iLike: Integrating Visual and Textual Features for Vertical Search

Yuxin Chen†, Nenghai Yu‡, Bo Luo†, Xue-wen Chen†
† Department of EECS, University of Kansas, Lawrence, KS, 66045, USA
‡ Department of EEIS, University of Science and Technology of China, Hefei, China
yxchen@ku.edu, ynh@ustc.edu.cn, bluo@ku.edu, xwchen@ku.edu

ABSTRACT

Content-based image search on the Internet is a challenging problem, mostly due to the semantic gap between low-level visual features and high-level content, as well as the excessive computation brought by huge amount of images and high dimensional features. In this paper, we present iLike, a new approach to truly combine textual features from web pages, and visual features from image content for better image search in a vertical search engine. We tackle the first problem by trying to capture the meaning of each text term in the visual feature space, and re-weight visual features according to their significance to the query content. Our experimental results in product search for apparels and accessories demonstrate the effectiveness of iLike and its capability of bridging semantic gaps between visual features and abstract concepts.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

General Terms

Algorithms, Design

1. INTRODUCTION

With the Internet explosion, tremendous amounts of multimedia information, such as images, videos, and flashes, become available on the Web. Unlike the great success of text-based web search, the research community is still struggling with content-based indexing and searching of multimedia information over the Internet. Image search engines still rely on text-based methods, i.e. retrieve and rank images based on surrounding text or human-submitted annotations. On the other hand, most existing content-based image retrieval (CBIR) prototypes still use offline image databases that are not comparable with the scale of the Web. Besides various research efforts that aim to directly employ visual content based retrieval for Web images, some recent approaches have proposed alternative routes, e.g. [24, 15, 35, 6]. In this paper, we take a different approach, which focuses on truly integrating textual and visual features for vertical search engines.

A vertical search engine, a.k.a. niche search engine, is a domain-specific search engine that works on a smaller subgraph of the Web. Examples of vertical search include scientific publications search (e.g. Google Scholar, CiteSeer), product search (e.g. Google Product, Yahoo! Shopping), Blog search, source code search, local search, etc. Vertical search engines have shown better performance than general Web search engines (e.g. precision, ranking), because they are more focused and optimized with domain knowledge [19].

In the scenario of vertical search, we have a better chance to truly integrate visual features from images and textual features from text contents. First, text contexts are better organized, hence focused crawlers/parsers are able to generate data patterns and structured data, instead of free text. Second, we are able to associate text content with images with high confidence. In general Internet image retrieval, one problem is that texts surrounding images may not necessarily describe the image content. However, in some vertical search engines, the focused crawlers are able to connect text contents with corresponding image(s), e.g. product images and product descriptions, paintings and introductions, etc. Third, with the knowledge of the focused domain, we are able to select image features and similarity measures that are more effective for the domain. Finally, computation issues become less critical for a smaller data set.

In this paper, our goal is to **explore the possibilities of integrating visual and textual features to improve search performance in the scenario of vertical search**. Particularly, we focus on the domain of product search for apparels and accessories. In this domain, we propose to utilize both textual features (from product description) and visual features (from product images) for items, and try to understand and mimic human perception of "similarity". More specifically, we try to understand the inherent connections between text (keywords) and visual features, to build a bridge over the semantic gap. We start from predicting user intention, which is implicitly carried with search terms. For instance, the query "blue shirt" indicates that the user concerns more on color than any other aspect. However, such clearly expressible intention is not available for all the terms, and such human perception is yet to be mapped to low-level feature spaces. We further assess the perception/intention behind each keyword in the visual feature space, and use

MM'10, October 25–29, 2010, Firenze, Italy.

them for searching as well as re-weighting visual features according to their significance to the query. These novel ideas allow us to integrate textual and visual features in accordance with user perception, and develop a similarity measure and ranking method that better fits user intention.

Our major contributions are three-fold: (1) we demonstrate that truly integrating textual and visual features could significantly improve ranking in vertical search, especially in the domains where visual contents are equally significant to text contents. It also improves overall recall by yielding items that would otherwise be missed by searching with either type of the features. (2) We are able to infer users' (visual) intention behind search terms, and apply such intention to improve relevance assessment and ranking through textual-feature-guided visual feature selection and weighting. (3) Our approach also assesses representations of keywords in the visual feature space, and computes the semantic relationships of the terms. In this way, we are able to automatically generate a thesaurus based on the "visual semantics" of words.

The rest of the paper is organized as follows: in Section 2, we review the related literature. In Section 3, we give an overview of our approach: iLike. We present the detailed algorithm of integrating textual and visual features in Section 4. We then present our visual thesaurus in Section 5. Next, we demonstrate our experimental results and further discuss the strength and weakness of different approaches in Section 6. Finally, we conclude the paper and discuss future works.

2. RELATED-WORK AND BACKGROUND

2.1 Content-based image retrieval

In early information retrieval systems, images were manually annotated with meta-data, and retrieved using text-based methods (e.g. library of congress catalog records for prints and photographs). However, it is too expensive, if not impossible, to create meta-data for a huge image databases. Meanwhile, not all image contents could be accurately and indisputably described by keywords. Discrepancies may also exist between query keywords and tag keywords (e.g. query "car" would not yield images annotated with "automobile"). Therefore, content based image retrieval (CBIR) was proposed to tackle such problems. It retrieves images with visual features such as color, texture, and shape. Comprehensive surveys on CBIR could be found at [30, 21, 8].

The primary goal, as well as the major challenge of content-based image retrieval research, is to bridge the semantic gap, which is the gap between high level image content and low level visual features. On the other hand, we are also expecting a major breakthrough to the computational issue, especially high dimensional indexing. Such challenges have prevented CBIR from being widely adopted in web search.

2.2 Image search on the web

At present, commercially available general-purpose search engines on the web still mostly depend on text methods for image search. Such text-based image search engines include Google Image Search (http://images.google.com/), Yahoo image search (http://images.search.yahoo.com/), etc. They take keyword queries, and match them against metadata (e.g. file name, URL, link text, etc) and surrounding text extracted from the webpages which contain the images. On the other hand, user-generated labels are employed to improve search quality, e.g. [23, 34], Google Image Labeler (http://images.google.com/imagelabeler/), Flickr image tags (http://www.flickr.com/photos/tags/), etc. Some approaches use more aggressive text methods on surrounding texts to better associate semantics to images, e.g. [1, 29]. On the other hand, link analysis has also been employed to improve search performance [20, 2].

