Visual Query Attributes Suggestion

Jingwen Bian
National University of Singapore, Singapore
bian_jingwen@nus.edu.sg

Zheng-Jun Zha
National University of Singapore, Singapore
zhazj@comp.nus.edu.sg

Hanwang Zhang
National University of Singapore, Singapore
hanwang@comp.nus.edu.sg

Qi Tian
University of Texas at San Antonio, USA
qitian@cs.utsa.edu

Tat-Seng Chua
National University of Singapore, Singapore
chuats@comp.nus.edu.sg

ABSTRACT

Query suggestion is an effective solution to help users deliver their search intent. While many query suggestion approaches have been proposed for test-based image retrieval with query-by-keywords, query suggestion for content-based image retrieval (CBIR) with query-by-example (QBE) has been seldom studied. QBE usually suffers from the *"intention gap"* problem, especially when the user fails to get an appropriate query image to express his search intention precisely. In this paper, we propose a novel query suggestion scheme named Visual Query Attributes Suggestion (VQAS) for image search with QBE. Given a query image, informative attributes are suggested to the user as complements to the query. These attributes reflect the visual properties and key components of the query. By selecting some suggested attributes, the user can provide more precise search intent which is not captured by the query image. The evaluation results on two real-world image datasets show the effectiveness of VQAS in terms of retrieval performance and the quality of query suggestions.

Categories and Subject Descriptors

H.3.3 [**Information Storage**]: Information search and retrieval—*Query formulation*

General Terms

Algorithms, Experimentation, Human Factors

Keywords

Query Suggestion, Attribute, Image Search

1. INTRODUCTION

With the explosive growth of media resources on the Web, and the increasingly stringent requirements in many media applications [5, 6, 7, 10], Content Based Image Retrieval (CBIR) has attracted significant attention in both academia and industry. Most CBIR systems allow users to specify their search intention by uploading an example image as query, called query-by-example (QBE). Although QBE has shown encouraging potential in image retrieval, it usually suffers from the *"intention gap"* problem between user's search intention and the query, especially when the submitted query image is ambiguous. Generally, the *"intention gap"* arises from the following two problems: **a)** the query image contains other components that are not part of user's interests. Given such a query image, the true intention of the user is hidden behind the noise, leading to undesired search results; **b)** the query image does not contain sufficient components or fails to capture some important properties of the intended objects. Figure 1 shows an example of these two problems. Suppose a user has an image of a car and wants to see more examples of it. But the user can only find a query image of "a car with a model". While noisy content, i.e., *the model*, appears in the query, some components of car, such as *wheels* and *side mirrors*, etc., are not included. The first problem can be partially solved by drawing a bounding box around the focused part of the query image [4]. However, due to the complexity of shapes and positions, it is sometimes hard to draw a proper bounding box. Besides, query by bounding box is not able to overcome the second problem.

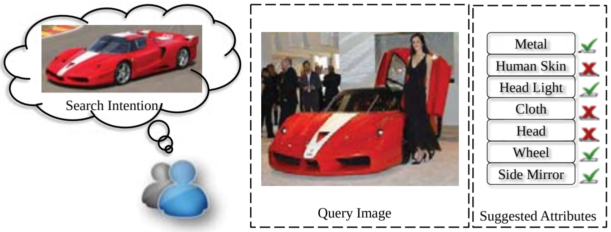


Figure 1: Example search procedure using VQAS.

On the other hand, query suggestion has been found as an effective solution to narrow down the *intention gap*. Query suggestion has been widely studied for image retrieval with query-by-keywords [8, 9]; however, it has been seldom studied for content based image retrieval (CBIR) with query-by-example (QBE). In order to perform query suggestion for QBE, we should be able to suggest topics related to the

MM'12, October 29–November 2, 2012, Nara, Japan.

query image. However, it is usually difficult to understand the semantic meanings of an image. Recently, *attributes* have shown encouraging capacity in describing objects and distinguishing different objects. As a kind of intermediate level semantic descriptors, attributes refers to visual properties (e.g., "round" as shape, "metallic" as texture), components (e.g., "has wheel", "has leg") and functionalities (e.g.,"can fly", "man-made") of objects. The advantages of attributes make it possible to formulate intention-specific query in terms of attributes.

