ROBUST SEMANTIC SKETCH BASED SPECIFIC IMAGE RETRIEVAL

Cailiang Liu[1] *Dong Wang*[2] *Xiaobing Liu*[1] *Changhu Wang*[3] *Lei Zhang*[3] *Bo Zhang*[1]

[1]State Key Lab of Intelligent Tech & Sys, Tsinghua National TNList Lab, Dept. Computer Sci & Tech, Tsinghua University, Beijing 100084 China {cl-liu07,liuxb02}@mails.tsinghua.edu.cn dcszb@mail.tsinghua.edu.cn
[2]Hulu LLC, 14/F, Building A, ChuangXin Plaza, Tsinghua Science Park, Beijing 100084 China wangdong@hulu.com
[3]Web Search & Mining Group, Microsoft Research Asia, Beijing 100190 China {chw, leizhang}@microsoft.com

ABSTRACT

Specific images refer to images one has a certain episodic memory about, e.g. a picture one has ever seen before. Specific image retrieval is a frequent daily information need and the episodic memory is the key to find a specific image. In this paper, we propose a novel semantic sketch-based interface to incorporate the episodic memory for specific image retrieval. The interface allows a user to specify the semantic category and rough area/color of the objects in his memory. To bridge the semantic gap between the query sketch and database images, in the back end, a sampling method selects exemplars from a reference dataset which contains many object instances with user-provided tags and bounding boxes. After that, an exemplar matching algorithm ranks images to retrieve the target image to match the user's memory. In practice, we have observed that query sketches are usually error prone. That is, the position or the color of an object may not be accurate. Meanwhile, the annotations in the reference dataset are also noisy. Thus, the search algorithm has to handle two kinds of errors: 1) reference dataset label noise; 2) user sketch error such as position or scale. For the former, we propose a robust sampling method. For the latter, we derive an efficient spatial reranking algorithm to tolerate inaccurate user sketches. Detailed experimental results on the LabelMe dataset show that the proposed approach is robust to both kinds of errors.

Keywords— specific image retrieval, query by semantic sketch, robustness, episodic memory

1. INTRODUCTION

Specific images refer to images one has a certain episodic memory [1] about, e.g. photos taken by oneself or news pictures one has ever seen before. Typically, the episodic memory [1] is not exactly appearance-based; rather, it is a meaningful account of an episode, documenting the relative positions and sizes of a few keypersons/objects and background scenes. Specific image retrieval (SIR) is a frequent information need, e.g. personal photo management or news image retrieval. To the best of our knowledge, most existing image search algorithms are either keyword-based or example-based, aiming at solving the general image retrieval (GIR) problem. However, these algorithms cannot be readily applied to SIR as the example-based retrieval method requires an unavailable target image and the keyword-based retrieval is ineffective when a user cannot express his query using only a few keywords.

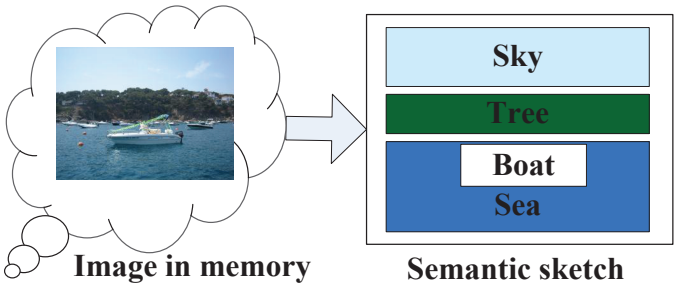

Fig. 1. Semantic sketch for one specific image to be searched.