Meanwhile, there has been prototypes of content-based image search for the web, e.g. [17, 12, 27, 28, 3]. Pure visual content based image search engines suffer from two major disadvantages inherited from CBIR: semantic gap and computation. To tackle such problems, alternative approaches have been proposed. In the web search scenario, both images and text contents are available, which provide opportunities to bridge the semantic gap and better indexing by integrating features from both sides. A two stage hybrid approach has been introduced in [24]. They first use text-based search to generate an intermediate answer set with high recall and low precision, and then apply CBIR methods to cluster or re-rank the results. Although their approach suffers from over simplified image features and clustering methods, the idea of applying CBIR after text search appears to be viable. More complicate re-ranking algorithms have been proposed for better search performance and user experience [15, 35]. Most recently, Bing image search (http://www.bing.com/images/) have started to employ CBIR methods to re-rank search results, when users click on "show similar images" [6, 5]. Meanwhile, other types of text-image interaction have been proposed, e.g.: [14, 22, 32, 40] use visual information to help annotating images. Our approach is significantly different from existing approaches in the way that we integrate textual and visual features.

Content-based image retrieval over the general web is a hard problem. On the other hand, some researchers have proposed to apply CBIR in vertical search. Vertical search engines work on specific (sub)domains of the Internet. They use focused crawlers to crawl constrained subsets of the general web, and evaluate user queries against such domain-specific collections of documents. Besides the benefits of working on much smaller datasets, they are also able to incorporate domain knowledge to help with relevance assessment and results ranking. Examples of vertical image search includes: photo album search [38], product search (http://www.like.com/, http://www.riya.com/), airplane image search (http://www.airliners.net/), etc. On the other hand, there are also off-line image retrieval systems that work on domain-specific collections of images, such as personal album search [39, 7], leaf images search [36, 10], fine arts image search [37], etc. These approaches utilize domain knowledge in image pre-processing, feature selection and similarity assessment. For instance, leaf image retrieval puts emphasis on shape and texture features, while personal album search usually employs face recognition methods.

3. SYSTEM OVERVIEW

3.1 System architecture

Since our goal is to integrate textual and visual features in vertical search, it is of our interest to select a domain where text content is directly associated with image content. Online shopping, especially clothing shopping, is a good exam-

ple of such domains. In shopping websites, text descriptions are always available with item images, and are usually faithful descriptions of the image contents. Moreover, we believe that both text descriptions and product images are equally important since: (1) from users' perspectives, they can only issue keyword queries for product search; on the other hand, while browsing the results, users focus more on visual presentations than the text specifications. (2) Due to different personal tastes, the descriptions of fashionable items are very subjective, hence traditional text-based search on such descriptions may not yield satisfactory results. Especially, the recall can be very low when there is a discrepancy between user's and narrator's tastes or vocabularies. (3) In many cases, two items may have similar style in human perception, but we see huge difference in the visual features. Hence, pure content-based image search will not yield high recall either. Note that our arguments are based on fashion shopping, but they are also true in many other shopping categories. Therefore, our system could be migrated to other categories with minimum modification.

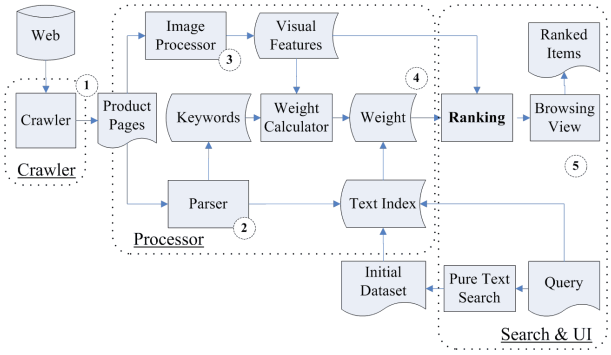

Figure 1: system architecture of iLike.

The system of iLike is comprised of three major components: the Crawler, the (Pre-)Processor, and the Search and UI component. As shown in Figure 1: (1) the Crawler fetches web pages from retailer websites, where structured text descriptions and item images are both available. (2) The text parser preprocesses pages using a customized parser, and fits item information (e.g. title, description) into a predefined universal schema. Using classic text retrieval methods, text processor generates term dictionary and text index. (3) Simultaneously, the image processor segments product images and calculates low level visual features. (4) Next, we integrate textual and visual features by calculating a "centroid" and a weight vector in the visual feature space for each text term. Such vectors are further utilized in item ranking. (5) Finally, the User Interface provides query interface, as well as browsing views of search results.

In iLike , a user starts with a traditional text query (since query-by-example is not really practical in this scenario), and the system returns a ranked list of relevant items (namely the *initial result set*) using classic text retrieval methods (TF/IDF) [26]. For each result in the initial result set, we construct a new query by integrating textual and visual features from item images. Each expanded query is evaluated to find more "similar" items. More importantly, a weight vector which represents the "visual perception" behind the text query is enforced during evaluation of the expanded queries. For instance, with a query "silky blouse", the weight factor will increase the significance of some texture features, and fade out irrelevant features, hence correctly interpret the visual meaning behind search term "silky". The philosophy of our approach is to infer *user intentions* from a text query and enhance the corresponding visual features that are implicitly favored in the intentions (e.g. query term "yellow" implicitly favors color features).

3.2 Crawling and feature extraction

Data Acquisition. In the prototype, we have initially crawled a total of 20788 product items from six online retailers: Banana Republic, Old Navy, Gap, Athleta, Piperlime and Macy's. They all provide mid-sized hi-quality images and well structured textual description. We use focused crawlers to harvest both text and images. Please note that the system is easily expandable by implementing more customized crawlers and parsers.

For each product, we record the name, category, class, online product ID, local path of the main image, original URL, detailed textual description, color tags, and size information, if available. We use an unique id for each product item, to identify both the database record and the image file. Text information is stored in a MySQL database (Version 5.0)[1], and all the customized software (e.g. focused crawlers) are written in C# programming language.

Visual Features. In order to make a sufficient coverage of an image's semantic meaning, we attempt to diversify the part of feature selection. In our experiment, a set of 263 commonly used texture, shape, intensity and color features are extracted to represent the low-level visual features of images.

