Chapter 4
Similarity

Abstract How to account for similarity between two data instances is fundamental for any data management, retrieval, and analysis tasks. This chapter[†] shows that traditional distance functions such as the Minkowski metric and weighted Minkowski are not effective in accounting similarity. Through mining a large set of visual data, we discovered a perceptual distance function, which works much more effectively for finding similar images than the Minkowski family. We call the discovered function *dynamic partial function* (DPF). We demonstrate the effectiveness of DPF through empirical studies and explain why it works better by cognitive theories.

Keywords Cognitive theory · Distance function · DPF · Perceptual similarity

4.1 Introduction

To achieve effective management, retrieval, and analysis, an image/video system must be able to accurately characterize and quantify perceptual similarity. However, a fundamental challenge—how to measure perceptual similarity—remains largely unanswered. Various distance functions, such as the Minkowski metric [2], earth mover distance [3], histogram Cosine distance [4], and fuzzy logic [5], have been used to measure similarity between feature vectors representing images (and hence video frames). Unfortunately, our experiments show that they frequently overlook obviously similar objects and hence are not adequate for measuring perceptual similarity.

[†] © Springer, 2003. This chapter is a minor revision of the author's work with Beitao Li and Yi-Leh Wu [1] published in ACM Multimedia Systems'03. Permission to publish this chapter is granted under copyright license 2591350681815.

E. Y. Chang, *Foundations of Large-Scale Multimedia Information Management and Retrieval*, DOI: 10.1007/978-3-642-20429-6_4,

Quantifying perceptual similarity is a difficult problem. Indeed, we may be decades away from fully understanding how human perception works (as we have discussed in Chap. 2). In this chapter, we show how we employed a data-driven approach to analyze the characteristics of similar data instances, and how that led to our formulation of a new distance function. Our mining hypothesis is this: suppose most of the similar data instances can be clustered in a feature space. We can then claim with high confidence that (1) the feature space can adequately capture the characteristics of those data instances, and (2) the distance function used for clustering data instances in that feature space can accurately model similarity. Our target task was to formulate a distance function that can keep similar data instances in the same cluster, while keeping dissimilar ones away.

We performed our *discovery through mining* operation in two stages. In the first stage, we isolate the distance function factor (we used the Euclidean distance) to find a reasonable feature set. In the second stage, we froze the features to discover a perceptual distance function that could better cluster similar data instances in the feature space. We call the discovered function *dynamic partial distance function* (DPF). When we empirically compare DPF to Minkowski-type distance functions in image retrieval, video shot-transition detection, and new-article near-duplicate detection, DPF performs significantly better.

Similarity is one of the central theoretical constructs in psychology [6, 7], probably related to human survival instincts. We believe that being able to quantify similarity accurately must also hold a central place in theories of information management and retrieval. Our excitement in discovering DPF does not arise merely from the practical effectiveness we found in three applications. More importantly, we find that DPF has roots in cognitive psychology. While we will discuss the links between DPF and some *similarity theories* in cognitive psychology in Sect. 4.5, let us use an example to explain both the *dynamic* and *partial* aspects. Suppose we are asked to name two places that are similar to England. Among several possibilities, Scotland and New England could be two reasonable answers. However, the respects England is similar to Scotland differ from those in which England is similar to New England. If we use the shared attributes of England and Scotland to compare England and New England, the latter pair might not be similar, and vice versa. Objects can be similar to the query object in different respects. A distance function using a fixed set of respects cannot capture objects that are similar in different sets of respects. A distance function for measuring a pair of objects is formulated only after the objects are compared, not before the comparison is made. The respects for the comparison are activated in this formulation process. The activated respects are more likely to be those that can support coherence between the compared objects.

The rest of his chapter is organized as follows:

1. We first show our data mining process to determine a reasonable feature space. In that feature space, we find distinct patterns of similar and dissimilar images, which lead to the discovery of DPF.
2. We derive DPF based on the observed patterns, and we provide methods for finding the optimal settings for the function's parameters.

3. Through case studies, we demonstrate that DPF is very effective in finding images that have been transformed by rotation, scaling, downsampling, and cropping, as well as images that are perceptually similar to the query image. Applying DPF to video shot-transition detection and new-article near-duplicate detection, we show that DPF is also more effective than the Minkowski metric.

4.2 Mining Image Feature Set

This section depicts how the mining dataset was constructed in three steps: testbed setup (Sect. 4.2.1), feature extraction (Sect. 4.2.2), and feature selection (Sect. 4.2.3).

4.2.1 Image Testbed Setup

To ensure that sound inferences can be drawn from our mining results, we carefully construct the dataset. First, we prepare for a dataset that is comprehensive enough to cover a diversified set of images. To achieve this goal, we collect 60,000 JPEG images from Corel CDs and from the Internet. Second, we define "similarity" in a slightly restrictive way so that individuals' subjectivity can be excluded.[1] For each image in the 60,000-image set, we perform 24 transformations (described shortly), and hence form 60,000 similar-image sets. The total number of images in the testbed is 1.5 million.

The 24 image transformations we perform include the following:

1. Scaling.
    - *Scale up then down*. We scale each image up by 4 and 16 times, respectively, and then scale it back to the original size.
    - *Scale down then up*. We scale each image down by factors of 2, 4, and 8, respectively, then scale it back to the original size.
2. *Downsampling*. We downsample each image by seven different percentages: 10–50, 70, and 90%.
3. *Cropping*. We evenly remove the outer borders to reduce each image by 5%, 10–70%, respectively, and then scale it back up to the original size.
4. *Rotation*. We rotate each image by 90, 180, and 270°.

[1] We have considered adding images taken under different lighting conditions or with different camera parameters. We decided not to include them because they cannot be automatically generated from an image. Nevertheless, our experimental results (see Sect. 4.4) show that the perceptual distance function discovered during the mining process can be used effectively to find other perceptually similar images. In other words, our testbed consists of a good representation of similar images, and the mining results (i.e., training results) can be generalized to testing data consisting of perceptually similar images produced by other methods.

Table 4.1 Multi-resolution color feature

Filter name	Resolution	Representation
Masks	Coarse	Appearance of culture colors
Spread	Coarse	Spatial concentration of a color
Elongation	Coarse	Shape of a color
Histograms	Medium	Distribution of colors
Average	Medium	Similarity comparison within the same culture color
Variance	Fine	Similarity comparison within the same culture color

5. *Format transformation*. We obtain the GIF version of each JPEG image.

