Object-Layout-Aware Image Retrieval for Personal Album Management

Mengyou Li*[12], Zheng Song*[1], Qiang Chen[1], Liang Lin[2], Zhongyang Huang[3], and Shuicheng Yan[1]

[1]National University of Singapore
[2]Sun Yat-Sen University [3]Panasonic Singapore Laboratory

Abstract. This demo shows a real-time object-layout-aware image retrieval system for personal album management. The query of the system is image's object layout and the system retrieves images based on the layout similarity of concerned objects in the query.
Using automatic object localization algorithms, we propose to index the personal album with the object positions and scales in the 2D image plane, which can reflect their layout in 3D scene and hence be used to measure the object layout similarity.
To query from the database, the users need describe their query by placing objects onto a canvas and configuring the position and scale of the objects. Then the system searches the album database and returns photos with the similar object layout. This system can facilitate people in finding the photos of which they have impression on the layout of certain objects.

Key words: Object Layout, Image Search, Photo Search

1 Introduction

While more and more visual recognition techniques are integrated into image retrieval system, most systems still index the database at image level, namely only provide tags about the whole image instead of detailed descriptions such as object location, human-object interaction, etc.

However, there is growing need in sensing semantically meaningful objects in the images and searching by the layout of human and key objects. In this proposed system, we adopt the recent works on object localization [1] and present an image search technique effectively utilizing object interactional information.

In our demo system, we first apply object localization algorithms to the images and extend the previous image-level indexing scheme to an object-level indexing scheme. We also propose to extend the visual recognition topics to quite a few daily objects and consequently build comprehensive object-level descriptions to the images.

To achieve the proposed new system, we require the users provide the expected object layout positions and scales in the 2D image geometry in order to

* indicates equal contributions

reproduce the 3D layout of objects in the image scene. This requirement is consistent with common sense since if one person remember the image content, it can always memorize the layout of the content as well. It provides more flexibility for users to define the images as "how it looks like".

Using the layout of objects as the query is also an appropriate compensation to the current yet not accurate object localization algorithms. To integrate object localization with object layout search, we set a low threshold to the object localization algorithm to produce high recall yet high false positive rate detection results. Then we simultaneously match the positions and scales of all the objects in the query with the detected objects in the database images and find the most confident combinations of the detection results.

2 Client Interface

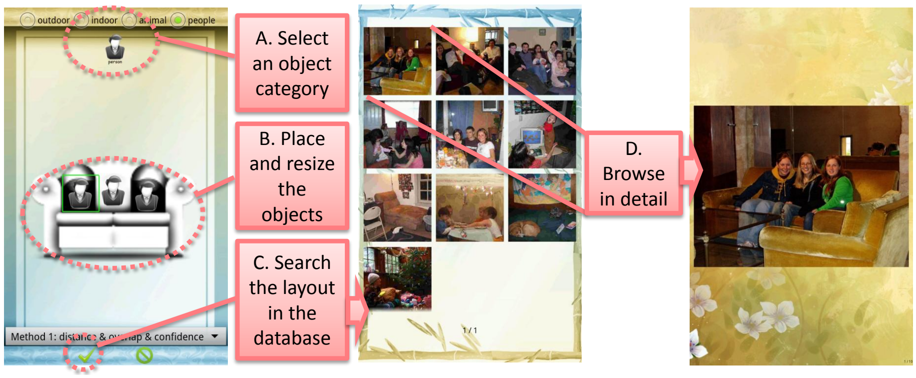

Fig. 1. Interface of the search client. The user first draws the object layout as the query, then search and browse the matched images.

As the prevalence of tablet mobile devices, the specialized search interface on such devices will be more convenient for personal purpose album management and image browsing. In this demo, we develop the search client based on Android devices.

The interface of the proposed search client is shown in Figure 1. To construct a query, users need place the key objects on the canvas, move them to the desired location, and zoom them to the desired size. Then the system can automatically return the specified photos.

3 Database and Indexing

To index the images with object layout information, an automatic object localization algorithm is applied to all images in the photo database. In our current system, we use the 20 kinds of objects introduced by the Pascal VOC Challenge [2], and use one state-of-the-art object detector [1] trained on the VOC dataset.

The object localization algorithm estimates the object positions, scales and confidences of existence in each image. Then this information is interpreted into image indices and saved to the database for the object-layout-aware image retrieval.