We use gray level co-occurrence matrix (GLCM)[13] to capture the basic texture information: contrast, correlation, energy, and homogeneity of the grayscale images are calculated, each of which generating a 4-scale feature vector. Image coarseness and direction are obtained by calculating 3 dimensions of Tamura texture features [31]. To extract the shape information, we represent the contour of an image in terms of 7 geometric invariant moments[11], which are invariant under rotation, scale, translation and reflection of images. To capture texture patterns in frequency domain, we apply Gabor wavelet filters in 8 directions and 5 scales, acquiring a vector of 40 texture features. Besides, fourier descriptors[33] are also employed, contributing 9-dimensional feature vector to our feature set. As part of shape features, the edge orientation is represented by phase congruency features(PC) [18] and moments of characteristic function: A three-level Daubechies wavelet decomposition of the test image is first carried out. At each level, the first four moments of phases, which are generated by Sobel edge detector, are obtained, together with the first three moments of the characteristic function, yielding a 28-dimensional feature vector. We demonstrate the image intensity using 48 statistics of 4 by 4 block histograms, with 16 dimensions in each of the R, G, B components. The color features are generated by color quantization approach. We map the original image into the HSV color space, and implement color quantization using 72 colors(8 levels for H channel, 3 levels for S channel and 3 levels for V channel).

The chosen features have been proved to work well for image classification in literature [25]. On the other hand,

[1]http://www.mysql.com/

we do not want our search performance to be overwhelmed by very complicate and computationally intensive visual features. Meanwhile, a comparative study [9] has shown that the effectiveness of visual features is dependent on the particular task. However, such a specifically optimized system cannot be easily migrated to other domains, due to the labor-intensive manual feature selection process. Instead, in iLike features are automatically weighted based on their significance to the user intent (implicitly carried by the query). Less important features are faded out, while more important features are enhanced. Therefore, unlike other CBIR approaches, the "quality" of low-level visual features is not the key factor in our system. As a side effect, our method is robust: the ranking quality is less sensitive to the selection of low-level image features.

Segmentation. Our database contains images of products in all shapes and sizes. Various retailers have different specifications of their product demo, some of which have introduced non-ignorable errors to feature extraction. For instance, the presence of a lingerie model could significantly influence the feature distribution. To simplify and clean the representation of product images and minimize the error of features, we perform an "YCbCr Skin-color Model"[16]-based image segmentation to remove the skin area and highlight product items.

Normalization. Our system uses diverse types of image features. However, features from different categories are not comparable with each other, since they take values from different domains. Without any normalization, search results will be dominated by those features taking larger values. To reduce the interferences brought by different feature types and scopes, we map the range of each feature $\vec{x}$ to $(0, 1)$:

$$y_i = \frac{x_i - min(\vec{x})}{max(\vec{x}) - min(\vec{x})} \quad (1)$$

in which i indicates the i-th item.

After normalizing, all the features are mapped into $\vec{y}$ with the same scale, and become comparable.

4. THE METHOD

If we consider the vast source of web data being distributed in a metadata space, to a certain extent, the role of semantic subspaces and visual subspaces are complementary. It is well known that textual information can better represent the semantic meaning while visual knowledge plays a dominant role at the physical level. In this subsection, we will discuss a native approach to bridge the "semantic gap", which allows easy transformation from one subspace to another.

4.1 Representing keywords

To some extent, textual description is a projection of human perception. Unlike visual features, where there's always a semantic gap, text directly represents the narrater's perception. However, there are difficulties using only text features for our goal: (1) perception is a subjective matter; (2) the same impression could be described through different words; and (3) calculating text similarity (or distance) is difficult - distance measurements (such as cosine distance in TF/IDF space) do NOT perfectly represent the distances in human perception. For instance, from a customer's perspective, 'relaxed-cut' is similar to 'regular-cut' and quite different from 'slim-cut'. However, they are equally different in terms of textual representation (e.g. in vector space model).

Figure 2: some items that has the keyword "floral" in their descriptions.

To make up for the deficiency of pure text search, we try to map keywords into visual subspace. Since the text description represents the narrator's perception of the visual features, we assume that: *items share the same keyword(s) may also share some consistency in* ***selected*** *visual features.* Moreover, *if the consistency is observed over a significant number of items described by the same keyword, such a set of features and their values may represent the human "visual" perception of the keyword.*

For instance, let's look at the items with the keyword "floral" (some examples are shown in Figure 2). Although they come from different categories and different venders, they all share very unique texture features. On the other hand, they all differ a lot in other features, such as color and shape. It indicates that the term "floral" is particularly used to describe certain texture features. When a user searches with this term, her intension is to find such texture features, not about color or shape. In this way, many terms could be connected with such a "visual meaning". Now let us discover such "visual meanings" automatically.

Base representation. Suppose there are N items sharing the same keyword, and each item is represented by a M-dimensional visual feature vector: $\vec{X}_k = (x_{k_1}, x_{k_2}, \ldots, x_{k_M})^T$, where $k \in [1, N]$. The mean vector of the N feature vectors could be utilized as a *base representation* of the keyword in the visual feature space:

$$\vec{\mu} = (\frac{1}{N}\sum_{k=1}^{N} x_{k_1}, \frac{1}{N}\sum_{k=1}^{N} x_{k_2}, \ldots, \frac{1}{N}\sum_{k=1}^{N} x_{k_M})^T$$

In the above equation, if N is large enough, $\vec{\mu}$ will preserve the common dimensions of the feature matrix and smooth over the various sections. In such a manner, the mean vector is rendered as a good representation of the keyword. However, those N feature vectors only share consistency over *selected* features, hence, not all dimensions of the mean vector makes sense. As shown in the "floral" example, those items are only similar in some texture features, while they differ a lot in color and shape features. Such consistancy/inconsistancy on the feature is a better indicator of

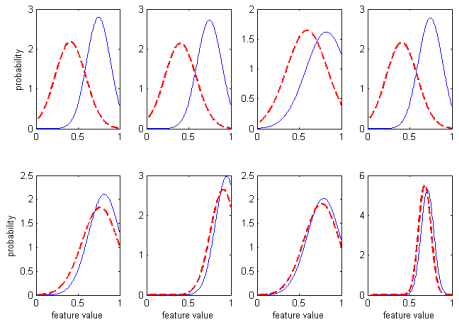

Figure 3: Examples of feature distributions.

the significance of the feature towards human perception of the keyword. Therefore, a more important task is to quantify such consistency or inconsistency.

