Similarity-invariant Sketch-based Image Retrieval in Large Databases

Sarthak Parui and Anurag Mittal

Computer Vision Lab, Dept. of CSE
Indian Institue of Technology Madras, Chennai, India

Abstract. Proliferation of touch-based devices has made the idea of sketch-based image retrieval practical. While many methods exist for sketch-based image retrieval on small datasets, little work has been done on large (web)-scale image retrieval. In this paper, we present an efficient approach for image retrieval from millions of images based on user-drawn sketches. Unlike existing methods which are sensitive to even translation or scale variations, our method handles translation, scale, rotation (similarity) and small deformations. To make online retrieval fast, each database image is preprocessed to extract sequences of contour segments (chains) that capture sufficient shape information which are represented by succinct variable length descriptors. Chain similarities are computed by a fast Dynamic Programming-based *approximate substring* matching algorithm, which enables partial matching of chains. Finally, hierarchical k-medoids based indexing is used for very fast retrieval in a few seconds on databases with millions of images. Qualitative and quantitative results clearly demonstrate superiority of the approach over existing methods.

Keywords: Image Retrieval, Shape Representation and Matching

1 Introduction

The explosive growth of digital images on the web has substantially increased the need of an accurate, efficient and user-friendly large scale image retrieval system. With the growing popularity of touch-based smart computing devices and the consequent ease and simplicity of querying images via hand-drawn sketches on touch screens [21], sketch-based image retrieval has emerged as an interesting problem. The standard mechanism of text-based querying could be imprecise due to wide demographic variations and it faces the issue of availability, authenticity and ambiguity in the tag and text information surrounding an image [35,37]. Sketch-based image retrieval, on the other hand, being a far more expressive way of image search, either alone or in conjunction with other retrieval mechanisms such as text, may yield better results. For instance, it may be possible to build a sketch in an on-line manner using the first few results of a text query system [3,20,24] and use this sketch for retrieving images that may not have any associated tag information. Image tag information may also be improved via an off-line process of sketch-based retrieval.

Several approaches have been proposed in the literature for sketch-based Object Detection and Retrieval. Ferrari *et al.* [15] describe a scale-invariant local shape feature that uses chains of k-connected Adjacent contour Segments (k-AS). To capture

the global shape properties as well, Felzenszwalb *et al.* [14] use a *shape-tree* to form a hierarchical structure of contour segments and devise an efficient Dynamic Programming (DP)-based matching algorithm to match to the given sketch. Riemenschneider *et al.* [31] describe a set of highly-overlapping translation and rotation-invariant contour descriptors that measure the relative angles amongst a set of fixed number of sampled points along a contour. However, all of these methods and many other state-of-the-art methods for Object Detection and Retrieval [4,18,22,30,36,43] perform costly online matching operations based on complex shape features to enhance the detection performance on relatively small-sized datasets such as ETHZ [16] and MPEG-7 [19]. However, for a dataset with millions of images with a desired retrieval time of at most a few seconds, these methods are inapplicable/insufficient and efficient pre-processing and fast online retrieval are necessary features for large (web)-scale Image Retrieval.

Relatively fewer attempts have been made on the problem of sketch query-based image retrieval on large databases. Eitz *et al.* [13], Cao *et al.* [8] and Bozas and Izquierdo [6] measure the sketch-to-image similarity by comparing the edges and their directions at approximately the same location in the sketch and the image after scale normalization. For fast search, Cao *et al.* [8] build an inverted index structure based on the edge pixel locations and orientations of all the database images. However, all these approaches rely on a strong assumption that the user wants only spatially consistent images as the search result. Thus, they would miss images having the sketched object at a different translation, scale or rotation. Zhou *et al.* [44] determine the most "salient" object in the image and measure image similarity based on a descriptor built on the object. However, determining saliency is a very hard problem and the accuracy of even the state-of-the-art saliency methods is low. Riemenschneider *et al.* [32] extend their idea of [31] to large scale retrieval, where to make the processing fast, invariance to scale and rotation is compromised. Furthermore, due to using high-overlapping descriptors, the computational complexity is still very high for very large datasets.

In this paper, we propose a large scale sketch-based image retrieval approach that enables efficient similarity-invariant, deformation handling matching even for datasets with millions of images unlike any relevant existing work. First, the essential shape information of all the database images is represented in a similarity-invariant way in an offline process. This is accomplished by extracting long sequences of contour segments (chains) from each image and storing them succinctly using variable length descriptors in a similarity preserving way (Sec. 2). Second, an efficient DP-based *approximate substring* matching algorithm is proposed for fast matching of such chains between a sketch and an image or between two images. Note that, variability in the length of the descriptors makes the formulation unique and more challenging. Furthermore, partial matching is allowed to accommodate intra-class variations, small occlusions and the presence of non-object portions in the chains (Sec. 3). Third, a hierarchical indexing tree structure of the chain descriptors of the entire image database is built offline to facilitate fast online search by matching the chains along the tree (Sec. 4). Finally, a geometric verification scheme is devised for an on-the-fly elimination of false positives that may accidentally receive a high matching score due to partial shape similarity (Sec. 5). Qualitative and quantitative comparisons with the state-of-the-art on a dataset of 1.2 million images clearly indicate superior performance and advantages of our approach.

Fig. 1: Creation of chains: (a) Original image, (b) Berkley edge-map [1], (c) Salient contours [45] extracted, (d) Extracted straight line-like segments, (e) Final chains obtained (black and blue).

2 From Images to Contour Chains

In this section, we describe the offline preprocessing of database images with an objective of having a compact representation which can be used to efficiently match the images with a query sketch. Since a user typically draws object boundaries, an image representation based on contour information would be appropriate in this scenario.

