Scalable Sketch-based Image Retrieval using Color Gradient Features

Tu Bui and John Collomosse
Centre for Vision Speech and Signal Processing (CVSSP)
University of Surrey
Guildford, United Kingdom.
{t.bui | j.collomosse}@surrey.ac.uk

Abstract

We present a scalable system for sketch-based image retrieval (SBIR), extending the state of the art Gradient Field HoG (GF-HoG) retrieval framework through two technical contributions. First, we extend GF-HoG to enable color-shape retrieval and comprehensively evaluate several early- and late-fusion approaches for integrating the modality of color, considering both the accuracy and speed of sketch retrieval. Second, we propose an efficient inverse-index representation for GF-HoG that delivers scalable search with interactive query times over millions of images.
A mobile app demo accompanies this paper (Android).[1]

1. Introduction

The volume of visual data consumed on mobile devices is growing exponentially [4]. Gestural interaction methods, such as sketch, provide a convenient and intuitive modality for interacting with visual content on such devices, where touch-screens are the primary interaction methods. Yet despite regaining significant traction in the research community in recent years, sketch based image retrieval (SBIR) has not yet seen wide-spread adoption for visual search. Possible explanations for this include a focus primarily on shape (structure) alone in SBR, and a lack of scalability with most techniques able to index only a few thousand images to remain within practical interactive query times (i.e. sub-second response).

This paper addresses the challenge of delivering scalable SBIR at interactive speeds. We extend a state of the art framework for SBIR, via the Gradient Field HoG (GF-HoG) descriptor [11], through two key technical contributions. First, we extend GF-HoG to enable color shape retrieval (Fig. 1(b-c)). We report a comprehensive investigation exploring integration points for color as a new modality within the GF-HoG descriptor and index representation. We make several recommendations on how this second, novel modality of color can be integrated for maximum accuracy and search efficiency (speed). Second, we show how the slow linear and kd-tree based search strategies currently proposed for GF-HoG can be substituted for an efficient inverse index structure enabling scalability of GF-HoG to over three million images (several orders of magnitude greater than largest image dataset previously demonstrated for this framework) whilst retaining retrieval speeds of less than one second.

[1] https://play.google.com/store/apps/details?id=com.collomosse.sketcher

2. Related Work

Early SBIR work can be categorized by the appearance of the query; either a color blob-based or line-art sketch. Blob based techniques match on coarse attributes of color, texture and shape within the users' sketch [13] and often take into account the spatial relationships of blobs via region adjacency *e.g.* QBIC [9]. Spectral representations such as the Haar Wavelet decomposition [12] and a coarse spatial grid of 2D Fourier Transforms [18] have also been applied to blob-based SBIR.

Line-art sketches comprise a set of lines and curves the spatial arrangement of which encoded structural that is used to match sketched queries to photographs. Early approaches to line-art SBIR focused upon optimization strategies, in which the sketched contours are deformed to fit each image to assess its support for the sketch. Bimbo and Pala [1] proposed the first of such methods using elastic template matching to compensate for the imprecision in sketches. Model fitting strategies have also been explored for video search by Collomosse *et al*. [5]. Due to the computational expense of optimization, global feature extraction and matching have also been explored for line-based SBIR. Eitz *et al*. use structure tensors within a regular spatial grid over the image [7]. Some approaches seek to introduce affine invariance *e.g.* Chalechale *et al*. [3] partition the image radially, computing a edge distribution within each sector in the frequency domain for rotational invariance.

Following the success of the Bag of Visual Words (BoVW) framework for photographic search in the mid-2000s, BoVW was first extended to SBIR by Hu *et al*. [10]. Hu *et al*. [10, 11] extrapolated edge orientations from

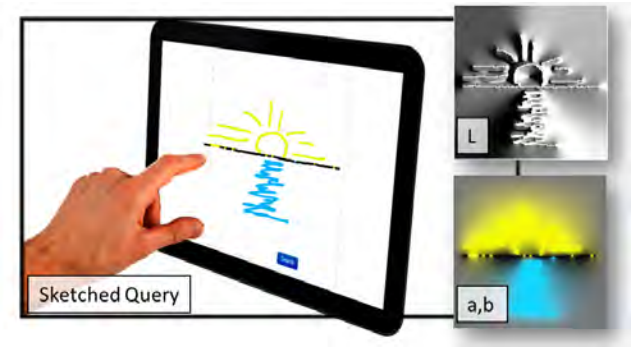

Figure 1. Dense field interpolations in Luminance (L*) and opponent color (a*,b*) space for a representative query sketch.

strokes to produce a dense gradient field, computing multi-scale HoG descriptors local to strokes that were then quantized within a hard-assignment BoVW pipeline. Perceptually simplified edge maps were later combined with HoG and BoVW in [15]. Eitz *et al*. [8] encoded the radial distribution of edge fragments local to key-points, encoded via a BoVW framework. Both techniques were shown to exhibit invariance to translation and scale, and graceful decay under rotation. Local structures were also explored in Saavedra *et al*.'s key-shapes [17, 16], where the Hungarian algorithm was used to match spatial distributions of local stroke shapes. Geometric parsing of contours and algorithms for their piecewise matching have also been explored for SBIR [14]. The disadvantage of such processes is increased computational complexity in the matching step which limits scalability to a few thousand images for practical search times. This can be ameliorated for datasets of up to a million images by introducing an initial bulk discard step (*e.g*. restricting search to a single cluster of the dataset via k-mediods [14]) but coarse restriction of search options in this manner often results in many false negatives.

