Tag Tagging: Towards More Descriptive Keywords of Image Content

Kuiyuan Yang*, Xian-Sheng Hua, Member, IEEE, Meng Wang, and Hong-Jiang Zhang, Fellow, IEEE

Abstract—Tags have been demonstrated to be effective and efficient for organizing and searching social image content. However, these human-provided keywords are far from a comprehensive description of the image content, which limits their effectiveness in tag-based image search. In this paper, we propose an automatic scheme called tag tagging to supplement semantic image descriptions by associating a group of property tags with each existing tag. For example, an initial tag “tiger” may be further tagged with “white”, “stripes” and “bottom-right” along three tag properties: color, texture and location, respectively. In this way, the descriptive ability of the existing tags can be greatly enhanced. In the proposed scheme, a lazy learning approach is first applied to estimate the corresponding image regions of each initial tag, and then a set of property tags that correspond to six properties, including location, color, texture, size, shape and dominance, are derived for each initial tag. These tag properties enable much more precise image search especially when certain tag properties are included in the query. The results of the empirical evaluation show that tag properties remarkably boost the performance of social image retrieval.

Index Terms—Tag Tagging, property tags, image retrieval.

I. INTRODUCTION

RECENT years have witnessed a rapid growth of digital image collections on the Internet. The massive image data are not easily accessible to average users unless they are well organized. Indexing is an efficient way to organize massive data, which can be roughly divided into two categories in image retrieval area, one is visual-based and the other is textual-based. Visual-based indexing is typically used in content-based image retrieval (CBIR), in which images are represented by low-level feature vectors, and high dimensional indexing techniques are adopted to index these feature vectors [2], [41]. In CBIR system, users are generally required to indirectly provide low-level features as a query by giving an exemplar image or sketching out their search intention [26], [34], which seriously limits its usability. In addition, it only supports finding similar images in the low-level features, which are not necessarily “similar” in semantic level.

In textual-based indexing schemes, images are either automatically or manually annotated with a set of keywords. Although automatic image annotation [16], [35], [36], which is often based on machine learning techniques, has achieved notable success recently, the accuracy and scalability is still far from satisfactory [3]. Manual labeling has gained increasing attention due to the relatively higher tag quality, as well as its potential of leveraging massive Internet users [12]. Besides, different incentive schemes have made the tedious labeling work more interesting and acceptable, such as rewarding [28] and gaming [32]. For example, Amazon Mechanical Turk is an online platform where one can distribute labeling tasks to Internet users. EPS Game accomplishes the annotation via letting users play a carefully designed game. At online photo sharing websites such as Flickr, users around the world are self-motivated to annotate photos with free-form tags for photo organization and social sharing [1], [37], [42]. Hundreds of millions of photos are uploaded and tagged by more than 8.5 million registered users, and these numbers are keeping growing [25]. These tags make the massive photos semantically searchable.

However, there is still a gap between the manually-input tags and the users’ queries. A main reason is that the set of so-called basic-level tags [21] are much more frequently used when tagging the images. The basic-level theory [24] states that terms can be cognitively structured in a hierarchical system with three different levels of specificity: (1) the superordinate level, (2) the basic level and (3) the subordinate level. The basic level often contains terms that are mostly demonstrative but not specific. Rorissa investigates user behaviors regarding image tagging [23]. It was found that the basic level is generally used for the description of a single object and the superordinate level for description of an object group: “When describing images, people mainly made references to objects (mainly named at the basic level) in images much more than other types of attributes.” Goodrum and Spink also analyzed the search behavior during the retrieval of visual information and observed that quite often users initially search for basic-level terms, but then they will restrict the search results using the so-called “refiners” [9]. Thus the more specific and subordinate level queries are employed during image search. For example, terms like “red” and “round” may serve to refine a general term such as “balloon” into a more specific search request.

To reduce the gap between the tags and queries, in this paper we propose a so-called tag tagging scheme, which automatically adds a set of property tags to each of the existing initial tags. Six exemplary tag properties will be studied in this paper, including location, color, shape, texture, size and dominance. Tags provided by users can be classified into two
The rest of the paper is organized as follows. Section II reviews the detail formulation of Lazy Diverse Density algorithm; (3) tagging, object detection and attribute learning; (2) we present comprehensive survey on related work along three directions: enhancements in the following aspects: (1) We conducted more property tags. Region association. Descriptive information for the initial tags.

The main contributions of this paper can be summarized as automatically expanding tags with comprehensive information. To our knowledge, this work represents the first attempt towards image search can sufficiently leverage the tag properties. To the entire procedure of scheme. After property tags are added, image search can sufficiently leverage the tag properties. To our knowledge, this work represents the first attempt towards automatically expanding tags with comprehensive information. The main contributions of this paper can be summarized as follows:

1. We proposed a new “tag tagging” scheme to mine more descriptive information for the initial tags.
2. We employ a lazy diverse density approach for fast tag-region association.
3. We investigate more precise image search based on the property tags.