2.1 Obtaining Salient Contours

At first, all the database images are normalized to a standard size taking the longest side size as 256 pixels. Then, the Berkeley Edge Detector [1] is used to generate a probabilistic edge-map of the image since it gives superior edges compared to traditional approaches such as Canny [7] by considering texture along with color and brightness in an image. Since such an edge map typically contains a lot of clutter edges (Fig. 1(b)), an intelligent grouping of edge pixels can yield better contours that have a higher chance of belonging to an object boundary. The method proposed by Zhu *et al.* [45] groups edge pixels by considering long connected edge sequences that have as little bends as possible, especially at the junction points. Contours that satisfy such a constraint are called *salient* contours in their work and this method is used to obtain a set of *salient* contours from each database image (Fig. 1(c)).

2.2 Creating Segments

The salient contours thus obtained may still contain some bends in them. Some articulation should be allowed at such bends since it has been observed that object shape perspective remains relatively unchanged under articulations at such bend points [5]. These bend points along the contour are determined as the local maxima of the curvature. The curvature of a point p_c is obtained using m points on either side of it as:

$$\kappa_{p_c} = \sum_{i=1}^{m} w_i \cdot \angle p_{c-i} p_c p_{c+i} \tag{1}$$

where w_i is the weight defined by a Gaussian function centered at p_c. This function robustly estimates the curvature at point p_c at a given scale m. The salient contours are split into different segments at such high curvature points and as a result, a set of straight line-like segments are obtained for an image (Fig. 1(d)).

2.3 Chaining the Segments

Given a set of straight line-like contour segments in an image, we design compact representation of an image by considering ordered sequences of segments that utilize the connectedness of the object boundary. The connectivity among the segments suggests an underlying graph structure. Ferrari *et al.* [15] utilize this by constructing an unweighted *contour segment network* which links nearby edge segments, and then extracting $k(\leq 4)$ adjacent contour segments(k-AS) for a large number of image windows. They trace the object boundary by linking individual small k-AS at the multi-scale detection phase. Although such an approach performs well in clutter, it leads to a costly online matching operation, which motivates us to represent an object with much longer segment chains a priori for each image rather than with very small contour fragments.

It has been observed that long sequences of segments typically have a large intersection with important object boundaries. Therefore, in our approach, we try to extract the long sequences. To obtain such long sequences, in contrast to [15], a weighted (rather than an unweighted) graph is constructed where each end of a contour segment is considered as a vertex and the edge weight is equal to the length of the segment. Vertices from two different contour segments are also joined by an edge if they are spatially close. The weight of such an edge is taken as $\lambda_d \cdot \exp(-d/D)$, where d is the spatial distance between the two end points, D is the diagonal length of the normalized image and λ_d is a constant factor that provides a trade-off between the segment length and the inter-segment gap.

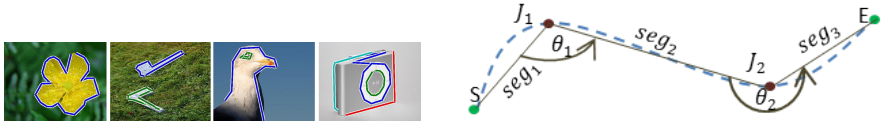
The weight of an edge in the graph represents the spatial extent of the segments and the connectedness between them. Therefore, a long path in the graph based on the edge weights relates to a long and closely connected sequence of contour segments which is what we desire in the image. As the graph may contain cycles, to get a unique long path, the maximum spanning tree[1] is constructed for each connected component using a standard minimum spanning tree algorithm [9] and the longest paths from such trees are determined using Depth First Search [9].

A long path thus obtained may deviate from the object boundary at the junction points. Ideally, to capture maximum shape information, all possible sequences through such junction points should be considered. However, this leads to an exponential blowup in the representation and is therefore impractical for a database of millions of images. Hence, as a trade-off between representative power and compactness, we follow a greedy approach by considering only edge-disjoint long sequences in the graph. These are determined by sequentially finding and removing the longest contour in the graph. Finally, an image is represented as a set of such non-overlapping long sequences of segments (Fig. 1(e)), and we call each of these sequences a *chain*. Fig. 2(a) shows the chains thus obtained in some common images.

2.4 Creating Descriptors for Each Chain

In order to efficiently match two chains in a similarity-invariant way, we require a compact descriptor that captures the shape information of the extracted chains in a

[1] Maximum spanning tree of a graph can be computed by negating the edge weights and computing the minimum spanning tree.



(a) (b)

Fig. 2: Image Representation: (a) Chains extracted for some images. Different chains are represented using different colors. (b) The chain for the curve SE is composed of three line segments. The descriptor for this chain is $\Psi = \left\langle \gamma_i = \frac{l_{seg_i}}{l_{seg_{i+1}}}, \theta_i \mid i \in \{1, 2\} \right\rangle$.

similarity-invariant way. Towards this goal, the local shape information is captured at the joints in a scale, in-plane rotation and position invariant way. For the i^{th} joint of chain k (J_i^k), the segment length ratio $\gamma_i = \frac{l_{seg_i}}{l_{seg_{i+1}}}$ (l_{seg_i} denotes the length of the i^{th} segment) and the anti-clockwise angle θ_i (range: [0, 2π]) between the corresponding pair of segments seg_i and seg_{i+1} are determined, as shown in Fig. 2(b). The descriptor Ψ^k for a chain k with N segments is then defined as an ordered sequence of such similarity-invariant quantities:

$$\Psi^k = \langle \gamma_i, \theta_i \mid i \in \{1 \dots N-1\} \rangle \tag{2}$$

Note that, Riemenschneider *et al.* [31] also use joint information by measuring the relative angles among all pairs of sampled points along a contour. However, their representation is not scale invariant which leads to a costly online multi-scale matching phase. In contrast, the proposed descriptor is insensitive to similarities and succinct enough for efficiently representing and matching millions of images.

