This ICCV2013 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

# 3D Sub-Query Expansion for Improving Sketch-based Multi-View Image Retrieval

Yen-Liang Lin, Cheng-Yu Huang, Hao-Jeng Wang, Winston Hsu
National Taiwan University, Taipei, Taiwan
(yenliang,lairoeaes,enjoylife1021)@cmlab.csie.ntu.edu.tw,whsu@ntu.edu.tw

## Abstract

*We propose a 3D sub-query expansion approach for boosting sketch-based multi-view image retrieval. The core idea of our method is to automatically convert two (guided) 2D sketches into an approximated 3D sketch model, and then generate multi-view sketches as expanded sub-queries to improve the retrieval performance. To learn the weights among synthesized views (sub-queries), we present a new multi-query feature to model the similarity between sub-queries and dataset images, and formulate it into a convex optimization problem. Our approach shows superior performance compared with the state-of-the-art approach on a public multi-view image dataset. Moreover, we also conduct sensitivity tests to analyze the parameters of our approach based on the gathered user sketches.*

## 1. Introduction

With the proliferation of internet images and video collections, content-based image retrieval (CBIR) approaches have been investigated to analyze and manage such exponentially growing media collections. However, example images may not be always at hand while searching, which motivates sketch-based image retrieval (SBIR) research that uses simpler hand-drawn sketches as query images.

A large portion of SBIR approaches mainly employ (grid-based) global descriptors for cross-domain image matching (sketch vs. target) [13, 24, 7, 2, 6], and thus inherit the drawbacks that not being invariant to affine transformations. To offer solutions to partial affine invariance, some recent research [14, 16] explore local descriptors within a bag-of-visual-word (BoVW) model. Others build upon these frameworks and tailor these features for their own applications (e.g., photo montage [9], sketch recognition [12]). However, these methods still suffer from the multi-view problem and can only retrieve images with similar viewing angles or partial affine transformations with the query sketch (Figure 1).

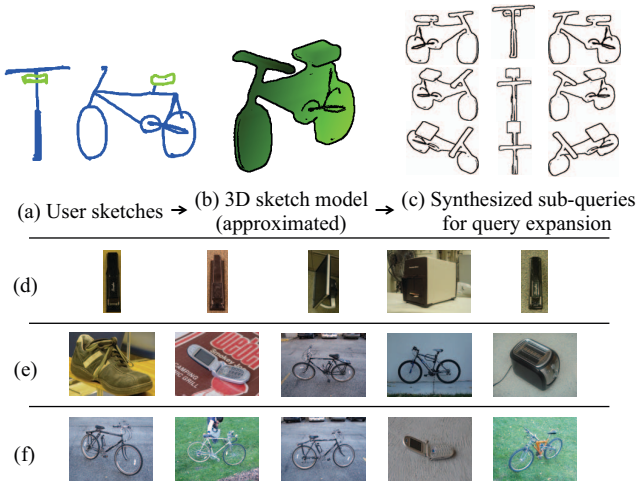



Figure 1. System overview. Given two (2D) user sketches (different colors indicate different parts), our system can automatically reconstruct an approximated 3D sketch model (b) and generate a set of synthesized views (9 views for illustration) as expanded sub-queries. Our system will use these synthesized sketches for query expansion to retrieve multi-view images. (d)(e) show the top five researching results using the state-of-the-art method [14] with the (2D) frontal and side view as the input sketch respectively. (f) shows the retrieval results with the proposed method, which obtains more accurate and diversified images.

Sketch is a promising and intuitive manner to express user intension for the target images they intend to retrieve. However, current sketch-based methods only retrieve images with similar 2D contours (or edges), it is still very challenging to retrieve images under large pose transformations (e.g., rotation, scale and translation), which is very common in those online shopping websites (e.g., Amazon) or image collections (e.g., Flickr).

The goal of this paper is to retrieve all relevant images under large viewing angle variations. Motivated by query expansion technique used in image retrieval [10], we bring the concept of query expansion into sketch-based image retrieval. We expand the original input sketches by a set