In comparison with a preliminary version in [43], we have enhancements in the following aspects: (1) We conducted more comprehensive survey on related work along three directions: tagging, object detection and attribute learning; (2) we present the detail formulation of Lazy Diverse Density algorithm; (3) more results, discussions, analyses and evaluations are provided; and (4) more comprehensive experiments are provided. The rest of the paper is organized as follows. Section II reviews the related work. Section III and IV, introduce the tag tagging scheme and image search with tag properties, respectively. Section V presents the experimental results. Finally, we conclude the paper in Section VI.

II. RELATED WORK

The proposed system is related to research efforts along three directions: tagging, object detection and attribute learning.

A. Tagging

Property tags can also be regarded as descriptive keywords that describe specific aspects of an image. However, property tags are more about the properties or characteristics of certain objects or scenes, which are seldom tagged by neither human being nor automatic annotation/tagging algorithms.

Another related work on tagging is tag enrichment through tag propagation (from another point of view, it can be regarded as annotation by search) [13], [17], [38]. However, although this scheme can supplement some missing tags by examining duplicate or near-duplicate images, tag with properties seldom exists in the initial tags thus cannot be propagated and enriched.

B. Object Detection

The tag-to-region component in our approach is related to object detection which aims to assign labels to image regions. Object detection is a challenging task because of the semantic gap between low-level features and high-level concepts. Extensive efforts have been dedicated to learning the mapping from low-level features to high-level concepts, and these approaches can be roughly divided into three categories according to the supervision degree in the training data:

1. Strong supervision. It this case, labels and the corresponding locations (delineation, segmentation or bounding box) are known in training data, from which object models can be directly learned from [31].
2. Weak supervision. It refers to the cases that only labels are given without correspondence region information. Many EM-like approaches alternately estimate the correspondence and learn the object model [7], [39]. Recently, Liu et al. directly estimate the correspondence through bi-layer sparse coding [18].
Fig. 2: The schematic illustration of tag tagging scheme. First an image is over-segmented into a set of image patches; Next, a tag to region module reassigns image-level tags to image regions; Finally, property tags for each tag are derived according to the content of its corresponding region.

C. Attributes Learning

Visual attributes are widely used in multimedia systems and computer vision. There are two forms to describe the visual attributes, one is low-level features and the other is high-level linguistic labels. A lot of work in cognitive science has been dedicated to studying the visual attributes. In [29], Thompson provides a philosophical discussions of color. Thorsten et al. show that in human vision, color sensations are greatly modulated by high-level memory instead of incoming sensory data alone [11]. Two recent papers are specialized in learning high-level linguistic labels from weakly supervised images [8], [30]. Since high-level visual attributes can be shared among many object categories, the knowledge learned from one category can be transferred to some other category via attributes if they have the attributes in common [5], [15]. Different from the work mentioned above, the “tag tagging” scheme focuses on extending existing initial tags with informative property tags.

III. TAG TAGGING

As illustrated in Fig. 2, the tag tagging scheme mainly consists of two steps: tag to region and property tag generation. Tag to region is to find each tag’s corresponding image region through Lazy Diverse Density. Property tag generation is to derive property tags based on the image regions found in the first step. In this section, we will detail the two steps.

A. Overview of Problem and Solution

By considering each image as a bag, each of which contains a number of instances corresponding to regions obtained from image segmentation, tag to region is to estimate each instance’s tag and intuitively it can be converted to a multi-instance learning problem. Diverse Density is a general framework proposed by Maron [19] for solving multi-instance problem. Diverse Density (DD) at a point in the instance feature space is defined as a measure of how many different positive bags have instances near that point, and how far the negative instances are from that point. An optimization algorithm such as gradient descent approach or EM-DD [44] is used to search for the point that achieves maximal DD, and once a point with maximum DD is found, a new instance can be classified according to its distance to the maximum DD point. In DD, one and only one point is taken as a general model for each class. But this is actually not suitable for tag to region task, since one tag may correspond to multiple local maximum points in the feature space. We can consider the following facts:

(1) One tag frequently corresponds to more than one region in an image. For example, a car in an image can be segmented into wheels, windows and body, and each part may correspond to one maximum DD in the feature space.

(2) Some tags have more than one prototype, e.g., “moon” has different shapes at different phases.

(3) Some tags are ambiguous. For example, “apple” may indicate fruit or digital products.