Having extracted chains from images and compactly representing them in a similarity-invariant way, we next describe an approach for efficiently matching such chains.

3 Matching Two Chains

Standard vectorial type of distance measures are not applicable for matching due to variability in the lengths of the chains. This constraint makes the task more challenging since most of the fast indexing mechanisms for large scale retrieval exploit a metric structure. Further, note that the object boundary is typically captured by only a portion of the chain in the database image. Therefore, a partial matching strategy of such chains needs to be devised which can be smoothly integrated with an indexing structure to efficiently determine object shape similarity.

Since image chains are typically noisy, a chain that captures an object boundary may have non-object contour segments on either side of the object boundary portion. Furthermore, we assume that the object boundary is captured by a more or less contiguous portion of the chain and is not split by large gaps. Although such large split-ups may occur in certain circumstances, allowing such matches leads to a lot of false matches of images due to too much relaxation of the matching criteria. This is illustrated in

Fig. 3(a), where the split matches put together do not match with the intended shape structure. Thus, in our work, the similarity between two chains is measured by trying to determine the maximum (almost) contiguous matching portions of the sequences while leaving out the non-matching portions on either side from consideration (Fig. 3(b)). This is quite similar to the Longest Common Substring[2] problem [9] with some modifications. The matching strategy between two chains is formulated by first matching individual joints of two chains.

3.1 Joint Similarity

Since exact correspondence of the joints does not capture the deformation that an object may undergo, we provide a slack while matching and score the match between a pair of joints based on the deviation from exact correspondence. The score $S_{jnt}(x, y)$ for matching the x^{th} joint of chain C_1 to the y^{th} joint of chain C_2 is taken to be the product of three constituent scores:

$$S_{jnt}(x, y) = S_{lr}(x, y) \cdot S_{ang}(x, y) \cdot S_{sz}(x, y) \tag{3}$$

$S_{lr}(x, y)$ is the closeness in the segment length ratio at the x^{th} and the y^{th} joints of the two descriptors:

$$S_{lr}(x, y) = \exp\left(\lambda_{lr} \cdot \left(1 - \Omega\left(\gamma_x^{C_1}, \gamma_y^{C_2}\right)\right)\right) \tag{4}$$

where $\gamma_x = \frac{l_{seg_x}}{l_{seg_{x+1}}}$ as defined in Sec. 2.4, $\Omega(a, b) = \max(a/b, b/a), a, b \in \mathbb{R}_{>0}$ measures the relative similarity between two ratios ($\Omega(a, b) \in (0, 1]$) and $\lambda_{lr} (= 0.5)$ is a constant. $S_{ang}(x, y)$ determines the closeness of the angles at the x^{th} and y^{th} joints and is defined as:

$$S_{ang}(x, y) = \exp\left(-\lambda_{ang} \cdot \left|\theta_x^{C_1} - \theta_y^{C_2}\right|\right) \tag{5}$$

where, $\lambda_{ang} (= 2)$ is a constant. These two components measure the structure similarity between a pair of joints. Due to the insensitivity of the descriptor itself to scale, translation and rotation, these measures are invariant to such transformations. However, lengthy segments are more relevant to an object and should get a higher score. Thus, it is desirable to give a higher score to a pair of matched joints if the segment lengths corresponding to the joints are large. This is captured by S_{sz} and is defined as:

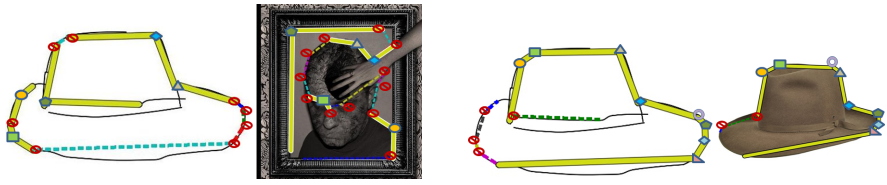
$$S_{sz}(x, y) = \min\left(\left(l_{seg_x}^{C_1} + l_{seg_{x+1}}^{C_1}\right), \left(l_{seg_y}^{C_2} + l_{seg_{y+1}}^{C_2}\right)\right) \tag{6}$$

where, seg_x and seg_{x+1} are the two segments on either side of a joint x. The information about individual segment lengths is also retained in the chain extraction stage for such a calculation.

3.2 Chain Matching

Given the scoring mechanism between a pair of joints, the match score between two chains can be determined by calculating the cumulative joint matching score of contiguous portions in the two chains. Although exact matching of such portions can be

[2] Substring, unlike subsequence, does not allow gap between successive tokens.