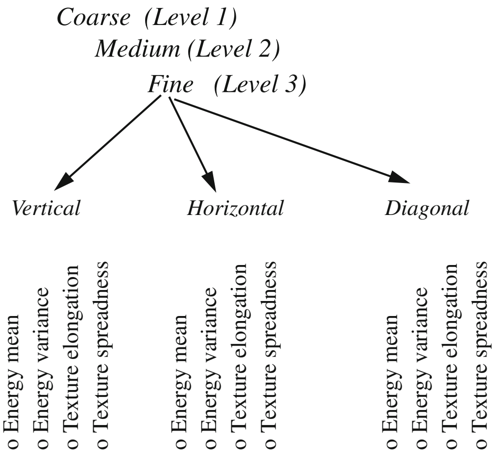
4.2.2 Feature Extraction

To describe images, we must find a set of features that can represent those images adequately. Finding a universal representative feature set can be very challenging, since different imaging applications may require different feature sets. For instance, the feature set that is suitable for finding tumors may not be effective for finding landscape images, and vice versa. However, we believe that by carefully separating perception from intelligence (i.e., domain knowledge), we can identify meaningful perceptual features. Chapter 2 shows both model-based and data-driven approaches for extracting features. We used a data-driven approach in this study to find useful features from a large set of feature candidates.

Psychologists and physiologists divide the human visual system into two parts: the *perceiving part*, and the *inference part* [8]. The perceiving part receives photons, converts electrical signals into neuro-chemical signals, and delivers the signals to our brains. The inference part then analyzes the perceived data based on our knowledge and experience. A baby and an adult have equal capability for perceiving, but differing capability for understanding what is perceived. Among adults, specially trained ones can interpret an X-ray film, but the untrained cannot. In short, the perceiving part of our visual system is task-independent, so it can be characterized in a domain-independent manner.

We extract features such as color, shape, and texture from images. In the color channel, we characterize color in multiple resolutions. We first divide color into 12 color bins including 11 bins for culture colors and one bin for outliers [9]. At the coarsest resolution, we characterize color using a color mask of 12 bits. To record color information at finer resolutions, we record nine additional features for each color. These nine features are color histograms, color means (in H, S and V channels), color variances (in H, S and V channels), and two shape characteristics: elongation and spreadness. Color elongation characterizes the shape of a color, and spreadness characterizes how that color scatters within the image [10]. Table 4.1 summarizes color features in coarse, medium and fine resolutions.

Fig. 4.1 Multi-resolution texture features



Texture is an important characteristic for image analysis. Studies [11–14] have shown that characterizing texture features in terms of structuredness, orientation, and scale (coarseness) fits well with models of human perception. From the wide variety of texture analysis methods proposed in the past, we choose a discrete wavelet transformation (DWT) using quadrature mirror filters [13] because of its computational efficiency.

Each wavelet decomposition on a 2D image yields four subimages: a$\frac{1}{2} \times \frac{1}{2}$ scaled-down image of the input image and its wavelets in three orientations: horizontal, vertical and diagonal. Decomposing the scaled-down image further, we obtain the tree-structured or wavelet packet decomposition. The wavelet image decomposition provides a representation that is easy to interpret. Every subimage contains information of a specific scale and orientation and also retains spatial information. We obtain nine texture combinations from subimages of three scales and three orientations. Since each subimage retains the spatial information of texture, we also compute elongation and spreadness for each texture channel. Figure 4.1 summarizes texture features.

4.2.3 Feature Selection

Once the testbed is set up and relevant features extracted, we fix the distance function to examine various feature combinations. For the time being, we employ the Euclidean distance function to quantify the similarity between two feature vectors. We use the Euclidean function because it is commonly used, and it achieves acceptable results. (However, we will offer a replacement distance function for the Euclidean distance in Sect. 4.3.)

Using different feature combinations, we employ the Euclidean function to find the distance rankings of the 24 images that are similar to the original image (i.e., the query image). If a feature set can adequately capture the characteristics of images, the 24 similar images should be among those closest to the query image. (In an ideal case, the 24 similar images should be the 24 images closest to the query image.)

Our experiments reveal that when only individual features (e.g., color histograms, color elongation, and color spreadness) are employed, the distance function cannot easily capture the similar images even among the top-100 nearest neighbors. For a top-100 query, all individual features suffer from a dismal recall lower than 30%. When we combine all color features, the top-100 recall improves slightly, to 45%. When both color and texture features are used, the recall improves to 60%.

At this stage, we can go in either of two directions to improve recall. One, we can add more features, and two, we can replace the Euclidean distance function. We will consider adding additional features in our future work. In this chapter, we focus on finding a perceptual distance function that improves upon the Euclidean Function.

4.3 Discovering the Dynamic Partial Distance Function

We first examine two most popular distance functions used for measuring image similarity: Minkowski function and weighted Minkowski function. Building upon those foundations, we explain the heuristics behind our new distance function—*Dynamic Partial Function* (*DPF*).

4.3.1 Minkowski Metric and its Limitations

The Minkowski metric is widely used for measuring similarity between objects (e.g., images). Suppose two objects X and Y are represented by two p dimensional vectors $(x_1, x_2, \dots, x_p)$ and $(y_1, y_2, \dots, y_p)$, respectively. The Minkowski metric $d(X, Y)$ is defined as

$$d(X, Y) = \left(\sum_{i=1}^{p} |x_i - y_i|^r \right)^{\frac{1}{r}}, \tag{4.1}$$

where r is the Minkowski factor for the norm. Particularly, when r is set as 2, it is the well-known Euclidean distance; when r is 1, it is the Manhattan distance (or L_1 distance). An object located a smaller distance from a query object is deemed more similar to the query object. Measuring similarity by the Minkowski metric is based on one assumption: that similar objects should be similar to the query object in all dimensions. This assumption is true for abstract points in mathematical space. However, for multimedia objects (e.g., images), this assumption may not hold.

Human perception of similarity may not strictly follow the rules of mathematical space [7].

A variant of the Minkowski function, the weighted Minkowski distance function, has also been applied to measure image similarity. The basic idea is to introduce weighting to identify important features. Assigning each feature a weighting coefficient w_i $(i = 1, 2, \ldots, p)$, the weighted Minkowski distance function is defined as:

$$d_w(X, Y) = \left(\sum_{i=1}^{p} w_i |x_i - y_i|^r \right)^{\frac{1}{r}}. \tag{4.2}$$

By applying a static weighting vector for measuring similarity, the weighted Minkowski distance function assumes that similar images resemble the query images in the same features. For example, when the function weights color features high and ignores texture features, this same weighting is applied to all pair-wise distance computation with the query image. We will show shortly that this *fixed* weighting method is restrictive in finding similar objects of different kinds.

We can summarize the assumptions of the traditional distance functions as follows:

- *Minkowski function*. All similar images must be similar in all features.
- *Weighted Minkowski function*. All similar images are similar in the same way (e.g., in the same set of features).

We questioned the above assumptions upon observing how similar objects are located in the feature space. For this purpose, we carried out extensive data mining work on a 1.5 M-image dataset introduced in Sect. 4.2. To better discuss our findings, we introduce a term we have found useful in our data mining work. We define the *feature distance* on the ith feature as

$$\delta_i = |x_i - y_i|. \quad (i = 1, 2, \ldots, p)$$

The expressions of (4.1) and (4.2) can be simplified into

$$d(X, Y) = \left(\sum_{i=1}^{p} {\delta_i}^r \right)^{\frac{1}{r}} \quad \text{and} \quad d_w(X, Y) = \left(\sum_{i=1}^{p} w_i {\delta_i}^r \right)^{\frac{1}{r}}.$$

In our mining work, we first tallied the feature distances between similar images (denoted as δ^+), and also those between dissimilar images (denoted as δ^-). Since we normalized feature values to be between zero and one, the ranges of both δ^+ and δ^- are between zero and one. Figure 4.2 presents the distributions of δ^+ and δ^-. The x-axis shows the possible value of δ, from zero to one. The y-axis (in logarithmic scale) shows the percentage of the features at different δ values.

