to different clusters (true negatives), FP is the number of image pairs with different class labels assigned to the same cluster (false positives), and FN is the number image pairs with the same class labels assigned to different clusters (false negatives). The Rand index (RI) measures the percentage of assignments that are correct.

F-measure is defined as:

$$RI = \frac{2.P.R}{P + R} \tag{5.4}$$

Where $p = \frac{TP}{TP+FP}$ is precision and $R = \frac{TP}{TP+FN}$ is recall.

5.5 Empirical results

In this section we first describe the set up for our clustering experiments followed by a presentation of empirical results in Section 5.5.2.

5.5.1 Experimental set up

As the clustering results of k-means and k-medoids are sensitive to the initial choice of random centroids or medoids, we conducted 40 runs of each experiment and reported the mean and standard deviation of RI and F over 40 runs. Because we know the number of class labels (C), the parameter k was set to C. RI and F were computed using class labels (ground truths). The parameter p in m_p and PDM2 was set as default to 2 to be consistent with ED. All experiments are performed using standard Matlab toolboxes.

5.5.2 Empirical results

In this section we compare the clustering performance of PDM2 with ED and m_p when being used as dissimilarity measures in k-medoids. Also, their performance will be compared to k-means as well. Results are presented in terms of average RI and F and standard deviation for k-means, k-medoids(ED), k-medoid(m_p) and k-medoids(PDM2).

Results in Tables 5.2 and 5.3 show that k-medoids (PDM2) outperformed all three competitors with seven wins, three draws and one loss in terms of RI. Also, it outperformed all other three competitors with 10 wins and one draw in terms of F measure. The good performance of PDM2 in clustering is consistent with our results in image retrieval. It is interesting to note that k-medoids (m_p) produced worse results than k-medoids (ED) in eight datasets. k-medoids (ED) and k-Means (ED) produced similar results because they are essentially the same method with the only difference of cluster centres being medoids and means, respectively. k-medoids(PDM2) has achieved better results due to the consideration of the perceptual effect of data distribution in calculating the distance between two instances in the feature space.

Table 5.2: The average $\pm$ standard deviation of RI over 40 runs

Datasets	RI			
	k-means	k-medoids(ED)	k-medoid(m_p)	k-medoids(PDM2)
Corel	$0.3984 \pm 4.10\text{e}{-4}$	$0.4090 \pm 1.64\text{e}{-4}$	$0.3480 \pm 1.07\text{e}{-2}$	$\mathbf{0.4121 \pm 2.46\text{e}{-4}}$
Texture	$0.4533 \pm 1.90\text{e}{-3}$	$0.4540 \pm 2.50\text{e}{-3}$	$0.4559 \pm 2.80\text{e}{-3}$	$\mathbf{0.4605 \pm 1.70\text{e}{-3}}$
eBay	$0.4344 \pm 1.92\text{e}{-4}$	$0.4331 \pm 4.17\text{e}{-4}$	$0.4158 \pm 7.20\text{e}{-3}$	$\mathbf{0.4373 \pm 2.08\text{e}{-4}}$
Iris	$0.4399 \pm 2.24\text{e}{-16}$	$0.4330 \pm 9.62\text{e}{-16}$	$0.2968 \pm 5.54\text{e}{-3}$	$\mathbf{0.4469 \pm 1.12\text{e}{-16}}$
Bank notes	$\mathbf{0.2624 \pm 1.68\text{e}{-16}}$	$0.2567 \pm 7.23\text{e}{-16}$	$0.2357 \pm 1.52\text{e}{-2}$	$0.2585 \pm 3.81\text{e}{-16}$
wdbc	$0.2654 \pm 3.73\text{e}{-16}$	$0.2654 \pm 3.73\text{e}{-16}$	$0.2650 \pm 3.73\text{e}{-16}$	$0.2654 \pm 1.78\text{e}{-16}$
Wine	$0.3258 \pm 1.68\text{e}{-16}$	$0.3361 \pm 2.48\text{e}{-16}$	$0.2975 \pm 1.68\text{e}{-16}$	$0.3361 \pm 2.24\text{e}{-16}$
Seeds	$0.4372 \pm 2.48\text{e}{-16}$	$0.4372 \pm 2.48\text{e}{-16}$	$0.3651 \pm 1.04\text{e}{-2}$	$\mathbf{0.4401 \pm 1.12\text{e}{-16}}$
Columns	$0.2785 \pm 2.48\text{e}{-16}$	$0.2785 \pm 2.48\text{e}{-16}$	$0.2694 \pm 2.48\text{e}{-16}$	$\mathbf{0.2961 \pm 0}$
Breast cancer	$0.3037 \pm 1.68\text{e}{-16}$	$0.3015 \pm 5.62\text{e}{-17}$	$0.3098 \pm 3.00\text{e}{-3}$	$\mathbf{0.3255 \pm 2.48\text{e}{-16}}$
Dim	0.5 ± 0	0.5 ± 0	$0.1852 \pm 4.60\text{e}{-3}$	0.5 ± 0

Table 5.3: The average $\pm$ standard deviation of F over 40 runs

Datasets	F			
	k-means	k-medoids(ED)	k-medoid(m_p)	k-medoids(PDM2)
Corel	$0.1950 \pm 5.40\text{e}{-4}$	$0.1939 \pm 9.20\text{e}{-4}$	$0.2003 \pm 4.10\text{e}{-3}$	$\mathbf{0.2297 \pm 3.06\text{e}{-4}}$
Texture	$0.2103 \pm 1.29\text{e}{-2}$	$0.2185 \pm 1.08\text{e}{-2}$	$0.2467 \pm 1.24\text{e}{-2}$	$\mathbf{0.2586 \pm 1.05\text{e}{-2}}$
eBay	$0.4137 \pm 3.01\text{e}{-4}$	$0.4172 \pm 1.75\text{e}{-4}$	$0.4131 \pm 2.90\text{e}{-3}$	$\mathbf{0.4192 \pm 1.65\text{e}{-5}}$
Iris	$0.8918 \pm 4.49\text{e}{-16}$	$0.8977 \pm 5.62\text{e}{-16}$	$0.5491 \pm 9.67\text{e}{-2}$	$\mathbf{0.8988 \pm 2.48\text{e}{-16}}$
Bank notes	$0.6026 \pm 2.48\text{e}{-16}$	$0.5952 \pm 2.48\text{e}{-16}$	$0.4149 \pm 2.88\text{e}{-2}$	$\mathbf{0.6166 \pm 5.62\text{e}{-16}}$
wdbc	$0.8443 \pm 8.99\text{e}{-16}$	$0.8443 \pm 8.99\text{e}{-16}$	$0.8443 \pm 8.99\text{e}{-16}$	$\mathbf{0.8731 \pm 5.34\text{e}{-16}}$
Wine	$0.7148 \pm 6.74\text{e}{-16}$	$0.7208 \pm 6.74\text{e}{-16}$	$0.4428 \pm 4.66\text{e}{-16}$	$\mathbf{0.7248 \pm 6.74\text{e}{-16}}$
Seeds	$0.8954 \pm 3.37\text{e}{-16}$	$0.8954 \pm 3.37\text{e}{-16}$	0.7258 ± 0	$\mathbf{0.8998 \pm 4.49\text{e}{-16}}$
Columns	$0.6744 \pm 4.49\text{e}{-16}$	$0.6744 \pm 4.49\text{e}{-16}$	0.6423 ± 0	$\mathbf{0.7314 \pm 6.74\text{e}{-16}}$
Breast cancer	$0.9584 \pm 6.74\text{e}{-16}$	0.9554 ± 0	$0.7368 \pm 9.93\text{e}{-4}$	$\mathbf{0.9594 \pm 4.49\text{e}{-16}}$
Dim	1 ± 0	1 ± 0	$0.1836 \pm 1.40\text{e}{-3}$	1 ± 0

In the previous chapter, Section 5.7, we discussed using PDM2 in clustering for codebook generation of SIFT BOW for Corel dataset. Here we use two sets of SIFT BOWs of Corel dataset to evaluate their performance in clustering. As presented in Tables 5.2 and 5.3, PDM2 has performed the best for clustering of Corel dataset, so we will use it as the dissimilarity measure for the comparison of two sets of SIFT BOWs of Corel dataset. The two sets of SIFT BOWs are produced using two different codebooks, one generated with conventional k-means and the other one, k-medoids with PDM2, as dissimilarity measure.

Table 5.4: The average $\pm$ standard deviation of RI and F over 40 runs for Corel dataset

	Codebook generated by k-means (ED)	Codebook generated by k-medoids (PDM2)
RI	$0.4121 \pm 2.46\text{e}{-04}$	$\mathbf{0.4193 \pm 2.08\text{e}{-04}}$
F	$0.2293 \pm 3.04\text{e}{-04}$	$\mathbf{0.2339 \pm 3.04\text{e}{-04}}$

Results in Table 5.4 show that SIFT BOW produced using the codebook generated by k-medoids(PDM2) yields better clustering results. This follows our discussion on using PDM2 for codebook generation in previous chapter and confirms the better retrieval results obtained by PDM2. In the following section we discuss the effect of PDM2 in improving the clustering results.

5.6 Performance study of PDM2 through visual examples

In this section we study the performance of PDM2 for clustering. Section 5.6.1 provides visual examples of image clustering from the eBay dataset. Section 5.6.2 presents our analysis on the PDM2 performance in clustering through the visual examples.