Instead of learning a general model for each tag, we adopt a lazy learning approach based on Diverse Density, that is Lazy Diverse Density. Since each point’s DD measures its positive degree, we directly estimate the DD of each instance and use the estimated DD to determine each instance’s tag. Due to the fact that each instance has its own best descriptive features, e.g., “flower” is well described by color and “zebra” is well described by texture etc, we compute the DD measure of each instance in three different feature spaces (color, shape and texture), and then the feature space with the highest DD
is regarded as best description for the instance. Finally, the instance’s tag is determined by comparing DD measurements computed with respect to tags from its bag. The Lazy Diverse Density approach for tag to region is detailed as follows.

Let \( D = \{ B_i, T_i \}_{i=1}^{N} \) denote the tagged image set, where \( N \) is the total image number, \( B_i = \{ B_{i1}, B_{i2}, \ldots, B_{in_i} \} \) represents an image, where \( B_{ij} \) is an over-segmented image patch by using graph-based segmentation algorithm in [6], and \( T_i = \{ T_{i1}, T_{i2}, \ldots, T_{im_i} \} \) is the tag set associated with the image \( B_i \). For each image patch, we extract three types of features, denoted by \( F \{ \text{color, shape, texture} \} \).

For an instance \( x \in B, T \) is the tag set associates with \( B \), and its DD with respect to tag \( t \in T \) in feature space \( f \) is defined as following:

\[
DD(x, t; f; D) = \sum_{1 \leq i \leq N, t \in T_i} P(x|B_i^{f}) + \sum_{1 \leq i \leq N, t \notin T_i} (1 - P(x|B_i^{f}))
\]

where \( DD(\cdot; D) \) means the DD is computed using bags from dataset \( D \), and \( P(x|B_i^{f}) \) is the “most-likely-cause” estimator which is defined as following,

\[
P(x|B_i^{f}) \propto \max_j \exp \left( -\frac{|B_{ij}^{f} - x|^{2}}{\sigma^{2}} \right)
\]

The computational cost of DD for all the instances is \( O(n^2 d) \), where \( n \) is the total instance number and \( d \) is the dimension of feature space. A solution to improve the computation efficiency is to approximately compute \( P(x|B_i^{f}) \). From Eq. 2, if all instances in bag \( B_i \) are far from \( x \) in feature space \( f \), then \( P(x|B_i^{f}) \approx 0 \), so we can approximate \( P(x|B_i^{f}) \) as,

\[
P(x|B_i^{f}) \propto \max_j \exp \left( -\frac{|B_{ij}^{f} - x|^{2}}{\sigma^{2}} \right) \approx \delta[x \sim B_i^{f}] \quad (3)
\]

where \( \delta[\text{expression}] \) is set to 1 when the expression is true, and 0 otherwise. Here \( x \sim B_i^{f} \) means bag \( B_i \) and \( x \) is “close” in feature space \( f \), where “close” means \( x \) has at least one instance of \( B_i \) as its \( k \)-Nearest Neighbor or at least one instance of \( B_i \) cites \( x \) as its \( k \)-Nearest Neighbor in feature space \( f \). Then \( DD(x, t, f; D) \) can be approximately computed as:

\[
DD(x, t, f; D) \approx \sum_{1 \leq i \leq N, t \in T_i} \delta[x \sim B_i^{f}] + \sum_{1 \leq i \leq N, t \notin T_i} (1 - \delta[x \sim B_i^{f}])
\]

\[
DD(x, t, f; D) \approx \frac{\sum \delta[x \sim B_i^{f}] + \sum (1 - \delta[x \sim B_i^{f}])}{N} \quad (4)
\]

The value embeds the information that how many bags tagged with \( t \) are close to \( x \) plus how many bags not tagged with \( t \) are far from \( x \) in feature space \( f \), which is consistent with the original definition of Diverse Density.

The feature space with maximum DD is regarded as best describes the instance, and then DD of instance \( x \) with respect to tag \( t \) is defined as,
Fig. 4: Example results on tag-to-region assignment. The original image and its corresponding region-tagged image are shown in pairs. The original image is illustrated in left and its region-tagged image is illustrated in right. Each color in the region-tagged images denotes one tag of localized region, white areas are tagged with background. The red contours overlapped on original images are boundaries of tagged regions before merging.

\[
DD(x, t; D) = \max_{f \in F} DD(x, t, f; D)
\]  

The tag of instance \(x\) is set to the one with the highest DD. Since tags are incomplete, some instances may have no corresponding tag in the bag’s tag set, so instance with highest DD below a threshold \(thresh\) is assigned with “background”,

\[
t(x) = \begin{cases} 
\text{background, if } & \max_{t \in T} DD(x, t; D) \leq \text{thresh} \\
\arg \max_{t \in T} DD(x, t; D), & \text{else} 
\end{cases}
\]

To visualize which instances achieved high DD for each tag, we sort the instances labeled with tag \(t\) according their DD, and then the top 10 instances for some tags are shown in Fig. 3, from which we can see that most instances with high DD are true positive instances.