The figure shows that δ^+ and δ^- have different distribution patterns. The distribution of δ^+ is much skewed toward small values (Fig. 4.2a), whereas the distribution of δ^- is more evenly distributed (Fig. 4.2b). We can also see from Fig. 4.2a that a

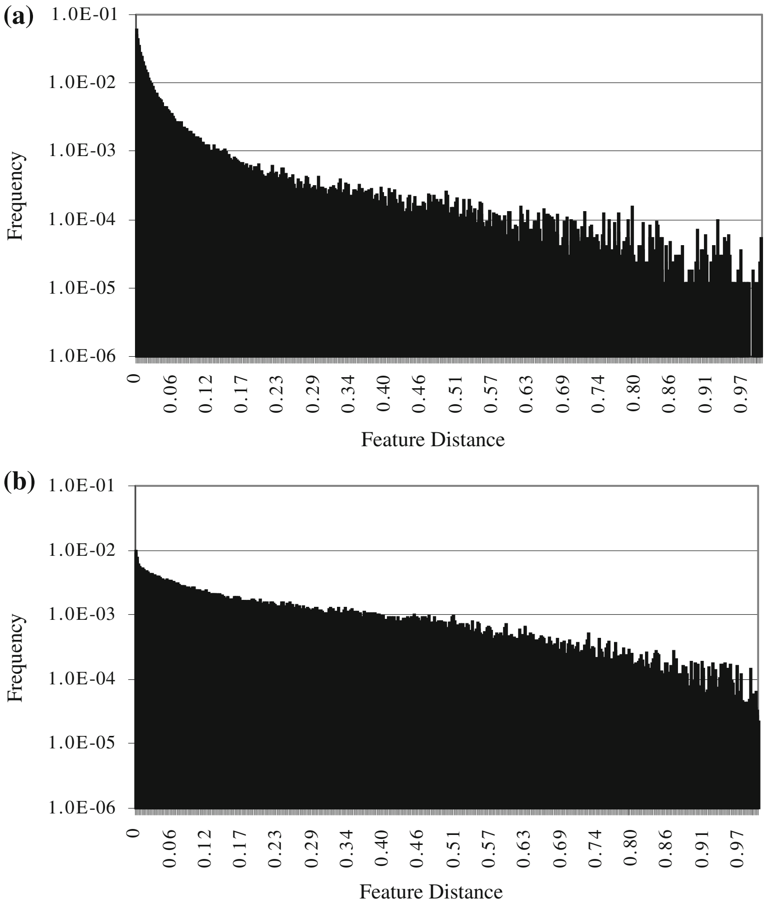


Fig. 4.2 The distributions of feature distances. **a** similar images, **b** Dissimilar images

moderate portion of δ^+ is in the high value range ($\geq$0.5), which indicates that similar images may be quite dissimilar in some features. From this observation, we infer that the assumption of the Minkowski metric is inaccurate. Similar images are not necessarily similar in all features.

Furthermore, we examined whether similar images resemble the query images in the same way. We tallied the *distance* (δ^+) of the 144 features for different kinds of image transformations. Figure 4.3 presents four representative transformations: GIF, cropped, rotated, and scaled. The x-axis of the figure depicts the feature numbers, from 1 to 144. The first 108 features are various color features, and the last 36 are texture features. The figure shows that various similar images can resemble the query images in very different ways. GIF images have larger δ^+ in color features (the first 108 features) than in texture features (the last 36 features). In contrast, cropped images have larger δ^+ in texture features than in color features. For rotated images,

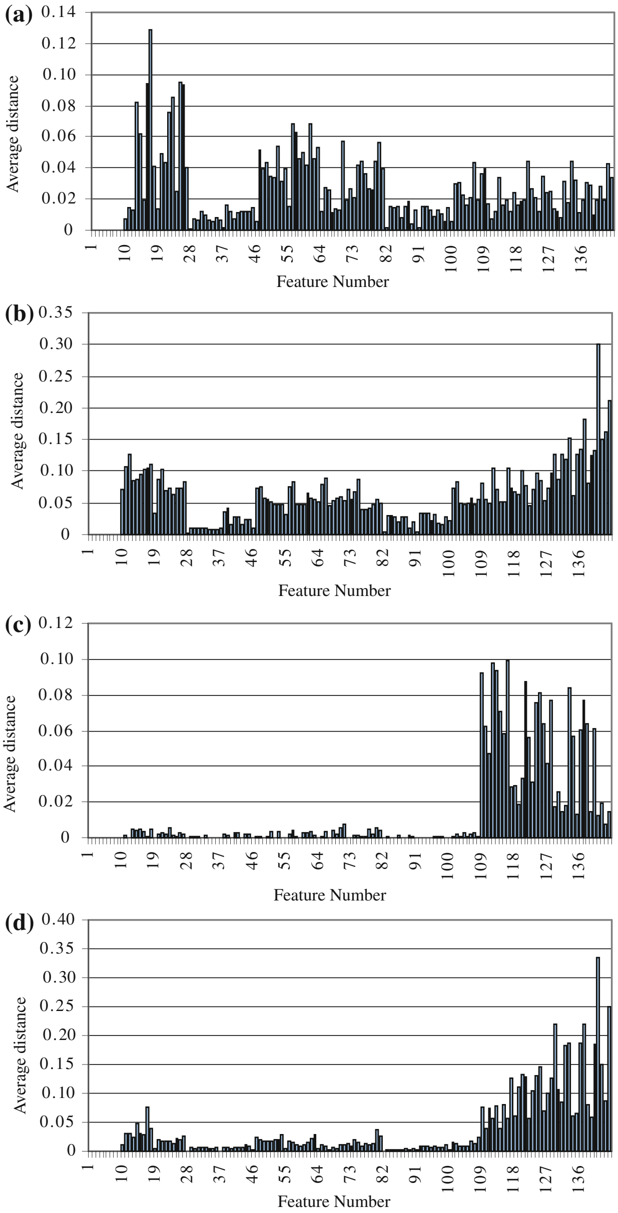


Fig. 4.3 The average feature distances. **a** Gif images **b** Cropped images **c** Rotational images **d** Scaled images

the δ^+ in colors comes close to zero, although its texture feature distance is much greater. A similar pattern appears in the scaled and the rotated images. However, the magnitude of the δ^+ of scaled images is very different from that of rotated images.

Our observations show that the assumptions made by the Minkowski and weighted Minkowski function are questionable.

1. Similar images do not resemble the query images in all features. Figure 4.2 shows that similar images different from a query image in many respects.
2. Images similar to the query images can be similar in differing features. Figure 4.3 shows that some images resemble the query image in texture, others in color.