5.6.1 Clustering visual examples

In this section, we present some visual examples from the eBay dataset where images are categorised based on colours. These visual examples will be used in the next section to provide further insight and analysis on PDM2 performance in clustering. We have used the HSV colour histogram to represent images in this dataset. We expect to get the clusters, which gathers images of one particular colour.

Figures 5.1- 5.8 show that the images gathered in clusters belong to white and orange colour from the four mentioned clustering methods. Therefore, in the white cluster images those that are not white are non-relevant and the same goes for the orange cluster. In the figures belonging to clustering results of k-medoids the first image is the medoid of a cluster that other images are assigned to. As we can see in Figure 5.1, k-medoids using PDM2 could gather 33 images of white colour in one cluster while this number for k-medoid(ED) is 22 and for k-means is 19 as shown in Figures 5.2- 5.3. Figure 5.4 shows the cluster that gathered all the white and yellow images using m_p as dissimilarity measure in k-medoids. Also, Figure 5.5 shows the k-medoids using PDM2 could gather 30 images of orange colour in one cluster while this number dropped to 18 in k-medoid(ED) and 21 for k-means in Figures 5.6- 5.7. k-medoids using m_p could also gather 22 images of orange colour in one cluster as shown in Figure 5.8.

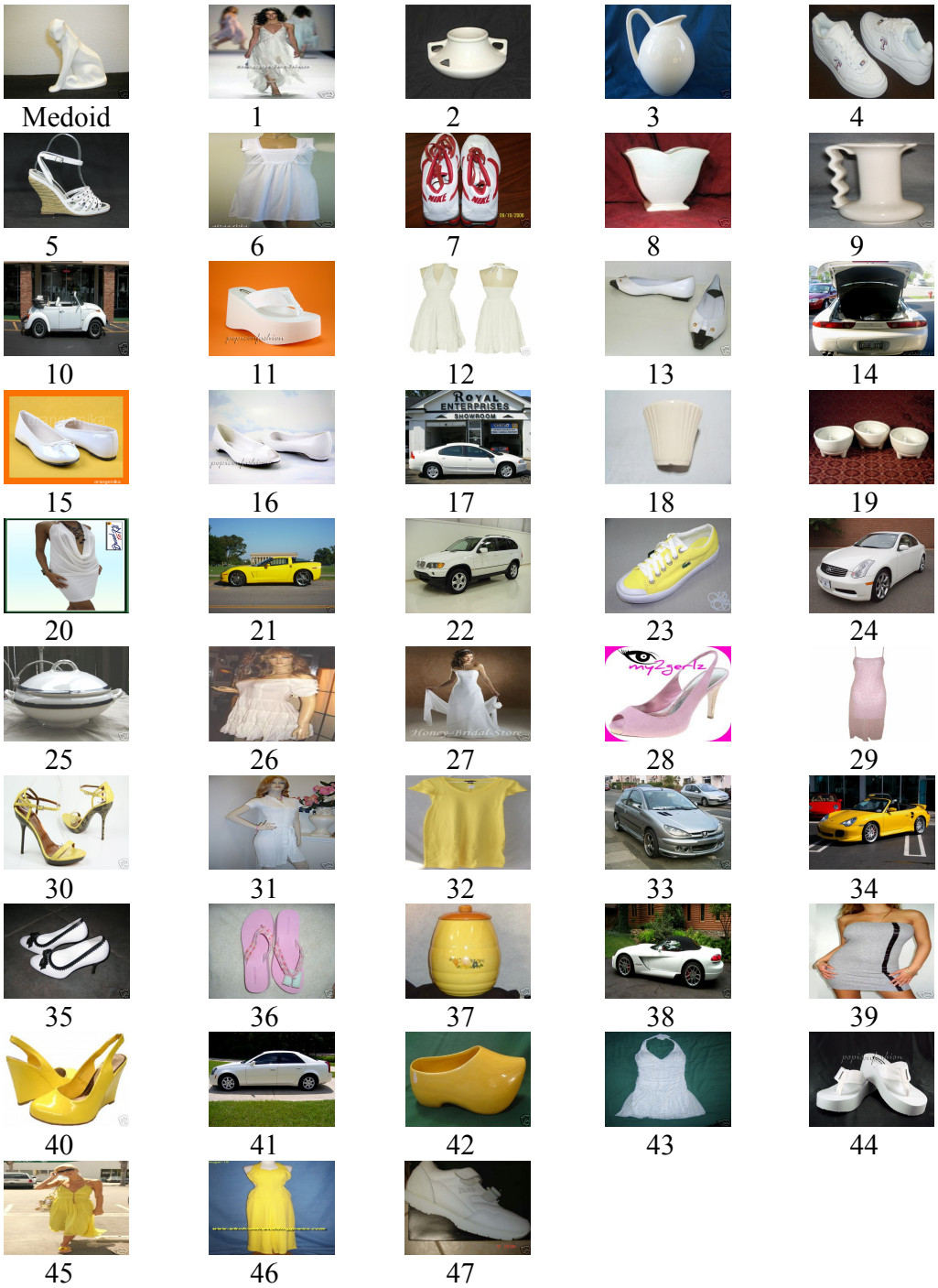


Figure 5.1: White cluster from k-medoids using PDM2

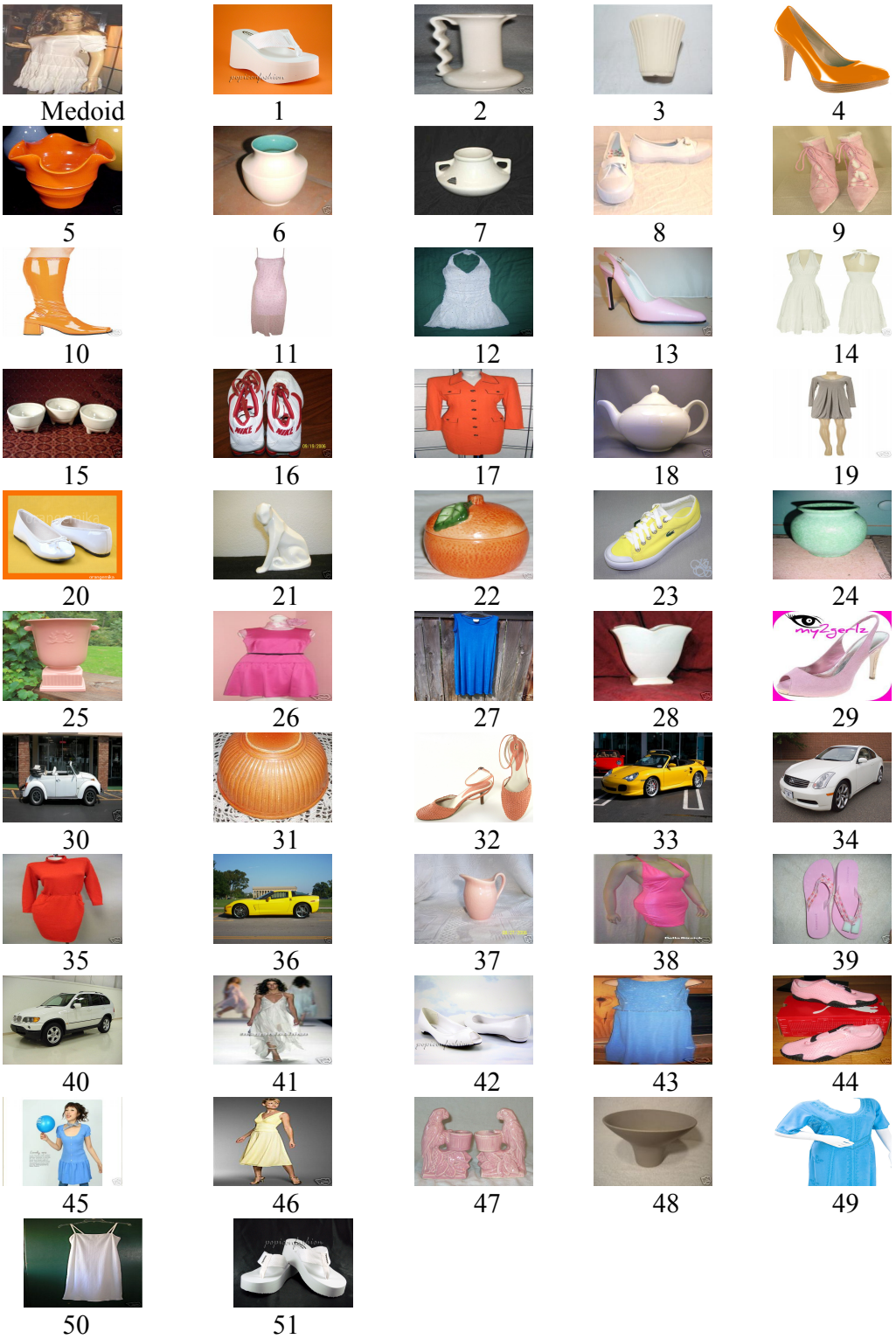


Figure 5.2: White cluster from k-medoids using ED

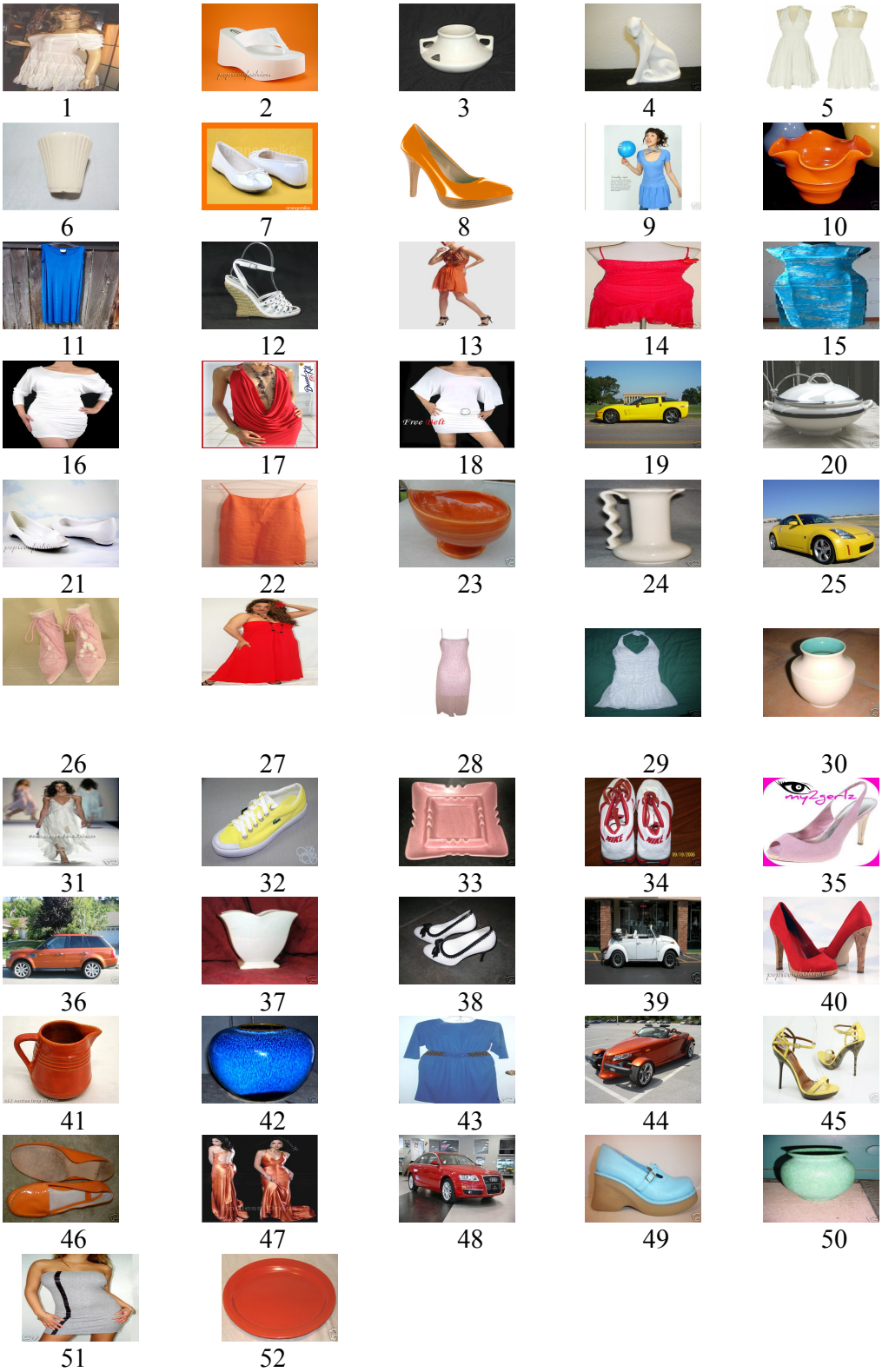
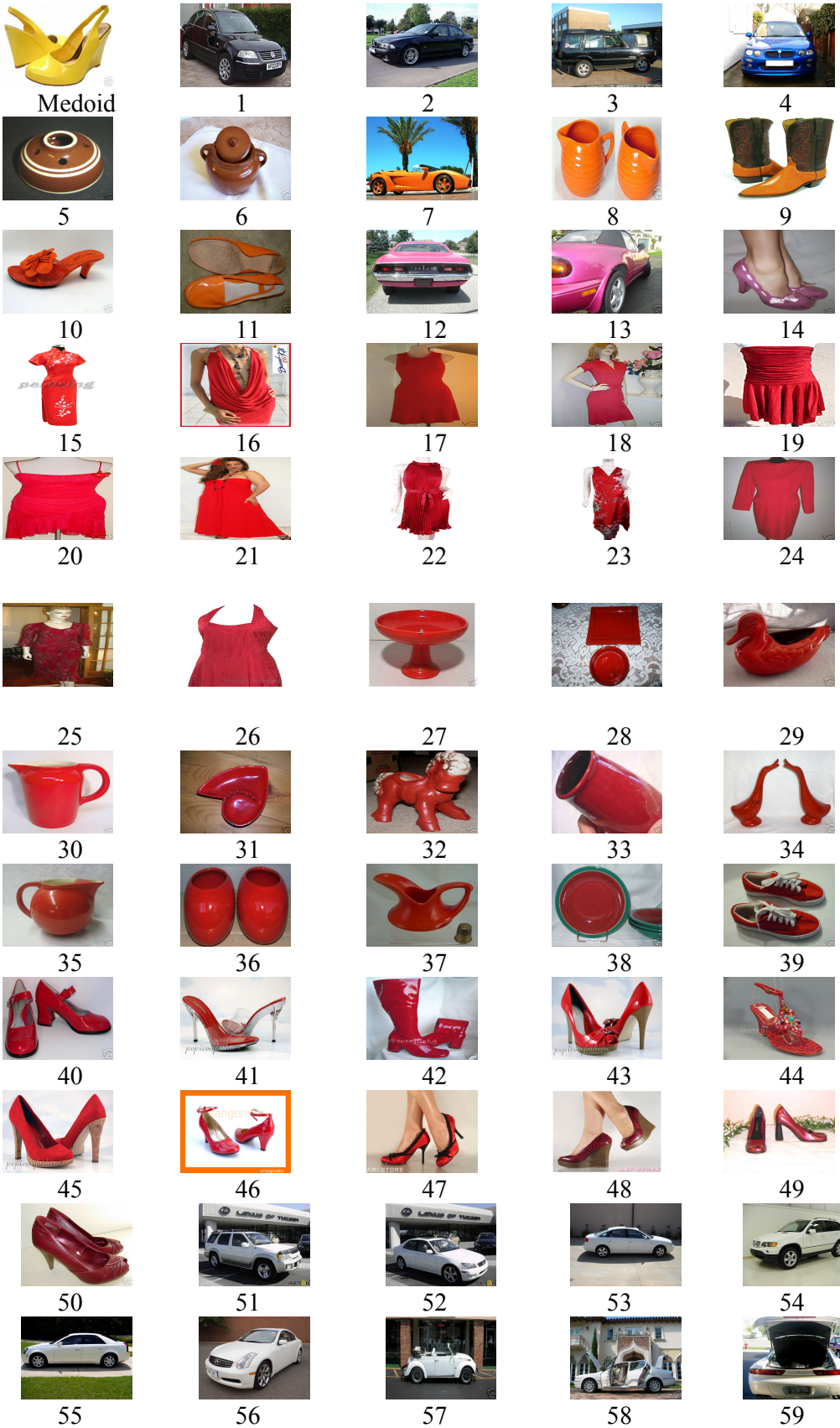