(a) (b)

Fig. 3: (a) A match when fragmented skips are allowed. (b) A match when only almost-contiguous matches are allowed. Matched joints are shown with the same marker in the sketch and the image. Unmatched portions of the chains are indicated by dashed lines.

considered, due to intra-class shape variations, small partial occlusion or noise, a few non-object joints may occur in the object boundary portion of the chain. To handle these non-object portions, some skips need to be allowed. Thus, the problem is formulated as one that finds *almost-contiguous* matches in the two descriptors that are to be matched. This is accomplished by applying a constant skip penalty α for the skips in the chain. To penalize lengthy skips more, the skip penalty is also weighted (ω_x) by the length of the segments on either side of a skipped joint x: $\omega_x = \left(l_{seg_x} + l_{seg_{x+1}}\right)$

Towards finding almost-contiguous matches, one can formulate the match score $M(p_1, q_1, p_2, q_2)$ for the portion of the chain between joints p_1 and q_1 in chain C_1 and joints p_2 and q_2 in chain C_2. Let the set J_1 and J_2 denote the set of joints of chains C_1 and C_2 respectively in this interval. Also let JM be a matching between J_1 and J_2 in this interval. We restrict JM to obey order constraint on the matches, i.e., if the joints a_1 and b_1 of the first chain are matched to the joints a_2 and b_2 respectively in the second chain, then a_1 occurring before b_1 implies that a_2 also occurs before b_2 and vice versa. Also let $X(\mathrm{JM}) = \{x|(x, y) \in \mathrm{JM}\}$ and $Y(\mathrm{JM}) = \{y|(x, y) \in \mathrm{JM}\}$ be the set of joints covered by JM. Then $M(p_1, q_1, p_2, q_2)$ is defined as:

$$M(p_1, q_1, p_2, q_2) = \max_{\substack{\mathrm{JM} \in \text{ ordered matchings in interval}(p_1,q_1) \text{ and}(p_2,q_2)}} \left(\sum_{\substack{(x,y)\in \\ \mathrm{JM}}} S_{jnt}(x, y) - \sum_{\substack{x\in \\ \mathrm{J}_1\setminus X(\mathrm{JM})}} \omega_x^1 \alpha^1 - \sum_{\substack{y\in \\ \mathrm{J}_2\setminus Y(\mathrm{JM})}} \omega_y^2 \alpha^2 \right) \tag{7}$$

Note that α^1 and α^2 may be different since while matching a sketch chain to an image chain, more penalty is given to a skip in the sketch chain ($\alpha = 0.07$) since it is considered cleaner and relatively more free from clutter compared to an image chain ($\alpha = 0.03$). Now, the maximum matching score ending at the joint q_1 of C_1 and q_2 of C_2 from any pair of starting joints, is defined as:

$$M(q_1, q_2) = \max_{p_1,p_2} M(p_1, q_1, p_2, q_2) \tag{8}$$

We take the matching score of a null set ($p_1 > q_1$ or $p_2 > q_2$) as zero which constrains $M(q_1, q_2)$ to take only non-negative values. Then, it is not difficult to prove that M can

be rewritten using the following recurrence relation:

$$M(q_1,q_2) = \begin{cases} 0, & \text{if } q_1, q_2 = 0 \\ \max \begin{cases} M(q_1-1,q_2-1) + S_{jnt}(q_1,q_2) \\ M(q_1-1,q_2) - \omega^1_{q_1}\alpha^1 \\ M(q_1,q_2-1) - \omega^2_{q_2}\alpha^2 \\ 0 \end{cases}, & \text{otherwise} \end{cases} \tag{9}$$

This formulation immediately leads to an efficient Dynamic Programming solution that computes M for all possible values of q_1 and q_2 starting from the first joints to the last ones. A search for the largest value of $M(q_1, q_2)$ over all possible q_1 and q_2 will then give us the best almost-contiguous matched portions between two chains C_1 and C_2 that have the highest matching score. Furthermore, to handle an object flip, we match by flipping one of the chains as well and determine the best matching score as the one that gives the highest score between the two directions. We call the final score between two chains C_1 and C_2 as the Chain Score $CS(C_1, C_2)$. The entire operation of matching two chains takes $\mathcal{O}(n_{C_1} * n_{C_2})$ time, where n_{C_1} and n_{C_2} are the number of joints in chains C_1 and C_2 respectively. It has been observed that a chain typically consists of 12-17 joints leading to a running time of approximately 100-400 units of joint matching, which is not very high. Note that, this DP formulation is similar to the Smith-Waterman algorithm (SW) [39], which aligns two protein sequences based on a fixed alphabet-set and predefined matching costs. Meltzer and Soatto [25] use SW to perform matching between two images under wide-baseline viewpoint changes. Our method is slightly different from this since it performs matching based on a continuous-space formulation that measures the deviation from exact correspondence to handle deformation.

This chain-to-chain matching strategy is used to match two image chains during indexing and a sketch chain to an image chain during image retrieval. A brief description of Image Indexing is given next.

4 Image Indexing

Given a chain descriptor, matching it online with all chains obtained from millions of images will take considerable amount of time. Therefore, for fast retrieval of images from a large scale dataset, an indexing mechanism is required. Due to the variability in the length of the descriptors, it is difficult to use metric-based data structures, such as *k-d tree* [26] or Vantage-Point tree [42]. Therefore, in this work, an approach similar to *hierarchical k-means* [26,27] is used, in which a hierarchical structure is constructed by splitting the set of chains into k different clusters using the *k-medoids* [28,40] algorithm. (Note that, because of the variable-length chain descriptors, *k-means* is inapplicable.) In our approach, k chains are chosen as the cluster centroids probabilistically using the initialization mechanism of *k-means++* [2] which increases both speed and accuracy. This operation is then recursively performed on the individual clusters to determine the clusters at different levels of the search tree. A leaf node of such a tree contains images in which at least one chain of each image matches to the medoid chain descriptor of that leaf node. Since an image has multiple chains, it can be present at multiple leaves.

5 Image Retrieval

A user typically draws the object boundary. From a touch-based device, the input order of the contour points of the object boundary is usually available. Therefore, sketch chains are trivially obtained and the corresponding descriptors are determined (Eq. 2). For each of these sketch chain descriptors, a search in the hierarchical k-medoids tree yields a small set of images in which at least one chain for each image matches with the query chain in the tree. Note that, for multiple sketch chains, we get multiple sets of images from the leaf nodes of the search tree, all of which are taken for the next step.