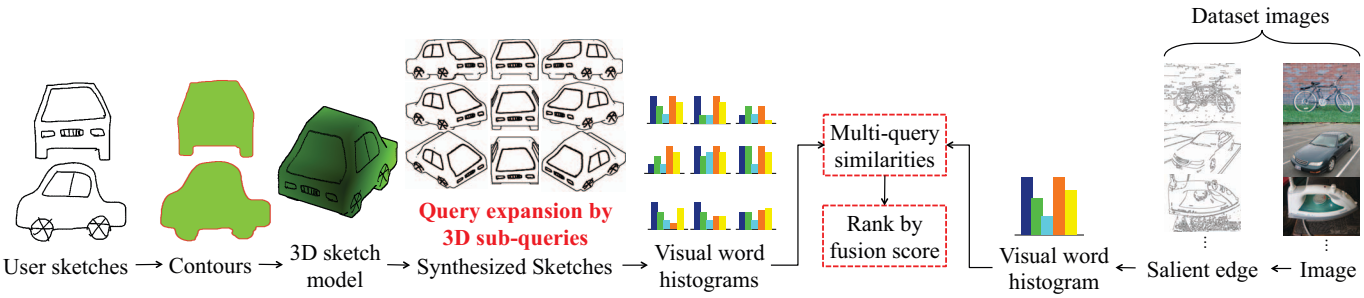



Figure 2. A schematic overview of the proposed system. Using the user-provided two (nearly) orthogonal sketches, the system reconstructs a 3D sketch model from the corresponding sketch contours and obtains a set of synthesized sketches for query expansion. Synthesized sketches and edge maps are represented by histogram of visual word frequency. A multi-query feature vector of each dataset image is created from the similarities between sub-queries and the database image. All dataset images are then ranked based on the final fusion score.

of synthesized multi-view sketches (as the expanded sub-queries) from the reconstructed 3D sketch model to boost the retrieval performance. The idea is intuitive: as a user query with keyword "Starbucks," query expansions methods might transparently enhance the search results with more semantically related keyword such as "coffee," "Seattle," etc., by variant methods [3, 22]. So is that for the proposed method.

To our best knowledge, this paper is the first approach that addresses the multi-view sketch-based image retrieval problem. Figure 1 illustrates the overview of our retrieval system. Figure 1 (a) shows an example (bicycle) of two sketches from the designated front and side views. Once two (nearly orthogonal) sketches have been specified, its 3D sketch model is automatically constructed (Figure 1 (b)). A set of sub-queries can be synthesized and expanded to match the possible multi-view candidate images in the data collection (Figure 1 (c)). Due to the space limitation, we only show 9 rendered views for illustration. For comparison, Figure 1(d)(e) show the top five searching results with the state-of-the-art method [14] using front-view and side-view as the input sketch respectively. The retrieval results with the proposed method are shown in Figure 1(f). It can be seen that the retrieved images from the proposed method are more accurate and presented in different orientations.

Our main contributions include:

- We propose a 3D-enhanced sketch-based system that generates multi-view sketches as expanded sub-queries to boost multi-view image retrieval performance. To our best knowledge, this is the first work that brings query expansion into sketch-based image retrieval.
- To learn the weights of synthesized sub-queries, a new multi-query feature representation is proposed to model the similarity between expanded query sketches and dataset images, and formulate it into a convex optimization problem.
- We compare our approach with the state-of-the-art method [14]. The experimental results show that our system achieves superior performances in terms of average precision on a public multi-view image dataset [23].

## 2. Related Work

### 2.1. Sketch-based Image Retrieval

While there exists a considerable amount of work on sketch-based image retrieval, most of the previous research mainly employ (grid-based) global descriptors for bridging the gap between cross-domain image matching. M. Eitz et al. [13] evaluate several state-of-the-art global feature descriptors (i.e., ARP [7], EHD [20], HoG [11] and structure tensor [19]) for SBIR. The method divides the image into a regular grid and computes descriptors from each cell, which are then concatenated into a global image descriptor. A. Shrivastava et al. [24] propose to learn data-driven uniqueness of each query image based on spatially-rigid HoG feature from a large set of negative samples. A scalable approach, MindFinder system [6], is the first to propose an efficient indexing structure for large-scale sketch retrieval, they build inverted index-like structure to speed up the sketch-based image search. However, these approaches (global based) are not invariant to affine transformations.

To offer the solutions to partial affine invariance, some recent research [14, 16] explore local descriptors within a BoVW framework. M. Eitz et al. [14] demonstrated the proposed SHoG descriptor outperforms the other global descriptors (e.g., ARP [7], EHD [20]) and local descriptors (e.g., shape contexts [2] and HoG [11]) in their experiments. However, these methods still suffer from the multi-view problem (i.e., they still restrict to images with similar views).

Different from prior methods that explore descriptors from pure 2D images. In this paper, we propose a 3D-enhanced approach that automatically reconstructs a 3D sketch model from (2D) user sketches, and expand the query sketches by a set of synthesized sketch (sub-queries) to boost the retrieval performance.

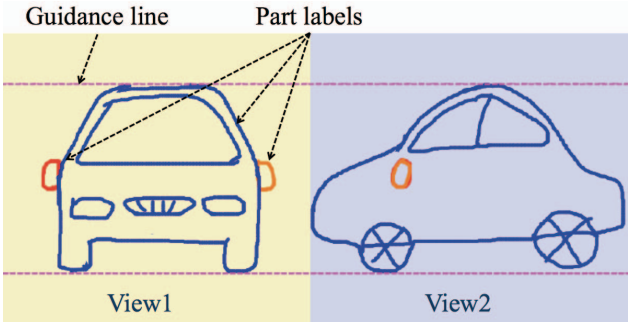



Figure 3. User interface of our system. A user can draw two orthogonal views [2] of target sketches at the corresponding panel. To draw different parts, a user can simply switch the brush color and system will automatically show a guidance line to help the user to align those parts from two views.