Figure 5.3: White cluster from k-means


Medoid
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59


60

61


62


63


64


65

66


67


68


69


70


71


72


73


74


75


76


77


78


79


80


81


82


83


84


85


86


87


88


89


90


91

92


93


94


95


96


97


98


99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

Figure 5.4: White cluster from k-medoids using m_p

Figure 5.5: Orange cluster from k-medoids using PDM2

Figure 5.6: Orange cluster from k-medoids using ED

Figure 5.7: Orange cluster from k-means

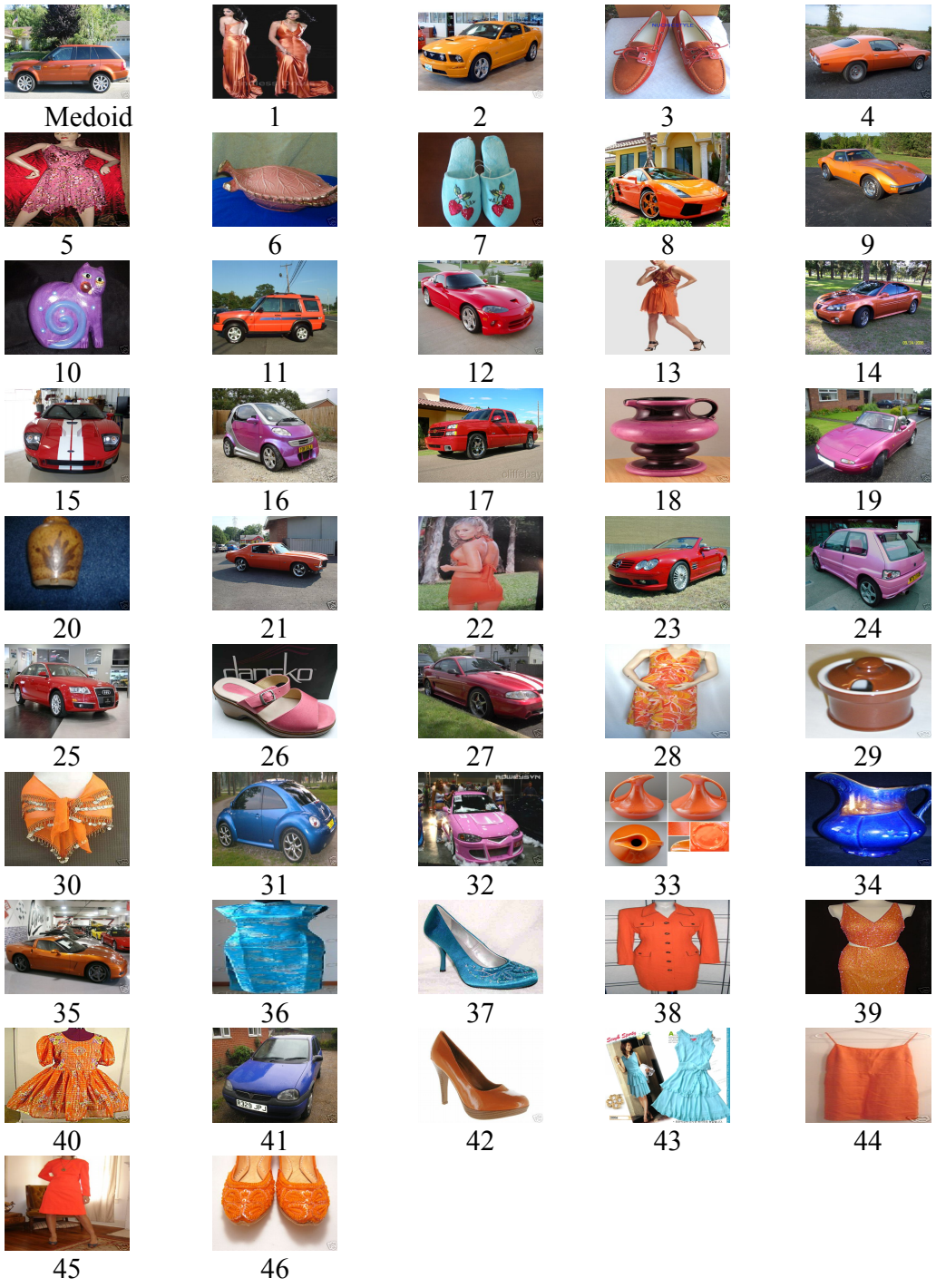


Figure 5.8: Orange cluster from k-medoids using m_p

5.6.2 Discussion on performance of PDM2 for clustering

In this section, we use the visual examples from the previous section to discuss the performance of PDM2 in clustering. We select specific examples from white and orange clusters and show the limitations of using ED or data mass in some selected dimensions of feature space. Then, we will show the effect of using PDM2 in those dimensions in addressing the discussed limitations. Generalising this effect to multiple dimensions will, thereby, explain the better performance of PDM2 compared to ED and m_p alone.

In the white cluster PDM2 has included a white shoe in Figure 5.1.4 (this is the image labeled as the fourth image in Figure 5.1), while this image has not been picked in white clusters by k-medoids using ED and k-means. However, an orange shoe has been included in the white clusters (Figure 5.2.4 and Figure 5.3.8) by both clustering methods of k-medoids(ED) and k-means while the PDM2 did not include that irrelevant image in the white cluster. This has occurred because the PDM2 considers the effect of region density on the ED between two instances. PDM2 uses the Log transformation of data mass as a proxy of region density to proportionally weight the distance between two instances. Therefore, if the region defined between two instances is denser, PDM2 assigns a higher weight to the distance between them and vice versa.

In the following section, we show the effect of region density on distance in our proposed dissimilarity measure through examples in some dimensions of feature space. The following discussion for two cases will be considered in the visual examples. In the first case, we select a few dimensions of feature vectors and show the distance between two similar images that were high in those dimensions, which, is counter intuitive. Then, we show the data mass between them in the same dimension which is low, which means the two instances are located in sparse region. Following this, we show how using the low data mass as the weight in PDM2 could moderate the large distance between two similar images in those dimensions.

In a similar trend, we show the examples of dimensions where the distance between two dissimilar images is small and is in conflict with human perception. However, they are located in a dense region. The high weight from high data mass of the region moderates the small distance between them and makes it closer to humans perception. Considering these effects in multiple dimension moderate the total dissimilarity between two images.

We will discuss the same scenarios for m_p as well and show the conflicts when using data mass for dissimilarity measurement between two similar images. Then, we will show how using PDM2 could address these situations.

Figure 5.9 (c) shows the distance between the white image assigned as the medoid of white cluster from k-medoids(PDM2) and another white shoe included in this cluster. Also Figure 5.9 (d) shows the distance between the same medoid and an orange shoe, which is not included in this cluster, however it has been assigned to the white cluster by k-medoids(ED) and k-means. There is a large distance of 0.89, between two white images (Figure 5.9 (c)) in Dimension 5, while they have been located in a sparse region where the data mass between them is 144. The log of data mass in this region is 4.9 (Figure 5.9 (e)). PDM2 assigned a lower weight to the ED between these two images in this dimension.

On the other hand, the distance between the white medoid and the orange shoe, Figure 5.9(d), in Dimension 10 is very small, at 0.02, while they have been located in a dense area with data mass of 384. Log of data mass in this region is 5.9 as shown in Figure 5.9 (f). In this case, PDM2 assigns a higher weight to the ED of these two images in this dimension. Considering this weighting in other dimensions in the feature space resulted in a higher dissimilarity of 15.3 between the white medoid and orange shoe compared to white shoe with the dissimilarity of 9.6. In k-medoids(PDM2) the orange shoe has not been assigned to the white medoid. However, in k-medoids(ED) and k-means, relying only on geometric position of two images in the feature space resulted in assigning the orange shoe to the white cluster.

Also, we can see that PDM2 moderates the effect of dominant dimensions in calculating dissimilarity using ED. As we discussed, in Figure 5.9 (c) in Dimension 5 of feature space the distance between two white images is large, 0.89. The large value of distance in this dimension and some other dimensions may dominate the others and effect the total distance calculated by ED. However, PDM2 helped to moderate this effect by weighting the distance using the Log of data mass.

Figure 5.10 shows another example of similar scenario where a blue pot has appeared in the orange cluster gathered by k-means and k-medoids(ED) while it has not been included in the orange cluster by k-medoids (PDM2). On the other hand, a relevant image of an orange dress, which has been included in the orange cluster by k-medoids(PDM2), has not been appeared in the orange clusters of k-medoids(ED) and k-means.

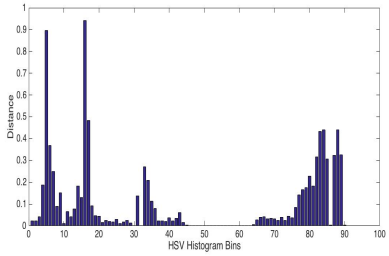
In Figure 5.10 (c) the distance between the medoid assigned for the orange cluster and the relevant image of the orange dress in Dimensions 81 is relatively large, at 0.73, while the region covering these two instances is sparse with data mass of 192. The Log of data mass in this region is 5.2. Therefore, PDM2 assigns lower weight to the ED between these two images in this dimension.

In the other case, Figure 5.10 (d) shows distance between the blue pot and a medoid from the orange cluster in Dimension 7 is very small, at 0.03, which is counter intuitive. However, they are located in a dense region with the a data mass of 384. The Log of data mass in this region is 5.9. Considering the high data mass, PDM2 assigned a higher weight to the distance between the medoid from orange cluster and the blue pot in this dimension. Applying this weighting in other dimensions resulted in the higher dissimilarity between the the medoid from orange cluster with blue pot, at 14.66, compared to the orange dress with a dissimilarity of 11.65. Considering the resulting dissimilarity, the blue pot has not been assigned to the orange cluster using the PDM2.

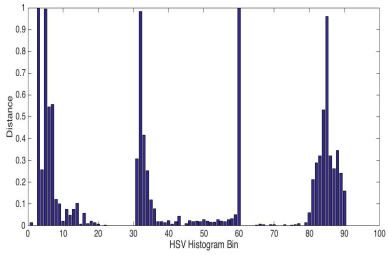
We can see that in Figure 5.10 (c) the effect of large distance in Dimension 81 in calculating the dissimilarity has been moderated by the weight from the low data mass in the region in PDM2.


(a) White medoid (left) white shoe using PDM2 (right)

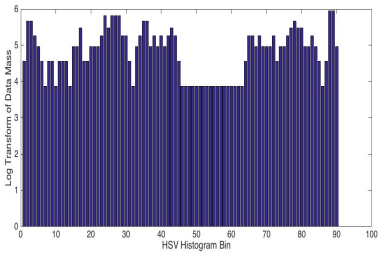

(b) White medoid (left) orange shoe from white cluster using ED (left)



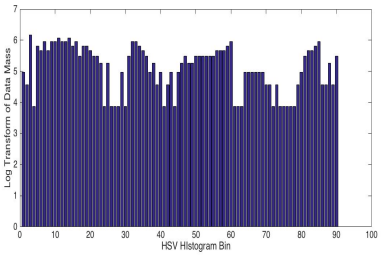
(c) Distance (ED) between images in (a)



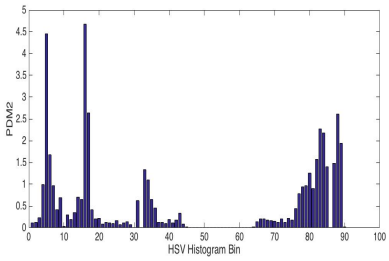
(d) Distance (ED) between images in (b)



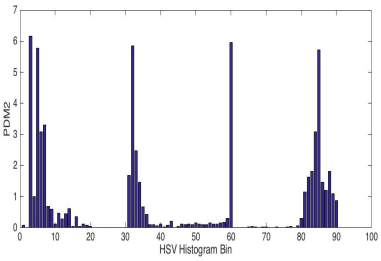
(e) Log of Data mass between images in (a)



(f) Log of Data mass between images in (b)