After each image region is assigned a tag, adjacent regions with the same tag are merged together. Some exemplar tag-to-region results are shown in Fig. 4. These results over various conditions validate the effectiveness of our proposed solution.

### B. Property Tag Generation

After finding the corresponding image regions for each tag, other property tags can be derived according to the content of the corresponding image region. Since one tag may correspond to multiple regions\(^1\) in one image, we use the one with the highest DD to derive its properties. Although we only describe a small number of exemplary tag properties here, it should be noted that we can easily extend to support other properties as long as we can develop an acceptable detector for the corresponding property. Some examples include motion (whether the tag-specified object is moving or not), view (for example, “frontal”, “top”, “bottom or “side”), and even affections (for example, “happy” and “sad”, which are generally more difficult to detect). Another noteworthy issue is that not all property tags are meaningful for a tag. For example, shape is meaningless for sunset. These cases can be detected by thresholding the output of the corresponding detector, in such cases, we may set a value “NA” to this tag property.

1) **Location Tagging**: Location tag tells where the target tag appears in the image. Different location descriptors can be used here, such as coordinate and orientation. In current implementation, each image is partitioned into 3-by-3 equal grids, and each grid corresponds to one location tag as shown in Fig. 5 and 9 location tags are defined in total. The location tag for each initial tag is given according to which grid the centroid of its corresponding image region falls into.

2) **Size Tagging**: Size measures how big the tag-specified regions appear in the image, which is measured by the sum of area-ratios of all regions tagged with tag \(t\),

\[
S(t) = \sum_{p \in P_{t}} A_{p}
\]

where \(P_{t}\) denotes all the regions tagged with \(t\), and \(A_{p}\) denotes the area-ratio of region \(p\).
3) **Dominance Tagging:** Diverse Density of an image region with respect to the tag can be regarded as its relevance degree to the tag. Dominance measures how significant a tag appears in one image, and it is related to both the relevance and size of the tag-specified regions. Here we estimate it as the summation of Diverse Densities weighted by the corresponding area-ratios,

\[ D(t) = \sum_{p \in P_t} DD(x_p, t; D) A_p \]  

where \( P_t \) denotes all regions tagged with \( t \).

4) **Color Tagging:** Color is one of the most widely used descriptor. We use 11 basic color terms as our color tags: \( T_c = \{ \text{"black", "blue", "brown", "gold", "green", "orange", "pink", "purple", "red", "white", "yellow"} \} \). The color tag for each target tag is derived according to its region found in the image.

Intuitively, color tag can be set by comparing the RGB value of the image region with each color tag’s definition in RGB color space. However, these definitions are under ideal lighting on a neutral background may greatly differ from the color in the real world. To make the color tag more consistent with real world, we propose to learn the color tags from real-world training samples.

We select 50 true positive images for each of the 11 color names from search results returned by an image search engine, and denote the collected image set as \( T_c = \{ B_i, c_i \}_{i=1}^{N_c} \), where \( B_i \) represents a bag obtained the same as bags used in tag to region and \( c_i \in T_c \) is its color tag. For the target tag \( t_o \), we denote its corresponding image region as \( x \). Denoted by \( DD(x, c; \mathcal{C}) \) the Diverse Density of \( x \) computed with respect to color tag \( c \in T_c \) in the color feature space using the bags of image set \( \mathcal{C} \). It is computed using Eq. 4. Then the color tag for target tag \( t_o \) is set as,

\[ c(t_o) = \begin{cases} NA, if & \max_{c \in T_c} \arg \max_{c \in T_c} DD(x, c; \mathcal{C}) \leq \text{thresh}_c \\ \max_{c \in T_c} \arg \max_{c \in T_c} DD(x, c, color; \mathcal{C}), else \end{cases} \]  

where \( NA \) means we do not set color tag for this target tag if the DD below threshold \( \text{thresh}_c \).

5) **Texture Tagging:** Texture refers to the visual patterns that have properties of homogeneity resulting from the presence of more than one color or intensity [27]. It is an innate property of virtually all surfaces, including clouds, bricks, etc. There are many different texture description schemes available and here we define 13 common texture tags, i.e., \( T_s = \{ \text{"stripes", "spots", "skin", "furry", "clouds", "water", "grass", "tree", "bricks", "check", "tough_skin", "rock", "wood"} \}. Since there is no numerical definition of texture tags in low-level feature space, so we learn these texture tags by exploiting real-world training samples.