### 2.2. 3D Model Reconstruction From Line Drawings

Previous methods of 3D reconstruction from line drawings are mainly based on a set of heuristic rules [5], yet those rules are not always satisfied in the imperfect sketch drawings. Also, to ensure interactive response, it is impossible to take minutes to recover the 3D model [27]. Some approaches employed Teddy [17] to convert user sketches into 3D objects. However it hypothesizes the models must have a spherical topology, which is not suitable for arbitrary input sketches. Fortunately, most of the man-made objects or natural objects are (roughly) axis-aligned and the underlying 3D models can be reconstructed by 2D lines from different orthogonal views (e.g., front, side or top views) [21]. Motivated by the method [21], we reconstruct our 3D sketch model using two orthogonal sketches. However, different from their approach for reconstructing a perfect 3D model, we tailor their method to reconstruct an approximated 3D sketch model and simplify the complexity of user sketches.

## 3. Proposed Approach

Figure 2 shows the system flow of the proposed method. Our system consists of two main steps. The first offline step consists of building visual word vocabulary from a set of feature descriptors, which are extracted from random locations on each edge map of dataset images. Each image is then converted to an visual word histogram representation and stored. In the second online step, as a user draws sketches, our system automatically reconstructs the corresponding 3D model from the sketch contours and generates a set of synthesized (expanded) sketches to cover a more dense viewing range. Then each synthesized sketch is similarly encoded by a visual word histogram. The similarity between each synthesized sketch and dataset image is computed and concatenated into a long dimensional vector (multi-query feature vector). Then a fusion function is designed and applied to the multi-query feature vector, all dataset images are ranked with the final fusion score.

### 3.1. 3D Sketch Model Reconstruction

To generate (approximated) 3D model from 2D sketches in the least effort, we propose to derive the results by two (nearly-orthogonal) sketches. In this section, we briefly introduce the 3D reconstruction algorithm proposed in [21] and show how this can be adapted to create our 3D sketch model. In their approach, each 3D model is assembled of parts and each part is specified with two or three silhouettes from front, side, or top views. The core idea of their algorithm is to convert a 3D Constructive Solid Geometry (CSG) problem into more simplified 2D operations, profoundly reducing the computational cost and thus the 3D model can be recovered in real-time after the silhouettes of each part are specified.

Given two input sketches, our algorithm first estimates the object contours by using the algorithm [26], and then create a rough 3D model by using the method described above. The sketch contours are then mapped back from 2D space onto the surface of the 3D model, hidden sketches are removed by testing against a depth map rendered from the reconstructed 3D model.

Different from their method for creating a sophisticated 3D model, a approximated 3D sketch model is sufficient for our system to estimate the 3D positions from input 2D sketches. Thus, some simplifications can be leveraged to mitigate the burden from users, which is considered crucial in sketch-based image retrieval. Since users may not want to spend lots of time to generate a 3D sketch model. In our system, a user can only draw those parts that may be occluded from the others. For example, in bicycle case (Figure 1), two wheels and the body could be regarded as a single part (blue) and reconstructed unitedly, while the saddle (green) might be occluded by the stem at the front view and is reconstructed separately. Although some curved objects (e.g., face, Figure 10) may not be axis-aligned as mentioned in [21] and there may generate some defects in the synthesized sketches. However, from our experiments, the roughly estimated 3D geometry of such kind of curved object still bring benefits to the searching results.

We currently implement our user interface on *iPad* platform for pilot study (Figure 3), a more user-friendly interface will be investigated in our future work.

### 3.2. View Synthesis as Sketch Sub-Queries

A fully viewpoint-independent retrieval system would require densely sampling the entire viewing sphere. To reduce the computational complexity and storage cost of the process of view synthesis, we discretize the viewing sphere into several viewpoints controlled by the parameters $(a, e)$,

[2] Two orthogonal views could be $< front, side >$, $< top, side >$ or other orthogonal views specified by users depending on the characteristic of the target object.