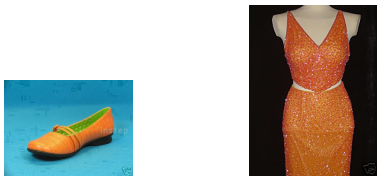


(g) PDM2 between images in (a)

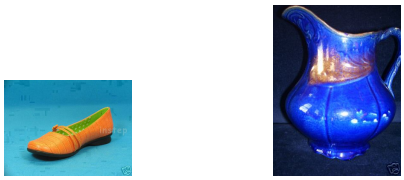


(h) PDM2 between images in (b)

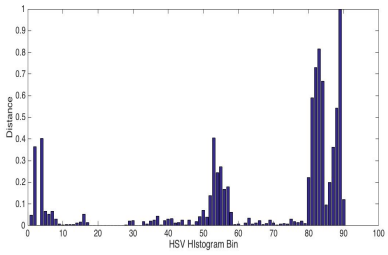
Figure 5.9: Comparison of white shoe from white cluster in k-medoids using PDM2 and orange shoe from white cluster in k-means and k-medoids using ED



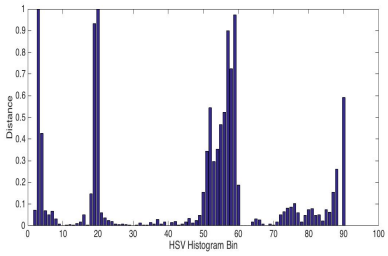
(a) Orange medoid (left) orange dress using PDM2 (right)



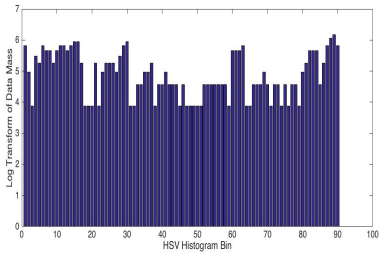
(b) Orange medoid (left) blue pot from orange cluster using ED (right)



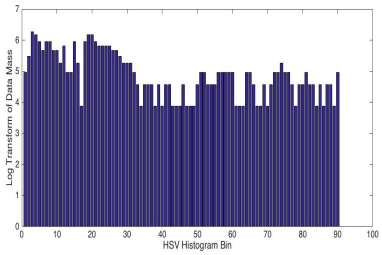
(C) Distance (ED) between images in (a)



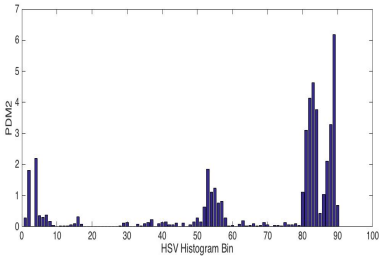
(d) Distance (ED) between images in (b)



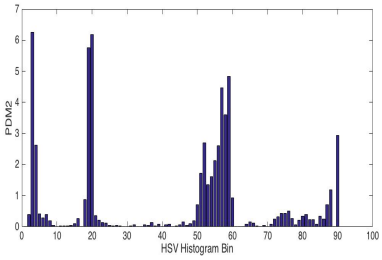
(e) Log of Data mass between images in (a)



(f) Log of Data mass between images in (b)



(g) PDM2 between images in (a)

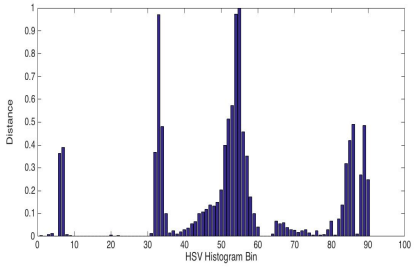


(h) PDM2 between images in (b)

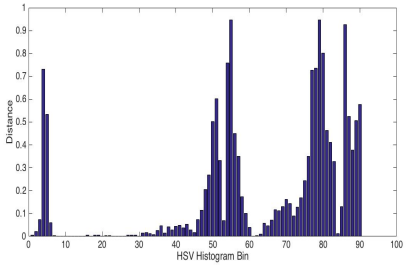
Figure 5.10: Comparison of orange shoe from orange cluster in k-medoids using PDM2 and blue pot from orange cluster in k-means and k-medoids using ED

(a) Medoid of yellow cluster (left) White pot using m_p (right)

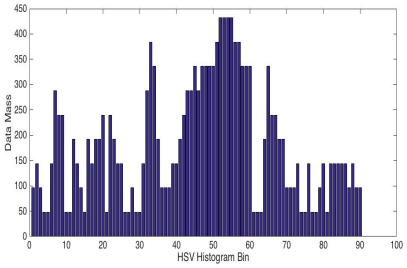
(b) Medoid of yellow cluster (left) yellow car using m_p (right)



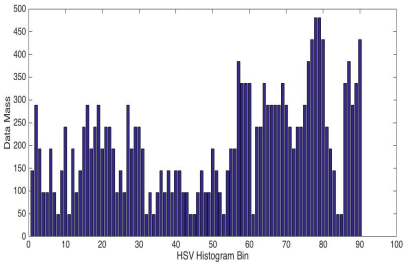
(c) Distance (ED) between images in (a)



(d) Distance (ED) between images in (b)



(e) Data mass between images in (a)

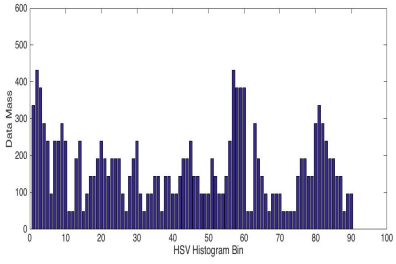


(f) Data mass between images in (b)

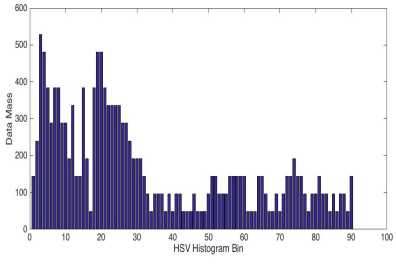
Figure 5.11: Comparison of white pot and yellow car from yellow cluster in k-medoids using m_p

(a) Medoid of orange cluster PDM2 (left) blue pot using m_p (right)

(b) Medoid of orange cluster (left) orange car using PDM2 (right)



(c) Data mass between images in (a)



(d) Data mass between images in (b)

Figure 5.12: Comparison of data mass between orange medoid using k-meodids PDM2, orange car using PDM2 and blue pot from k-medoids using m_p

Figure 5.4 shows that in k-medoids(m_p) there is an expanded cluster that includes big numbers of images from different colours, especially white, yellow and red. In the following section, through exploring examples of data masses in some dimensions we will show why relying only on data mass, as in m_p may result in such a cluster.

In Figure 5.11, we can see that an image of a yellow shoe has been assigned as the medoid for the cluster that gathers all the white and yellow images, along with many red ones and a few other colours. Relying only on data distribution and ignoring the distance between two instances made this happen. For example in Dimension 89 of Figure 5.11 (e), we can see that the data mass between the yellow medoid and irrelevant white image is as low as 96. However, the distance between them is quite considerable, at 0.48, as shown in Figure 5.11 (c). In this case the low data mass as a basis for calculation in m_p suggests a small dissimilarity between the yellow shoe and the white pot in this dimension. Considering the similar conflict in other dimensions resulted in finding many white images closer to the yellow shoe as the medoid of this cluster.

The blue pot shown in Figure 5.12 (a) has been assigned to the orange cluster using m_p but it has not been collected in orange cluster of k-medoids(PDM2). Figure 5.12 (c) shows the data mass between the medoid from the orange cluster of k-medoids(PDM2) and a blue pot, which has not been collected in the orange cluster. Sum of data mass in all dimensions is 14982. Figure 5.12 (d) shows the data mass between the same medoid with an orange car that has been appeared in the orange cluster of k-meodis(PDM2) but not included in orange cluster of k-medoids(m_p). Sum of data masses between these two in all dimensions is 16,608. In the event of relying only on the data mass as in m_p, the blue pot with lower data mass should be closer to the medoid of orange cluster of k-meodis(PDM2). However, the weighting system proposed in the PDM2 as previously discussed through Figure 5.10 (b) resulted in it not picking the blue pot in the orange cluster of k-medoids(PDM2).

5.7 Improving Codebook generation in BOW using the PDM2

In this section we present the results of using PDM in codebook generation of BOW for Corel dataset. As discussed in Chapter 3, Section 3.3.2.3, in the BOW feature extraction procedure there is a stage called 'feature quanti-

sation' which produces the codebook. In this stage, the dense features extracted from dataset images are clustered in to N numbers of clusters. The representative of each cluster is considered as a visual word. As a result it produces N number of visual words which is known as codebook or visual dictionary. In the next stage, BOW builds a histogram from frequency of occurrence of each the visual words in the codebook to represent each image.

The most widely used clustering method used in codebook generation is k-means, which uses ED as its dissimilarity measure [18, 105, 120, 129, 141]. Following the promising results of PDM2 for image retrieval, we have proposed to use PDM2 as the dissimilarity measure in clustering to generate the codebook. We have used PDM2 in codebook generation for SIFT BOW of Corel dataset. To evaluate the performance of the new codebook generated by PDM2 for image retrieval, we have compared it with SIFT BOW produced using conventional codebook.

Figure 5.13 shows the retrieval results of two produced sets of SIFT BOWs for Corel dataset, where PDM2 has been used as the dissimilarity measure for retrieval purpose.