4.2 Weighting visual features

As shown in the "floral" example, features coherent with the human perception of the keyword tends to have consistent values; while other features are more likely to be diverse. To put it another way, suppose that we have two groups of samples: (a) *positive*: N_1 items that have the keyword in their descriptions, and (b) *negative:* N_2 items that do not contain the keyword. In this way, if the meaning of a keyword is coherent with a visual feature, its N_1 values in the positive group should demonstrate a different distribution than the N_2 values in the negative group. Moreover, the feature values in the positive group tends to demonstrate a small variance, while values in the negative group are diversified.

Figure 3 demonstrates the value distribution of eight different features for the keyword "floral". In the figure, blue line represents distribution of the positive samples, while red represents negative samples. Note that both sample sets are fitted to normal distributions for better presentation in the figure. However, when we quantitatively compare both distributions, we do not make such assumption. For the first four texture features, distributions of the positive samples are significantly different from negative samples (e.g. items described by the keyword is statistically different from other items in these features). On the contrary, the two distributions are indistinguishable for the other four features (selected from color and shape).

Please note that we still have overlaps between the distributions of positive and negative samples. This indicates that there are items visually similar to the positive items on those "good" features, but they do not have the particular keyword (e.g. "floral") in their descriptions. In the experimental results in Section 6, we will show that iLike is able to yield back such items without getting false hits (e.g. items with similar colors to the positive samples, but not the "floral" texture).

The difference between two distributions could be quantitatively captured by running Kolmogorov-Smirnov test (K-S test) [4] across each dimension of feature vectors. The two sample K-S test is commonly used for comparing two data sets because it is nonparametric and does not make any assumption on the distribution. The null hypothesis for this test is that the two samples are drawn from the same distribution. For n i.i.d samples $X_1, X_2, ...X_n$ with unknown distribution, an empirical distribution function can be defined as follows:

$$S_n(x) = \begin{cases} 0, & \text{if } x < X_{(1)}, \\ \frac{k}{n}, & \text{if } X_{(k)} \le x < X_{(k+1)}, \text{ for } k = 1, 2, \cdots, n-1 \\ 1, & \text{if } x \ge X_{(n)}, \end{cases}$$

where $X_{(1)}, X_{(2)}, \cdots X_{(n)}$ are ascending values. The K-S statistic for a given function $S(x)$ is

$$D_n = \max_x |S_n(x) - S(x)|$$

The cumulative distribution function of Kolmogorov distribution is

$$K(x) = 1 - 2\sum_{i=1}^{\infty}(-1)^{i-1}e^{-2i^2x^2} = \frac{2\pi}{x}\sum_{i=1}^{\infty}e^{-(2i-1)^2\pi^2/(8x^2)}.$$

It can be proved that $\sqrt{n}D_n = \sqrt{n}\max_x |S_n(x) - S(x)|$ will converge to the Kolmogorov distribution [4]. Therefore if $\sqrt{n}D_n > K_\alpha = Pr(K \le K_\alpha) = 1 - \alpha$, the null hypothesis for the K-S test will be rejected at confidence level α.

Similarly, to determine whether the distributions of two data sets differ significantly, the K-S statistic is

$$D_{n,m} = \max_x |S_n(x) - S_m(x)|$$

and the null hypothesis will be rejected at level α if

$$\sqrt{\frac{nm}{n+m}}D_{n,m} > K_\alpha \tag{2}$$

The P-value from the K-S test is used to measure the confidence of the comparison results against the null hypothesis. Back to our scenario, for each keyword, a P-value is calculated at each dimension of the feature vector. Features with lower P-values demonstrate statistically significant difference between positive and negative groups. For instance, the P-values for the features shown in Figure 3 row 1 are: 1.532×10^{-10}, 1.524×10^{-10}, 1.899×10^{-8}, 1.761×10^{-10}; and for Figure 3 row 2 are: 2.518×10^{-1}, 3.770×10^{-3}, 4.350×10^{-1}, 5.839×10^{-2}. As we can see, items described by the keywords have significantly different values in those features, compared with items that are not described by the keyword. Therefore, such features are more likely to be coherent with visual meaning of the keyword, and hence more important to the human perception of the keyword. On the contrary, items with and without the keyword have statistically indistinguishable values on other visual features, showing that such features are irrelevant with the keyword.

In this way, we can use the inverted P-value of the K-S test as the weight of each visual feature for each keyword. Note that P-values are usually extremely small, so it is necessary to map the value to a normal scale before using it as weight. Ideally, the mapping function should satisfy the following requirements: (1) it should be a monotone decreasing function: lower P-values should give higher weight; (2) when the variable decreases under a threshold (conceptually, small enough to be determined as "statistically significant"), the function value decreases slower. Therefore, we apply

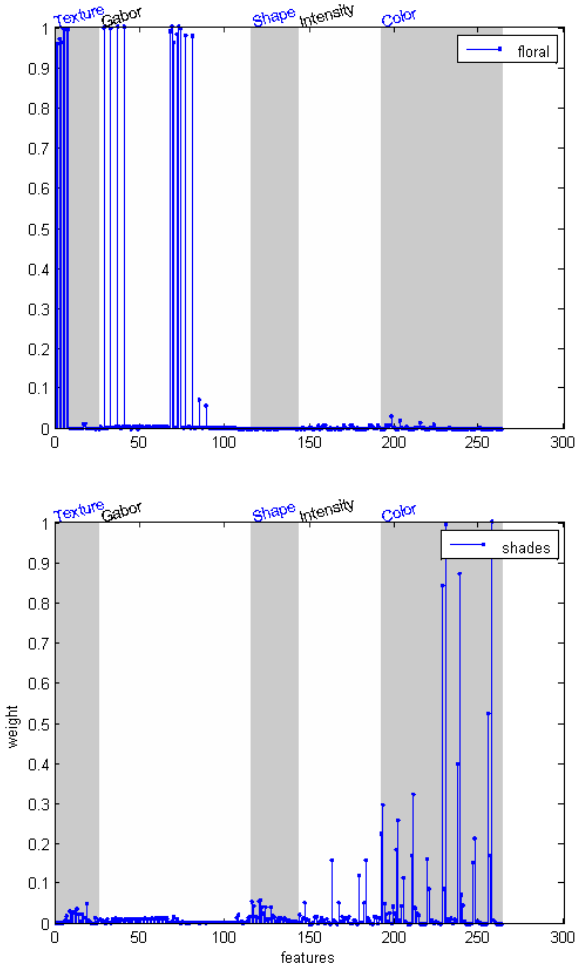

Figure 4: Weight vectors for terms "floral" and "shades".

two steps of normalization. First, we designed a mapping function:

$$f(x) = \frac{\arctan(-log(x) - C) + \arctan(C)}{\pi}$$

where $C = (max(x) - min(x))/2$. It is then followed by a linear scaling to map the data range from to (0, 1), rendering itself as the weight vector of the keyword.