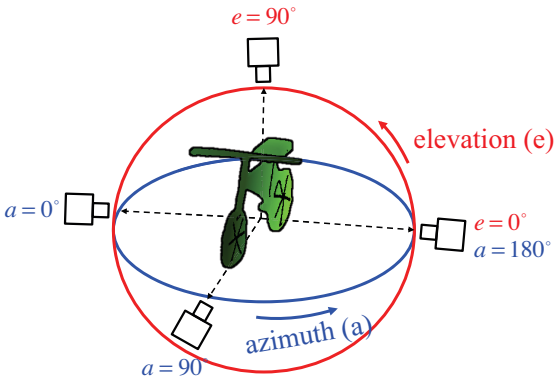



Figure 4. Viewing sphere.

where $a$ and $e$ represents for azimuth and elevation respectively (i.e., the angular coordinates of the camera located on the viewing sphere, Figure 4). The missing viewpoints in the viewing sphere are made up by affine invariant local descriptors. The choices of the discretization step need to consider the degree of invariance of adopted local feature descriptors and underlying viewpoint distribution of image dataset. In the experimental section, we conduct a set of sensitivity tests to optimize the parameters of our system. Figure 10 shows some synthesized object sketches defined in 3D object dataset.

### 3.3. Bag-of-Visual-Word Model

As mentioned in [14], bag-of-visual-word approaches generally outperform the other global descriptors in the literature. From their experiments, SHoG local descriptor shows the better performance than the other descriptors, which applies HoG descriptors on the edge map rather than in the original image space. Note that the HoG used in their paper is based on the localized variant used in the SIFT descriptor that computes the dominant orientations for each window (4x4 spatial bins and 8 bins for gradient orientation).

We learn our visual codebook model from a set of dataset images from a multi-view object dataset [23]. For each dataset image, we find those salient edges that are most likely drawn by a user by applying Canny edge detector [4] and HoG descriptors are extracted at 500 random locations on each Canny edge map. The visual codebook is constructed via hierarchical k-means clustering method. The window size of HoG descriptors and codebook size are set to 50 (in the percent of the minimum of image width and height) and 1000 with the best retrieval results in our image dataset.

In our experiments, we also compare the performance between rotation invariant and rotation variant HoG descriptor. Interestingly, we found that rotation variant HoG shows better retrieval performance in overall. The reason is rotation invariant HoG might brings more ambiguities, since those man-made objects will have similar shapes in some viewpoints. Also, our approach has automatically generated a range of views, which already offer the solutions to in- and out-plane rotations. The role of local descriptors is to bridge the gap for those views that are not fully covered by our synthesized data.

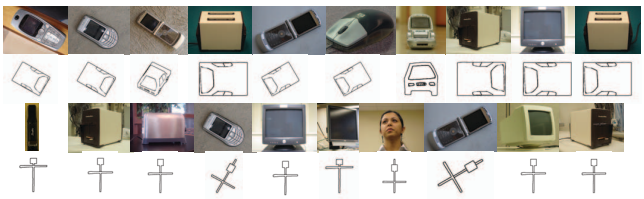


Figure 5. Examples of the ambiguity problem. 1st and 3rd rows are the top 10 retrieved samples using 28 synthesized car and bicycle query sketches respectively with "max" fusion scheme. 2nd and 4th rows are the corresponding best matching query sketches. The results reveal that some views of an category are less discriminative and could be confused with other objects.

A visual word histogram is constructed for each synthesized sub-query sketch and dataset image, and the similarity between each synthesized sub-query sketch and dataset image can be computed according to histogram distance, and finally is concatenated into a long dimensional feature vector.

### 3.4. Fusion Function

Formally, given a set of synthesized sub-query sketches $Q = \{q^{(1)}, q^{(2)}, ..., q^{(m)}\}$ and a set of dataset images $I = \{I_1, I_2, ..., I_n\}$. A multi-query feature vector for each dataset image $x_j = \{s_j^{(1)}, s_j^{(2)}, ..., s_j^{(m)}\}$, $j = 1, 2, ..., n$, is created from each sub-query-image pair, where $s_j^{(i)}$ represent the visual similarity between each sub-query $q^{(i)}$ and dataset image $I_j$. Each $s_j^{(i)}$ is defined as 1 - $d(H_{q^{(i)}}, H_{I_j})$, where $d(H_{q^{(i)}}, H_{I_j})$ is the normalized visual word histogram distance (i.e., L1). We then create a fusion function $f$; for each dataset image, it outputs the score $f(x_j)$. The dataset images are then ranked by the fusion score.

There have been several commonly used fusion functions, e.g., average or max fusion scheme, that averages the similarity scores or pick the best one as the final fusion score. However, these simple fusion methods obtain poor retrieval performance (Section 4.2) due to the ambiguity problem. That is, views within each category are not equally important and some might be confused with the other categories (Figure 5). This motivates us to learn the weights of different views of an object category. In other words, those views exhibit high discriminative power should discriminate the object than the rest of views. We set up the learning problem using those dataset images with the same category label as query sketch as positive samples, and the others as negative samples. We learn the weight $w$