The scalability of SBIR is discussed in Cao *et al*.'s MindFinder [2]. Oriented Chamfer Matching (OCM) is used to match sketches to edge maps, extracting edgels from images and storing these within an inverted index (associative array) for scalable search. MindFinder reports query times of one second over a two million image database but has no affine invariance which affects performance. The contribution of our work is to contrast several strategies for combining the modalities of color and shape into an efficient inverse index structure. For the first time, we extend the state of the art GF-HoG [11] to use such an indexing structure, and to use color line-art query sketches. With the exception of an early work-in-progress paper that lacks any implementation details [19], color SBIR has not yet been addressed in the literature.

3. Scalable color Sketch Retrieval

We build upon the work of Hu *et al*. [10, 11], who construct a dense field of edge orientations from a sparse set of strokes (in sketches) or Canny edge pixels (in photographs). This synthesized gradient field (GF) is treated as synthetic texture from which Histogram of Oriented Gradients (HoG) descriptors are computed at multiple scales and passed through a vector quantization (codebooking) process to build an image descriptor that is invariant to depictive style. In this section, we briefly recap on this dense field interpolation process (subsec. 3.1) as it remains central to the proposed SBIR approach. We then explain how color is encoded and represented within our index representation (subsec. 3.2-3.3), contrasting several such strategies.

3.1. Classical Gradient Field HoG (GF-HoG)

The Gradient field HoG (GF-HoG) method [10, 11] takes as input a binary edge field $M(\mathbf{x}) = [0, 1]$ where $\mathbf{x} \in \Omega$ the set of image pixel coordinates, comprising known $M(\mathbf{x}) = 1$ and unknown $M(\mathbf{x}) = 0$ pixels. At each known pixel a local estimate of edge orientation is available as:

$$\theta\left[\mathbf{x}\right] \mapsto \arctan\left(\frac{\delta M}{\delta y} / \frac{\delta M}{\delta x}\right), \quad \forall \mathbf{x}\ M(\mathbf{x}) = 1. \tag{1}$$

A dense orientation field $\Theta(x)$ over the image is required to sample the HoG descriptors, for which several interpolation strategies may be applied from the sparse x but Hu *et al*. identified a Laplacian smoothness constraint as producing the highest accuracy results:

$$\Theta_\Omega = \underset{\Theta}{\operatorname{argmin}} \int\int_\Omega \left(\nabla\Theta - \mathbf{v}\right)^2 \quad s.t.\ \Theta\mid_{\delta\Omega} = \theta\mid_{\delta\Omega}. \tag{2}$$

The equation is solved in closed form via Poisson's equation with Dirichlet boundary conditions *i.e*. $\triangle\Theta = \text{div}\mathbf{v}$ over Ω $s.t.$ $\Theta\mid_{\delta\Omega} = \theta\mid_{\delta\Omega}$, where $\mathbf{v}$ is the guidance field derived from θ. In practice we solve for $v = \frac{\delta M}{\delta x}$ and $v = \frac{\delta M}{\delta y}$ separately as single channel interpolation problems, then combine to obtain θ via eq. 1. The single channel solution is given by solving the linear system:

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & & & & & \vdots \\ -1 & 4 & -1 & \dots & -1 & -1 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_u \\ \vdots \\ x'_n \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ 0 \\ \vdots \\ v_n \end{bmatrix}. \tag{3}$$

where the first two and last rows of the design matrix above indicate known pixels *i.e*. $M(x_i) = 1$, for which we know the corresponding value from the guidance field v_i via eq. 1 and the middle row indicates an unknown pixel for which we wish to derive a value x'_u. In this case elements of the designed matrix row are set to 4 at the position corresponding to x_u and -1 at the four locations north, south, east and west of x_u. Thus the linear system comprises an $n \times n$ design matrix for an n pixel image; in practice we reduce complexity by resizing M preserving aspect ratio, such that Ω is a 2D image domain with the longest side of 200 pixels.

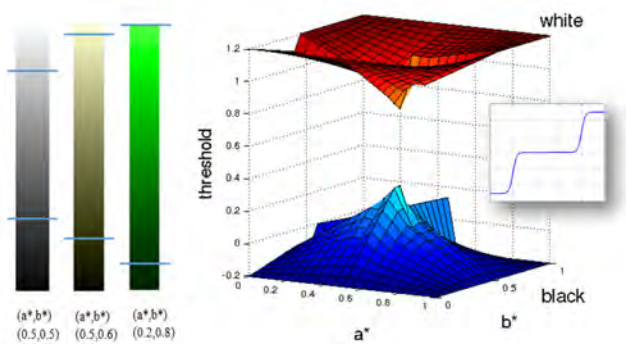

Figure 2. Double-sigmoid filter design for L^* channel: (left) different colors (a^*, b^*) have different perceptual black and white thresholds on L^*; (right) Decision surface visualising threshold values for black (bottom) and white (top) across the (a^*, b^*) space. Inset: Illustrating shape of the double-sigmoid transfer function.

Eq. 3 is solvable via a sparse linear solver *e.g.* LAPACK in only a few milliseconds for n of this order. This yields values for all x', constraining $x'_i = x_i$ if $M(x_i) = 1$.