We select 50 true positive images for each of the 13 texture names from search results returned by an image search engine, and denote the collected image set as \( T_s = \{ B_i, t_i \}_{i=1}^{N_s} \), where \( t_i \in T_s \). Similar to color tagging, \( DD(x, t, texture; T) \) is the Diverse Density of \( x \) computed with respect to texture tag \( t \in T_t \) in the texture feature space using the bags of image set \( T \), then the texture tag for target tag \( t_o \) is set as,

\[ t(t_o) = \begin{cases} NA, if & \max_{t \in T_s} DD(x, t, texture; T) \leq \text{thresh}_t \\ \arg \max_{t \in T_s} DD(x, t, texture; T), else \end{cases} \]  

where \( NA \) means we do not set texture tag for this target tag if the DD below threshold \( \text{thresh}_t \). In Fig. 6, the second column shows some results of color tagging.

6) **Shape Tagging:** Shape is also an important description for perceptual objects. We define 5 common shape tags, i.e., \( T_s = \{ \text{"round", "rectangle", "triangle", "diamond", "heart"} \}, similar to the shape filters used by Picitup. Also due to shapes in real world are different with ideal defined ones, we learn the shape tags with real-world images.

---

*Fig. 5: Image is partitioned into 3-by-3 equal grids and each grid corresponds to one location tag.*

*Fig. 6: Example results on color tagging, texture tagging and shape tagging.*
We select 50 true positive images for each of the 5 shape names from search results returned by an image search engine, and denote the collected image set as \( S = \{ B_i, s_i \}_{i=1}^{N_s} \), where \( s_i \in T_s \). Similar to color tagging, \( DD(x, s, shape; S) \) is the Diverse Density of \( x \) computed with respect to shape tag \( s \in T_s \) in the shape feature space using the bags of image set \( S \), then shape tag for target tag \( t_o \) is set as,

\[
s(t_o) = \begin{cases} 
NA, \text{ if max } DD(x, s, shape; S) \leq \text{thresh}_s \\
\arg \max_{s \in T_s} DD(x, s, shape; S), \text{ else} 
\end{cases}
\]

where \( NA \) means we do not set shape tag for this target tag if the DD below threshold \( \text{thresh}_s \). In Fig. 6, the last column shows some results of shape tagging.

IV. IMAGE SEARCH WITH TAG PROPERTIES

To demonstrate the effectiveness of tag properties, in the section, we introduce a simple BM25-based [22] image search scheme that takes advantage of tag properties.

Each image is represented by its original tags and their associated property tags, which can be regarded as a short document. Okapi BM25 [22] is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document. The BM25 score of a image \( B_j \) with respect to a keyword \( q \) is define as,

\[
BM25(B_j, q) = IDF(q) \frac{tf(B_j, q)(k_1 + 1)}{tf(B_j, q) + k_1(1 - b + \frac{b \cdot |B_j|}{avgdl})}
\]

where \( tf(B_j, q) \) is \( q \)'s soft term frequency in the image \( B_j \), \( |B_j| \) is the total number of tags associated with \( B_j \), and \( avgdl \) is the average tag number of the image collection. The parameters \( k_1 \) and \( b \) are empirically set to \( k_1 = 2 \) and \( b = 0.75 \) in this work. The term \( IDF(q) \) is the IDF (Inverse Document Frequency) weight of query term \( q \). It is computed as:

\[
IDF(q) = \log \frac{N - n(q) + 0.5}{n(q) + 0.5}
\]

where \( N \) is the total number of images in the collection, and \( n(q) \) is \( q \)'s soft frequency in the collection,

\[
n(q) = \sum_{j=1}^{N} tf(B_j, q)
\]

Given a query \( Q \), contains keywords \( q_1, \ldots, q_n \), the BM25 score of an image \( B_j \) is:

\[
score(B_j, Q) = BM25(B_j, Q) + \sum_{i=1}^{n} BM25(B_j, q_i)
\]

where we add the first term in the r.h.s to take some compound queries into consideration, e.g., “red apple”. After obtaining the BM25 scores, images are sorted with BM25 scores in non-increasing order.

Since users often only focus on top results [14], images relevant to user queries should be ranked as high as possible. We adopt the precision of top results \( Prec@X \) as our performance evaluation metric,

\[
Prec@X = \frac{\text{# relevant images in top } X}{X}
\]

---

Fig. 7: The frequency distribution of the 601 tags and several examples.

V. EXPERIMENTS

In this section, we systematically evaluate the effectiveness of property tags on a publicly available dataset.