Compared with GIR, specific image retrieval (SIR) is more challenging because: 1) usually only one image instead of a set of images can satisfy the user's information need; 2) specific object instances in the target image are more difficult to describe than general objects. Thus, as a kind of prior knowledge, the episodic memory is the key to SIR. An intuitive and natural user interface is to allow a user to draw sketches to express his query need. In a query panel, the user can simply draw a few bounding boxes to indicate the location/size of several objects, possibly with certain background scenes (street, kitchen, etc), and further type in some keywords as the names of the objects in an image. Object color can also be easily added if the user wants. Allowing users to input keywords effectively prevents the system from the semantic gap problem. We call such a sketch as semantic sketch to differentiate from the traditional query sketches without semantic keywords. See Fig. 1 for an illustration. We can see that the semantic sketch is an intuitive and easy to use interface to incorporate the user's episodic memory.

Query by sketch (QBS) [2] can be traced back to the early days of CBIR. Query sketches typically comprise blobs of color [3] or edge [4]. Appearance based features like shape descriptors [2] and user feedback [5] are also considered. As such

Cailiang Liu performed this work while being a research intern at Microsoft Research Asia. The first, third and last authors are partially supported by the National Natural Science Foundation of China under Grant Nos.90820305 and the National Basic Research Program (973 Program) of China under Grant Nos.2007CB311003.

bottom-up approaches force users to adapt to the system by inputting low-level features directly. However, users may lack the required talent or patience to find an exact color or draw a detailed shape sketch. In contrast, the top-down query by semantic sketch gives users more freedom in formulating their queries with semantic categories and their respective spatial layout.

Empirically, we have observed that user sketches are rarely accurate account of the episodic memory. Actually, they are error prone in nature. For example, the position/size of a specific object instance is usually inaccurate. If a user provides the object color, it is very likely that the chosen color differs from the actual object color. Thus it still remains challenging how to combat with these kinds of errors efficiently. Besides, annotations (tags and bounding boxes) in the reference dataset are also quite noisy and this labeling noise significantly deteriorates the retrieval performance.

In this paper, we propose a novel semantic sketch based search method that allows user to specify object semantic and extent to construct a query. The proposed method first samples multiple object exemplars from an annotated (with object bounding boxes) reference dataset (e.g. the LabelMe dataset) to represent one missing target object in a query sketch. After that, by using a deliberately designed low level matching algorithm, it efficiently and effectively ranks the target images in less than top 50^{th} resulting images. More concretely, we tackle the following two kinds of inaccuracy: 1) reference dataset label noise; 2) user sketch errors of position and size. We propose a robust sampling method to deal with reference dataset label noise. It selects multiple object exemplars from the reference dataset to represent one missing sketched target object. Images are then matched against possible object exemplar combinations with a parallel local matching algorithm. Meanwhile, we derive an efficient spatial reranking algorithm to offset inaccuracy from user sketches. The spatial reranking algorithm selects the best response in a set of local bounding boxes by an efficient branch-and-bound procedure on multiple object exemplars simultaneously. Finally, reranked object retrieval results are combined to produce the final ranking.

Extensive experiments on the LabelMe [6] dataset show the robustness and efficiency of the proposed formulation. Note that SIR is quite challenging, as validated by the results of text-based and example image-based approaches, the results we have obtained are promising and show great potential of sketch-based image retrieval.

2. QUERY BY SEMANTIC SKETCH FOR SIR

The overall search process consists of the following steps as shown in Fig. 2. First, the user is allowed to depict the query sketch with specific objects. Given the query sketch, we sample multiple object exemplars from an annotated reference dataset $\mathcal{D}_R$ to represent the specific objects which user is imagining in his mind. Ideally, in search of the best possible match, each image in the target dataset $\mathcal{D}_T$ should be matched to the localized image features derived from all possible object exemplar combinations. However, it would be too computational to implement

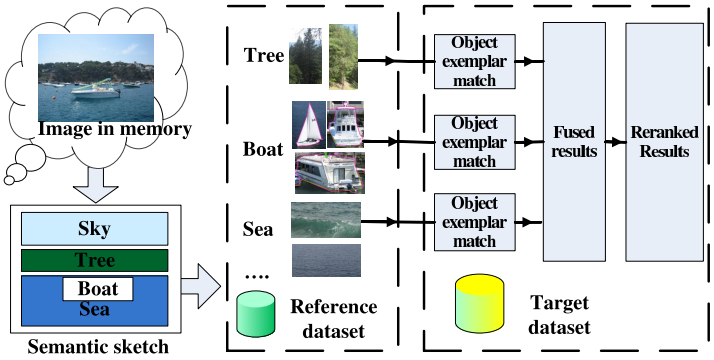

Fig. 2. Search process overview using the semantic sketch query.