Inspired by the above observations, we propose a novel query suggestion approach for CBIR with QBE, named visual query attributes suggestion (VQAS). Attributes are exploited to help users deliver their search intention precisely. Given a query image, the most informative attributes related to the query are suggested to the users. Two kinds of attributes are considered as informative: a) the attributes with high probability of presence in the query. These attributes express the content of the query. Through feedbacks on these attributes, users can state what content is or not in their search intention; and b) the attributes frequently co-occurred with those shown in the query image. These attributes are likely to be of user's interest. By selecting them, users can indicate the desired content that is not included in the query. For example, given the query image in Figure 1, VQAS suggests *"Metal"*, *"Head Light"*, *"Human Skin"*, *"Cloth"*, and *"Head"* which appear in the query, as well as *"Wheel"* and *"Side Mirror"* which are not included in the query but is highly likely to be user's interest. By suggesting such attributes, VQAS can help users to formulate a more intention-specific query. Specifically, the workflow of VQAS is as follows. Offline, VQAS learns a set of binary classifiers, each of which predicts the presence of an attribute in an image. When a query image is submitted online, VQAS utilizes the classifiers to predict the presence of the attributes in the query. The above two kind of informative attributes are then discovered and presented for user selection. VQAS then generates search results by simultaneously exploiting both low-level visual features as well as the selected attributes.

2. APPROACH

2.1 Attribute Learning

As aforementioned, attributes refer to visual properties (e.g., "round" as shape, "metallic" as texture), components (e.g., "has wheel", "has leg") and functionalities (e.g., "can fly", "man-made") of objects. Here, we focus on the first two kinds of attributes.

Given a training set of images which are labeled with attributes, one can easily train the classifiers such as SVM. However, due to the large visual variance of some attributes (e.g., the "wing" of a plane looks very different from the "wing" of a chicken), standard training methods cannot be directly applied because they may be trained by irrelevant feature dimensions. For example, the "wing" classifier may be unfortunately trained based on the feature dimensions related to "sky". Therefore, we need to select feature dimensions that are most informative and effective to the target attributes.

In this paper, we apply the feature selection method as described in [1]. The selected feature dimensions of an attribute are the union of feature dimensions that are most discriminative for sub-categories. For example, in order to train the "wing" classifier, we first find "chicken" with and without "wing" as training samples, and train a preliminary *linear* classifier called "chicken wing". Then, we may train another preliminary classifier called "plane wing" and so on. By doing so, we finally obtain a set of parameters of such preliminary classifiers. Particularly, we use the ℓ_1-norm regression [3] as the preliminary classifiers due to its good performance on sparsity of the parameters. The features are then selected by pooling the union of indices of the sparse nonzeros entries in those parameters. Once we have selected the feature dimensions, we can apply standard training methods to learn an overall attribute classifier (e.g., SVM).

2.2 Attribute Suggestion

Given a query image, attribute suggestion aims to offer a list of most informative attributes to help users specify their search intention. As aforementioned, an attribute is considered to be informative if it has high probability of presence in the query image or it is not in the query but is likely to be user's interest. Users' feedbacks on these attributes compose a semantic description of the search intention, either filtering noisy attributes, emphasizing attributes of interest in the query, or indicating missing attributes of interest. In particular, we first exploit the responses of attribute classifiers to find a set of attributes with high probabilities of presence in the query. Then, we discover the missing attributes of interest based on their co-occurrence to the attributes with high probabilities. All these two types of attributes are finally presented for user selection. Let $\mathcal{A} = \{a_1, a_2, \cdots, a_m\}$ denote the set of m attributes. The two types of informative attributes are discovered as follows:

Attributes of high confidence $\mathcal{A}^c$: These are attributes with the highest probability of appearing in the query image. The confidence of an attribute a_i on query image q is defined as

$$conf(a_i|q) = P(a_i|q) \times conf(c_i) \tag{1}$$

where c_i is the classifier for attribute a_i. $P(a_i|q)$ is the response of c_i on q. $conf(c_i)$ is the reliability of c_i. $conf(c_i)$ plays as a kind of prior knowledge to assist discover the informative attributes. We here define $conf(c_i)$ as the AUC value of c_i. n attributes with the highest confidence scores are selected and denoted as $\mathcal{A}^c$.