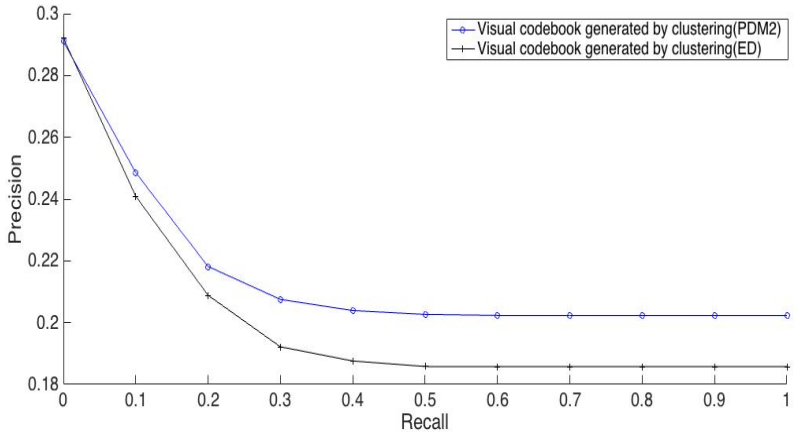


Figure 5.13: Image retrieval results on Corel dataset with SIFT BOW from two sets of generated codebooks using ED and PDM2.

As shown in Figure 5.13, PDM2 as an alternative to ED in generating the codebook produces features that improved the retrieval results. Previously, in Chapter 3 and 4, we have analysed the limitations of ED and m_p in measuring dissimilarity between two features vectors as both of them rely only on distance or data mass between two instances. Also,

we have discussed the effect of considering region density on human similarity judgment of distance. Following the same scenario, conventional k-means used for codebook generation inherits the limitations of ED in measuring the dissimilarity between two features vectors.

In SIFT feature quantisation, k-means uses ED as the dissimilarity measure to calculate distance of each sampled SIFT vector with each of the k randomly selected cluster centres. The centre of each of the resulted clusters is considered as a visual word in the codebook. We have discussed the central role of ED in cluster assignments for each vector. ED has been identified with limitations of ignoring the role of data distribution in dissimilarity calculation. Also, ED calculation of distance is under the effect of dominant dimensions of features.

Exploiting PDM2 as the alternative to ED, weights the distance between two feature vectors in each dimension by log of data mass in the region covering them. This results in a codebook that the perceptual effect of data distribution has been considered in choosing its visual words. Image features produced from these visual words may be better aligned with human judgment of similarity.

5.8 Summary

The performance of image clustering algorithms depends on the effectiveness of dissimilarity measure used to compute pairwise similarities of images. A dissimilarity measure that relies solely on distance and ignores the data distribution, or one that completely ignores distance does not lead to the best effectiveness. The combination of both in a PDM2 results in more effective similarity in clustering. In this chapter, we have compared the effectiveness of ED, m_p and PDM2 in the image clustering task using the simplest and most widely used family of cluster algorithms: k-means and k-medoids. We also used other standard benchmark clustering datasets to evaluate the performance of PDM2. Our experimental results suggest that the PDM2 produces better clustering results than ED (using distance only) and m_p-dissimilarity (using mass only). Also we have used PDM2 as an alternative to ED in clustering to generate codebook for SIFT BOW. Our results confirm the improvement in retrieval results using the proposed method for codebook generation of SIFT BOW.

6 Thesis conclusions and future work

This chapter concludes this thesis in the following sections and provides potential future works in Section 6.2.

6.1 Thesis conclusions

The work presented in this thesis mainly focuses on developing a new dissimilarity measure that incorporates the perceptual effect of data distribution with geometric distance. Also, we aimed to study the performance of such a dissimilarity measure for image retrieval and clustering. The following sections present the conclusions of this thesis.

6.1.1 Investigation of m_p suitability for image retrieval

This thesis has investigated the suitability of m_p in image retrieval. The retrieval results and analysis show that m_p may not be suitable dissimilarity choice in some situations. Consider having two pairs of instances, the first pair are perceptually similar and located in a dense region and the second pair is perceptually dissimilar but located in a sparse region. m_p will find the second pair more similar compared to the first pair, contrary to the human perception. In these situation, m_p alone may not retrieve an accurate result.

Similar to ED, m_p gives equal weight to all the dimensions in aggregation of data masses for total dissimilarity calculation. This results in situations where a few dimensions with relatively much higher data masses influence the overall dissimilarity.

6.1.2 Proposed perceptual dissimilarity measure

This thesis has introduced a new dissimilarity measure, called 'perceptual dissimilarity measure' (PDM), which address the limitations of both ED and m_p. Both ED and m_p have their strengths in measuring the dissimilarity from different aspects that partially complies with humans perception. PDM combines the strengths of m_p and ED to yield more accurate results. Two variants of PDM have been proposed PDM1 and PDM2.

PDM1 focuses on improving m_p in situations where it may not retrieve accurate results. PDM1 defines a threshold to identify the situations where measurements from m_p and ED are conflicting (the discussed Cases 2 and 4). PDM1 uses the ED to proportionally weight the data mass in these situations. As a result, PDM1 assigns higher weight to the low data mass between two data instances in Case 2, using the distance between them. In case 4, PDM1 assigns a lower weight to the high data mass between two data instances by weighting it using the normalised distance of them.

PDM2 considers the perceptual effect of region density on the perceived distance between two instances. PDM2 uses log transform of data mass in the region covering two instances as a proxy of region density. It weights the distance between two instances by the log of transform of their data mass in each dimension. As a result, the distance would be perceived differently depending on whether it has been measured in a dense or sparse region.

PDM1 and PDM2 have the characteristic of combining the perceptual effect of data distribution and ED to yield the results that are more consistent with human perception. Additionally, the proposed weighting in PDM1 and PDM2 alleviate the effect of the domination of a few dimensions in total calculation of dissimilarity with ED and m_p.

6.1.3 Improving image retrieval using the perceptual dissimilarity measure

PDM has been proposed based on the strengths of ED and m_p in measuring the dissimilarity. It addresses the limitations with these two measures which has been discussed before. This enables us to exploit PDM in image retrieval to improve the accuracy of the results. We have performed the experiments using three image datasets: eBay, Texture and Corel.

Images in the eBay dataset are represented using HSV colour histograms.

This thesis proposed a weighting system for extracting HSV colour histograms that let each of the HSV components contribute proportional based on their importance for human perception. The weighting system assigns higher weight to hue and equal weights to saturation and value. The results from our experiments show 60% H, 20% S and 20% V components yield the best retrieval results.

To evaluate the retrieval performance of PDM, it has been compared with ED and m_p. Throughout the image retrieval experiments, PDM2 has consistently performed the best, while PDM1 has performed the second best in the Texture and Corel datasets.

PDM1 has not outperformed ED in the eBay dataset. The performance of PDM1 has been affected by the choice of the threshold in defining dense/ sparse regions and small/large distance. The threshold has been defined as the mid point between the minimum and maximum of data masses/ distances between query and all dataset instances. This may raise the limitation that border points will be identified, as in sparse/dense region or small and large distance, although their actual difference does not reflect that distinction.

6.1.4 Improving image clustering using perceptual dissimilarity measure

Following the promising retrieval results of PDM2, this thesis has exploited PDM2 as an alternative to ED to improve clustering results. Due to the restriction of k-means in working with ED, k-medoids have been used aas they are capable of working with arbitrary dissimilarity measures. The performance of PDM2 has been compared with k-medoids (ED) and k-medoids (mp). Also, these results are compared with k-means. This thesis, for the first time, used mp as the dissimilarity measure in the clustering. mp has not achieved good results, as it could outperform the k-means and k-medoids (ED) only in one dataset.

Clustering results show that PDM2 could outperformed the other two dissimilarity measures, ED and m_p with seven wins, three draws and one loss in terms of RI. Also, it outperformed ED and m_p with 10 wins and one draw in terms of F measure. Considering the perceptual effect of data distribution along with ED in PDM2, it has resulted in gathering more similar data instances in clusters.

In addition to this, following the promising results from PDM2, this the-

sis has exploited PDM2 in clustering as an alternative to ED to generate the codebook. We compared the SIFT BOW produced from codebooks generated by k-means(PDM2) and k-means. Results show that the SIFT BOW of the Corel dataset produced from the codebook generated by k-means(PDM2) has yielded better retrievals. This occurred because of the use of k-means(PDM2) for generation of codebook considers the perceptual effect of data distribution along with ED in selecting the visual words.

6.2 Future work

Dissimilarity measurement is fundamental in many applications such as information retrieval, clustering, classification and anomaly detection. PDM2 has shown promising results in both image retrieval and clustering.

In machine learning, classification is the problem of identifying which category a new data instances belongs to, based on the training set of data instances whose category membership is known. Classifiers use a dissimilarity measure to compare a new data with the trained data to identify it's membership. Minkowski distances are widely used in this classification. We will study the performance of PDM2 for classification in the future.

Anomaly detection is an important task in data mining and it is the identification of instances or observations that do not conform to an expected pattern in a given database. Some of the examples are bank fraud, errors in a text and identification of a medical issue. Minkowski distances are popular choices in this field as well, and it would be interesting to study the effect of using PDM2 for anomaly detection in the future.

Previously, we have described the DBSCAN algorithm for clustering in Chapter 2. Identifying core points is fundamental in DBSCAN. ED is the distance measure used to identify the core points in this algorithm. DBSCAN suffers from finding the clusters with varying densities. Kai Ming et al [114] proposed a new method to address this limitation by using mass instead of ED. However, considering data mass and distance together has not been explored in this clustering method. In the future we will propose a variation of PDM that considers both data mass and distance and suits DBSCAN clustering.