using this ideal approach. Therefore, for efficiency, the matching process is approximately designed and is decomposed into several independent object exemplar matching processes. Then, for each object exemplar, reranking is performed to combat possible user sketch errors. Finally, the ranking lists for different object exemplars are fused together to get the final retrieval results.

Semantic Sketch The sketch contains multiple specific objects $\{o_j\}$ which may overlap with each other, e.g. one person riding a bicycle.[1] During sketching, users are asked to indicate which object occludes other ones in their drawings (cf. Fig 1). Let us denote an object by $o = \{c, B\}$, where c is the object category and $B = \{x, y, w, h\}$ is a bounding box which specifies its center (x, y), width w and height h. We use relative position here with respect to the whole image size. Sketch is often spatially under-specified since only a few salient objects are depicted. Additional object attributes like color composition, orientation or shape can also be determined by either simple selection or by choosing exemplars in an interactive manner.

Object Representation To enforce spatial constraints, we adopt the local feature based object representation. More concretely, the color sift (CSIFT) descriptor is chosen for its good performance in recent object classification benchmarks [7]. Give an image I, a set of CSIFT points are extracted. The bag of keypoints model converts the variable number of local feature points to a fixed length histogram vector H_I. The CSIFT codebook contains $|V|$ "visual words" which are obtained by K-means clustering and serves as the centers for the histogram bins. We denote H_B for a localized CSIFT histogram with bound box B.

Reference Dataset $\mathcal{D}_R$ contains various object exemplars. Each exemplar e_c^k has the spatial boundary of $B = \{x, y, w, h\}$ and its semantic category. If the original localized annotations are not in the bound box format, conversion is performed.

Exemplar Sampling Given one specific object o, a few representative object exemplars $\{e^i\}_{i=1}^m$ are sampled from its category c_o in $\mathcal{D}_R$. In automatic setting, we adopt a principled GRASSHOPPER algorithm as detailed in Sec. 3 to select representative exemplars and avoid noisy ones. In an interactive

[1] Currently, we focus on objects and leave the scene to future work.

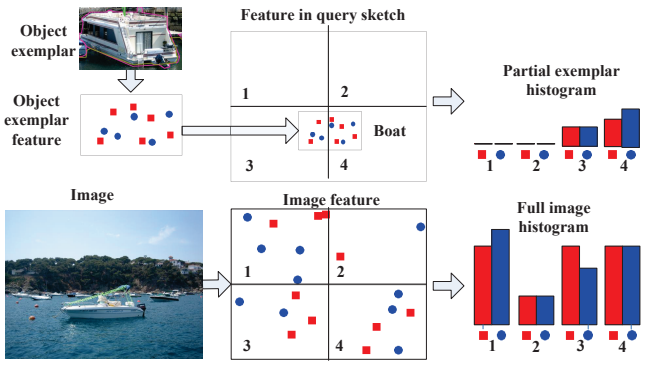

Fig. 3. Partial match for one specific object exemplar.

retrieval setting, user may further manually select a few exemplars for each object from the sampled exemplars.