In detail, the following information will be stored in the database table for each image:

- The number of detected objects for each class
- The confidence of each detected object instance, which is normalized to [0,1]
- The detected rectangle bounding box of each detected object instance

And note that the bounding box coordinate is represented by the ratio of the coordinates to the image size in order to normalize all images. For example, let *width* and *height* be the width and height of the original image and (x_1, y_1) be the top-left coordinate of detected bounding box (all above are measured in pixels), then the stored bounding box top-left coordinate will be $(\frac{x_1}{width}, \frac{y_1}{height})$.

The computation of the object localization algorithm consumes much computer resource. Thus this database construction step should be done off-line on the server side.

4 Image Retrieval

As aforementioned, each query consists of the names, positions and the sizes of the expected objects. The search target of the system is to find images with similar object layout.

Providing a query, the system first retrieves all the images with similar kind and amount of objects. Then the system calculates the distance and overlap between the matched objects in the query and the retrieved images to score the images. For each matched object i, $distance(i)$ is the distance between the object center point in the query and the retrieved images. And $overlap(i)$ is the intersection area of the object in the query and retrieved images. With such elements, the score used for ordering the result photos can be calculated as Equation 1:

$$score = \sum_{i=1}^{N} (A \times overlap(i) + B \times (1 - distance(i))) \times confidence(i), \quad (1)$$

while A and B are constant coefficients to control the weight of overlap and distance. In this demo, we set A = B = 1.

The retrieval result is sorted with descending order of the scores and returned to the search client. Some example retrieval results are shown in Figure 2.

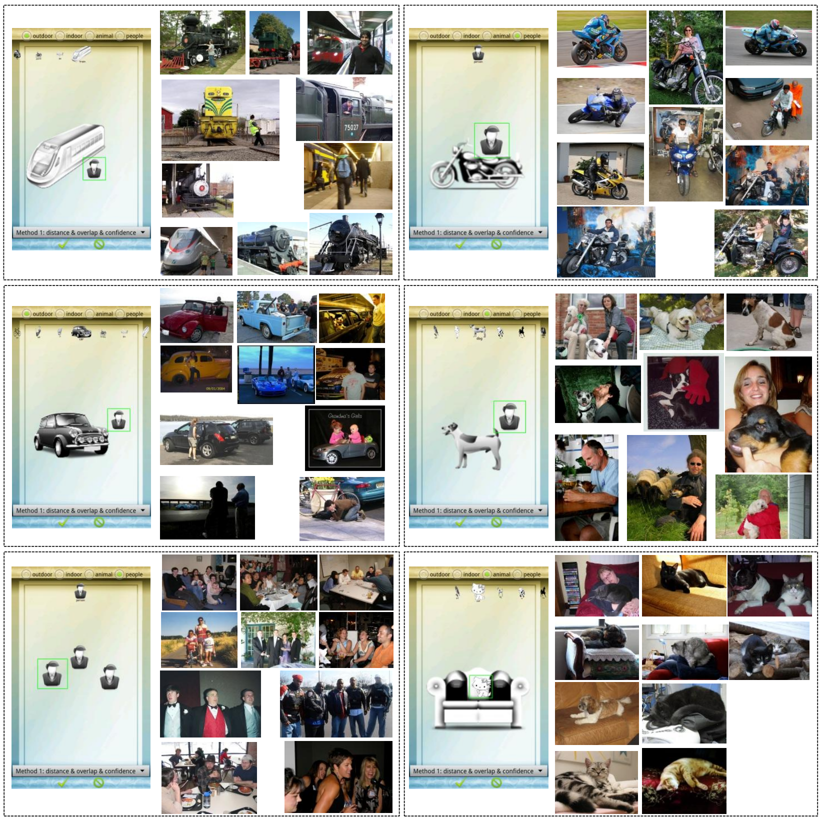

Fig. 2. Example Results. The experiment is performed on the VOC [2] image database. For each result, the left image is the user query and the right are top retrieved images.

5 Future work

In the future, we plan to introduce different significance of objects. For example, if we search for a bicycle and a person, the bicycle may have a higher weight, while the person may have the lower one, because persons are found in most photos and might not be the key object. Thus we plan to use machine learning to get a series of weight parameters indicating the significant of the objects, which can lead to a better searching result according with human perception.

References

1. Zhu, L., Chen, Y., Torralba, A., Freeman, W., Yuille, A.: Latent Hierarchical Structural Learning for Object Detection. In: CVPR. (2011)
2. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal Visual Object Classes (VOC) Challenge. IJCV **88** (2010) 303–338