Given a set of retrieved images and corresponding matched chains, we devise a sketch-to-image matching strategy to rank the images. The matching score of an image for a given sketch is calculated based on the cumulative matching scores of individual matched chain pairs between the sketch and the image. Since the actual object boundary may be split across multiple chains, it is necessary to consider geometric consistency of the matched portions of multiple chains for correct retrievals. Although such geometric consistency has been studied previously in the literature [29,33,41], this is considered in a new context in this work.

5.1 Geometric consistency between matched chains

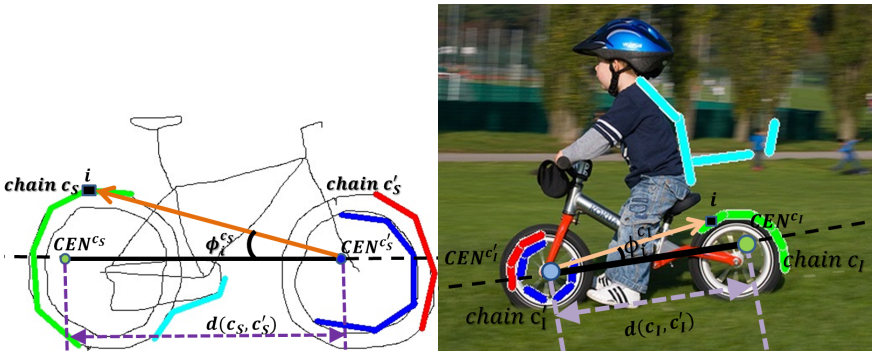
The geometric consistency of the matched portions of a pair of chains $\mathbf{p} = (m(C_S), m(C_I))$ with respect to that of another chain pair $\mathbf{p}' = (m(C'_S), m(C'_I))$, where C_S and C'_S are the sketch chains and C_I, C'_I are the image chains, is measured based on two factors: i) *distance-consistency* $G_d(\mathbf{p}, \mathbf{p}')$ and ii) *angular-consistency* $G_a(\mathbf{p}, \mathbf{p}')$. Since only small skips are allowed while matching the object portion, the distance between the centroids of the matched chain portions remains relatively robust to the presence of noise. Therefore, $G_d(\mathbf{p}, \mathbf{p}')$ is defined in terms of the closeness of the distances between the chain centroids $d(m(C_S), m(C'_S))$ in the sketch and $d(m(C_I), m(C'_I))$ in the database image (Fig. 4). To achieve scale insensitivity, the distances are normalized by the total length of the matched portions of the corresponding chains.

$$G_d(\mathbf{p}, \mathbf{p}') = \exp\left(\lambda_c \cdot \left(1 - \Omega\left(\frac{d(m(C_S), m(C'_S))}{L_S}, \frac{d(m(C_I), m(C'_I))}{L_I}\right)\right)\right) \tag{10}$$

where, L_S=length $(m(C_S))$+ length$(m(C'_S))$, L_I=length $(m(C_I))$+ length $(m(C'_I))$ and λ_c(=1) is a scalar constant and Ω is defined in Eq. 4. The next factor G_a measures *angular-consistency*. To achieve rotational invariance, the line joining the corresponding chain centers is considered as the reference axis and the angle difference at the i^{th} joint is determined (Fig. 4). $G_a(\mathbf{p}, \mathbf{p}')$ is defined using the average angle difference of all the individual matched joints in a chain:

$$G_a(\mathbf{p}, \mathbf{p}') = \exp\left(-\lambda_a \cdot \frac{1}{N^{J_\mathbf{p}}} \sum_{i=1}^{N^{J_\mathbf{p}}} \left|\phi_i^{C_S} - \phi_i^{C_I}\right|\right) \tag{11}$$

where, $N^{J_\mathbf{p}}$ is the number of matched joints between C_S and C_I and λ_a(=2) is a scalar constant. Since, both G_d and G_a are should be high for consistent matching, we consider the pairwise geometric consistency $G(\mathbf{p}, \mathbf{p}')$ as a product of the constituent factors: $G(\mathbf{p}, \mathbf{p}') = G_d(\mathbf{p}, \mathbf{p}') \cdot G_a(\mathbf{p}, \mathbf{p}')$.



(a) (b)

Fig. 4: Pairwise geometric consistency of the matched portions of a chain pair $\mathbf{p} = (C_S, C_I)$ with respect to $\mathbf{p}' = (C'_S, C'_I)$ uses (i) the distances $d(C_S, C'_S)$ and $d(C_I, C'_I)$ between their centroids (CEN) and (ii) the difference of angles $\left|\phi_i^{C_S} - \phi_i^{C_I}\right|$.

Erroneously matched chains are typically geometrically inconsistent with others and one may have both geometrically consistent and inconsistent pairs in a group of matched pairs between a sketch and an image. Therefore, the geometric consistency $GC(\mathbf{p})$ for a matched pair $\mathbf{p}$ is taken to be the maximum of $G(\mathbf{p}, \mathbf{p}')$ with respect to all other matched pairs $\mathbf{p}'$: $GC(\mathbf{p}) = \max_{\mathbf{p}'} G(\mathbf{p}, \mathbf{p}')$. This allows us to neglect the false matches while determining the consistent matched pairs. Finally, the similarity score of a database image I with respect to a sketch query S is determined as:

$$Score(I) = \sum_{\mathbf{p} \in \mathrm{P}} GC(\mathbf{p}) \cdot CS(\mathbf{p}) \tag{12}$$

where $CS(\mathbf{p})$ is the *Chain Score* for the match of the chain pair $\mathbf{p}$ (Sec. 3.2) and P is the set of all matched pairs of chains between a sketch S and an image I. Since erroneously matched chains get very low score for consistency, effectively only the geometrically consistent chains are given weight for scoring an image. This score is used to determine the final matching of the database images, which can be used for ranked display of such images. Results of the experiments performed are presented next.