Exemplar Matching Given a set of objects $S = \{o_j\}$, a matching score $f(I, \{o_j\})$ needs to be assigned to image $I \in \mathcal{D}_T$. We now represent each object o_j with a set of exemplars $E_j = \{e_j^k\}$ from the same category because o_j is not available. Thus, $s(I, \{o_j\})$ should reflect possible similarity between image I and exemplar combinations of different objects. The maximum matching score over all exemplar combinations is a good candidate for s. Although the average matching score is another alternative, it requires exponential computation time. By carefully choosing the histogram intersection (*HI*) matching function where the bounded property of $s(H_I, \sum_j H_j) \leq \sum_j s(H_I, H_j)$ holds, we approximate the maximum matching score by taking the maximum for each object independently. Fig. 3 illustrates the process of generating exemplar and image histograms. We first warp the exemplar keypoints into the object boundary given by user's sketch, then divide the image into a $r \times c$ grid, and finally generate sift histogram for each cell. By using *HI* matching function, it ensures that only keypoints in the object boundary are matched.

With this, it is necessary to compute local histograms on all images from $\mathcal{D}_T$ at query time since object o_j can occupy anywhere in the query sketch. However, the *sparsity* of the histogram feature could provide further speedup. By partitioning images in $\mathcal{D}_T$ into fine grids (4×4 grids perform best) and computing local histogram for each grid cell as inverted lists on disk, we further approximate and accelerate the local match process from sublinear to $|\mathcal{D}_T|$. The reasoning is that the inverted lists are sorted and many images are never accessed.

Therefore, we can approximate the maximum score over all possible object exemplar combinations by calculating the HI matching process for each exemplar independently from sublinear to $|\mathcal{D}_T|$. To reduce the possible influence of noisy exemplars, practically we can take average of matching scores of top M ($M = 10$ in this study) exemplars for image $I \in \mathcal{D}_T$.

Fusion Given one image, the local matching scores for different objects in the query sketch are combined together to produce the final ranking with a simple linear score fusion. The weights for different objects are simply set to their respective area occupied. Similarly, the reranked scores are also fused in the same way.

Exemplar Reranking As [8] had found, sketches drawn by users are often imprecise. Thus real target object located nearby may be lost if searching only within the box provided by the user. Although it would be ideal if we can search in all possible images for all object exemplars, for efficiency considerations we only rerank top 2000 images from the fusion step. Reranking is done in a per object style like in the matching step.

3. ROBUST EXEMPLAR SAMPLING

The proposed algorithm is motivated by the random walk on graph (RWoG) model. Given a normalized adjacency graph $G = [g_{ij}]_{i,j=1}^m$ which is a valid transition probability matrix, or equivalently a Markov chain model with m states, it can be considered as the similarity matrix for exemplars. An input initial probability distribution r can be assigned as the similarity to exemplar and object o. A parameter α of restart probability controls the balance between these two kinds of similarities. As stated by Zhou et al. [9], the stable distribution x^* for the RWoG model minimizes the following cost function with respect to x

$$C(x) = \alpha \sum_{i,j=1}^{m} g_{ij}(\frac{x_i}{g_{ii}} - \frac{x_j}{g_{jj}})^2 + (1-\alpha)\sum_{i=1}^{m}(x_i - r_i)^2.$$

It is clear that x^* balances the similarity between exemplars and that between exemplar and object o. We have $x^* = (1-\alpha)(I-\alpha G)^{-1}r$ as a closed form solution. An efficient iterative solution is given by $x_{n+1} = \alpha P x_n + (1-\alpha)r$, which converges exponentially. Although x^* provides a good start point for sampling exemplars. it is still influenced by strongly concentrated groups of exemplars in G as exemplars at group centers have higher probabilities, and tighter groups have overall higher probabilities in x^*.

To further correct the group concentration effect and to generate more diverse exemplars, we introduce an absorbing RWoG algorithm [10] for sampling. This algorithm reduces the probability of choosing nearby exemplars after sampling exemplar e^i to allow for diversity. Nicked as GRASSHOPPER, it iteratively selects the exemplar with the largest *expected number of visits* to other states in the Markov chain defined by G, and sets the corresponding state to absorbing state subsequently. Although the computation of *expected number of visits* requires matrix inversion each time, it can be done iteratively with the Sherman-Morrison-Woodbury matrix inversion lemma (cf. [10] for details). Then we only need to invert the matrix once in the first iteration which brings a significant speedup.