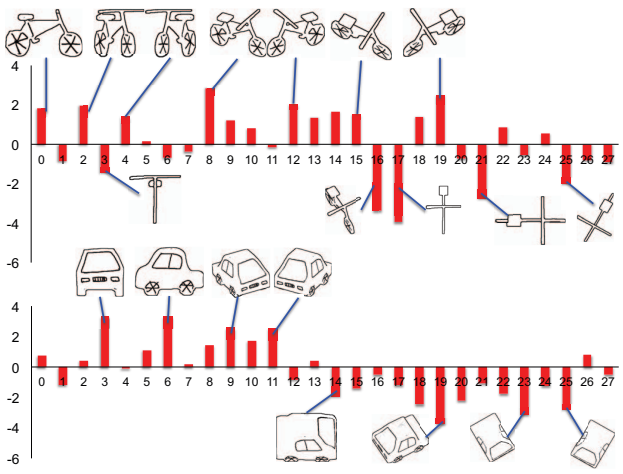


Figure 6. Visualization of the optimal weights of different views for car and bicycle categories. Our approach automatically learns those discriminative views while down-weighs the less discriminative ones.

that minimizes the following optimization problem:

$$\min_{w} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi(w; x_i, y_i) \qquad (1)$$

We use LIBLINEAR [15] for learning $w$ with L2-regularized Logistic Regression solver, the loss function $\xi(w; x_i, y_i)$ is defined as $\log(1 + e^{-y_i w^T x_i})$. We use LIBLINEAR to train our model as it achieves the comparable performance with non-linear one (e.g., RBF kernel) in LIBSVM [8] in our preliminary experiments, while it is much more efficient in testing and training phrases. Given the learned weight $w$, the final fusion score is defined as:

$$f(x_j) = \frac{1}{1 + e^{-(w^T x_j)}}, \qquad (2)$$

where $x_j$ is the feature vector for each dataset image as defined above. Figure 6 visualizes the linear weights learned from our gathered user sketches and the image dataset. The result confirms our idea, it down-weights those less discriminative views such as the (nearly) top and frontal views of the car and bicycle categories. Those more discriminative views would contribute more to the final fusion score and thus improve the retrieval performance.

# 4. Experiments

In the experiments, we compare the performances of our algorithm with the state-of-the art approach [14], SHoG local descriptor within a bag-of-visual-word model, which has been shown to outperform the other descriptors as mentioned in Section. 2.1. The details of bag-of-visual-word model construction are described in Section 3.3. The baseline method is based on a single-view query sketch, either front (top) or side view in our experiments. The front (top) and side views are selected manually based on the object characteristics. Some examples of two views used in our experiments are shown in Figure 10. Average and max fusion scheme are also evaluated, which averages the similarity scores or pick the best one as the final fusion score. Our approach further learns the weights of synthesized views to highlight those more discriminative sub-queries.

## 4.1. Dataset and Query Sketches

Due to the lack of established research in 3D sketch, it is difficult to collect a standard dataset to compare with. Thus, we evaluate the retrieval performance on a public multi-view image dataset [23], which is commonly used for evaluating pose estimation and object detection tasks. The dataset comprises images of 10 different object categories, each of which contains 10 different instances captured under a large pose variation. The total number of poses of each instance in this dataset is 72: 8 viewing angles, 3 heights and 3 scales.

For fairly comparing our approach with the baseline method, we select 5 viewing angles and the largest scale to evaluate the performance; since the backside information is unknown as users usually draw those head-on views. These 5 viewing angles can be mapped into our sphere space with the range: azimuth = $0^\circ \sim 180^\circ$ and elevation = $0^\circ \sim 90^\circ$. To generate synthesized sketches, we sample those viewpoints within this range with the defined azimuth and elevation steps. The choices of the steps will be discussed in Section 4.3.

We conducted the experiment with 10 subjects, each subject is asked to draw 10 categories as defined in this multi-view image dataset. In our study, the users were firstly explained with the rules for drawing two orthogonal views, and briefed with example images for each category, sketches are then drawn by their memory. Figure 7 shows example sketches of the car category from 10 subjects.