The above observations not only refute the assumptions of Minkowski-type distance functions, but also provide hints as to how a good distance function would work. The first point is that a distance function does not need to consider all features equally, since similar images may match only some features of the query images. The second point is that a distance function should weight features dynamically, since various similar images may resemble the query image in differing ways. These points lead to the design of the *dynamic partial* distance function.

4.3.2 Dynamic Partial Distance Function

Based on the observations explained above, we designed a distance function to better represent the perceptual similarity. Let $\delta_i = |x_i - y_i|$, for $i = 1, 2, \ldots, p$. We first define sets Δ_m as

$$\Delta_m = \{\text{The smallest } m\ \delta' s \text{ of } (\delta_1, \ldots, \delta_p)\}.$$

Then we define the DPF as

$$d(m, r) = \left(\sum_{\delta_i \in \Delta_m} {\delta_i}^r \right)^{\frac{1}{r}}. \tag{4.3}$$

DPF has two adjustable parameters: m and r. Parameter m can range from 1 to p. When $m = p$, it degenerates to the Minkowski metric. When $m < p$, it counts only the smallest m feature distances between two objects, and the influence of the $(p-m)$ largest feature distances is eliminated. Note that DPF dynamically selects features to be considered for different pairs of objects. This is achieved by the introduction of Δ_m, which changes dynamically for different pairs of objects. In Sect. 4.4, we will show that if a proper value of m is chosen, it is possible to make similar images aggregate more compactly and locate closer to the query images, simultaneously keeping the dissimilar images away from the query images. In other words, similar and dissimilar images are better separated by DPF than by earlier methods.

The idea employed by DPF can also be generalized to improve the weighted Minkowski distance function. We modify the weighted Minkowski distance by defining the weighted DPF as

$$d_w(m, r) = \left(\sum_{\delta_i \in \Delta_m} w_i {\delta_i}^r \right)^{\frac{1}{r}}. \tag{4.4}$$

In Sect. 4.4, we will show that DPF can also improve the retrieval performance of the weighted Minkowski distance function.

4.3.3 Psychological Interpretation of Dynamic Partial Distance Function

The *Just Noticeable Difference* (JND) is the smallest difference between two stimuli that a person can detect. B. Goldstein [15] uses the following example to illustrate the JND: A person can detect the difference between a 100 g weight and a 105 g weight but cannot detect a smaller difference, so the JND for this person is 5 g. For our purpose, we introduce a new term. The term is *just not the same* (JNS). Using the same weight example, we may say that a 100 g weight is just not the same as a weight that is more than 120 g. So the JNS is 20 g. When the weight is between 105 and 120 g, we say that the weight is similar to a 100 g weight (to a degree).

Now, let us apply JND and JNS to our color perception. We can hardly tell the difference between *deep sky blue* (whose RGB is 0,191,255) and *dodger blue* (whose RGB is 30,144,255). The perceptual difference between these two colors is below JND. On the other hand, we can tell that blue is different from green, and yellow is different from red. In both cases, the colors are perceived as JNS.

For an image search engine, JND and JNS indicate that using Euclidean distance for measuring color difference may not be appropriate. First, JND reveals that when the difference between two colors is insignificant, the two colors are perceived as the same. Second, JNS reveals that when the difference is significant, we say two colors are not the same, and it may not be meaningful to account the full magnitude of difference. (E.g., saying that blue is more different from red than from green is meaningless for our purpose.)

The JND and JNS values for each feature can be obtained only through extensive psychological experiments. Moreover, different people may have different subjective values of JND and JNS. Being aware of the practical difficulty of obtaining exact values of JND and JNS for each feature, DPF addresses this issue reasoning as follows:

- JND is not vital for designing a perceptual distance function, since a feature distance below JND usually is very small and has little effect on the aggregated distance. It does not make much difference to consider it as zero or as a small value.
- JNS is vital for designing a perceptual distance function. A feature distance greater than JNS can introduce significant noise on the aggregated distance.

Though it is difficult to obtain the exact value of JNS for each feature, DPF circumvents this difficulty by taking a probabilistic view: the largest $(p-m)$ feature distances are likely to exceed their JNS values. Removing the $(p-m)$ largest feature distances from the final aggregated distance between objects can reduce the noise above JNS. First, the distances of the $(p-m)$ features are all scaled back to their respective JNS. Second, removing these JNS from the aggregated distance does not affect the relative distance between objects.

In short, DPF considers only the m smallest feature distances and does not count the $(p - m)$ largest feature distances. In this sense, DPF provides a good approximation to consider JND and JNS.

4.4 Empirical Study

We conducted an empirical study to examine the effectiveness of DPF. Our experiments consisted of three parts.

1. We compared DPF with the Euclidean distance function and L_1 distance function, the most widely used similarity functions in image retrieval. We also compared DPF with the histogram Cosine[2] distance function, which is also commonly used in information retrieval [4, 16] (Sect. 4.4.1).
2. We tested whether DPF can be generalized to video shot-transition detection, the foundation of video analysis and retrieval applications (Sect. 4.2.2).
3. We experimented DPF with a set of news articles to identify near-duplicates.
4. In addition to the unweighted versions, we also examined whether the weighted DPF is effective for enhancing the performance of the weighted Minkowski distance function (Sect. 4.4.4).

4.4.1 Image Retrieval

Our empirical study of image retrieval consisted of two parts: training and testing. In the training part, we used the 1.5M-image dataset to predict the optimal m value for DPF. In the testing part, we set DPF with the optimal m value, and tested it on an independently constructed 50K-image dataset to examine its effectiveness.

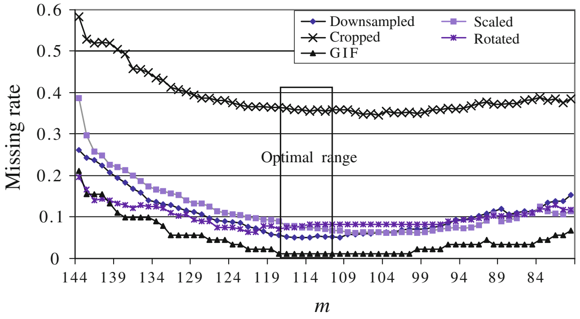
4.4.1.1 Predicting m Through Training

The design goal of DPF is to better separate similar images from dissimilar ones. To meet this design goal, we must judiciously select parameter m. (We take the Euclidean distance function as the baseline, thus we set $r = 2$ for both DPF and the Minkowski distance function.) Alternatively, we can set a JND threshold for selecting features

[2] The Cosine metric computes the direction difference between two feature vectors. Specifically, given two feature vectors $\mathbf{x}$ and $\mathbf{y}$, the Cosine metric is given as

$$D = 1 - \frac{\mathbf{x}^T \mathbf{y}}{|\mathbf{x}||\mathbf{y}|}.$$

Fig. 4.4 Training for the optimal *m* value



to be considered by DPF. If we find enough number of features between two images having a difference below JND, we can say the pair to be similar. One advantage of the threshold method is that the value of *m* is also pairwise dependent. Please see [17] for this threshold method.