Missing attributes of interest $\mathcal{A}^i$: These are attributes not shown in query image, but are part of users' search intention. Due to occlusion, some desired attributes may be missing (e.g., the occluded wheel of an intended car). Suppose we already have the set of attributes of high confidence $\mathcal{A}^c$, given the fact that attributes belonging to the same objects often occur simultaneously, we can find those missing attributes of interest by looking for attributes with highest co-occurrence with those in $\mathcal{A}^c$. We define the co-occurrence score of a missing attribute (i.e., attribute a which satisfies $a \in \mathcal{A}, a \notin \mathcal{A}^c$) a_j as:

$$cooc(a_j) = \max_{i=1,2,\ldots,n} P(a_j|a_i^c) \tag{2}$$

where n is the number of attributes in $\mathcal{A}^c$.

2.3 Image Search with Attribute Suggestion

The search results are generated by exploiting both low level visual features of images and the attributes selected by users. We first compute the visual relevance of images in database based on their visual similarities to the query image, as well as the attribute relevance based on the attribute features. The attribute feature of each image is represented as the responses of the attribute classifiers on the image. Based on the attributes selected by users, a new kind of relevance, termed attribute coherence, is then computed. The attribute coherence of an image is computed as an aggregation of its probability of containing the positive attributes (i.e., the attributes of interest) and that of not containing the negative attributes. All these relevance scores for an image are then aggregated to form the final relevance. Finally, VQAS presents the images sorted by their relevance with a descending order.

Let x denote a database image. Its three kinds of relevance scores are computed as follows:

Visual Relevance is computed based on the low-level features as

$$r_v(x,q) = e^{-\frac{\|x-q\|_2^2}{\sigma_v}} \tag{3}$$

where σ_v is the normalization parameter.

Attribute Relevance is computed based on the attribute features as

$$r_a(x,q) = e^{-\frac{\sum_{a_i \in \mathcal{A}} |P(a_i|q)-P(a_i|x)|^2}{\sigma_a}} \tag{4}$$

where $P(a_i|x)$ and $P(a_i|q)$ are the responses of a_i's classifier on x and q, respectively. They measure the possibilities of the presence of a_i in x and q.

Attribute Coherence is proposed to measure the relevance of x to user's feedbacks on the suggested attributes. Suppose the user gives feedbacks on K out of all the suggested attributes. Let $\mathcal{F} = \{f_1, f_2, \ldots, f_K\}, f_k \in \{0,1\}$ denote user's feedbacks. $f_k = 0$ or 1 indicates that the kth attribute is or not part of user's search intention. Here, we adopt the Direct Attribute Prediction (DAP) model [2] to compute the coherence of x to $\mathcal{F}$ as follows:

$$r_c(x) = \prod_{k=1}^{K} P(a_k = f_k|x) \tag{5}$$

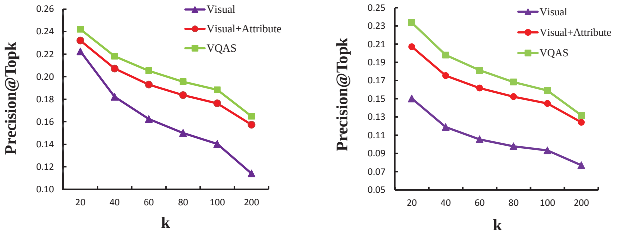
where $P(a_k = 1|x)$ is computed as $conf(a_k|x)$ and $P(a_k = 0|x)$ is computed as $1 - conf(a_k|x)$. We can see that the above formula gives high relevance score to the images whose attributes are consistent with user's selection.

3. EXPERIMENTS

3.1 Data Set Description

We conducted experiments over the following two real world image datasets: the Pascal-Yahoo! (PY) image corpus [1] and a Web Image collection downloaded from the Web.

Pascal-Yahoo!. This dataset contains 15,339 images collected from Pascal VOC 2008 (12,695 images) and Yahoo! image search engine (2,644 images). The images are from 32 object categories and all of them were annotated with 64 pre-defined attributes. In particular, 6,340 images from Pascal were used to train the classifiers. The rest of 6,355 images in Pascal together with the 2,644 images in Yahoo! were used for image retrieval. We randomly selected 10 images from each category as experimental queries. This gives rise to 320 queries in total.