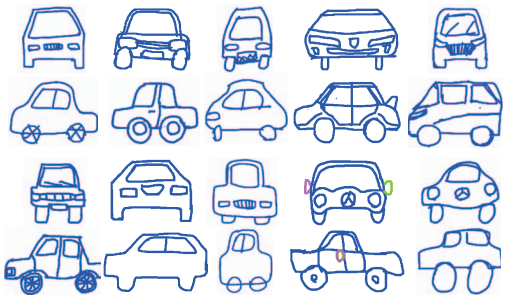
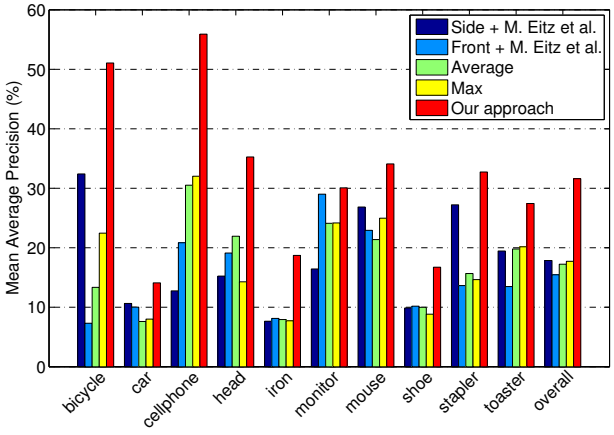
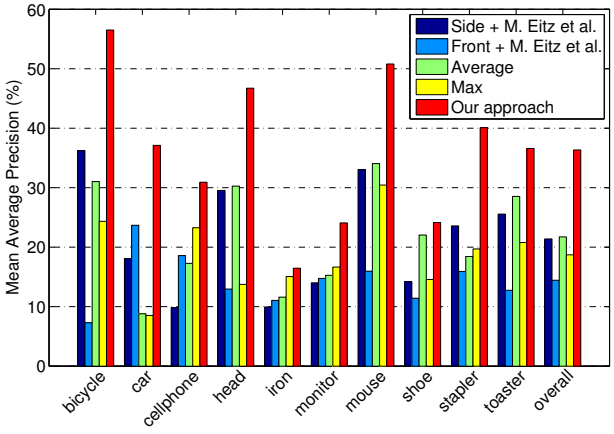


Figure 7. Examples of query sketches for car category. 1st and 3rd rows show the front views drawn by different users and the corresponded side views are shown in 2nd and 4th rows.




(a) Rotation invariant SHoG




(b) Rotation variant SHoG

Figure 8. Mean average precision (MAP) of our approach, state-of-the-art method: SHoG [14], and different fusion strategies.

### 4.2. Retrieval Performance

For quantitative evaluation, we evaluate the performance by mean average precision (MAP) with leave-one-out cross validation. In all cases, a disjunct set of query sketches from a single user are used as test samples, while the remaining sketches are training samples for learning our category-specific fusion function (cf. Eq. 1). In our case, we hypothesize that the category label of a test query is given and the dataset images are unlabeled in the testing phrase [3].

The MAP numbers reported in Figure 8 is based on 28 synthesized query sketches (azimuth and elevation steps are set to 30°). In the next section, we will show how the number of synthesized views influence the retrieval performance. Figure 8 summarizes results of different models on 10 categories.

As mentioned in Section 3.4, the ambiguity problem leads the max fusion scheme to have unacceptable even worser retrieval performance than the baseline approach. Average fusion scheme does not perform well either, since the views within a object category are not equally important and may include some noise responses from those less discriminative views.

We observed that the rotation invariant SHoG (Figure 8(a)) does not perform better than rotation variant SHoG (Figure 8(b)) in this multi-view image dataset. The reason is that rotation invariant SHoG brings more ambiguities for those outline sketches under a large pose variation, e.g., the edges of the side view of an upstanding monitor may be quite similar to the edges of a rotated sideward stapler.

The experimental results also show that the use of synthesized views with learned fusion function can significantly improve the retrieval performance and shows best MAP = 0.36 compared to the state-of-the method (MAP = 0.21 and 0.14 for frontal view and side view case). Figure 11 shows some example queries and the corresponding top 5 retrieval results for our approach and the baselines. It can be seen that our approach not only outperforms the baselines, but returns images with larger pose variation.

### 4.3. Sensitivity Test

We conduct sensitivity tests to evaluate the impact of number of synthesized views (controlled by azimuth (a) and elevation (e) steps) to the retrieval performance. Figure 9 shows the retrieval performance with different azimuth and elevation steps of our method. From the result, we found these values achieve the similar performance as adopting SHoG descriptors. The reason might be that SHoG can offer a partial solution to affine transformations. In addition, both increasing and decreasing the number of synthesized views resulted in a loss in performance due to over- or under- interpreting the pose distribution. We found that the parameters: azimuth $= 30°$ and elevation $= 30°$ achieve the best overall performance on this dataset.

## 5. Conclusions and Future work

In this paper, we propose the use of synthesized multi-view sketches as expanded sub-queries to retrieve multi-view images. Experimental results show our method outperforms the state-of-the-art and baseline methods on a public multi-view image dataset.

We currently implement the 3D reconstruction algorithm on a laptop with 1.7 GHz Intel Core i5 CPU and 4G 1333 MHz memory, it takes approximately 2 seconds on average to recovery a 3D sketch model. For image retrieval efficiency, some standard techniques (e.g., inverted index [25] and hash-based [1] methods) can be used. However, the efficiency issue is not the focus for this pilot study.

For the future work, we will design a more friendly interface to help users to draw two orthogonal views from

[3] In the text/image retrieval domain, there have shown some successful cases that use query-dependent (ranking) method to boost the performance [28, 18]. Meanwhile, it is also possible to automatically approximate query intension by adopting some recent sketch recognition system [12].