HoG descriptors [6] are computed local to key-points $M(x) = 1$ at multiple scales. Codeword frequency histograms are computed for each image to yield the final global image descriptor. Using a 9-bin angular quantization for HoG, and a 3×3 grid structure with each grid cell of side 7, 11, 25 pixels (for the multiple scales) we obtain a 81-D descriptor for the BoVW. Hu *et al.* identified in [11] that a dictionary size of $k = 3500$ yields optimal performance of 12% mean average precision (mAP) over the diverse Flickr15k dataset. To further improve this performance we report three minor enhancements. Although these do not form our core contributions, they establish a state of the art baseline upon which we build our color and scalability extensions.

1. We apply a bilateral filter [20] to the image database prior to extracting the $M(x)$. The anisotropic smoothing preserves edges whilst abstracting image content, removing fine texture clutter that distorts the GF. This is also beneficial for the color fields (subsec. 3.2).

2. Applying binary dilation to $M(x)$ produces a more robust, noise-free GF.

3. Forcing the guidance field v to zero at the single-pixel image perimeter causes detail near the image edges to be attenuated in Θ. This has the effect of "focusing" the descriptor on objects more central to the image reducing sensitivity to clutter.

These small improvements raise the best-reported performance for classic GF-HoG over Flickr15k from 12% to 16% mAP, an equivalent boost to a very recent GF-HoG adaptation using an expensive perceptual edge detection step in lieu of Canny [15].

3.2. color extraction

color is incorporated into the GF-HoG pipeline using the CIELab color space due to its approximation of perceptual uniformity. The chroma (a* and b*) channels are extracted from the color photograph, following an initial bilateral filtering pass. To process the sketch we wish to create a smooth interpolation of known color values at $M(\mathbf{x}) = 1$ across Ω (similar to eq. 2), and this is achieved by solving linear system eq. 3 using the a* or b* channel respectively as the scalar guidance field $\mathbf{v}$, to yield interpolated fields A_Ω.

3.2.1 Luminance Transfer Function

Working solely with a* and b* offers the convenience of brightness invariance, which is attractive as color choices in user sketches tend to be highly stereotypical (*e.g.* an artificial bright green to indicate the more subtle greens occurring in nature) and there is a natural tendency for users to sketch with a limited color palette. However, this 2D space prevents discrimination between black, gray, and white all of which manifest around the midpoint of the space $a^* = b^* = 0.5$. This is problematic as such colors commonly occur in both sketches and natural objects.

We therefore consider the luminance (L^*) channel of the input image, interpolated densely via the method of subsec. 3.1 using mask $M(\mathbf{x})$ as before but drawing our guidance field $\mathbf{v}$ from $L^*(x)$ rather than $\theta(x)$ as in GF-HoG. We pass the densely interpolated result (written L^*_Ω) through a soft min-max *i.e.* double-sigmoid operator to produce a tri-level signal that soft-clamps to a binary response at extremes of L^*_Ω but defaults to the midpoint of L^*_Ω to offer invariance to luminance difference for values of L^*_Ω toward the mid-range. Writing this transfer function $S(\lambda; a, b)$, where $\lambda = L^*_\Omega(\mathbf{x})$ is normalized pixel intensity sampled from the dense field and a, b are the chroma components of a given CIELab color:

$$f(\lambda; a, b) = \frac{0.5}{1 + \mathrm{e}^{-B(\lambda - M_1(a,b))}} + \frac{0.5}{1 + \mathrm{e}^{-B(\lambda - M_2(a,b))}}. \tag{4}$$

$$S(\lambda; a, b) = 0.5 + s \times (f(\lambda; a, b) - 0.5). \tag{5}$$

where B is the slope of the sigmoid function; $M_1(.)$ and $M_2(.)$ are black and white thresholds respectively set on a per-color basis as a function of (a,b); and the weight factor s decides the lower and upper bound of the function. In our implementation we select $B = 64$ and $s = 0.1$ (Fig. 2(b)). The values of $M_1(.)$ and $M_2(.)$ are color dependent, i.e. each pair (a*,b*) has different black and white thresholds (Fig. 2(a)). We specify these thresholds *a priori* via a manual calibration process. We construct a 2-tuple look-up table $(a^*, b^*) \mapsto (M_1, M_2)$ by manually selecting black and white thresholds for pairs of (a*,b*) sampled at regular intervals. At run-time linearly interpolating nearest look-up values yields thresholds for any color. The resulting decision surfaces are visualized in Fig. 2(c). Since the black and

white range is subjective the threshold mesh is also subjective and is estimated as a once-only pre-process by an individual with good color vision.

3.2.2 Feature Sampling

Sparse features are extracted local to $M(x) = 1$ using windows of size w. We establish a 3×3 grid within the $w \times w$ window and compute the mean of each of the three interpolated color channels within each cell. The resulting 9 values from each channel are concatenated to form 3 independent channel feature vectors, which are then concatenated to form a 27-D color descriptor named $\hat{H}_C$. Local to the same feature point we extract the 81-D GF-HoG shape descriptor ($\hat{H}_S$) following the method of subsec. 3.1, without proceeding to the BoVW stage. The compound descriptor for the sparse feature point x is formed as $v_i = \left[\hat{H}_C, \hat{H}_S\right]$ used in separated form by some of the fusion strategies later explored. The set of color-shape descriptors $\mathcal{V}_i = \{v_1, ..., v_n\}$ is collected over multiple scales, where $w = \{21, 33, 45\}$ pixels and $i = [1, |\mathcal{D}|]$ for the searchable set of images $\mathcal{D}$.