(a) (b)

Figure 2: Average precision on (a) Pascal-Yahoo! dataset (320 queries); (b) Web Image dataset (700 queries).

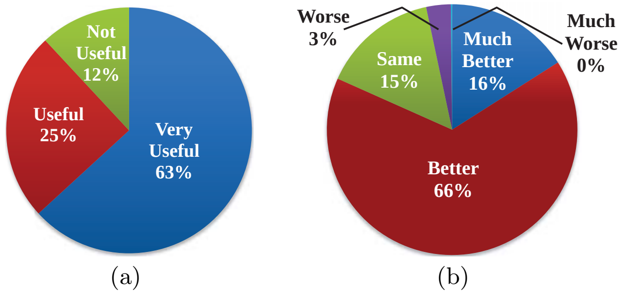
Web Image[1]. We collected another web image corpus from Microsoft Bing search engine with 120 popular text queries in *Animal* and *Object* domains. In total, there are 76,303 images, with 300-800 images per query. We defined 67 attributes by referring to the 64 attributes in **PY** data. We randomly selected 50 classes with 25,203 images as the development set for attribute learning, and the remaining 70 classes with 51,100 images as the test set for retrieval. Due to the high cost of labeling, we only manually annotated the 67 attributes on the development set. We use 80% of images in the development set as training samples in attribute learning, and the remaining 20% images were used for testing. In retrieval process, we randomly selected 10 images from each of the 70 classes as queries, and thus obtained 700 queries in total.

3.2 Visual Features

We used four types of features, including color and texture, which are good for material attributes; edge, which is useful for shape attributes; and scale-invariant feature transform (SIFT) descriptor, which is useful for part attributes. We used a bag-of-words style representation for each of these four features.

Color descriptors were densely extracted from each pixel as the 3-channel LAB values. The color descriptors of each image were then quantized into a 128-bin histogram. Texture descriptors were computed for each pixel as the 48-dimensional responses of texton filter banks. The texture descriptors of each image were then quantized into a 256-bin histogram. Edges were found using the standard canny edge detector and their orientations were quantized into 8 unsigned bins. This gives rise to a 8-bin edge histogram for each image. SIFT descriptors were densely extracted from the 8×8 neighboring block of each pixel with 4-pixel step size. The descriptors were quantized into a 1000-dimensional bag-of-words feature. Afterward, for each image, we concatenated these four type of features into a 1392-dimensional vector, which was used in image retrieval. Since attributes usually appear in one or more regions in an image, we further split each image into 2×3 grids and extracted the above four kinds of features from each grid respectively. Finally, we obtained a 9744-dimensional feature from the grids and a 1392-dimensional feature from the whole image. The 9744-dimensional feature was used in attribute learning.

[1] `http://www.comp.nus.edu.sg/~hanwang/data.html`



(a) (b)

Figure 3: Evaluation results over 600 queries from 30 users in terms of (a) the usefulness of suggested attributes and (b) the quality of search results generated by VQAS.

3.3 Experimental results

3.3.1 Evaluation on Retrieval Performance

We compared the proposed VQAS with the following two baseline methods: (a) image retrieval based on low-level visual features; and (b) image retrieval based on both visual and attribute features. The visual and attribute relevance scores of an image were computed according to Eq.3 and Eq.4, respectively. In our VQAS, seven attributes of high confidence and five missing attributes of interest were suggested for each query image on average. We adopted the precision@k as the performance metric, which measures the precision at the top k search results. We averaged precision@k over all the experimental queries and obtained the overall metric average precision@k (AP@k).

The performance of VQAS and the two baseline methods on the two datasets are illustrated in Figure 2(a) and Figure 2(b), respectively. We can see that VQAS outperforms the baseline methods with various k on the two datasets. By suggesting informative attributes, VQAS can help user deliver search intention more precisely, leading to more relevant search results.

3.3.2 Evaluation on Suggestion Quality

We conducted a user study of 30 people to evaluate the quality of query suggestion by VQAS. Each participant was asked to submit 20 queries and evaluate the following two aspects of VQAS:

The usefulness of the suggested attributes: The participants were asked to evaluate whether the suggested attributes were useful to help them express their search intention more precisely.