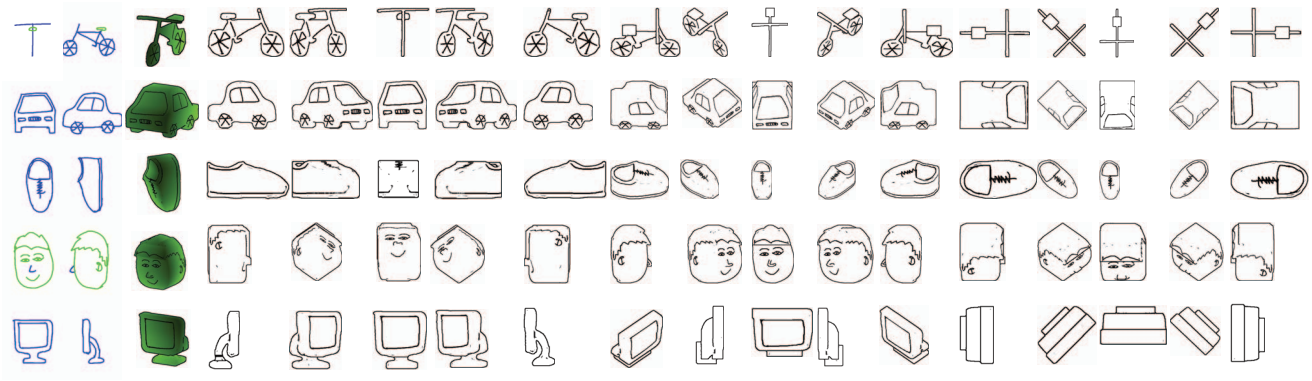


Figure 10. Some examples of user sketches and synthesized views with azimuth step = $45^\circ$ and elevation step = $45^\circ$.

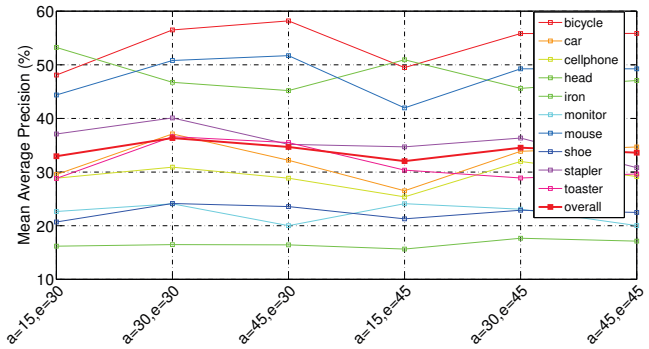



Figure 9. Sensitivity test with different choices of azimuth (a) and elevation (e) steps. The experimental results reveal that different parameters achieve similar performance as adopting a parietal affine invariant local descriptors (i.e., SHoG within a bag-of-visual-word model).

Human-Computer Interaction (HCI) aspect.

## References


[1] P. I. A. Gionis and R. Motwani. Similarity search in high dimensions via hashing. In *VLDB*, 1999.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 2002.

[3] C. Buckley. Automatic query expansion using smart:trec 3. In *In Proceedings of The third Text REtrieval Conference (TREC-3)*, pages 69–80.

[4] J. Canny. A computational approach to edge detection. *TPAMI*, 1986.

[5] L. Cao, J. Liu, and X. Tang. What the back of the object looks like: 3d reconstruction from line drawings without hidden lines. *TPAMI*, 2008.

[6] Y. Cao, W. Changhu, Z. Liqing, and L. Zhang. Edgel inverted index for large-scale sketch-based image search. In *CVPR*, 2011.

[7] A. Chalechale, G. Naghdy, and A. Mertins. Sketch-based image matching using angular partitioning. *TSMC - Part A*, 35:28–41, 2005.

[8] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[9] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: Internet image montage. *ACM Trans. Graph.*, 28:124:1–124:10, 2009.

[10] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007.

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2006.

[12] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? In *SIGGRAPH*, 2012.

[13] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. An evaluation of descriptors for large-scale image retrieval from sketched feature lines. *Computers & Graphics*, 34(5):482–498, 2010.

[14] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *TVCG*, 17(11):1624–1636, 2011.

[15] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[16] R. Hua and J. Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *CVIU*, 2013.

[17] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. In *ACM SIGGRAPH*, 1999.

[18] Y.-G. Jiang, J. Wang, and S.-F. Chang. Lost in binarization: Query-adaptive ranking for similar image search with compact codes. In *ICMR*, 2011.

[19] H. Knutsson. Representing local structure using tensors. Technical report, Computer Vision Laboratory, Linkoping University, 1989.

[20] B. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7: multimedia content description interface*. John Wiley & Sons, Inc., 2002.

[21] A. Rivers, F. Durand, and T. Igarashi. 3d modeling with silhouettes. In *ACM SIGGRAPH*, 2010.