3.3. Inverted Index

Inverted index structures originate in the retrieval of text documents and have become widely adopted in general content-based retrieval systems where each document can be hashed into a set of unordered tokens or 'codewords'. A simple linear search results in $O(N)$ index entries being visited per query, scaling linearly with document count. By contrast, an inverse index maintains a table of codewords and for each of these, a list of documents containing those words is maintained. As a usable hash will always result in a much smaller codeword count (k) than document count (N), leading to $O(k)$ scalability which is usually preferable as, being a hashing function, $k \ll N$.

We adopt a vector quantization (BoVW) strategy for image representation, after [11, 8, 10], and so can integrate forms of the inverse index in our algorithm. We first describe the similarity metric used to compute the ranking score in sub. 3.3.1. We then propose three approaches for representing and matching $\mathcal{V}$ in such an index. The three approaches – early fusion, hybrid inverted table and late fusion – differ in where to fuse the shape and color modalities together.

3.3.1 Similarity metric

During the process of image retrieval, only the visual words available in the query are visited. The images associated with these visual words are accumulated in the ranking scores via eq. 6 defined between a query document (sketch) Q and any image D_i [22, 21]. Eq. 6 represents the cosine rule fused with the inverse document frequency (IDF) rule which accounts for the weights of a visual word in the query, the candidate image and the whole database altogether:

$$Sim_1(Q,D_i)=\frac{1}{M_Q M_{D_i}} \sum_{p\in Q\cap D_i} (1+\ln f_{Q,p})(1+\ln f_{D_i,p})IDF_p^2. \quad (6)$$

where $M_Q=\sqrt{\sum_{p\in Q}(1+\ln f_{Q,p})^2}$, $M_{D_i}=\sqrt{\sum_{p\in D_i}(1+\ln f_{D_i,p})^2}$, $IDF_p=1+\ln\frac{N}{f_p}$. N is number of images in the whole database, f_p is number of images containing visual word W_p, $f_{Q,p}$ and $f_{D_i,p}$ are the counts of visual word W_p in the query and image respectively.

The structure of the inverted table allows multi-thread/multi-core processing to speed up the calculation. Many terms in eq. 6 can also be computed offline and the inverted table can be trivially appended as the database grows.

3.3.2 Early fusion

For each document $D_i \in \mathcal{D}$ the combined 108-D color-shape descriptors $\mathcal{V}_i$ (concatenation of the 27-D $\hat{H}_C$ and 81-D $\hat{H}_S$ components) for each key-point x $s.t.$ $M(x) = 1$ are collected for all windows over all scales (Fig.3(a)). Since the color and shape components of this descriptor are of differing dimension, we up-weight the color subspace (in our results by a factor of 2.0) to balance the modalities. Vector quantization is performed upon a random sub-sample of all $\mathcal{V}_{1..|\mathcal{D}|}$ via k-means resulting in a dictionary of codewords $\mathcal{W} = W_{1..k}$. Hard assignment of $\mathcal{V}_i$ to $\mathcal{W}$ yields corresponding frequency $(D_i, f_{D_i,p})$ where $f_{D_i,p}$ is number of occurrences of a given word W_p in image D_i.

3.3.3 Hybrid inverted table

Vector quantization is performed on the color $\hat{H}_C$ and shape $\hat{H}_S$ subspaces of $\mathcal{V}$ independently (two codebooks W^C and W^S are produced respectively). Thus, two codewords are associated via hard-assignment to each v_j, which we write W_p^C and W_q^S. A hybrid inverted table is built within space $W^C \times W^S$, yielding a hash entry for each hybrid 'word' *i.e.* concatenation of W_p^C and W_p^S. Fig. 3(b) illustrates this hybrid arrangement.

At query-time, visual words from the query are similarly extracted and concatenated into compound words. Writing each compound word $[W_p^C W_p^S]$ as W_Q and W_{D_i} for the query and document respectively, the similarity function eq. 6 may be directly applied to estimate the relevance of each document W_{D_i}.

3.3.4 Late fusion

In the late fusion strategy (Fig. 3(c)), separate codebooks are independently produced for the color $\hat{H}_C$ and shape $\hat{H}_S$ subspaces of $\mathcal{V}$. The similarity of Q to D_i is determined via eq. 6 for each modality using independent codebooks. Writing these intermediate scores $Sim_{1C}(Q, D_i)$ and $Sim_{1S}(Q, D_i)$, they are combined using a geometric mean:

$$Sim_2(Q, D_i) = Sim_{1S}(Q, D_i)^{1-w} Sim_{1C}(Q, D_i)^w \quad (7)$$

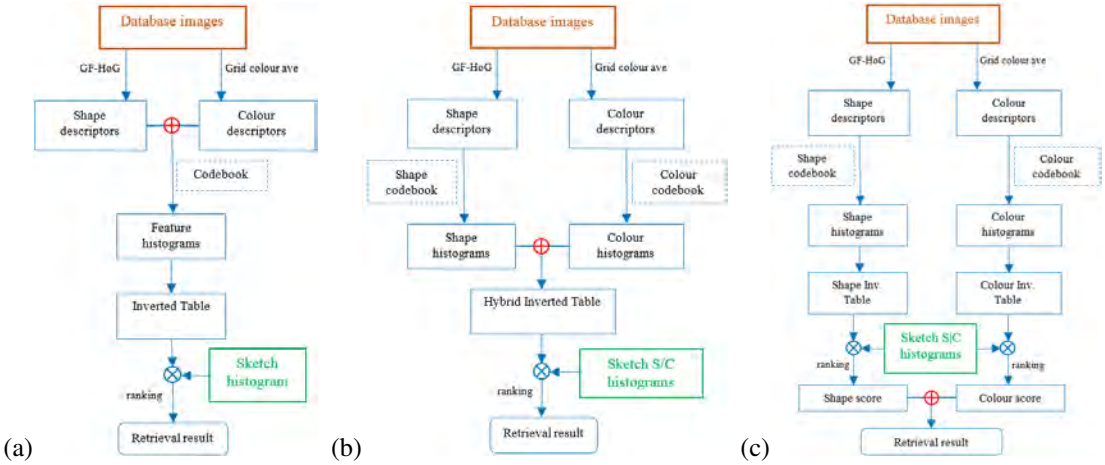

Figure 3. Summarizing the three approaches for color-shape indexing: (a) early fusion; (b) hybrid inverted S-C table; (c) Late fusion.

where w is the geometric weighting factor ($w = 0.2$ in our implementation *i.e.* 80% of the contribution to final ranking is derived from the shape score). Although w is constant for our results, it would be practical to enable user control over this value for relevance feedback.

4. Results and Discussion

We report several experiments evaluating each strategy for scalable SBIR indexing, considering first classical GF-HoG with an inverse index, and then the various color-shape representations proposed.

We use two separate datasets to evaluate the accuracy and computational cost. The public SBIR FlickR15k dataset [11] comprises of 15k color photographs with associated query sketches. The dataset is groundtruthed for 33 categories of shape and is augmented in this work with additional annotation for color [2]. We manually labeled each image to one of the 9 colors (black, white, gray, brown, red, yellow, blue, green, purple) according to the dominant color of the object of interest within the photograph. Second, we gathered a dataset of 3 million unique creative-commons licensed images from FlickR (FlickR3M). We used various keywords describing object classes echoing Flickr15k to direct the web-crawler and assemble this image set (*e.g.* "dog", "moon", "car"), landmarks (*e.g.* "Eiffel tower", "London bridge") , and scenes (*e.g.* "sun rise", "beach").

A 90 sketch query-set was formed to evaluate our approaches over both FlickR15k and FlickR3M. Using the 330 FlickR15k query sketches (10 per each of 33 categories) we recolored the sketches to each of our 9 colors resulting in 2970 sketches. We then manually culled sketches from the query-set that did not exist within the Flickr15k dataset. We considered both shape and color when making this decision (*e.g.* a purple swan). The result of this objective culling process was 90 sketches corresponding to the objects in the FlickR15k dataset. Several examples are shown in the first column of Fig. 8.

[2] Available for public download at http://X

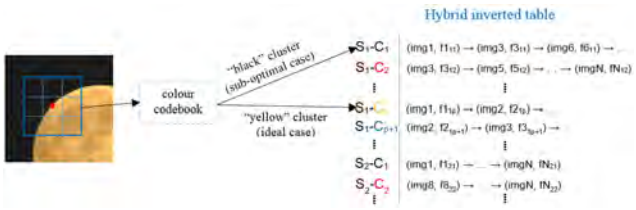

Figure 4. Interference of background color can have negative impact on the construction of the index.

4.1. Evaluation of Accuracy

We benchmark the performance of classical GF-HoG with linear search to our inverted index, using the histogram intersection (HI) metric reported to yield the highest mean average precision (mAP) in [11]. Both methods use an identical feature extraction process (i.e. using the shape-only GF-HoG descriptor outlined in subsec. 3.1), and the same dictionary size k=3500. The only difference lays in the indexing and retrieving steps. Fig. 5 (top-left) shows that the inverted index yields superior performance to linear search via HI by 1.5%. More importantly, the inverted index method performs fewer calculations than linear HI. On average a query visits less than 500k nodes of the inverted table, whilst the number of entries visited for exhaustive search is 52.5m (equal to $N \times k = 15000 \times 3500$).

Next, we evaluate the performance of the three proposed color-shape fusion strategies. We measure the mAPs of these approaches for a variety of shape and color (S-C) codebook sizes, which we denote as k_s and k_c respectively for the hybrid and late fusion approaches (note the early fusion approach uses a single codebook size k), visualized in Fig. 5. The early fusion approach has maximum mAP at $k = 5000$, whilst the hybrid and late fusion approaches ex-

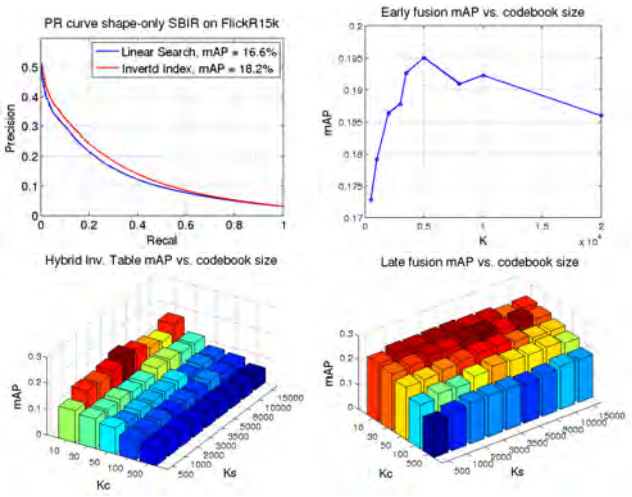

Figure 5. Top-left: Evaluating mAP for the Inverted Index vs Linear Search for classical GF-HoG over FlickR15k. Top-right: Performance of early fusion color-shape indexing strategy for varying codebook size (k). Bottom: Performance of hybrid (left) and late (right) fusion strategies for varying dictionary sizes (k_s and k_c).