By re-weighting visual features for each keyword, we amplified the features that are significant for the keyword, while faded out the others. As an example, Figure 4 shows the normalized weight vectors computed from keywords "floral" and "shades", respectively. In the figure, the X axis represents visual features (as introduced in Section 3): (1-26) are texture features: contrast, correlation, homogeneity, coarseness, direction, moment invariant etc.; (27-115) are texture features from the frequency domain: Gabor texture, Fourier descriptors, etc; (116-143) are shape features: phase congruency, edge; (144-191) are intensity features: block histogram statistics; and(192-263) are color features. In the figure, a large value (higher weight, lower P-value) are generated by statistically different positive and negative samples, indicating that the feature is more likely to have some kind of association with the human perception of the term. From the figures, we can see that some texture features show more significance in representing the keyword "floral", while the visual features of keyword "shades" is primarily captured by color features. In this way, when user queries with term "floral", we can infer that she is more interested in texture features, while local color and shape features are of less importance. Most importantly, we can further retrieve items with similar visual presentation in such features, but do not have the particular keyword in their descriptions.

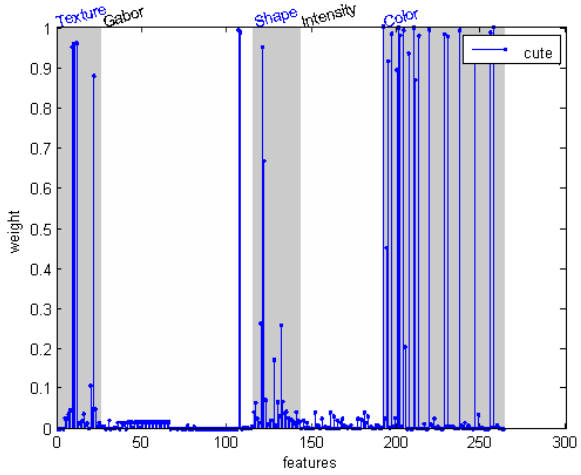

Figure 5: Weight vectors for terms "cute'.

On the other hand, it is difficult to imagine or describe the human visual perception for some keywords. Fortunately, our approach is still capable of assessing such perceptions. For instance, Figure 5 shows the weight vectors for term "cute". It is not easy for a user to summarize the characteristics of "cute" items. However, when we look at the figure, the visual meaning is obvious. "Cute" items share some distinctive distributions in the color and local textual features, while they are diversified in intensity and high frequency textual features.

As a conclusion, we have established a connection between terms and visual features: we have learned a representation of each term in the visual feature space from a large training sample set, and identified the feature components that have significance towards the visual perception of the term.

4.3 Feature quality

In CBIR, the entropy of low-level visual features is widely used for feature selection and image annotation. Effective as they are claimed, however, such algorithms share one common limitation: the semantic gap. In iLike, we reemploy this problem by utilizing the entropy of feature weights across all keywords.

In Section 4.2, we have generated a weight vector for each keyword, measuring the significance of each image feature dimension towards the keyword. Intuitively, a visual feature that is significant for a number of keywords is a "good" feature, while a visual feature that is in-significant for all keywords is a "bad" feature. Practically, we do not find any feature that is significant for (almost) all keywords. If such a feature existed, it would not be a good feature since it would not represent any distinctive visual meaning.

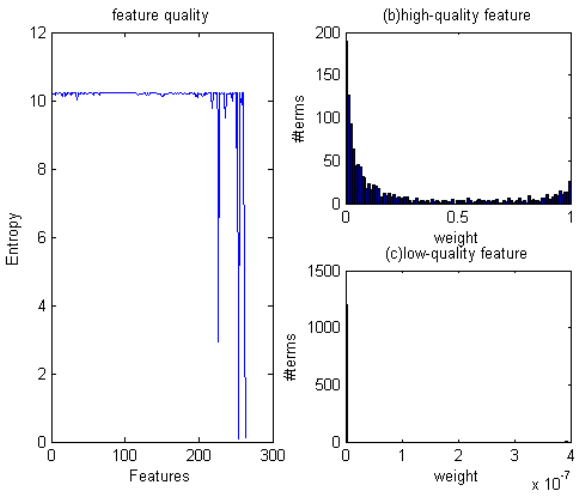

Figure 6: Feature Quality

In this way, for each feature, we collect weight values across all keywords (i.e. the ith component of all weight vectors). The entropy of each collection of weights is used as a quality assessment of the particular feature. The feature-quality curve is shown as Figure 6 (left).

Figure 6(b) and (c) demonstrates the weight histogram for two difference features. As we can see, the feature shown in Fig 6(b) has higher weights for some terms, while the feature in Fig 6(c) has low weights for all terms. That is to say, the first feature is able to distinguish the positive and negative sets for some terms, while the other feature does not work well for any term. The first feature is certainly better than the other one. Figure 6 also shows that our selected features demonstrate good quality, except for a few color features (e.g. those with much lower entropy in Figure 6 (left)). This is consistent with the CBIR literature.