The edge weights of the adjacency graph are computed by $g_{ij} = \exp(-\frac{dist_{ij}^2}{2\sigma^2})$, where $dist$ is the Euclid distance on 64-dim color histogram in HSV space and σ is set as the average distance among exemplar pairs. We set $\alpha = 0.25$ in this work.

4. ROBUST EXEMPLAR MATCHING BY RERANK

With spatial constraint in mind, we could run a localized spatial pyramid matching (SPM) on B between both the query and each image in the returned top N list. However, SPM simply ignores the keypoints outside the given box and might introduces additional errors. A better alternative is the recent proposed Efficient Subimage Retrieval (ESR) [11] algorithm. ESR works with a quality bound for a set of box regions and a branch-and-bound procedure to efficiently prune low quality box sets. We adapt ESR to our scenario to find a box with maximum score within a certain set $\mathcal{B}$ of boxes to tolerate position errors, e.g. size change $\pm\theta\%$. The image level branch-and-bound is removed since we aim at finding a box for each image in the short list. Instead, we extend the branch-and-bound heuristic to support multiple query sliding search. In this step, we use the normalized histogram intersection (NHI) as the sparse quality function.

More concretely, for the bounding step, we construct a quality bounding function $\hat{f}$ for a set $\mathcal{B} = \{B\}$ of possible boxes B that fulfills the conditions

$$\hat{f}(\mathcal{B}) \leq \max_{B \in \mathcal{B}} f(B), \text{ and } \hat{f}(\{B\}) = f(B)$$

where f is the quality function. The quality bound for NHI is $\hat{f}(\mathcal{B}) = \sum_l \min(\frac{H_{ql}}{||H_q||}, \frac{\overline{H_{\mathcal{B}l}}}{||\overline{H_{\mathcal{B}}}||})$, where $\overline{H_{\mathcal{B}}}$ is the histogram of the union of all boxes in $\mathcal{B}$, $\underline{H_{\mathcal{B}}}$ is the histogram of the intersect of all boxes in $\mathcal{B}$, and H_q is the histogram of an exemplar.

Different from the original ESR formulation, our objective is to find the best matched exemplar and best matched position, since we have multiple query exemplars for each object. To further reduce the time of reranking, we modify the branch-and-bound process to handle multiple exemplars, i.e. all the exemplars are put in the same priority queue. The upper bound remains the same when taking into account multiple object exemplars: $\hat{f}(\mathcal{B}_i) = \sum_l \min(\frac{H_{q_i l}}{||H_{q_i}||}, \frac{\overline{H_{\mathcal{B}_i l}}}{||\underline{H_{\mathcal{B}_i}}||})$. By sharing the priority queue, computations of low-match exemplars are saved. Still, multiple objects can be searched in parallel.

5. EXPERIMENTS

5.1. Experimental setup

We evaluate our algorithm on the LabelMe dataset since it contains a lot of personal photos which are specific in nature. Sequential video frames are removed to prevent the bias of near-duplicate images. We randomly split the left 43391 non-sequential images in two halves denoted as $\mathcal{D}_R$ and $\mathcal{D}_T$. The original polygon annotations provided by labelers are converted to a maximal bounding box. 200 query sketches are automatically generated in $\mathcal{D}_T$ by using the full set of converted object bounding boxes with corresponding semantic category directly. This provides a large query pool beyond human hand-generated query sketches and may allow us to better understand how different factors affect the search result. Besides, we manually select 14 diverse images from the 200 queries and draw sketches by hand for comparison.

Table 1. DCG for different retrieval approaches.

	QBK	QBE	semantic sketch
DCG	0.011	0.011	0.178

Table 2. DCG for different sampling methods and sample size.