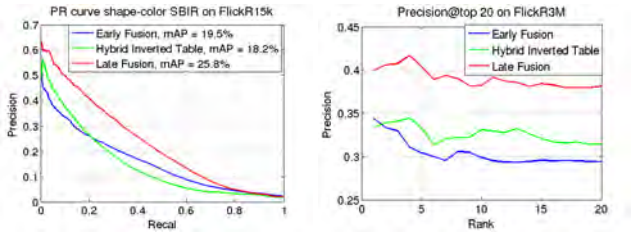

Figure 6. Performance of the three approaches (left) on FlickR15k via mAP and (right) FlickR3M via Precision@20.

hibit the highest mAP at lower color and mid-range shape codebook sizes; $(k_s, k_c) = (3500, 10)$ and $(5000, 30)$ for the two approaches respectively. The small color codebook size may be explained by the small number of color labels that we used to produce the groundtruth. Notably the mAP for the hybrid approach falls faster than the late fusion approach as k_c increases, and the mAP value is more sensitive to k_c than to k_s in both approaches.

Fig. 6(a) depicts the PR curves of the three strategies at their optimal codebook size combinations. Late fusion outperforms the others with $mAP = 26\%$. Interestingly, the hybrid strategy has a better performance than the early fusion for the top 3000 results (20% of the dataset) but has lower overall mAP. This indicates that the hybrid strategy results in a majority of the relevant images distributed both on the top and at the bottom of the ranking list (explaining the flat tail of the approach's P-R curve). One explanation is ambiguity between background and foreground colors occurring within the window used to cut features in subsec. 3.2.2, illustrated in Fig. 4. Here, a patch of a yellow object on black background might be clustered to either a

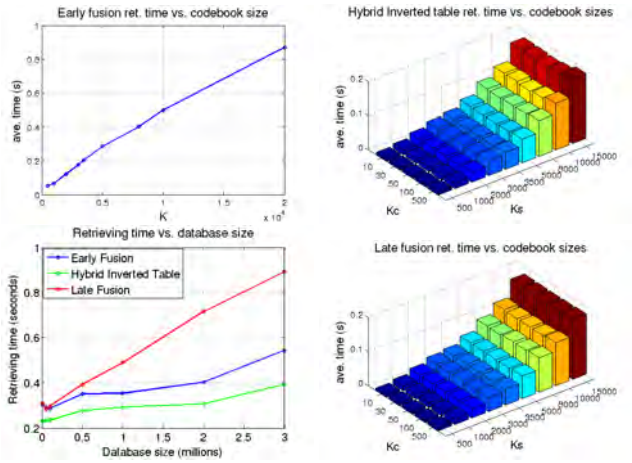

Figure 7. Effect on Query speed. Impact of increasing codebook size for the three approaches over FlickR15k: (top-left) early fusion; (top-right) hybrid inverted table; (bottom-right) late fusion. Impact of increasing dataset size for the three strategies (bottom-left).

"yellow" cluster (ideal scenario) or a "black" cluster (suboptimal scenario). The structure of the hybrid inverted table in the hybrid approach strictly enforces an "AND" relation of shape and color, *i.e*. such image patches will be associated with incorrect S-C entries.

Our experiments on the FlickR3M add further support for the late fusion strategy. Since it is difficult to label the whole 3M images we manually assess the precision of the approaches over the top 20 returned results *i.e*. producing a precision@20 metric, Fig. 6(b). Only the images which match both shape and color are considered as relevant. Also, we used "soft marking" in our assessment of shape. For example, if the user draws a circle to represent the moon, an image of "coin" or "ball" with the same color can be marked as relevant. The late fusion outperforms the others with 7% and 9% mAP boost for the top 20 results — Fig. 8 visually illustrates several success and failure cases.

4.2. Query-time Performance

We evaluate the impact of codebook size on query-time execution speed (Fig. 7). In the early fusion approach retrieval time increases linearly with k, while in the other approaches the relationship is exponential in K_s whilst no trend is obvious in color codebook size since the K_c range for functional retrieval is too small. As the codebook size increases, a sketch tends to have more non-zero bins in its histogram meaning more entries in the inverse index must be visited, increasing the retrieval time.

Fig. 7 (bottom-left) compares the scalability of the three approaches in terms of query execution time as database size grows. This has been measured by randomly subsampling the FlickR3M dataset to varying degrees. The hybrid approach exhibits the fastest growing (poorest scalability) as it has the sparsest inverted table ($k_s \times k_c$ num-