Bibliography

[1] Nuclear feature extraction for breast tumor diagnosis - Google Scholar.

[2] Aeberhard, S., Coomans, D., and Vel, O. D. (1993). Improvements to the classification performance of RDA. *Journal of chemometrics*, 7:99–115.

[3] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). OPTICS: ordering points to identify the clustering structure. In *ACM Sigmod record*, volume 28, pages 49–60. ACM.

[4] Arora, J., Khatter, K., and Tushir, M. (2018). Performance assessment for clustering techniques for image segmentation. In *Sensors and Image Processing*, pages 121–129. Springer.

[5] Aryal, S., Ting, K. M., Haffari, G., and Washio, T. (2014). Mp-Dissimilarity: A Data Dependent Dissimilarity Measure. pages 707–712. IEEE.

[6] Aryal, S., Ting, K. M., Washio, T., and Haffari, G. (2017). Data-dependent dissimilarity measure: an effective alternative to geometric distance measures. *Knowledge and Information Systems*, 53(2):479–506.

[7] Au, A. (2018). Sociology and science: the making of a social scientific method. *The American Sociologist*, 49(1):98–115.

[8] Bailey, R. A. (2008). *Design of comparative experiments*, volume 25. Cambridge University Press.

[9] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer.

[10] Beaudet, P. R. (1978). Rotationally invariant image operators. In *Proc. 4th Int. Joint Conf. Pattern Recog, Tokyo, Japan, 1978*.

[11] Benkrama, S., Zaoui, L., and Charrier, C. (2013). Accurate Image Search using Local descriptors into a compact image representation. *International Journal of Computer Science Issues (IJCSI)*, 10(1):220.

[12] Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer.

[13] Blassnigg, M. (2006). The Aegina Academy, A Forum for Art and Science: Light/Image/Illusion. *Leonardo*, 39(3):257–259.

[14] Bosch, A., Zisserman, A., and Muñoz, X. (2006). Scene classification via pLSA. In *European conference on computer vision*, pages 517–530. Springer.

[15] c.Faloutsos, Barber, R., and Flickner, M. Efficient and effective Querying by Image Content | SpringerLink.

[16] Chandrika, L. (2014). Implementation image retrieval and classification with SURF technique. *IJISET-International Journal of Innovative Science, Engineering & Technology*, 1(4).

[17] Chaudhuri, B. B. and Sarkar, N. (1995). Texture segmentation using fractal dimension. *IEEE transactions on pattern analysis and machine intelligence*, 17(1):72–77.

[18] Chen, X., Hu, X., and Shen, X. (2009). Spatial weighting for bag-of-visual-words and its application in content-based image retrieval. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 867–874. Springer.

[19] Chen, Y., Wang, J., and Krovetz, R. (2003). An unsupervised learning approach to content-based image retrieval. pages 197–200 vol.1. IEEE.

[20] Ch'ng, K., Vazquez, N., and Khatami, E. (2018). Unsupervised machine learning account of magnetic transitions in the Hubbard model. *Physical Review E*, 97(1):013306.

[21] Chowdhury, G. G. (2010). *Introduction to modern information retrieval*. Facet publishing.

[22] Christopher, D. M., Prabhakar, R., and Hinrich, S. (2008). Introduction to information retrieval. *An Introduction To Information Retrieval*, 151(177):5.

[23] Cox, D. R. (1958). Planning of experiments.

[24] Cross, G. R. and Jain, A. K. (1983). Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):25–39.

[25] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague.

[26] Datasets, U. Bank notes authentication dataset.

[27] Datasets, U. UCI Machine Learning Repository.

[28] Davis, J. and Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM.

[29] Defays, D. (1977). An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366.

[30] Djenouri, Y., Belhadi, A., Fournier-Viger, P., and Lin, J. C.-W. (2018). Fast and effective cluster-based information retrieval using frequent closed itemsets. *Information Sciences*, 453:154–167.

[31] Duygulu, P., Barnard, K., de Freitas, J. F., and Forsyth, D. A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European conference on computer vision*, pages 97–112. Springer.

[32] Eakins, J. (1999). Content-based image retrieval.

[33] Ertoz, L., Steinbach, M., and Kumar, V. (2002). A new shared nearest neighbor clustering algorithm and its applications. In *Workshop on clustering high dimensional data and its applications at 2nd SIAM international conference on data mining*, pages 105–115.

[34] Fan, J., Gao, Y., Luo, H., and Xu, G. (2004). Automatic image annotation by using concept-sensitive salient objects for image content representation. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 361–368. ACM.

[35] Feng, S. L., Manmatha, R., and Lavrenko, V. (2004). Multiple bernoulli relevance models for image and video annotation. In *null*, pages 1002–1009. IEEE.

[36] Fisher, D. (1996). Iterative optimization and simplification of hierarchical clusterings. *Journal of artificial intelligence research*, 4:147–178.

[37] Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine learning*, 2(2):139–172.

[38] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

[39] Foulger, D. (1979). Credibility and the statistical interaction variable: Speaking up for multiplication as a source of understanding. In *Annual Conference of the Speech Communication Association, San Antonio, TX. Retrieved from http://davis. foulger. info/papers/statisticalInteraction1979. htm*.

[40] Franti, P., Virmajoki, O., and Hautamaki, V. (2006). Fast agglomerative clustering using a k-nearest neighbor graph. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1875–1881.

[41] Förstner, W. and Gülch, E. (1987). A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, pages 281–305. Interlaken.

[42] Goh, K.-S., Chang, E. Y., and Li, B. (2005). Using one-class and two-class SVMs for multiclass image annotation. *IEEE Transactions on Knowledge and Data Engineering*, 17(10):1333–1346.

[43] Gowda, K. C. and Diday, E. (1991). Symbolic clustering using a new dissimilarity measure. *pattern recognition*, 24(6):567–578.

[44] Gowda, K. C. and Krishna, G. (1978). Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition*, 10(2):105–112.

[45] Guo, Y., Şengür, A., Akbulut, Y., and Shipley, A. (2018). An effective color image segmentation approach using neutrosophic adaptive mean shift clustering. *Measurement*, 119:28–40.

[46] Gupta, A. and Jain, R. VISUAL INFORMATION RETRIEVAL. *COMMUNICATIONS OF THE ACM*, 40(5):10.

[47] Han, J. and Ma, K.-K. (2002). Fuzzy color histogram and its use in color image retrieval. *IEEE Transactions on Image Processing*, 11(8):944–952.

[48] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer.

[49] Hartigan, J. A. and Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

[50] Islam, M. M., Zhang, D., and Lu, G. (2008). A geometric method to compute directionality features for texture images. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 1521–1524. IEEE.

[51] Jaccard, J., Turrisi, R., and Jaccard, J. (2003). *Interaction effects in multiple regression*. Number 72. Sage.

[52] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters*, 31(8):651–666.

[53] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

[54] Jain, A. K. and Vailaya, A. (1996). Image retrieval using color and shape. *Pattern recognition*, 29(8):1233–1244.

[55] Jarvis, R. A. and Patrick, E. A. (1973). Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on computers*, 100(11):1025–1034.

[56] Jeong, S., Won, C. S., and Gray, R. M. (2004). Image retrieval using color histograms generated by Gauss mixture vector quantization. *Computer Vision and Image Understanding*, 94(1-3):44–66.

[57] Jumb, V., Sohani, M., and Shrivas, A. (2014). Color image segmentation using K-means clustering and Otsu's adaptive thresholding. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 3(9):72–76.

[58] Jung, Y. (2001). *Design and evaluation of clustering criterion for optimal hierarchical agglomerative clustering*. University of Minnesota.

[59] Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

[60] Keene, O. N. (1995). The log transformation is special. *Statistics in medicine*, 14(8):811–819.

[61] King, B. (1967). Step-wise clustering procedures. *Journal of the American Statistical Association*, 62(317):86–101.

[62] Krumhansl, C. L. (1978). Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. *Psychological Review*, 85(5):445–463.

[63] Kumar, A. and Kumar, S. (2016). Color Image Segmentation via Improved K-Means Algorithm. *image*, 7(3).

[64] Larson, R. (2016). *Elementary linear algebra*. Nelson Education.

[65] Lazebnik, S., Schmid, C., and Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278.

[66] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *null*, pages 2169–2178. IEEE.

[67] Lee, K.-L. and Chen, L.-H. (2005). An efficient computation method for the texture browsing descriptor of MPEG-7. *Image and Vision Computing*, 23(5):479–489.

[68] Lei, L., Peng, J., and Yang, B. (2013). Image retrieval based on YCbCr color histogram. In *2013 IEEE 12th International Conference on Cognitive Informatics and Cognitive Computing*, pages 483–488.

[69] Li, J. and Wang, J. Z. (2003). Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1075–1088.

[70] Lindeberg, T. (1994). Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270.

[71] Liu, F. and Picard, R. W. (1996). Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 18(7):722–733.

[72] Liu, Y., Zhang, D., Lu, G., and Ma, W.-Y. (2007). A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282.

[73] Logothetis, N. K. and Sheinberg, D. L. (1996). Visual object recognition. *Annual review of neuroscience*, 19(1):577–621.

[74] Long, F., Zhang, H., and Feng, D. D. (2003). Fundamentals of content-based image retrieval. In *Multimedia Information Retrieval and Management*, pages 1–26. Springer.

[75] Lowe, D. (1999). Object recognition from local scale-invariant features. pages 1150–1157 vol.2. IEEE.