To find the optimal *m* value, we used the 60,000 original images to perform queries. we applied DPF of different *m* values to the 1.5 M-image dataset. The 24 images with the shortest distance from each query image were retrieved. For each of the five similar-image categories (i.e., GIF, cropped, downsampled, rotated, or scaled), we observed how many of them failed to appear in the top-24 results. Figure 4.4 presents the average rate of missed images for each similar-image category. The figure shows that when *m* is reduced from 144 to between 110 and 118, the rates of missing are near their minimum for all five similar-image categories. (Note that when $m = 144$, DPF degenerates into the Euclidean function.) DPF outperforms the Euclidean distance function by significant margins for all similar-image categories.

To investigate why DPF works effectively when *m* is reduced, we tallied the distances from these 60,000 queries to their similar images and their dissimilar images, respectively. We then computed the average and the standard deviation of these distances. We denote the average distance of the similar images to their queries as μ_d^+, of the dissimilar images as μ_d^-. We denote the standard deviation of the similar images' distances as σ_d^+, of the dissimilar images as σ_d^-.

Figure 4.5 depicts the effect of *m* (in the *x*-axis) on μ_d^+, μ_d^-, σ_d^+, and σ_d^-. Figure 4.5a shows that as *m* becomes smaller, both μ_d^+ and μ_d^- decrease. The average distance of similar images (μ_d^+), however, decreases at a faster pace than that of dissimilar images (μ_d^-). For instance, when we decrease *m* from 144 to 130, μ_d^+ decreases from 1.0 to about 0.3, a 70% decrease, whereas μ_d^- decreases from 3.2 to about 2.0, a 38% decrease. This gap indicates μ_d^+ is more sensitive to the *m* value than μ_d^-. Figure 4.5b shows that the standard deviations σ_d^+ and σ_d^- observe the same trend as the average distances do. When *m* decreases, similar images become more compact in the feature space at a faster pace than dissimilar images do.

To provide more detailed information, Fig. 4.6 depicts the distance distributions at four different *m* values. Figure 4.6a shows that when $m = 144$, a significant

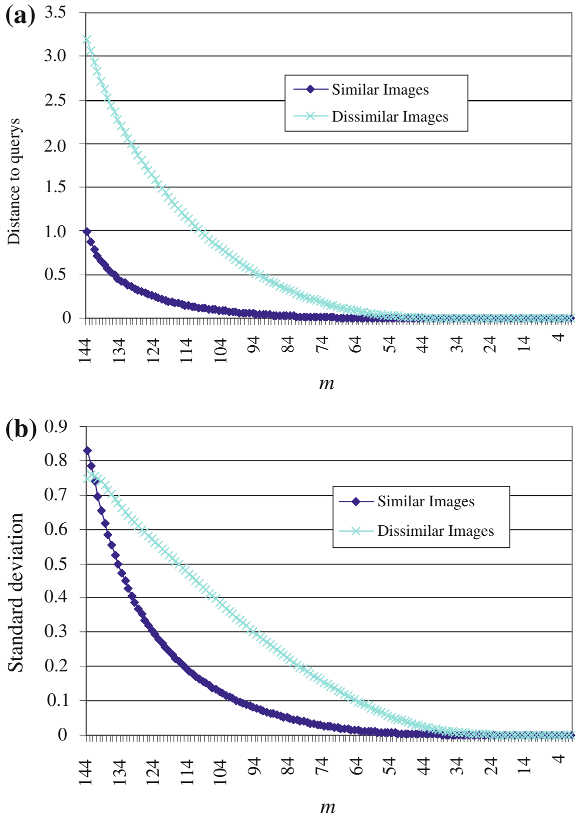


Fig. 4.5 The effect of DPF. **a** Average of distances, **b** Standard deviation of distances

overlap occurs between the distance distributions of similar and dissimilar images to the query images. (When $m = 144$, DPF degenerates to the Euclidean function.) In other words, many similar images and dissimilar images may reside about the same distance from their query image, which causes degraded search performance. When we decrease m to 124, Fig. 4.6b shows that both distributions shift toward the left. The distribution of similar images becomes more compact, and this leads to a better separation from dissimilar images. Further decreasing the m value moves both distributions leftward (as shown in Figs. 4.6c, d). When little room is left for the distance distribution of similar images to move leftward, the overlap can eventually increase. Our observations from these figures confirm that we need to find the optimal m value to achieve best separation for similar and dissimilar images.

4.4.1.2 Testing DPF

We tested our distance functions on a dataset that was independently constructed from the 1.5M-image dataset used for conducting mining and parameter training.

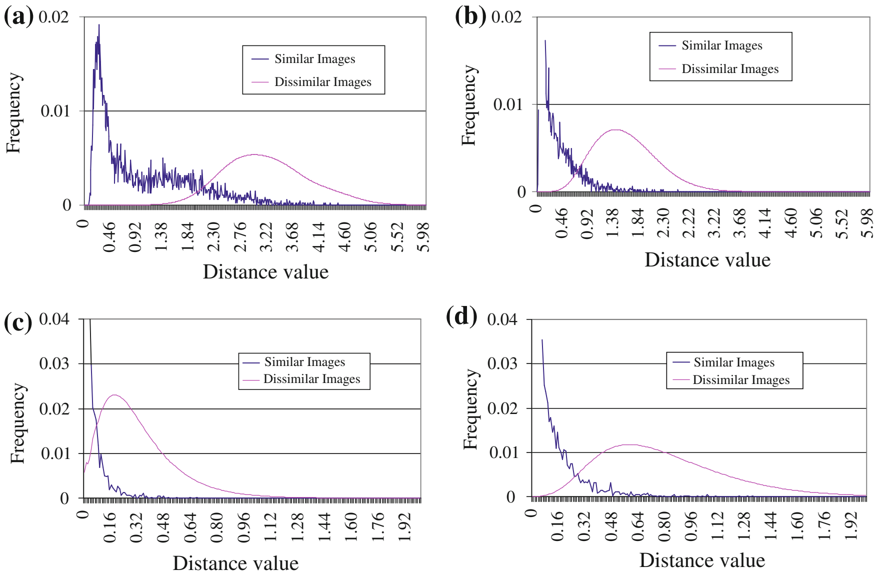


Fig. 4.6 Distance distributions versus m. **a** $m = 144$, **b** $m = 124$, **c** $m = 104$, **d** $m = 84$

The test dataset consisted of 50K randomly collected World Wide Web images. Among these images we identified 100 images as query images. For each query image, we generated 24 similar images using the transformation methods described in Sect. 4.2. We also visually identified three perceptually similar images for each query image. (See Fig. 4.7. for examples of visually-identified similar images).

We conducted 100 queries using the 100 query images. For each query, we recorded the distance ranks of its similar images. For DPF, we fixed m value as 114 based on the training results in Sect. 4.4.1.1. Figure 4.8 depicts the experimental results. The precision-recall curves in the figure show that the search performance of DPF is significantly better than the other traditional distance functions. For instance, to achieve a recall of 80%, the retrieval precision of DPF is 84%, whereas the precision of the L_1 distance, the Euclidean distance, and the histogram Cosine distance is 70, 50, and 25%, respectively.