A. Data Set Description

We conduct experiments on NUS-WIDE dataset [4], which contains 269,648 images that are collected from Flickr. There are 425,059 tags associated with these images originally. To deal with the noises and synonyms in the original tags, we keep tags that belong to the “physical entity” group in the WordNet, and we change all plural to singular. Synonyms are grouped using WordNet synsets and the tags appear with too low frequencies are filtered out. Finally 601 tags are kept after these processes, Figure 7 illustrates their frequency distribution. Images with no tag are not used and thus we obtain a subset with 66,015 images, and in average there are 3.48 tags associated with each image. Then images are segmented into image patches, and the patches that are smaller than 1% of the original image are not taken into consideration. This results in 1,108,285 legible patches in total. Most works for assigning tags or labels to regions now adopt the process of first performing over-segmentation and then merging adjacent regions with the same tag or label, and existing study have shown that, since there will be a step of merging adjacent regions with the same tag, the segmentation performance is not sensitive to the parameters in segmentation [18]. For each image patch, we extract 125-dimensional color histogram features, 80-dimensional edge histogram features and 256-dimensional LBP texture features [20]. The neighborhood size \( k \) used for computing Eq. 4 is empirically set to 50 for all instances. The threshold parameters \( \text{thresh}_c, \text{thresh}_t \) and \( \text{thresh}_s \) are useful for controlling the computational complexity when the image collection is huge. In this paper, these thresholds are empirically set to 0.8 and only a few images are assigned to “NA”.

---

Copyright (c) 2011 IEEE. Personal use is permitted. For any other purposes, Permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.
B. Property Tags

Following the proposed approach in Section III-B, a set of property tags are assigned to each tag. We illustrate the frequency distribution of these added property tags in Fig. 8, 9, 10 and 11. Fig. 8 shows the frequency distribution of location tags. From the distribution on “all tags”, it is interesting that in general the property tag “center” appears most frequently and the four corner location tags appear much less. In fact, this is consistent with people’s capture behavior: people tend to locate the their interesting objects at the center of the frame when they take a picture.

Also we show the distribution of the location tags for some exemplary tags which have typical spatial distribution in real world (Fig. 8), such as “sea” appears more frequently in the bottom of the image while “sky” most appears in the top of the image, and “church” and “bicycle” most appear in the center of the image. This is consistent with their general spatial distributions in real world. However, this does not mean location tag can be directly inferred from the “name” of the initial tag itself. The imbalanced distribution only indicates that the predicted locations are consistent with common sense.

In Fig. 9, we show the frequency distribution of color tags. Specifically we demonstrate the distributions for specific tags “sea”, “sky”, and “church” as examples. We can see that our associated color tags are consistent with their color in real world, e.g., “sea” are mostly assigned with “blue”, “gold” (under sun shining) or “white”.

In Fig. 10, we show the frequency distribution of shape tags. Many tags are not suitable to assign basic shape terms, such as “girl”, “sea” and “sky”. But when employing these queries, users will also rarely add their shape information for search. From the distribution on the tags with explicit basic shapes, we can see that our assigned shape tags are consistent with real-world cases. For example, most signs are rectangle or round. Rectangle shape dominates all other shapes, this is due to the fact that many regions are bounded by image borders, these regions are assigned to rectangle with much higher probability in comparison with other shapes. Fortunately, in most images the interesting objects are at the center and their shape analysis will not be affected by this fact.

In Fig. 11, we show the frequency distribution of texture tags. From the distribution on the tags with obvious texture, we can see that our assigned texture tags are also consistent with real-world cases, e.g. leopard is assigned with “spots”, zebra...
TABLE I: The basic queries used in the experiments.

<table>
<thead>
<tr>
<th>Basic Query</th>
<th>Tag</th>
<th>Tag</th>
<th>Tag</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>girl</td>
<td>boat</td>
<td>car</td>
<td>cloud</td>
<td>church</td>
</tr>
<tr>
<td>sky</td>
<td>portrait</td>
<td>sunset</td>
<td>flower</td>
<td>wildlife</td>
</tr>
<tr>
<td>sea</td>
<td>bird</td>
<td>beach</td>
<td>moon</td>
<td>cow</td>
</tr>
<tr>
<td>dog</td>
<td>tree</td>
<td>street</td>
<td>apple</td>
<td>architecture</td>
</tr>
<tr>
<td>cat</td>
<td>horse</td>
<td>chair</td>
<td>bicycle</td>
<td>window</td>
</tr>
<tr>
<td>sign</td>
<td>mountain</td>
<td>elephant</td>
<td>helicopter</td>
<td>grass</td>
</tr>
<tr>
<td>snow</td>
<td>rose</td>
<td>bicycle</td>
<td>sailboat</td>
<td>tiger</td>
</tr>
<tr>
<td>model</td>
<td>waterfall</td>
<td>umbrella</td>
<td>bus</td>
<td>camel</td>
</tr>
<tr>
<td>petal</td>
<td>water</td>
<td>book</td>
<td>glacier</td>
<td>penguin</td>
</tr>
<tr>
<td>eye</td>
<td>sheep</td>
<td>palm</td>
<td>bus</td>
<td>lily</td>
</tr>
<tr>
<td>heart</td>
<td>squirrel</td>
<td>glacier</td>
<td>bus</td>
<td>lily</td>
</tr>
<tr>
<td>lion</td>
<td>balloon</td>
<td>leopard</td>
<td>bus</td>
<td>lily</td>
</tr>
<tr>
<td>bull</td>
<td>wolf</td>
<td>penguin</td>
<td>bus</td>
<td>lily</td>
</tr>
<tr>
<td>zebra</td>
<td>panda</td>
<td>motorcycle</td>
<td>bus</td>
<td>lily</td>
</tr>
</tbody>
</table>