4.4 Query expansion and search

As we have introduced, in iLike, we first employ classic text-based search to obtain an initial set (since users could only provide text queries). For each keyword in the user query, the system loads its corresponding weight vector, which is generated off-line. Weight vectors from query terms are combined to construct the query weight vector $\vec{\omega}_Q$, which represents user intention in the visual feature space. For each item in the initial set, we use its visual features to construct a base query $\vec{q_i}$. We also obtain an expanded weight vector $\vec{\omega}_E$ from its textual description. Therefore, given a query q, the new query corresponding to the i-th item in the initial set is:

$$\vec{q'}(Item_i, Query) = \vec{q_i}. \times (\alpha \cdot \vec{\omega}_Q + \beta \cdot \vec{\omega}_E) \qquad (3)$$

where $.\times$ indicates component-wise multiplication. Practically, β is set to a much smaller value than α, to highlight the intension from users. In the new query, features that are insignificant to the search terms carry very small values. Hence the new query could be used to search the item database on the basis of their Euclidean distances, without further enforcing the weights.

5. VISUAL THESAURUS

As a bye-product of our approach, we are able to build a "visual thesaurus" based on the statistical similarities of the visual representations of the terms.

In our approach, two words are similar in terms of "visual semantics" if they are used to describe items that are visually similar. Since each term is used to describe many items, the similarity is assessed statistically. In our approach, the visual representation (mean vector) and weight vector for two terms t_1 and t_2 are denoted as M-dimensional vectors: $\vec{\mu}_{t_1}$, $\vec{\mu}_{t_2}$, $\vec{\omega}_{t1}$, $\vec{\omega}_{t_2}$, respectively. The similarity between t_1 and t_2 is calculated as:

$$sim(t_1, t_2) = \frac{\sum_{i=1}^{M}(\mu_{t_1,i} \times \omega_{t_1,i}) \times (\mu_{t_2,i} \times \omega_{t_2,i})}{(\sum_{i=1}^{M} \mu_{t_1,i} \times \omega_{t_1,i}) \times (\sum_{i=1}^{M} \mu_{t_2,i} \times \omega_{t_2,i})}$$

To make it simple, the above formula returns the cosine similarity of two weighted vectors $\vec{\mu}_{t_1}$ and $\vec{\mu}_{t_2}$. Each vector is weighted by its own weight vector through an element-by-element multiplication, i.e. a weight (significance indicator) is enforced on each feature for each keyword. On the other hand, we also observed that some terms are so popular that they demonstrate moderate similarity with many other terms. We eliminated the high frequency terms through post-processing. On the other hand, we are also able to compute antonyms: words with similar weight vectors, but very different mean vectors.

As a conclusion, we are able to compute the semantic similarities between text terms, and such semantic similarities are coherent with human visual perception in this particular application domain. Further more, we have constructed a domain-specific "visual thesaurus" or a "visual WordNet", as shown in Table 1. This thesaurus could be used for query expansion for existing text-based product search engines in similar domains.

6. SYSTEM EVALUATION

6.1 Settings

We have implemented our iLike prototype on a database crawled from six selected fashion shopping sites. We obtained a 263-dimensional visual feature vector from the main product image for each item. Both the visual and textual feature pre-processing are carried out on a off-line basis. For each user query, we calculate the initial result set based on text-based retrieval, and display in the title row of output. For each item in the initial set, we expand the user query with the textual and visual features from the item, and enforce the weight vector which infers user intension. The query expansion parameters α, β are set to 0.9, 0.1, respectively. The search results using expanded and weighted query is displayed in columns, with the original item (from initial result set) in the title row.

6.2 Results and comparison

Examples of our search results are shown in Figure 7(a). To evaluate iLike, we use traditional Content-Based Image Retrieval approach as a baseline. The baseline approach employs the same visual features and database as iLike does, with the single difference that it skips query expansion and feature weighting. With original image features, the baseline

Table 1: visual thesaurus

Words	First Few Words in Visual Thesaurus
\feminine	bandeau, hipster, breezy, pregnancy, hem, lifestyle, braid, comfy, femininity.
\flirty	flirt, bikini, vibrant, effortlessly, pointelle, dressy, edgy, splashy, swimsuit
\gingham	subtle, sparkly, floral, gauze, glamour, sassy, surplice, beautifully, pajama
\trendy	adorn, striking, playful, supersoft, shiny, nancy, ladylike, cuddly, closure
\pinstripe	smock, sporty, khaki, pleat, oxford, geometric, gauzy, ruffle, chic, thong
\embroider	suede, crochet, versatility, ultra, corduroy, spectrum, softness, faux, crease
\twill	complement, plaid, contour, logo, decorative, buckle, classically, tagless

Table 2: name of similar items

\Girls Floral Applique Dresses	\Women's Floral Watercolor Scarves	\Floral pleated dress
Girls Slub-Knit Jersey Sundresses	Women's Abstract Floral-Print Scarves	Gingham crinkle dress
Girls Gauzy Ruffle-Tiered Dresses	Women's Lightweight Applique Scarves	Crawler dress
Girls Surplice Babydoll Dresses	Women's Embroidered-Eyelet Scarves	Pleated pear dress
Girls Metallic-Embroidered Tiered Dresses	Women's Lightweight Tonal Scarves	Chevron striped jumper dress
Girls Smocked Metallic-Stripe Dresses	Women's Striped Linen-Blend Scarves	Drop-waist bubble dress
Girls Striped Jersey Tube Dresses	Women's Polka-Dot Silk-Blend Scarves	Pintucked eyelet dress

algorithm uses Euclidean distance between two vectors from the visual space. The results are shown in Figure 7(b).

In the demonstration, we present the search results for query "floral" across the entire database. As is shown in Section 3, the term "floral" preserves significant importance in local texture features, especially in the domain of frequency. Therefore query with "floral" should put emphasis on the local texture features, while decrease the weight of other features such as color, shape and intensity features. Figure 7(a) turn out to be a convincing illustration of our assumption. Compare the first column of (a) and (b), we can see that the first seven items retrieved by iLike share high frequency local texture features, while items in the Baseline result set deviate their emphasis onto the local color and shape features. Moreover, without semantic restriction in the visual space, the second column of the baseline system are dominated by color features, which can explain the first four Cambridge blue images. The last two columns are essentially similar to the condition of the first group, with one simple difference: iLike successfully filters out the background noises, and hence it gives a higher recall.

To further evaluate our system, we first conduct TBIR with random keywords. Then we manually judge the quality of the top 30 similar items across 50 items in the initial result set (as first row of the interface), and mark each item with a boolean value, based on its relevance to the query. After that, we calculate the system precision and recall. In the same way, we can also obtain the quantitative evaluation of the Baseline approach. Precision-Recall Curve is shown as Figure 8.