6 Experiments and Results

To evaluate the performance of our system, we have created a database of 1.2 million images, which contains 1 million Flickr images taken from the MIRFLICKR-1M image collection [17]. In addition, we included images from Imagenet [10] database in order to have some common object images in our database. We asked 5 random subjects to draw sketches on a touch-based tablet and collected 75 sketches, which, along with 100 sketches from a crowd-sourced sketch database [11], containing 24 different categories

Table 1: Precision (expressed as % of true positives) at different ranks for 175 retrieval tasks in 24 categories on a dataset of 1.2 million images. B: Best, W: Worst, A: Average performances are computed among sketches for each category and then averaged. CS+GC and CS indicate the performances with and without geometric verification respectively.

Method	Top 5			Top 10			Top 25			Top 50			Top 100			Top 250		
	B	W	A	B	W	A	B	W	A	B	W	A	B	W	A	B	W	A
TENSOR [12]	30.8	7.5	14.7	30	7.1	13.7	24.8	7	12.9	20.8	7	12.3	16.5	5.8	10.2	9.4	3	5.7
EI [8]	36.7	20.8	23.4	34.2	17.9	21.5	30	15.3	19.5	27	13.8	17.5	22.2	11.2	14.8	15.7	7.8	10.5
CS	50	11.7	26.1	42.1	10.8	22.9	33.5	7.8	18.5	27.8	5.7	15	23.1	5.1	12.9	18.3	4	9.7
CS+GC	**80.8**	**42.5**	**60.8**	**72.5**	**38.3**	**53.6**	**54.7**	**29.3**	**39.5**	**40.3**	**20.7**	**28.5**	**31.8**	**16.3**	**22.2**	**23**	**12.5**	**16.5**

in total, are used for retrieval. In the experiments, the hierarchical index for 1.2 million images is generated with a branching factor of 32 and a maximum leaf node size of 100, which leads to a maximum tree depth of 6. This is used to reduce the search space to around 1500 similar images for a given sketch, for which geometric consistency (Eq. 12) is measured to rank the list of retrieved images from the search tree. The whole operation for a given sketch typically takes $1-5$ seconds on a single thread running on an Intel Core i7-3770 3.40GHz CPU, with the running time depending on the number of chains in the sketch and almost the whole processing time is consumed by the geometric verification phase. This time can be scaled down almost linearly with the number of cores as the geometric consistency check on each image can be done in parallel. The hierarchical index for our dataset required only around 150 MB of memory. We observed a memory footprint of approximately 6.5 GB while also storing the chain descriptors for all 1.2M images.

Visual results for 14 sketches of different categories of varying complexity are shown in Fig. 5. These clearly indicate insensitivity of our approach to similarity transforms (e.g 1^{st} and 3^{rd} retrieved image of the swan sketch). Furthermore, due to our partial matching scheme, an object is retrieved even under a viewpoint change if a portion of the distinguishing shape structure of the object is matched (e.g 8^{th} image for swan). Global invariance to similarities as well as matching with flipped objects can be seen in the results for the bi-cycle sketch (7^{th} and 10^{th} retrieved image). False matches (e.g duck, parrot for the sketch of swan; face, wall-clock for lightbulb in Fig. 5) typically occur due to some shape similarity between the sketch and an object in the image, the probability of which is higher when the sketch is simple and/or contains only one chain (e.g lightbulb).

Quantitative measurement of the performance of a large scale retrieval system is not easy due to difficulty in obtaining the ground truth data, which is currently unavailable for a web-scale dataset. Common metrics to measure retrieval performances (F-measure, Mean Average Precision [23] etc.) use recall which is impossible to compute without a full annotation of the dataset. Therefore, to evaluate the performance of our approach quantitatively, we use the Precision-at-K measure for different rank levels (K) for the retrieval results (Table 1). This is an acceptable measure since an end-user of a large scale Image Retrieval system typically cares only about the top results.

Unavailability of public implementation of any prior work makes it difficult to have a comparative study. Even though a Windows phone App (*Sketch Match*) [38] based on [8] is available, the database is not available to make a fair comparison to other

Fig. 5: Top retrieved images for 14 sketches from 1.2 million images. Retrieved images indicate similarity insensitivity and deformation handling of our approach. Chains are embedded on the retrieved images to illustrate the location of matchings. Multiple matched chains are shown using different colors. Correct, similar and false matches are illustrated by green, yellow and red boxes respectively (Best viewed in color).

algorithms. Hence, we re-implemented this algorithm [8] (EI) as well as another by Eitz *et al.* [12] (TENSOR) and tested their algorithms on our database for the purpose of comparison. Zhou *et al.* [44] did not provide complete implementation details in their publication and it is not trivial to make [32] run efficiently on a very large database. Furthermore, [32] did not show any result on a large scale dataset and [44] shows results only for 3 sketches. Hence, these methods were not compared against.