We then further analyze the sensitivity of our property tag generation with respect to the number of positive examples. We conduct three experiments, color tagging, shape tagging and texture tagging, under identical conditions, and here we introduce color tagging in details. For each color tag, we collect 150 positive images from an image search engine as training data and label another 50 positive regions as testing data. For each color tag, we vary the number of positive examples from 10 to 100, and the Lazy Diverse Density for predicting the tag of testing data is repeated 10 times for each value (using different randomly selected training images). The prediction accuracy is used to measure the performance. Figure 12 shows the results of the color tagging, shape tagging and texture tagging respectively. The prediction accuracy generally increases when more positive examples are used. But the performance curves become steady when the number is above 50. The results also demonstrate that it generally requires fewer training examples to obtain satisfactory classification results for these properties, which is also consistent with the facts that these properties are with less visual variability compared with general categories such as “cat” and “dog”.

C. Image Search

We conduct image search on the 66,015 images described in Section IV to verify the effectiveness of property tags. We first select 58 basic queries (see Table I) from the kept tag list after filtering according popularity and diversity. For each basic query, we add meaningful modifiers to form specific queries such as “girl on the left” and “red balloon”. There are 804 queries generated in total. For simplicity, we use 1 to 804 to denote the IDs of these queries.

We compare the image search results based on the following indexing methods:

1. Image search without tag properties, i.e., index the images with original tags, the soft term frequency in Eq. 12 is simply set as the term frequency.

2. Image search with tag properties, i.e., index the images with both the original tags and the property tags added by our method. The soft term frequency in Eq. 12 is estimated...
Image search without properties
- green apple
- red apple
- blue apple
- rectangle
- apple
- round

Image search with properties
- round
- green
- apple
- round
- white
- apple

Fig. 13: The top 5 results of “apple”-related queries obtained by two different indexing methods. Each row corresponds to one query. The first column shows specific queries by adding modifiers to “apple”. The second column shows top image search results with tag properties, and image search results with original tags are shown in the last column. When “green”, “red” and “blue” are added as modifiers, our results of “green apple” and “red apple” are fruit, and “blue apple” are digital products for most these products have blue screens, when “rectangle” and “round” are used to restrict the search results, “rectangle apple” are digital products, “round apple” returns apple logo or fruit, and “round green apple” finds fruit and “round white apple” finds out the images with apple logo.

by dominance score in Eq. 8. It is worth mention here that dominance does not simply mean relevance. It depends on many other factors, such as the simplicity of the image, other objects in the image and so forth. But dominance is closely related to relevance. Intuitively, if an object is large and clear in an image, it should be highly relevant, and meanwhile our method will also assign it a high dominance score. Our experimental results will also demonstrate the effectiveness of integrating dominance in ranking.

Fig. 13 illustrates the top 5 results of the exemplar query “apple” with different added modifiers. From the figure we can see that the results of “image search with properties” are much more relevant than “image search without properties”. This shows the effectiveness of property tags in precise image search. The relatively lower performance of “image search without properties” is understandable since it only uses original tags, as mentioned before, many property tags are missed in original tags. Even occasionally there are some property tags in the original tag list of the image, they seldom have correspondence with other tags in the tag list, e.g., one image tagged with “red, apple, bag”, it does not tell whether there is a red apple or a red bag.

Fig. 15: Average Prec@X comparison with varied X of image search results using different methods. (a) Average performance comparison on the modified queries; (b) Average performance comparison on the basic queries.
To quantitatively compare the image search results, we invited 10 subjects to manually label the relevance of the top 40 images of each query and each method. Fig. 14 illustrates the \( \text{Prec}@40 \) measurements obtained by the two methods on the modified queries. We observe that “image search with tag properties” performs better on 673 of the all 804 queries. The average \( \text{Prec}@X \) results with varied \( X \) are illustrated in Fig. 15. Here we illustrate the performance comparison of the two methods “image search without tag properties” and “image search with tag properties” on two sets of queries, the modified queries and the basic queries. It can be observed that our method achieves much better search performance on both the modified queries and the basic queries. The better results on the modified queries clearly demonstrate the effectiveness of the added property tags, i.e. the added property tags can help better match to the modified queries. For the basic queries, since they only contain single term, the relevance scores are estimated by Eq. 12 which is a monotone increasing function of \( tf(B_j, q) \). Images are actually sorted by \( tf(B_j, q) \) in non-increasing order. For “image search with tag properties”, \( tf(B_j, q) \) is estimated by the dominance score. Therefore, the better results on the basic queries actually demonstrate that dominance scores are able to provide a better relevance clue for image search.