[76] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[77] Lu, Z.-M., Li, S.-Z., and Burkhardt, H. (2006). A content-based image retrieval scheme in JPEG compressed domain. *International Journal of Innovative Computing, Information and Control*, 2(4):831–839.

[78] Ma, W.-Y. and Manjunath, B. S. NeTra: A toolbox for navigating large image databases. page 15.

[79] Malik, F. and Baharudin, B. (2013). Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the DCT domain. *Journal of king saud university-computer and information sciences*, 25(2):207–218.

[80] Manjunath, B., Salembier, P., and Sikora, T. (2002). *Introduction to MPEG 7: Multimedia Content Description Language. Ed*. Wiley.

[81] Materka, A. and Strzelecki, M. (1998). Texture analysis methods–a review. *Technical university of lodz, institute of electronics, COST B11 report, Brussels*, pages 9–11.

[82] Mezaris, V., Kompatsiaris, I., and Strintzis, M. (2003). An ontology approach to object-based image retrieval. volume 3, pages II–511–14. IEEE.

[83] Mikolajczyk, K. and Schmid, C. (2001). Indexing based on scale invariant interest points. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 525–531. IEEE.

[84] Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86.

[85] Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359.

[86] Murthy, V., Vamsidhar, E., Kumar, J. S., and Rao, P. S. (2010). Content based image retrieval using Hierarchical and K-means clustering techniques. *International Journal of Engineering Science and Technology*, 2(3):209–212.

[87] Ng, H. P., Ong, S. H., Foong, K. W. C., Goh, P. S., and Nowinski, W. L. (2006). Medical image segmentation using k-means clustering and improved watershed algorithm. In *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*, pages 61–65. IEEE.

[88] Ng, R. T. and Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, 14(5):1003–1016.

[89] Nowak, E., Jurie, F., and Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In *European conference on computer vision*, pages 490–503. Springer.

[90] Ojala, T., Pietikainen, M., and Harwood, D. (1994). Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 582–585. IEEE.

[91] Osborne, J. W. (2010). Improving your data transformations: Applying the Box-Cox transformation. *Practical Assessment, Research & Evaluation*, 15(12):2.

[92] Park, S. B., Lee, J. W., and Kim, S. K. (2004). Content-based image classification using a neural network. *Pattern Recognition Letters*, 25(3):287–300.

[93] Pass, G. and Zabih, R. (1999). Comparing images using joint histograms. *Multimedia Systems*, 7(3):234–240.

[94] Passalis, N. and Tefas, A. (2018). Information clustering using manifold-based optimization of the bag-of-features representation. *IEEE transactions on cybernetics*, 48(1):52–63.

[95] Pentland, A., Picard, R. W., and Sclaroff, S. (1995). *Photobook: Content-Based Manipulation of Image Databases*.

[96] Perrotta, C. and Williamson, B. (2018). The social life of Learning Analytics: cluster analysis and the 'performance'of algorithmic education. *Learning, Media and Technology*, 43(1):3–16.

[97] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

[98] Saad, M. H., Saleh, H. I., Konbor, H., and Ashour, M. (2011). Image retrieval based on integration between YCbCr color histogram and texture feature. *International Journal of Computer Theory and Engineering*, 3(5):701–706.

[99] Schalkoff, R. J. (1989). *Digital image processing and computer vision*, volume 286. Wiley New York.

[100] Selvarajah, S. and Kodituwakku, S. R. (2011). Analysis and comparison of texture features for content based image retrieval. *International Journal of Latest Trends in Computing*, 2(1).

[101] Sergyan, S. (2008). Color histogram features based image classification in content-based image retrieval systems. In *Applied Machine Intelligence and Informatics, 2008. SAMI 2008. 6th International Symposium on*, pages 221–224. IEEE.

[102] Sethi, I. K., Coman, I. L., and Stan, D. (2001). Mining association rules between low-level image features and high-level concepts. pages 279–290.

[103] Sharif, U., Mehmood, Z., Mahmood, T., Javid, M. A., Rehman, A., and Saba, T. (2018). Scene analysis and search using local features and support vector machine for effective content-based image retrieval. *Artificial Intelligence Review*, pages 1–25.

[104] Sharma, M., Purohit, G. N., and Mukherjee, S. (2018). Information retrieves from brain MRI images for tumor detection using hybrid technique K-means and artificial neural network (KMANN). In *Networking Communication and Data Knowledge Engineering*, pages 145–157. Springer.

[105] Shekhar, R. and Jawahar, C. V. (2012). Word image retrieval using bag of visual words. In *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, pages 297–301. IEEE.

[106] Silakari, S., Motwani, M., and Maheshwari, M. (2009). Color image clustering using block truncation algorithm. *arXiv preprint arXiv:0910.1849*.

[107] Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE.

[108] Sneath, P. H. and Sokal, R. R. (1973). *Numerical taxonomy. The principles and practice of numerical classification*.

[109] Sural, S., Vadivel, A., and Majumdar, A. K. (2005). Histogram generation from the HSV color space. In *Encyclopedia of Information Science and Technology, First Edition*, pages 1333–1337. IGI Global.

[110] Surv, C., Murty, M. N., Flynn, P. J., Jain, A. K., and Flynn, P. J. (1999). *And*.

[111] Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International journal of computer vision*, 7(1):11–32.

[112] Tamura, H., Mori, S., and Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions on Systems, man, and cybernetics*, 8(6):460–473.

[113] Teague, M. R. (1980). Image analysis via the general theory of moments. *JOSA*, 70(8):920–930.

[114] Ting, K. M., Zhu, Y., Carman, M., Zhu, Y., and Zhou, Z.-H. (2016). Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1205–1214. ACM.

[115] Town, C. and Sinclair, D. (2000). *Content based image retrieval using semantic visual categories*. Society of Manufacturing Engineers.

[116] Tuytelaars, T. (2010). Dense interest points. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2281–2288. IEEE.

[117] Tversky, A. (1977). Features of similarity. *Psychological Review*, 84(4):327–352.

[118] Vailaya, A., Figueiredo, M. A., Jain, A. K., and Zhang, H.-J. (2001). Image classification for content-based indexing. *IEEE transactions on image processing*, 10(1):117–130.

[119] Van De Weijer, J., Schmid, C., and Verbeek, J. (2007). Learning color names from real-world images. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

[120] Van Gemert, J. C. (2011). Exploiting photographic style for category-level image classification by generalizing the spatial pyramid. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 14. ACM.

[121] Wang, J. Z., Li, J., and Wiederhold, G. (2001). SIMPLIcity: semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963.

[122] Wang, L. and He, D.-C. (1990). Texture classification using texture spectrum. *Pattern Recognition*, 23(8):905–910.

[123] Wang, X., Han, T. X., and Yan, S. (2009). An HOG-LBP human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE.

[124] Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

[125] Warton, D. I. and Hui, F. K. (2011). The arcsine is asinine: the analysis of proportions in ecology. *Ecology*, 92(1):3–10.

[126] Wishart, D. (1969). 256. Note: An algorithm for hierarchical classifications. *Biometrics*, pages 165–170.

[127] Wolberg, W. H. and Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the national academy of sciences*, 87(23):9193–9196.

[128] Yang, C., Dong, M., and Fotouhi, F. (2005). Image content annotation using bayesian framework and complement components analysis. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–1193. IEEE.

[129] Yang, J., Jiang, Y.-G., Hauptmann, A. G., and Ngo, C.-W. (2007). Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206. ACM.

[130] Yavlinsky, A., Schofield, E., and Rüger, S. (2005). Automated image annotation using global features and robust nonparametric density estimation. In *International Conference on Image and Video Retrieval*, pages 507–517. Springer.

[131] Yeung, K. Y. and Ruzzo, W. L. (2001). Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774.

[132] Yousuf, M., Mehmood, Z., Habib, H. A., Mahmood, T., Saba, T., Rehman, A., and Rashid, M. (2018). A novel technique based on visual words fusion analysis of sparse features for effective content-based image retrieval. *Mathematical Problems in Engineering*, 2018.

[133] Yu, J., Qin, Z., Wan, T., and Zhang, X. (2013). Feature integration analysis of bag-of-features model for image retrieval. *Neurocomputing*, 120:355–364.

[134] Zhang, D., Islam, M. M., and Lu, G. (2012). A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362.

[135] Zhang, D. and Lu, G. (2003). Evaluation of similarity measurement for image retrieval. In *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*, volume 2, pages 928–931 Vol.2.

[136] Zhang, D. and Lu, G. (2004). Review of shape representation and description techniques. *Pattern recognition*, 37(1):1–19.

[137] Zhang, D., Wong, A., Indrawan, M., and Lu, G. (2000). Content-based image retrieval using Gabor texture features. *IEEE Transactions PAMI*, pages 13–15.

[138] Zhang, J. (1992). Selecting typical instances in instance-based learning. In *Machine Learning Proceedings 1992*, pages 470–479. Elsevier.

[139] Zhang, K. (1995). Algorithms for the constrained editing distance between ordered labeled trees and related problems. *Pattern Recognition*, 28(3):463–474.

[140] Zhang, R., Zhang, Z., Li, M., Ma, W.-Y., and Zhang, H.-J. (2005). A probabilistic semantic model for image annotation and multimodal image retrieval. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 846–851. IEEE.

[141] Zhang, S., Tian, Q., Hua, G., Huang, Q., and Li, S. (2009). Descriptive visual words and visual phrases for image applications. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 75–84. ACM.

[142] Zhou, X. S. and Huang, T. S. (2000). CBIR: From Low-Level Features to HighLevel Semantics. In *Proc. SPIE Image and Video Communication and Processing*, pages 24–28.