We were particularly interested in the retrieval performance of the visually identified similar images, which were not included into the training-image dataset. Figure 4.9. compares the retrieval performance of DPF and traditional distances for the visually identified similar images. The precision-recall curves indicate that, even though the visually identified similar images were not included in the training-image dataset, DPF could still find them effectively in the testing phase. This indicates that the trained DPF parameters can be generalized to find similar images produced by methods other than those for producing the training dataset.

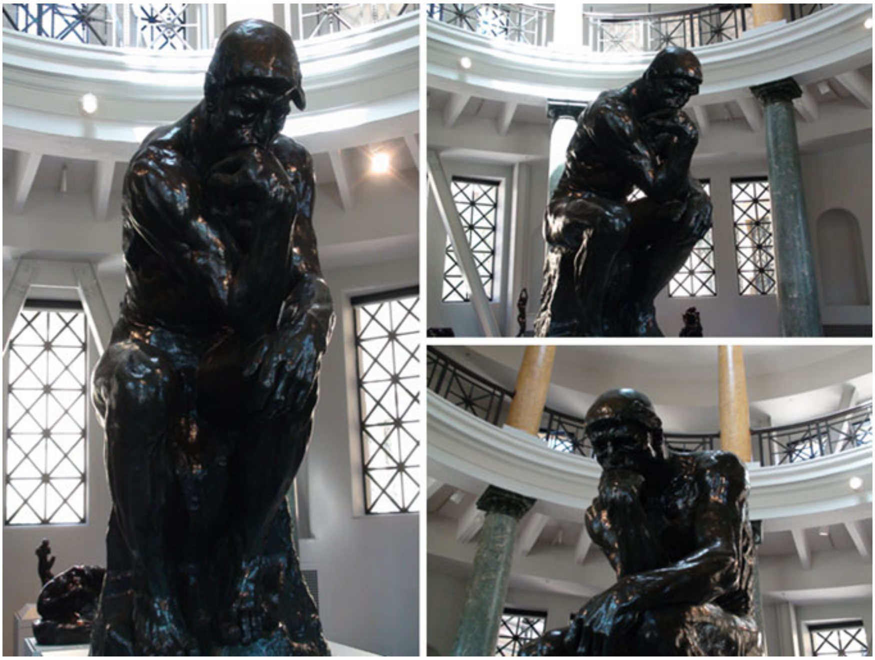
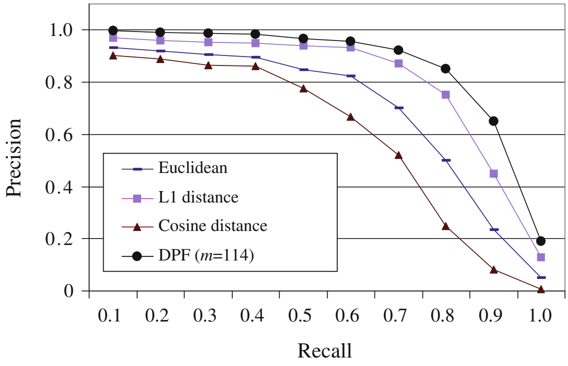


Fig. 4.7 Three perceptually similar images

Fig. 4.8 Precision/recall for similar images



4.4.2 Video Shot-Transition Detection

To further examine the generality of the DPF, we experimented DPF in another application—video shot-transition detection. Our video dataset consisted of 150 video clips which contained thousands of shots. The videos covered the following subjects:

- *Cartoon*: 30 clips, each clip lasting for 50 s (from commercial CDs).

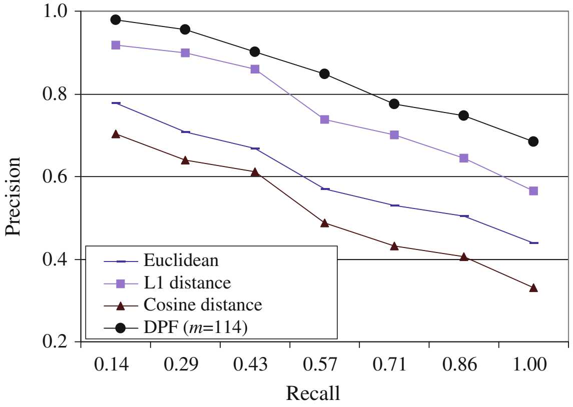


Fig. 4.9 Precision/recall for visually identified similar images

- *Comedy*: 50 clips, each lasting for up to 30 s.
- *Documentary*: 70 clips, each lasting for 2–5 min [18].

For characterizing a frame, we extracted the same set of 144 features for each frame, since these features can represent images to a reasonable extent. Our experiments had two goals. The first was to find the optimal parameter m settings for DPF (Sect. 4.4.2.1). The second was to compare the shot detection accuracy between employing DPF and employing the Minkowski metric as the inter-frame distance function (Sect. 4.4.2.2).

4.4.2.1 Parameter *m*

We fixed $r = 2$ in our empirical study. Then we took a machine learning approach to train the value of m. We sampled 40% of the video clips as the training data to discover a good m. We then used the remaining 60% of video clips as testing data to examine the effectiveness of the learned m.

In the training phase, we labeled the accurate positions of shot boundaries. We then experimented with different values of m on three video datasets (cartoon, comedy, and documentary). Figure 4.10. shows that for all three video types, the false detection rates are reduced to a minimum as m is reduced from 144 to between 115 and 120. (Recall that when $m = 144$, DPF degenerates into the Minkowski distance function.) It is evident that the Minkowski distance function is not the best choice for our purpose.

4.4.2.2 DPF Versus Minkowski

We next compared two inter-frame distance functions, DPF and Euclidean, on the testing data. For DPF, we set $m = 117$ based on the training results in Sect. 4.4.2.1.

Fig. 4.10 Optimal *m*

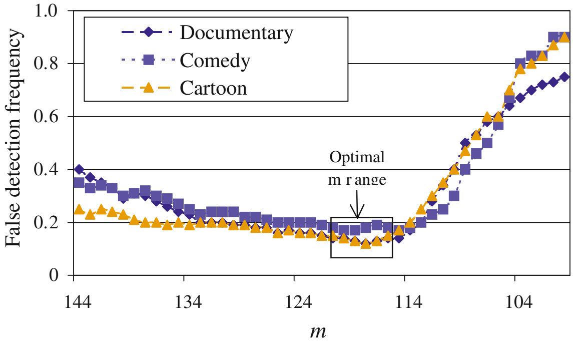


Table 4.2 Precision and recall

Distance functions	Video type # of Shot Boundaries	Comedy 425	Cartoon 167	Documentary 793
Euclidean	# of false	93	39	192
	# of miss	97	37	183
	Precision (%)	78.1%	76.6%	75.8%
	Recall (%)	77.2%	77.8%	76.9%
DPF	# of false	61	26	140
	# of miss	67	25	129
	Precision (%)	85.6%	84.4%	82.3%
	Recall (%)	84.2%	85.0%	83.7%

Fig. 4.11 Overall precision and recall comparison

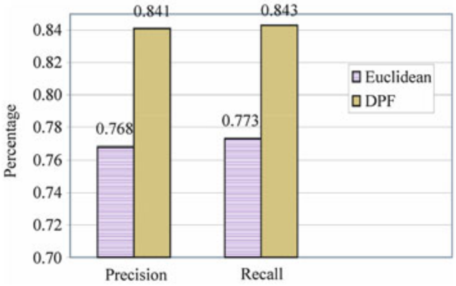


Table 4.2 shows that DPF improves the detection accuracy over the Euclidean distance function on both precision and recall for all video categories. The average improvement as shown in Fig. 4.11 is about 7% in both recall and precision. In other words, for every 100 shot transitions to be detected, DPF makes seven fewer detection errors, a marked improvement.