**VI. CONCLUSIONS AND FUTURE WORK**

This paper proposed a tag tagging scheme aiming at expanding existing tags with a set of property tags to supplement the missed descriptive information. With the popularity of photo sharing websites, community-contributed images with tags are much easier to obtain, and the keyword-based semantic image search can greatly benefit from our proposed technique. The experiments also demonstrate that search with property tags significantly improves search relevance. It is worth noting that though we only use Flickr data in this work, the proposed method is a general approach and can be applied for other data sources, such as dealing with tags of Youtube’s video, and property tags can be added like location in temporal, motion, etc. We can also extend the approach to handle the contextual text information of images, such as surrounding text, titles and comments. In addition, property tags can also be used to generate diverse search results. For example, we can diversify search results for a query by illustrating the results that are with different properties.

**REFERENCES**


Kuiyuan Yang received the B.E. degree from the University of Science and Technology of China in Automation, Hefei, China, in 2007. He is now with the Media Computing Group, Microsoft Research Asia, Beijing, China, as a Research Intern. His research interests include computer vision, multimedia, and machine learning, especially content-based image/video retrieval, analysis, management, and sharing.

Xian-Sheng Hua (M’05) received the B.S. and Ph.D. degrees from Peking University, Beijing, China, in 1996 and 2001, respectively, both in applied mathematics. When he was in Peking University, his major research interests were in the areas of image processing and multimedia watermarking. Since 2001, he has been with Microsoft Research Asia, Beijing, where he is currently a Lead Researcher with the internet media group. His current interests are in the areas of video content analysis, multimedia search, management, authoring, sharing, and advertising.

Dr. Hua is a senior member of the Association for Computing Machinery, a member of the Visual Signal Processing and Communications Technical Committee (VSPC TC) and Multimedia Systems and Applications Technical Committee (MAS TC) of the IEEE Circuits and Systems Society. He is an associate editor of the IEEE TRANSACTIONS ON MULTIMEDIA. He was the recipient of the Best Paper Award and Best Demonstration Award at ACM Multimedia Conference in 2007, the Best Poster Award at the 2008 IEEE International Workshop on Multimedia Signal Processing, the Best Student Paper Award at the ACM Conference on Information and Knowledge Management in 2009, and the Best Paper Award at the International Conference on Multimedia Modeling in 2010. He was also the recipient of the 2008 MIT Technology Review TR35 Young Innovator Award and named as one of the “Business Elites of People under 40 to Watch” by Global Entrepreneur.

Meng Wang (M’09) received the B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 1999 and 2008, respectively. From 2008 to 2010, he was with Microsoft Research Asia, Beijing, China, as a Research Staff Member. In 2010, he joined Akira Media Systems, Palo Alto, CA, as a Research Scientist. He has authored or coauthored over 60 papers and book chapters. He served as an editorial board member, guest editor, or program committee member of numerous international journals and conferences. His current research interests include multimedia content analysis, management, search, mining, and large-scale computing.

Dr. Wang is a member of the Association for Computing Machinery. He was the recipient of the Best Paper Award at the ACM International Conference on Multimedia 2009 and the Best Paper Award at the International Multimedia Modeling Conference 2010.

Hong-Jiang Zhang (M’91–SM’97–Fellow’03) received his Ph.D from the Technical University of Denmark, Lyngby, in 1991, and his BS from Zhengzhou University, Henan, China, 1982, both in Electrical Engineering, respectively.

From 1992 to 1995, he was with the Institute of Systems Science, National University of Singapore, where he led several projects in video and image content analysis and retrieval and computer vision. From 1995 to 1999, he was a research manager at Hewlett-Packard Labs, Palo Alto where he was responsible for research and development in the areas of multimedia management and intelligent image processing. In 1999, he joined Microsoft Research, where he is currently the Managing Director of Advanced Technology Center in Beijing.

Dr. Zhang is a Fellow of IEEE and ACM. He has coauthored/co-edited 4 books, over 350 papers and book chapters, numerous special issues of international journals on image and video processing, content-based media retrieval, and computer vision as well as over 60 granted patents. He currently serves as the Editor In Chief of IEEE Transactions on Multimedia and on the editorial board of Proceedings of IEEE.

Copyright (c) 2011 IEEE. Personal use is permitted. For any other purposes, Permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.