DCG	Random	Heuristic	Grasshopper
20 samples	0.121	0.126	0.125
50 samples	0.117	0.120	0.128

We adopt the Discounted Cumulative Gain (DCG) like metric[2] for a query pool as the performance measurement to highly reward early returned results. $DCG = \frac{1}{|Q|}\sum_{q=1}^{|Q|} \frac{1}{\log_2(1+r_q)}$, where $|Q|$ is the number of the queries and r_q the rank of the only target image of query q in the search results.

We set the CSIFT code book size $V = 512$ and partition the image to 4×4 grids to obtain the localized histograms for CSIFT points. 20% size/position error in user sketch is allowed by setting $\theta = 10$. $N = 2000$ image results are returned for each sketch query.

5.2. Comparison across methods

We compare the proposed approach with representative GIR algorithms like QBK and QBE for SIR first. QBK is performed by running a tf-idf based search with the object/scene name in the query sketch as keywords. And QBE is performed by searching top images from QBK results. Searching with the relevant image is unfair but we lack alternative image examples. Thus a round-about is designed which reranks the QBK results by sorting images with respect to their distances to a feature point. This feature point is chosen as the mean of features for top 100 images in the QBK result. For query by semantic sketch, GRASSHOPPER with location rerank is performed (Detailed parameter are explained later on). Results in Table 1 illustrate the significant superiority of semantic sketch over both QBK and QBE.

Someone may advocate for a tag based retrieval and visual rerank approach. However, this is not always possible for SIR problem since the user-provided labels are too sparse and important objects may lack the necessary label for tag based retrieval. Currently only a small number of tags co-occurs often with another small number of tags. For example, there are many image containing both building and person, but most of them are annotated only by building. This can also explain why QBK and subsequently QBE performs inferior.

[2]It is a little different from standard DCG, since we only care about the specific image ranking. It is actually equivalent to DCG@ $\inf$.

Table 3. DCG for automatic/manual sampling with respective reranked results.

DCG	Random	Grasshopper	Manual	Color
Original	0.118	0.128	0.130	0.138
Reranked	0.133	0.178	0.193	0.192

5.3. Tolerance to label/sketch error

We have observed that significant label noise exists in the LabelMe dataset. For example, a bird may often be annotated as "bird mid front forest". When forest is searched for, we may falsely select bird exemplars for ranking if the sampling method is not robust.

Here we compare three sampling methods for their robustness to label noise: 1) a Random method which simply selects $|E_j|$ exemplars at random; 2) a Heuristic method which selects the top ranked results in the random walk; 3) the Grasshopper method introduced in Sec. 3. The exemplar size can also impact the ranking results and thus is evaluated at the same time. The results are shown in Table 2. Though Grasshopper performs only 9% better than Random, as shown in Table 3, the reranked results are significantly better. One possible reason for this is that Grasshopper provides more diverse exemplars and are better at tolerating object location errors. Grasshopper can select more diverse exemplars than Heuristic as the exemplar number increases from 20 to 50. This is reasonable since Heuristic is the input vector to run Grasshopper for diversity. Table 2 also shows that Grasshopper with 50 exemplars is slightly better than that with 20. Thus 50 exemplars are used subsequently.

It is also interesting to see how Grasshopper behaves compared with manual methods. To this end, we manually assign 10 exemplars for each object (Manual) for comparison. We also cluster the exemplars based on 64-dim color histogram in HSV space and then manually select 10 representative exemplars in the produced clusters (Color). It is excessive to manually assigning 10 exemplars for each object since it is too tedious for common users. The results in Table 3 show that the fully automatic Grasshopper method is on par with the Manual method and additional color cue with extra user interaction does not improve the result significantly.

The results for reranked results of both automatic and manual sampling methods are also summarized in Table 3. It is evident that the reranked result is significantly better than the original one. For example, after reranked, DCG for Grasshopper and Manual increase 40% and 48% respectively. Thus location based reranking is necessary for combating the user sketch errors.