[22] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 1999.

[23] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV*, 2007.

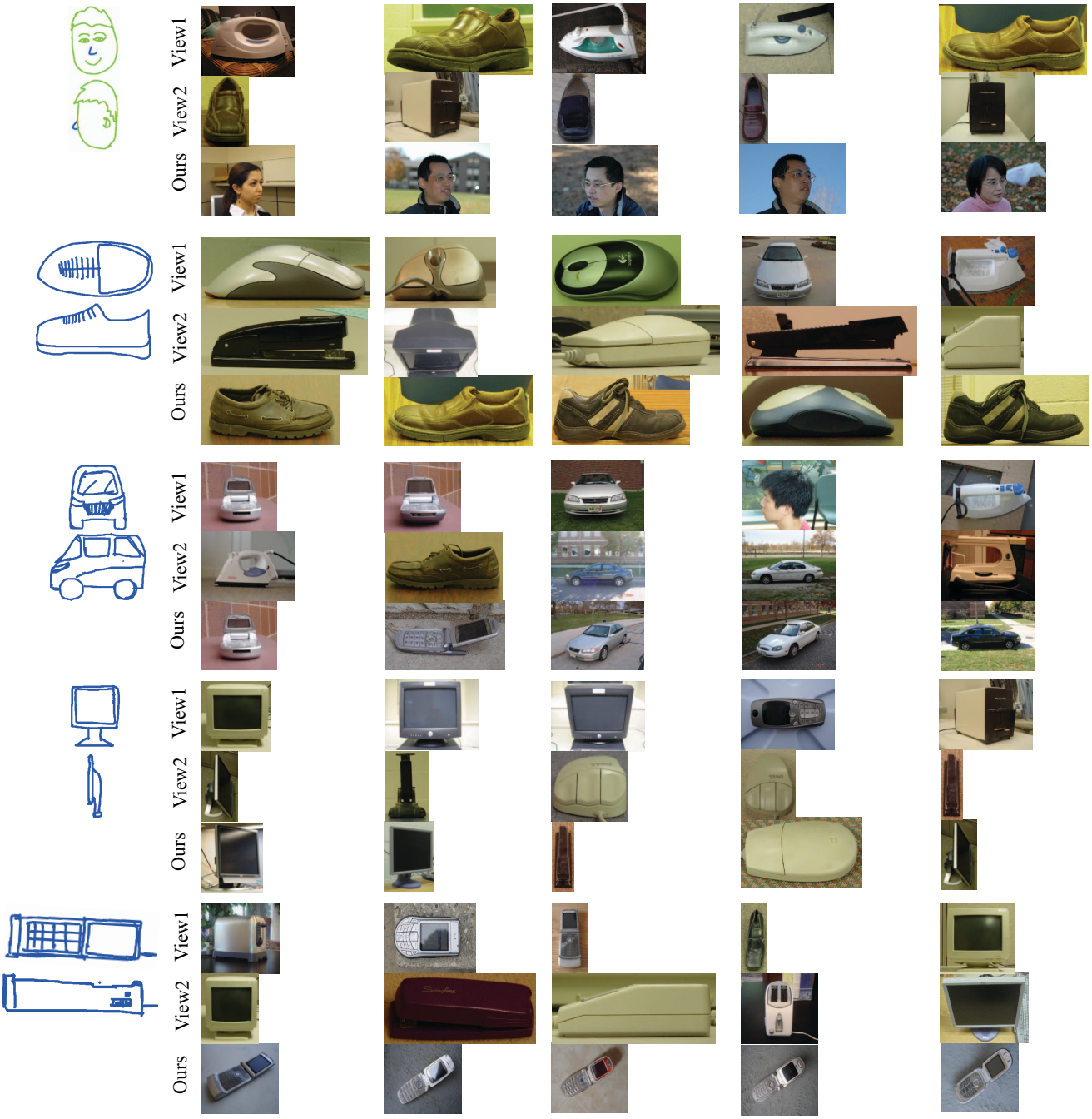



Figure 11. Qualitative comparison of our approach against the state-of-the-art method [14] based on a single-view query sketch, front or side view in our experiments. It can be seen that the proposed method, 3D sub-query expansion (capturing more information than a single query) and fusion function (emphasizing on those more discriminative sub-queries), can get more accurate and diversified results.


[24] A. Shrivastava, T. Malisiewic, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. In *ACM SIGGRAPH ASIA*, 2011.

[25] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[26] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. *CVIU*, pages 32–46, 1985.

[27] J. L. Tianfan Xue and X. Tang. Example-based 3d object reconstruction from line drawings. In *CVPR*, 2012.

[28] T. Q. A. Xiubo Geng, Tie-Yan Liu. Query dependent ranking using k-nearest neighbor. In *SIGIR*, 2012.