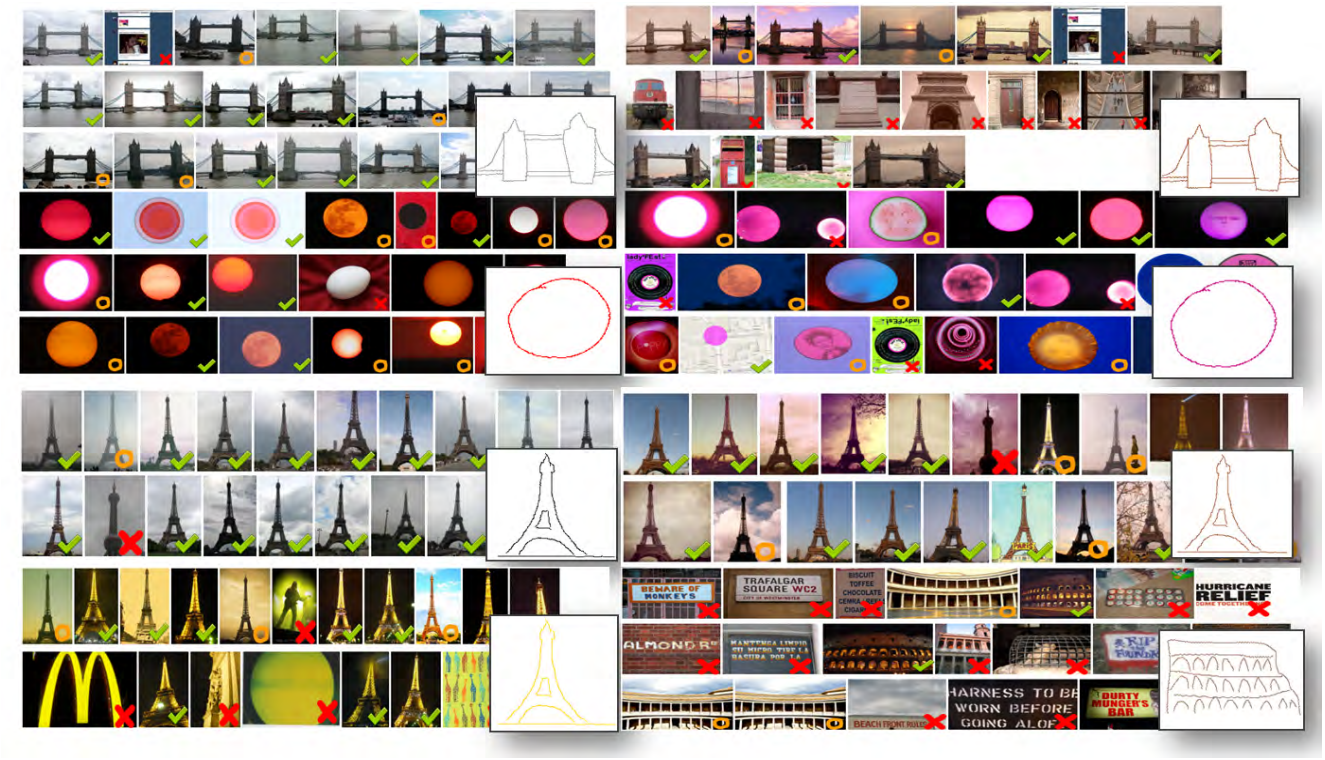

Figure 8. Top 20 retrieval results for 8 examples from the query set for FlickR3M, including both relevant (tick) and irrelevant (cross) results. Semi-relevant results (*e.g.* images having similar spatial structure, or the same shape but different color) are marked with an orange circle. Failure case in bottom-right. Queries took between 200-700ms over 3 million images.

ber of entries). By contrast, the late fusion approach has to process the shape and color inverted tables separately and thus is slower yet scales close to linear. However, even on the 3-million dataset the late fusion technique exhibits highest accuracy takes less than one second (typically 200-700ms) to process a query on a commodity 2.6Ghz AMD workstation. It is worth noting that at least half of this execution time is spent to sorting the similarity scores (with complexity $O(NlogN)$) to yield the ranked list, so the actual time spent searching the inverted table(s) is less than half of a second. In general, the querying time is shorter for those sketches with higher level of abstraction and little color variance since their shape and color histograms have fewer empty codeword bins. As a further experiment we explored application of our approach over the public ImageNet datasets; 12 million diverse images typically used for object recognition (Fig. 9). Search times for this very large dataset averaged at 3.27 ± 0.62 seconds for the six queries shown. These timings are for a single instance running on a commodity PC; clearly map-reduce or similar distributed engineering frameworks could deliver time savings.

5. Conclusions

We have reported several experiments seeking to determine the best strategy (in terms of accuracy and speed) for incorporating both the modality of color, and the efficient inverse index representation within the state of the art GF-HoG SBIR framework. Neither has been demonstrated with GF-HoG before, and more broadly, scalable SBIR has been demonstrated only with shape (not color-shape) previously. Our experiments with 3 million images demonstrate the scalability of GF-HoG which had previously been demonstrated only with 15k images. As a secondary contribution, we extended the sketch query-set and ground-truth annotation of the FlickR15k database with color labels. Concluding that the best strategy for color-shape search is a late fusion strategy with independent inverse index structure for each modality, and vector quantization applied coarsely for the color modality and finely for the shape modality. Future extensions of this work will build upon the convenience of weighting parameter w appearing after the codebooking processes, which could be interactively varied by the user for relevance feedback, enabling the user to easily re-rank results by rebalancing the two modalities.