Hand vs. Automatic Sketch So far we are testing on 14 hand generated semantic sketches only. It is desirable to see how the proposed framework works when using all large query pool. To this end, we compare the retrieval results of 14 hand generated semantic sketches (Hand 14) with 200 additional automatically generated semantic sketches (Automatic 200) in Table 4. The reranked performance of Automatic 200 significantly outperforms that of Hand 14. One possible reason may be that the maximal boxes generated from the user provided polygon annotations lead to the oversize of automatic bounding boxes, which results in its priority over Hand 14. Therefore, reranking is helpful even with automatically generated query.

Table 4. DCG for hand and automatic sketch queries.

DCG	Hand 14	Automatic 200
Original	0.128	0.147
Reranked	0.178	0.204

Result Visualization We also show the reranked object retrieval result for two representative hand generated query sketches in Fig. 4. The target image may occur in the first page. But if there exists similar images to the target image in the target dataset, as shown in the second query in Fig. 4, the results are still unsatisfactory. This is surely one situation the current exemplar based search approach can not handle.

6. CONCLUSIONS

This paper identifies the specific image retrieval problem and proposes an initial attempt towards solving it. The key finding is that one's episodic memory is very helpful in retrieving the target specific image, at least for a medium sized dataset. Secondly, it is viable to represent specific objects by general exemplars under strong spatial constraints. With the semantic query sketch interface, the sampling algorithm effectively generates concrete object exemplars from a well annotated reference dataset. Thirdly, the proposed formulation is robust to both annotation and sketch errors. It can run efficiently and is also easy to implement. Generally, we can rank the target image to the 50^{th} position currently, a usable result for a medium-sized dataset such as personal photo corpus.

Possible extensions to the current study can be divided into two parts: for the user interface, we will incorporate more semantics by incorporating visual query suggestion and adding local shape support with an interactive interface; for the retrieval system, we will increase the scale of the dataset, explore larger and more realistic datasets like Flickr, and incorporate different kinds of models for specific objects, e.g. faces or city landmarks.

7. REFERENCES

[1] E. Tulving, *Elements of episodic memory*, Oxford University Press New York, 1983.

[2] A. del Bimbo and P. Pala, "Visual image retrieval by elastic matching of user sketches," *PAMI*, vol. 19, no. 2, pp. 121–132, February 1997.

[3] Z.N. Li, O.R. Zaiane, and Z. Tauber, "Illumination invariance and object model in content-based image and video

Fig. 4. Search result visualization. For each query, the 1st and 2nd rows are different exemplars (marked in thin yellow box) of different objects sampled in the reference dataset, and the 3rd and 4th rows are search results before/after reranking where green boxes denote the target image and yellow ones denote similar yet false matches.

retrieval," *JVCIR*, vol. 10, no. 3, pp. 219–244, September 1999.

[4] R.K. Rajendran and S.F. Chang, "Image retrieval with sketches and compositions," in *ICME00*, 2000.

[5] Eugenio Di Sciascio, G. Mingolla, and Marina Mongiello, "Content-based image retrieval over the web using query by sketch and relevance feedback," in *VISUAL '99*, London, UK, 1999, pp. 123–130, Springer-Verlag.

[6] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," Tech. Rep., Tech. Rep. MIT-CSAIL-TR-2005-056, Massachusetts Institute of Technology, 2005.

[7] G. J. Burghouts and J. M. Geusebroek, "Performance evaluation of local colour invariants," *Computer Vision and Image Understanding*, vol. 113, pp. 48–62, 2009.

[8] J P Collomosse, G McNeill, and L Watts, "Free-hand sketch grouping for video retrieval," in *Intl. Conf on Pattern Recognition (ICPR)*, 2008.

[9] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf, "Learning with local and global consistency," in *NIPS(15)*, 2003, pp. 237–244.

[10] Xiaojin Zhu, Andrew B. Goldberg, Jurgen Van Gael, and David Andrzejewski, "Improving diversity in ranking using absorbing random walks," in *HLT-NAACL*, 2007, pp. 97–104.

[11] Christoph H. Lampert, "Detecting objects in large image collections and videos by efficient subimage retrieval," .