The quality of search results with VQAS: The participants were asked to compare the search results with and without VQAS.

The experimental results are shown in Figure 3. These results demonstrate that VQAS is capable of suggesting useful attributes for most of the queries, and generating search results which are more relevant to users' search intention.

4. CONCLUSIONS

In this paper, we proposed a novel query suggestion approach for content-based image retrieval with QBE, named Visual Query Attribute Suggestion (VQAS). VQAS suggests a list of informative attributes based on users' query image and is able to help users specify and deliver their search intention in a more precise way. We conducted extensive experiments to evaluated the proposed VQAS. The experimental results demonstrated the effectiveness of VQAS in terms of retrieval performance and the quality of query suggestion.

5. ACKNOWLEDGMENTS

This work was supported by NUS-Tsinghua Extreme Search (NExT) project under the grant number: R-252-300-001-490.

6. REFERENCES

[1] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[2] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[3] A. Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *ICML*, 2004.

[4] K. Vu, K. Hua, and W. Tavanapong. Image retrieval based on regions of interest. *TKDE*, 2003.

[5] M. Wang, X. Hua, J. Tang, and R. Hong. Beyond distance measurement: Constructing neighborhood similarity for video annotation. *TMM*, 2009.

[6] M. Wang, K. Yang, X. Hua, and H. Zhang. Towards a relevant and diverse search of social images. *TMM*, 2010.

[7] Y. Yang, Y. Yang, Z. Huang, H. Shen, and F. Nie. Tag localization with spatial correlations and joint group sparsity. In *CVPR*, 2011.

[8] Z. Zha, L. Yang, T. Mei, M. Wang, and Z. Wang. Visual query suggestion. In *MM*, 2009.

[9] Z. Zha, L. Yang, T. Mei, M. Wang, Z. Wang, T. Chua, and X. Hua. Visual query suggestion: Towards capturing user intent in internet image search. *TOMCCAP*, 2010.

[10] S. Zhang, Q. Tian, G. Hua, Q. Huang, and S. Li. Descriptive visual words and visual phrases for image applications. In *MM*, 2009.

APPENDIX

Queries used in PY dataset: *aeroplane, bag, bicycle, bird, boat, bottle, building, bus, car, carriage, cat, centaur, chair, cow, diningtable, dog, donkey, goat, horse, jetski, monkey, motorbike, mug, person, pottedplant, sheep, sofa, statue, train, tvmonitor, wolf, zebra.*

Queries used in Web Image dataset: *arrow, banana, beer, billiard, birthday cake, boomerang, bouquet, building, butterfly, cactus, camion, canard, capuche, cattle, champagne, cherry, chess, chevy, chopper, corvette, cricket, crocodile, crown, diamond, dragonfly, ferrari, ferret, gemstone, giraffe, goggle, great dane, haricot, helmet, high heel, ice sculpture, jeans, jeep, jug, kangaroo, kayak, kilt, lamborghini, lip, lizard, lobster, monkey, moon, new year card, olive, orange, owl, palm tree, pencil, penguin, pizza, popcorn, python, quail, raven, reindeer, rhinoceros, rodent, shark, shoe, skull, spear, unicorn, virus, voiture, wolf.*

Attributes defined on PY dataset: *2D_Boxy, 3D_Boxy, Arm, Beak, Clear, Cloth, Door, Ear, Engine, Exhaust, Eye, Face, Feather, Flower, Foot_or_Shoe, Furn_Arm, Furn_Back, Furn_Leg, Furn_Seat, Furry, Glass, Hair, Hand, Handlebars, Head, Headlight, Horiz_Cyl, Horn, Jet_engine, Label, Leaf, Leather, Leg, Mast, Metal, Mouth, Nose, Occluded, Pedal, Plastic, Pot, Propeller, Rein, Round, Row_Wind, Saddle, Sail, Screen, Shiny, Side_mirror, Skin, Snout, Stem_or_Trunk, Tail, Taillight, Text, Torso, Vegetation, Vert_Cyl, Wheel, Window, Wing, Wood, Wool.*

Attributes defined on Web Image dataset: *Refer to the attributes defined on PY dataset, except that we remove "2D_Boxy" and "3D_Boxy", and add "Transparent", "ToughSkin", "Dotted", "Knit" and "Smooth".*