Figure 4.12 illustrates why DPF can better detect shot boundaries than Euclidean distance, from the signal/noise ratio perspective. The *x*-axis of the figure depicts the frame number; the *y*-axis depicts the inter-frame distance between the ith and the $(i+1)$th frames. We mark each real shot boundary with a circle and a false detection

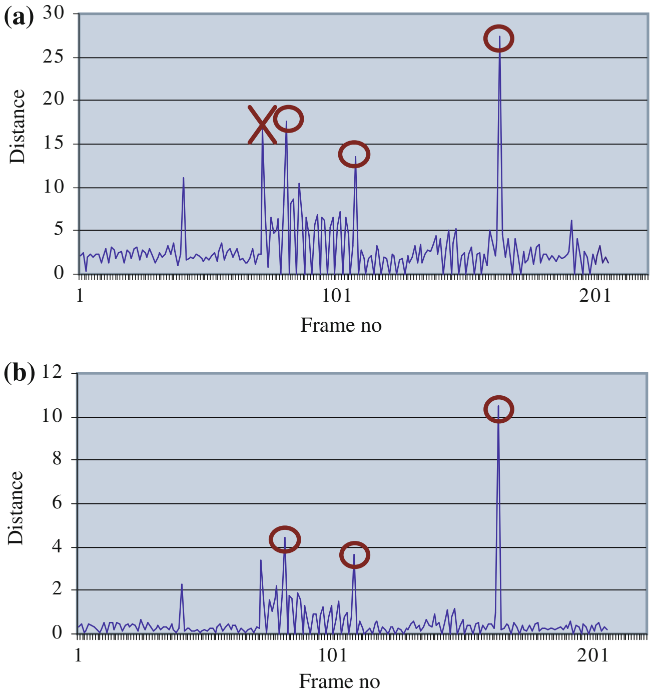


Fig. 4.12 Euclidean versus DPF. **a** Euclidean, **b** DPF

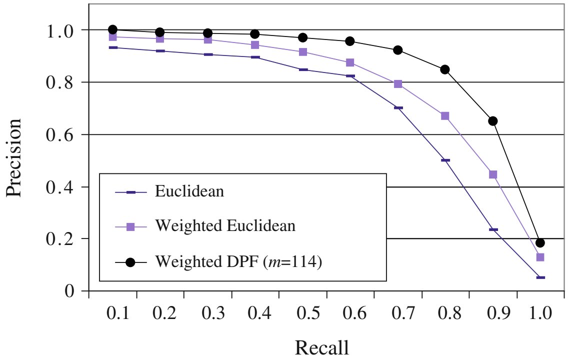
with a cross. Figure 4.12a shows that the Euclidean distance function identified four shot boundaries, in which the left-most one was a false positive. Figure 4.12b shows that DPF separates the distances between shot boundaries and non-boundaries better, and hence eliminates the one mis-detection. DPF improves the signal/noise ratio, and therefore, it is more effective in detecting shot transitions.

4.4.3 Near Duplicated Articles

A piece of news is often quoted or even included by several articles. For instance, a piece of new released by the Reuters may be included in some Blogger posts. A search engine would like to cluster all near-duplicated articles and present them together to avoid information redundancy.

We compared two distance functions on Google News in 2006. Between DPF and a hashing algorithm very similar to LSH, DPF outperforms the hash algorithm by about 10% in both precision and recall. However, since the computation complexity of hashing is linear but DPF quadratic. When the number of candidate articles is very large, DPF encounters scalability problem. To deal with this practical deployment

Fig. 4.13 Comparison of weighted functions



challenge, the work of Dyndex [19] proposes an approximate indexing method to speed up similar-instance lookup. The basic idea is to ignore the non-metric nature of DPF, or using the full Euclidean space to perform indexing. A lookup is performed in the Euclidean space. Though precision/recall may be degraded, this approximation compromises slightly degraded accuracy for speedup. For details, please consult reference GohLC02.

4.4.4 Weighted DPF Versus Weighted Euclidean

We were also interested in applying weighted DPF to improve the weighted Minkowski distance function, which has been used extensively to personalize similarity measures. For weighted Minkowski distance, a weighting vector is learned for each query. Usually, the weight of a feature is set as the inverse of the variance of its values among similar images. Here, we allowed the weighted Euclidean distance function to work under the ideal condition—that is, it knows all similar images a priori and can compute the ideal weighting vector for each query. Figure 4.13 shows that the weighted Euclidean function outperforms its unweighted counterpart. This result confirms that the weighted version [20, 21] is indeed a better choice than the unweighted version (provided that the appropriate weighting can be learned). However, there is still much room for improvement. When we applied weighted DPF using the same weighting vector, its retrieval performance was better than that of the weighted Euclidean distance function. For instance, at 80% recall rate, the retrieval precision of the weighted Euclidean distance is about 68%, whereas the weighted DPF could achieve a precision of above 85%. Again, our empirical study shows that the generalized form of DPF, weighted DPF, can be used to markedly enhance the weighted Minkowski distance for measuring image similarity.

4.4.5 Observations

We summarize the results of our experiments as follows:

1. DPF is more effective than some most representative distance functions used in the CBIR community (e.g., Minkowski-like and histogram Cosine distance functions) for measuring image similarity and for detecting shot transitions.
2. The weighted version of DPF outperforms the weighted version of the Euclidean distance function.
3. We believe that DPF can be generalized to find similar images of some other ways, and that DPF can be effective when a different set of low-level features are employed. Our belief is partially supported by our empirical results, and partially justified by similar theories in cognitive science, which we discuss next.

4.5 Related Reading

Similarity is one of the most central theoretical constructs in psychology [6, 7]. Its also plays a central role in information categorization and retrieval. Here we summarize related work in similarity distance functions. Using our experimental results, together with theories and examples in cognitive psychology, we explain why DPF works effectively as we discuss the progress of the following three similarity paradigms in cognitive psychology.