In comparison with traditional text-based search, iLike has a clear advantage over search recall. Particularly, iLike is able to retrieve items that do not contain query terms in their description. To compare, we gather the text information of all the items returned by iLike. Table 2 shows three group of items retrieved by keyword "floral". Except for the initial results set, there is only one item that contains the query term (in both title and description fields).

To sum up, most of the results demonstrate patterns that fits our perception of the query terms. Especially, (1) not all the returned items have the term in the descriptions;

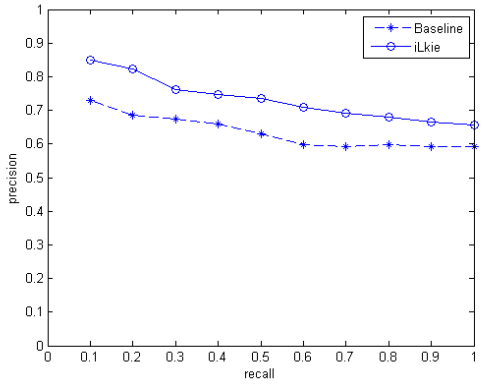

Figure 8: Precision-Recall Curve of iLike and Baseline approach.

they are retrieved by visual features. (2) if we only use the visual features from initial result set (row 1) as the query, the results will drift away from user intension. Many other items has higher overall visual similarity with the items in the initial set. Thanks to the weighting approach, we are able to infer the implicit user intension behind the query term, pick up a smaller subset of visual features that are significant to such intension, and yield better results.

7. CONCLUSION AND DISCUSSIONS

In this paper, we present iLike, a vertical search engine for apparel shopping. We aim to integrate textual and visual features for better search performance. We have represented text terms in the visual feature space, and developed a text-guided weighting scheme for visual features. Such weighting scheme infers user intension from query terms, and enhances the visual features that are significant towards such intension. Experimental results show that iLike is effective and capable of bridging the semantic gap.

In our experiments, the current version of iLike has demonstrated outstanding performance for a large number of de-

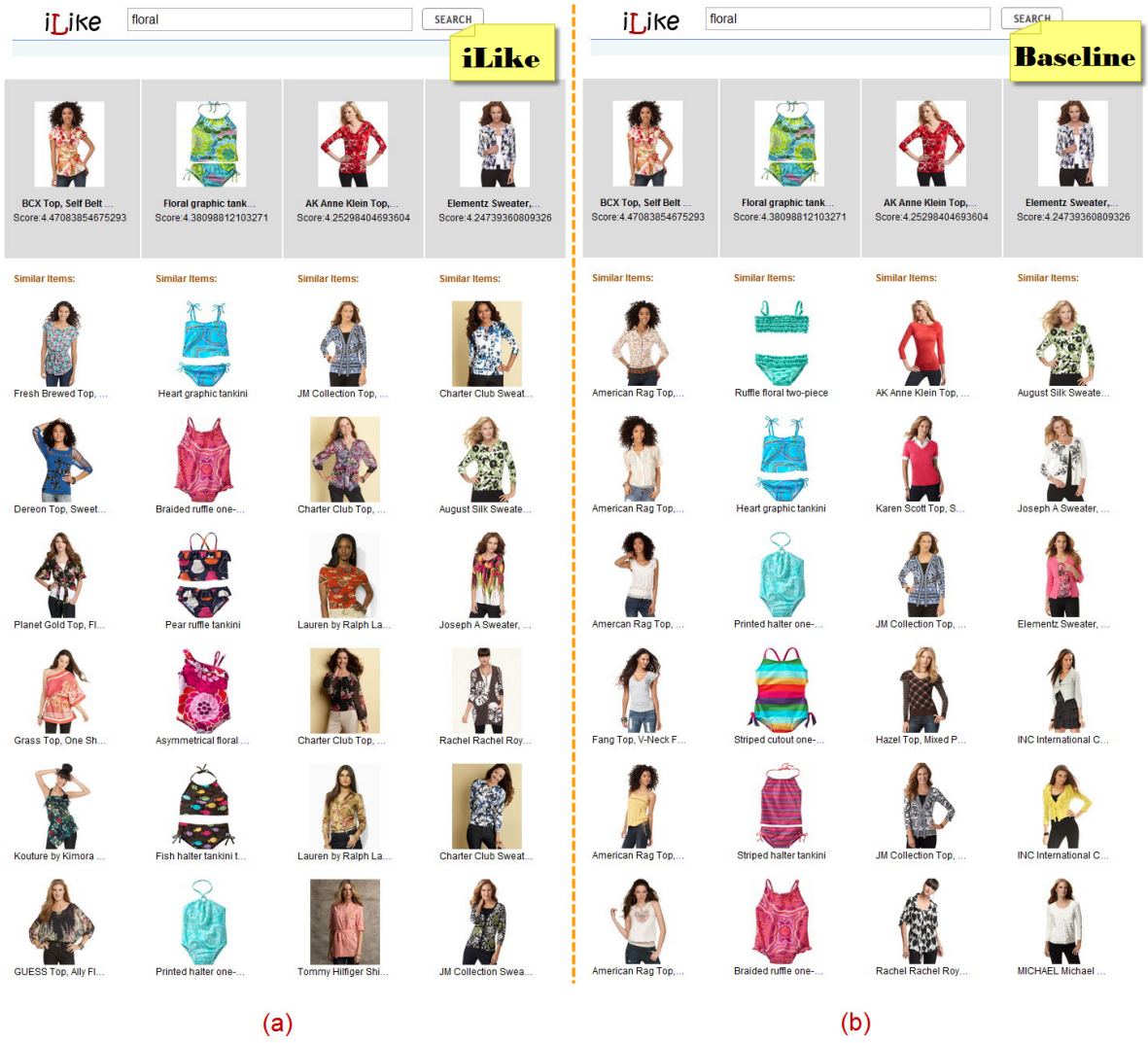

Figure 7: (a) iLike search results for keyword "floral"; (b) Baseline search results for keyword "floral";

scriptive terms. However, it does not work well for some keywords (mostly non-adjectives). Many of such words are very unlikely to be included in user queries (e.g. zip, logo). Meanwhile, we are still working on improving the performance of iLike. First, we will enlarge our database by implementing more focused crawlers and parsers. With more a larger database, we will be able to better assess visual meanings of text terms. Second, more advanced statistical learning approaches will be employed to manipulate the large number of samples. And finally, we plan to employ more visual features for product images. Due to the effectiveness of text-guided visual feature discrimination (weighting), we are able to simply add all kinds of visual features, and let iLike pick "good" ones.