Table 1 shows the best, worst and average retrieval scores (among multiple sketches of a given object category, averaged over all 24 categories). The significant deviation between the best and the worst retrieval performances indicate the diversity in the quality of the user sketches and the system response to it. It can be observed from Table 1 that our method significantly outperforms the other two methods on this large dataset. Both TENSOR [12] and EI [8] consider edge matchings approximately at the same location in an image as that of the sketch and therefore, the retrieved images from their system contain the sketched shape only at the same position, scale and orientation while images containing the sketched object at a different scale, orientation and/or position are missed leading to false retrievals (Fig. 6). Similar performance was observed by us on the *Sketch Match* app [38], a direct comparison with which is inappropriate since the databases are different. To evaluate the advantage of the geometry-based verification step, we also show the retrieval performance with and without this step and it can be observed that the geometric consistency check improves our results substantially. Note that, due to the unavailability of a fully annotated dataset of a million images, it is difficult to use an automated parameter learning algorithm. Hence, parameters are chosen emperically by trying out a few variations. Proper parameter learning/tuning could possibly improve the results further.

To provide easy comparisons on a standard dataset and compute the recall which is difficult for a large dataset, we tested our system on the ETHZ extended shape dataset [34] consisting of 385 images of 7 different categories with significant scale, translation and rotation variance. Out of 175 sketches used for evaluation in the large scale dataset, 63 sketches fall into different categories of ETHZ [34] and these are used

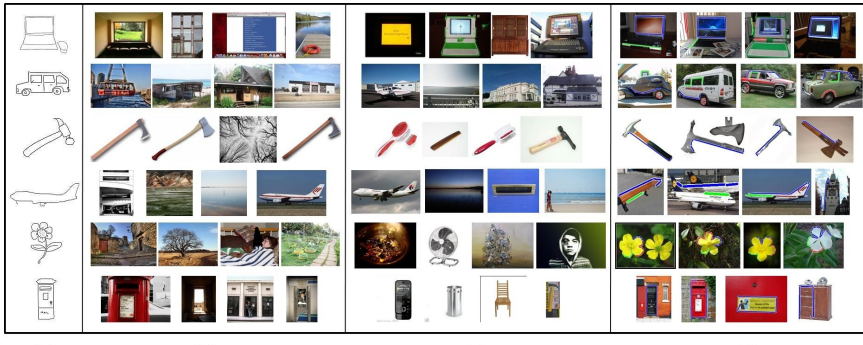


Fig. 6: Top 4 results by (b) Eitz *et al.* [12], (c) Cao *et al.* [8] and (d) our system on a 1.2 million image dataset for some sample sketches (a).

for evaluation here. Although standard sketch-to-image matching algorithms for Object Detection that perform time consuming online processing would perform better than our approach on this small dataset, such comparison would be unfair since the objectives are different. Hence, we compare only against TENSOR [12] and EI [8]. In this dataset, we measure the percentage of positive retrievals in top 20 retrieved results which also gives an idea of recall of various approaches since the number of true positives is fixed. Table 2 shows the best, worst and average performance for the different sketches in a category (as for the previous dataset) while Fig 7 details the performance of our approach for different categories (more results in the supplementary section). It can be seen that our method performs much better than other methods on this dataset as well. The performance on ETHZ models [16] is better than the average performance, which is expected since those sketches are computer generated and are therefore cleaner.

Table 2: Comparison of % of true positive retrievals in top 20 using our 63 sketches and ETHZ models [16] on ETHZ extended dataset [34]

Method	Our Sketches			ETHZ
	Best	Worst	Avg	Models [16]
TENSOR [12]	20	8.6	13.8	13.6
EI [8]	49.3	5	23.5	27.9
CS	51.4	14.3	35.3	33.6
CS+GC	**53.6**	**28.6**	**38.8**	**49.3**

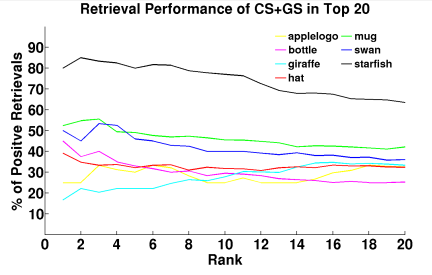


Fig. 7: Retrieval performance of the proposed algorithm (CS+GC) for different categories of the ETHZ extended shape dataset [34]

7 Conclusion

We have proposed a sketch-based fast image retrieval approach for large datasets that, unlike any prior work, handles similarity transformations and deformations of the object shape. This is achieved by preprocessing all the database images in which the essential shape information is extracted using multiple but a small number of variable length descriptors from contour chains. These descriptors are efficiently matched using a Dynamic Programming-based *approximate substring* matching algorithm that is used for chain indexing and then efficiently searching for matching image chains in a hierarchical k-medoids tree structure. Geometric verification on candidate images helps reducing the false positives. Extensive experiments performed on a 1.2 million Image Database indicate significant performance improvement over other existing methods. Our method, augmented by other techniques, could also be used for tagging images in an offline fashion or for improving online results.