References

[1] A. D. Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(2):121–132, 1997. 1

[2] Y. Cao, C. Wang, L. Zhang, and L. Zhang. Edgel index for large-scale sketch-based image search. In *Proc. CVPR*, pages

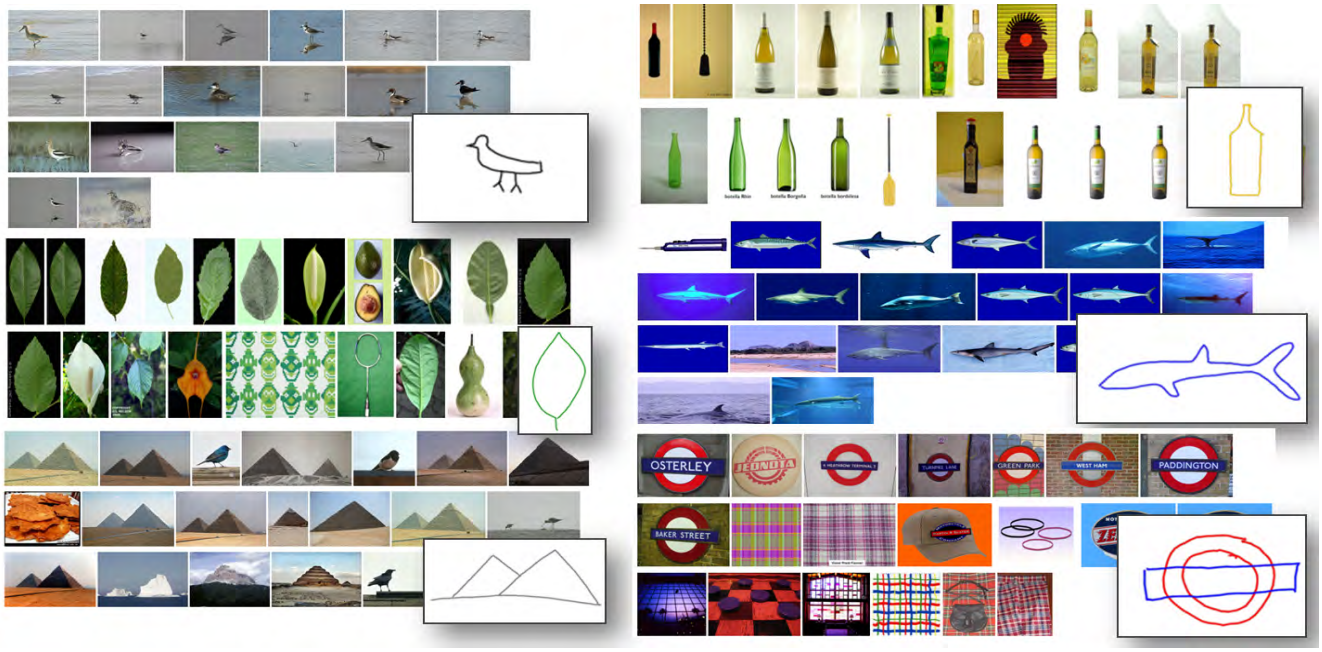

Figure 9. Representative color SBIR results for free-hand queries on ImageNet. Queries took approximately 2-3s over 12 million images.

761–768. IEEE, 2011. 2

[3] A. Chalechale, G. Naghdy, and A. Mertins. Sketch-based image matching using angular partitioning. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(1):28–41, 2005. 1

[4] Cicso. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20142019 White Paper*, 2014. 1

[5] J. P. Collomosse, G. McNeill, and Y. Qian. Storyboard sketches for content based video retrieval. In *Proc. ICCV*, pages 245–252, 2009. 1

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, volume 1, pages 886–893, 2005. 3

[7] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. A descriptor for large scale image retrieval based on sketched feature lines. In *Proc. SBIM*, pages 29–36, 2009. 1

[8] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE Trans. Visualization and Computer Graphics*, 17(11):1624–1636, 2011. 2, 4

[9] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, and D. Petkovic. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, 1995. 1

[10] R. Hu, M. Barnard, and J. P. Collomosse. Gradient field descriptor for sketch based retrieval and localization. In *Proc. ICIP*, volume 10, pages 1025–1028, 2010. 1, 2, 4

[11] R. Hu and J. Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *Computer Vision and Image Understanding*, 117(7):790–806, 2013. 1, 2, 3, 4, 5

[12] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *Proc. ACM SIGGRAPH*, pages 277–286, 1995. 1

[13] T. Kato. Database architecture for content-based image retrieval. In *SPIEIS&T 1992 Symposium on Electronic Imaging: Science and Technology*, pages 112–123. International Society for Optics and Photonics, 1992. 1

[14] S. Parui and A. Mittal. *Similarity-Invariant Sketch-Based Image Retrieval in Large Databases*, pages 398–414. Proc. ECCV. Springer, 2014. 2

[15] Y. Qi, Y.-Z. Song, T. Xiang, H. Zhang, T. Hospedales, Y. Li, and J. Guo. Making better use of edges via perceptual grouping. In *Proc. CVPR*, 2015. 2, 3

[16] J. M. Saavedra and J. M. Barrios. Sketch based image retrieval using learned keyshapes. In *Proc. BMVC*, 2015. 2

[17] J. M. Saavedra and B. Bustos. Sketch-based image retrieval using keyshapes. *Multimedia Tools and Applications*, 73(3):2033–2062, 2014. 2

[18] E. D. Sciascio, G. Mingolla, and M. Mongiello. Content-based image retrieval over the web using query by sketch and relevance feedback. In *Visual Information and Information Systems*, pages 123–130. Springer, 1999. 1

[19] X. Sun, C. Wang, A. Sud, C. Xu, and L. Zhang. Magicbrush: Image search by color sketch (demo paper). In *Proc. ACM Multimedia*, 2013. 2

[20] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Pfoc. ICCV*, pages 839–846, 1998. 3

[21] I. H. Witten, A. Moffat, and T. C. Bell. *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann, 1999. 4

[22] Q.-F. Zheng, W.-Q. Wang, and W. Gao. Effective and efficient object-based image retrieval using visual phrases. In *Proc. ACM Multimedia*, pages 77–80, 2006. 4