8. ACKNOWLEDGEMENTS

This material is based upon work partially supported by the US National Science Foundation under Grant No. 0644366.

9. REFERENCES

[1] Y. A. Aslandogan, C. Thier, C. T. Yu, J. Zou, and N. Rishe. Using semantic contents and wordnet in image retrieval. In *ACM SIGIR conference on Research and development in information retrieval*, 1997.

[2] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen. Hierarchical clustering of www image search results using visual, textual and link information. In *ACM international conference on Multimedia*, 2004.

[3] Z. Chen, L. Wenyin, C. Hu, M. Li, and H.-J. Zhang. ifind: a web image search engine. In *ACM SIGIR conference on Research and development in information retrieval*, 2001.

[4] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, December 1998.

[5] J. Cui, F. Wen, and X. Tang. Intentsearch: interactive on-line image search re-ranking. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 997–998, 2008.

[6] J. Cui, F. Wen, and X. Tang. Real time google and live image search re-ranking. In *ACM international conference on Multimedia*, 2008.

[7] J. Cui, F. Wen, R. Xiao, Y. Tian, and X. Tang. Easyalbum: an interactive photo annotation system based on face clustering and re-ranking. In *SIGCHI conference on Human factors in computing systems*, 2007.

[8] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, 2008.

[9] T. Deselaers, D. Keysers, and H. Ney. Features for image retrieval – a quantitative comparison. In *DAGM 2004*, 2004.

[10] J.-X. Dua, X.-F. Wang, and G.-J. Zhang. Leaf shape based plant species recognition. *Applied Mathematics and Computation*, 185(2):883–893, February 2007.

[11] S. A. Dudani, K. J. Breeding, and R. B. McGhee. Aircraft identification by moment invariants. *IEEE Trans. Computers*, 26(1):39–46, 1977.

[12] C. Frankel, M. J. Swain, and V. Athitsos. Webseer: An image search engine for the world wide web. Technical report, Chicago, IL, USA, 1996.

[13] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.

[14] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM SIGIR*, 2003.

[15] F. Jing, C. Wang, Y. Yao, K. Deng, L. Zhang, and W.-Y. Ma. Igroup: web image search results clustering. In *ACM international conference on Multimedia*, 2006.

[16] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern Recogn.*, 40(3):1106–1122, 2007.

[17] I. Kompatsiaris, E. Triantafyllou, and M. Strintzis. A world wide web region-based image search engine. *International Conference on Image Analysis and Processing*, 0, 2001.

[18] P. Kovesi. Image features from phase congruency. *Journal of Computer Vision Research*, 1(3), 1999.

[19] S. Lawrence. Context in web search. *IEEE Data Engineering Bulletin*, 23:25–32, 2000.

[20] R. Lempel and A. Soffer. Picashow: pictorial authority search by hyperlinks on the web. In *10th international conference on World Wide Web*, 2001.

[21] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, 2006.

[22] X. Li, L. Chen, L. Zhang, F. Lin, and W.-Y. Ma. Image annotation by large-scale content-based image retrieval. In *ACM international conference on Multimedia*, 2006.

[23] H. Lieberman, E. Rozenweig, and P. Singh. Aria: An agent for annotating and retrieving images. *Computer*, 34(7):57–62, 2001.

[24] B. Luo, X. Wang, and X. Tang. A world wide web based image search engine using text and image content features. In *IS&T/SPIE Electronic Imaging, Internet Imaging IV*, 2003.

[25] W.-Y. Ma and H.-J. Zhang. Content-based image indexing and retrieval. *Handbook of multimedia computing*, pages 227–253, 1998.

[26] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge Univ. Press, New York, NY, 2008.

[27] S. Mukherjea, K. Hirata, and Y. Hara. Amore: A world wide web image retrieval engine. *World Wide Web*, 2(3):115–132, 1999.

[28] S. Sclaroff, L. Taycher, and M. L. Cascia. Imagerover: A content-based image browser for the world wide web. In *Workshop on Content-Based Access of Image and Video Libraries (CBAIVL)*, 1997.

[29] H. T. Shen, B. C. Ooi, and K.-L. Tan. Giving meanings to www images. In *ACM international conference on Multimedia*, 2000.

[30] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.

[31] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Trans. Systems, Man, and Cybernetics*, 8(6), 1978.

[32] V. S. Tseng, J.-H. Su, B.-W. Wang, and Y.-M. Lin. Web image annotation by fusing visual features and textual information. In *ACM symposium on Applied computing (SAC)*, 2007.

[33] A. Vijay and M. Bhattacharya. Content-based medical image retrieval using the generic fourier descriptor with brightness. *Machine Vision, International Conference on*, 0:330–332, 2009.

[34] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *SIGCHI conference on Human factors in computing systems (CHI)*, 2004.

[35] S. Wang, F. Jing, J. He, Q. Du, and L. Zhang. Igroup: presenting web image search results in semantic clusters. In *SIGCHI conference on Human factors in computing systems (CHI)*, 2007.

[36] Z. Wang, Z. Chi, and D. Feng. Fuzzy integral for leaf image retrieval. In *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*, 2002.

[37] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *SIGCHI conference on Human factors in computing systems (CHI)*, 2003.

[38] L. Zhang, L. Chen, F. Jing, K. Deng, and W.-Y. Ma. Enjoyphoto: a vertical image search engine for enjoying high-quality photos. In *ACM international conference on Multimedia*, 2006.

[39] L. Zhang, Y. Hu, M. Li, W. Ma, and H. Zhang. Efficient propagation for face annotation in family albums. In *ACM international conference on Multimedia*, 2004.

[40] X. Zhou, M. Wang, Q. Zhang, J. Zhang, and B. Shi. Automatic image annotation by an iterative approach: incorporating keyword correlations and region matching. In *ACM international conference on Image and video retrieval (CIVR)*, 2007.