1. *Similarity is a measure of all respects*. As we discussed in Sect. 4.3, a Minkowski-like metric accounts for all respects (i.e., all features) when it is employed to measure similarity between two objects. Our mining result shown in Fig. 4.2. is just one of a large number of counter-examples demonstrating that the assumption of the Minkowski-like metric is questionable. The psychology studies of [6, 7] present examples showing that the Minkowski model appears to violate human similarity judgements.
2. *Similarity is a measure of a fixed set of respects*. Substantial work on similarity has been carried out by cognitive psychologists. The most influential work is perhaps that of Tversky [7], who suggests that similarity is determined by matching features of compared objects, and integrating these features by the formula

$$S(A, B) = \theta f(A \cap B) - \alpha f(A - B) - \beta f(B - A). \tag{4.5}$$

   The similarity of A to B, $S(A, B)$, is expressed as a linear combination of the common and distinct features. The term $(A \cap B)$ represents the common features of A and B. $(A - B)$ represents the features that A has but B does not; $(B - A)$ represents the features that B has but A does not. The terms $\theta, \alpha,$ and β reflect the weights given to the common and distinctive components, and function f is often assumed to be additive [6] The weighted Minkowski function [22] and the quadratic-form distances [23, 24] are the two representative distance functions that match the spirit of (4.5). The weights of the distance functions can be learned via techniques such as relevance feedback [20, 22], principal component analysis, and discriminative analysis [25]. Given some similar and some dissimilar objects, the weights can be adjusted so that similar objects can be better distinguished from other objects.

3. *Similarity is a process that provides respects for measuring similarity* Murphy and Medin [26] provide early insights into how similarity works in human perception: "The explanatory work is on the level of determining which attributes will be selected, with similarity being at least as much a consequence as a cause of a concept coherence." Goldstone [27] explains that similarity is the process that determines the respects for measuring similarity. In other words, a distance function for measuring a pair of objects is formulated only after the objects are compared, not before the comparison is made. The respects for the comparison are activated in this formulation process. The activated respects are more likely to be those that can support coherence between the compared objects.

With those paradigms in mind, let us re-examine how DPF works. DPF activates different features for different object pairs. The activated features are those with minimum differences—those which provide coherence between the objects. If coherence can be maintained (because sufficient a number of features are similar), then the objects paired are perceived as similar. Cognitive psychology seems able to explain much of the effectiveness of DPF.

4.6 Concluding Remarks

We have presented DPF, its formulation via data mining and its explanation in cognitive theories. There are several avenues to improve DPF. First, the activation of respects is believed to be context-sensitive [28–30]. Also, certain respects may be more salient than others, and hence additional weighting factors should be considered. In Chap. 5 we discuss how weights can be learned from user feedback via some supervised approach. As we discussed in the chapter, the parameters of DPF can be learned using a threshold method, and the quadratic nature of DPF can be alleviated through an approximate indexing scheme. For details, please consult [17, 19].

References

1. B. Li, E.Y. Chang, Y.L. Wu, Discovery of a perceptual distance function for measuring image similarity. Multimedia Syst. **8**(6), 512–522 (2003)
2. M.W. Richardson, Multidimensional psychophysics. Psychol. Bull. **35**, 659–660 (1938)
3. Y. Rubner, C. Tomasi, L. Guibas, in *Adaptive color-image embedding for database navigation*, in *Proceedings of the Asian Conference on Computer Vision*, pp. 104–111, Jan 1998

4. I. Witten, A. Moffat, T. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*. (Van Nostrand Reinhold, New York, 1994)
5. J. Li , J.Z. Wang , G. Wiederhold, Irm: Integrated region matching for image retrieval, in *Proceedings of ACM Multimedia*, pp. 147–156, Oct 2000
6. D.L. Medin, R.L. Goldstone, D. Gentner, Respects for similarity. Psychol. Rev. **100**(2), 254–278 (1993)
7. A. Tversky, Feature of similarity. Psychol. Rev. **84**, 327–352 (1977)
8. B. Wandell, *Foundations of Vision*, (Sinauer, MA, 1995)
9. K.A. Hua, K. Vu, J.H. Oh, Sammatch: a flexible and efficient sampling-based image retrieval technique for image databases, in *Proceedings of ACM Multimedia*, pp. 225–234, 1999
10. J.G. Leu, Computing a shape's moments from its boundary. Pattern Recognit **24**(10), 949–957 (1991)
11. W.Y. Ma, H. Zhang, Benchmarking of image features for content-based retrieval. in *Proceedings of Asilomar Conference on Signal, Systems and Computers*, 1998
12. B. Manjunath, P. Wu, S. Newsam, H. Shin, A texture descriptor for browsing and similarity retrieval. Signal Process. Image Commun. 2001
13. J. R. Smith, S.-F. Chang, VisualSEEk: A Fully Automated Content-Based Image Query System. ACM Multimedia, 1996, pp. 87–98
14. H. Tamura, S. Mori, T. Yamawaki, Texture features corresponding to visual perception. IEEE Trans. Syst. Man Cybern. (SMC), 1978
15. E. Goldstein, S. Fink, Selective attention in vision. J. Exp. Psychol. **7**, 954–967 (1981)
16. J.R. Smith, Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression. Ph.D. Thesis, Columbia University, 1997
17. A. Qamra, Y. Meng, E.Y. Chang, Enhanced perceptual distance functions and indexing for image replica recognition. IEEE Trans. Pattern Anal. Mach. Intell. **27**(3), 379–391 (2005)
18. http://www-nlpir.nist.gov/projects/t01v/t01v.html
19. K. Goh, B. Li, E.Y. Chang, Dyndex: a dynamic and non-metric space indexer. In *Proceedings of ACM International Conference on Multimedia*, pp. 466–475, 2002
20. K. Porkaew, S. Mehrota, M. Ortega, Query reformulation for content based multimedia retrieval in mars, in *Proceedings of ICMCS*, pp. 747–751, 1999
21. M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, T.S. Huang, Supporting similarity queries in mars, in *Proceedings of ACM International Conference on Multimedia*, pp. 403–413, 1997
22. J. Rocchio, in *Relevance Feedback in Information Retrieval*, ed. by G. Salton, The SMART Retrival System: Experiments in Automatic Document Processing. (Prentice-Hall, New Jersey, 1971)
23. M. Flickner, H. Sawhney, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, Query by image and video content: the QBIC system. IEEE Comput. **28**(9), 23–32 (1995)
24. Y. Ishikawa, R. Subramanya, C. Faloutsos, Mindreader: querying databases through multiple examples, in *Proceedings of VLDB*, pp. 218–227, 1998
25. X.S. Zhou, T.S. Huang, Comparing discriminating transformations and svm for learning during multimedia retrieval, in *Proceedings of ACM Conference on Multimedia*, pp. 137–146, 2001
26. G. Murphy, D. Medin, The role of theories in conceptual coherence. Psychol. Rev. **92**, 289–316 (1985)
27. R.L. Goldstone, Similarity, interactive activation, and mapping. J. Exp. Psychol. Learning Mem. Cogn. **20**, 3–28 (1994)
28. I. Jurisica, J.I. Glasgow, Improving performance of case-based classification using context based relevance. IJAIT **6**(4), 511–536 (1997)
29. P.G. Schyns, R.L. Goldstone, J.P. Thibaut, The development of features in object concepts. Behav. Brain Sci. **21**, 1–54 (1998)
30. S. Tong, E. Chang, Support vector machine active learning for image retrieval, in *Proceedings of ACM International Conference on Multimedia*, pp. 107–118, October 2001.