References

1. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33(5), 898–

916 (2011) 3
2. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: ACM-SIAM Symposium on Discrete algorithms. pp. 1027–1035 (2007) 8
3. Bagon, S., Brostovski, O., Galun, M., Irani, M.: Detecting and sketching the common. In: Computer Vision and Pattern Recognition, IEEE Conference on. pp. 33–40. IEEE (2010) 1
4. Bai, X., Latecki, L.J.: Path similarity skeleton graph matching. Pattern Analysis and Machine Intelligence, IEEE Transactions on 30(7), 1282–1292 (2008) 2
5. Basri, R., Costa, L., Geiger, D., Jacobs, D.: Determining the similarity of deformable shapes. Vision Research 38(15), 2365–2385 (1998) 3
6. Bozas, K., Izquierdo, E.: Large scale sketch based image retrieval using patch hashing. In: Advances in Visual Computing, pp. 210–219. Springer (2012) 2
7. Canny, J.: A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on (6), 679–698 (1986) 3
8. Cao, Y., Wang, C., Zhang, L., Zhang, L.: Edgel index for large-scale sketch-based image search. In: Computer Vision and Pattern Recognition, IEEE Conference on. pp. 761–768. IEEE (2011) 2, 11, 13, 14
9. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Third Edition. The MIT Press (2009) 4, 6
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, IEEE Conference on. pp. 248–255. IEEE (2009) 10
11. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Transactions on Graphics (Proc. SIGGRAPH) 31(4), 44 (2012) 10
12. Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: A descriptor for large scale image retrieval based on sketched feature lines. In: Eurographics Symposium on Sketch-Based Interfaces and Modeling. pp. 29–38 (2009) 11, 13, 14
13. Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: An evaluation of descriptors for large-scale image retrieval from sketched feature lines. Computers & Graphics 34(5), 482–498 (2010) 2
14. Felzenszwalb, P.F., Schwartz, J.D.: Hierarchical matching of deformable shapes. In: Computer Vision and Pattern Recognition, IEEE Conference on. pp. 1–8. IEEE (2007) 2
15. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on 30(1), 36–51 (2008) 1, 4
16. Ferrari, V., Tuytelaars, T., Van Gool, L.: Object detection by contour segment networks. In: European Conference on Computer Vision, pp. 14–28. Springer (2006) 2, 14
17. Huiskes, M.J., Lew, M.S.: The MIR flickr retrieval evaluation. In: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval. pp. 39–43. ACM (2008) 10
18. Kokkinos, I., Yuille, A.: Inference and learning with hierarchical shape models. International Journal of Computer Vision 93(2), 201–225 (2011) 2
19. Latecki, L.J., Lakamper, R., Eckhardt, T.: Shape descriptors for non-rigid shapes with a single closed contour. In: Computer Vision and Pattern Recognition, IEEE Conference on. pp. 424–429. IEEE (2000) 2
20. Lee, Y.J., Grauman, K.: Shape discovery from unlabeled image collections. In: Computer Vision and Pattern Recognition, IEEE Conference on. pp. 2254–2261. IEEE (2009) 1
21. Lee, Y.J., Zitnick, C.L., Cohen, M.F.: Shadowdraw: real-time user guidance for freehand drawing. In: ACM Transactions on Graphics (Proc. SIGGRAPH). vol. 30, p. 27. ACM (2011) 1

22. Ma, T., Latecki, L.J.: From partial shape matching through local deformation to robust global shape similarity for object detection. In: Computer Vision and Pattern Recognition, IEEE Conference on. pp. 1441–1448. IEEE (2011) 2
23. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval, vol. 1. Cambridge University Press (2008) 11
24. Marvaniya, S., Bhattacharjee, S., Manickavasagam, V., Mittal, A.: Drawing an automatic sketch of deformable objects using only a few images. In: European Conference on Computer Vision. Workshops and Demonstrations. pp. 63–72. Springer (2012) 1
25. Meltzer, J., Soatto, S.: Edge descriptors for robust wide-baseline correspondence. In: Computer Vision and Pattern Recognition, IEEE Conference on. pp. 1–8. IEEE (2008) 8
26. Muja, M., Lowe, D.G.: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In: International Conference on Computer Vision Theory and Application (VISSAPP). pp. 331–340. INSTICC Press (2009) 8
27. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Computer Vision and Pattern Recognition, IEEE Conference on. vol. 2, pp. 2161–2168. IEEE (2006) 8
28. Opelt, A., Pinz, A., Zisserman, A.: Learning an alphabet of shape and appearance for multi-class object detection. International Journal of Computer Vision 80(1), 16–44 (2008) 8
29. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Computer Vision and Pattern Recognition, IEEE Conference on. pp. 1–8. IEEE (2007) 9
30. Ravishankar, S., Jain, A., Mittal, A.: Multi-stage contour based detection of deformable objects. In: European Conference on Computer Vision, pp. 483–496. Springer (2008) 2
31. Riemenschneider, H., Donoser, M., Bischof, H.: Using partial edge contour matches for efficient object category localization. In: European Conference on Computer Vision, pp. 29–42. Springer (2010) 2, 5
32. Riemenschneider, H., Donoser, M., Bischof, H.: Image retrieval by shape-focused sketching of objects. In: 16th Computer Vision Winter Workshop. p. 35 (2011) 2, 13
33. Sattler, T., Leibe, B., Kobbelt, L.: SCRAMSAC: Improving RANSAC's efficiency with a spatial consistency filter. In: Computer Vision and Pattern Recognition, IEEE Conference on. pp. 2090–2097. IEEE (2009) 9
34. Schindler, K., Suter, D.: Object detection by global contour shape. Pattern Recognition 41(12), 3736–3748 (2008) 13, 14
35. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting image databases from the web. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33(4), 754–766 (2011) 1
36. Scott, C., Nowak, R.: Robust contour matching via the order-preserving assignment problem. Image Processing, IEEE Transactions on 15(7), 1831–1838 (2006) 2
37. Sigurbjörnsson, B., Van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: Proceedings of the 17th international conference on World Wide Web. pp. 327–336. ACM (2008) 1
38. SketchMatch: http://research.microsoft.com/en-us/projects/sketchmatch/ 11, 13
39. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. Journal of molecular biology 147(1), 195–197 (1981) 8
40. Toyama, K., Blake, A.: Probabilistic tracking with exemplars in a metric space. International Journal of Computer Vision 48(1), 9–19 (2002) 8
41. Tsai, S.S., Chen, D., Takacs, G., Chandrasekhar, V., Vedantham, R., Grzeszczuk, R., Girod, B.: Fast geometric re-ranking for image-based retrieval. In: Image Processing, 17th IEEE International Conference on. pp. 1029–1032. IEEE (2010) 9
42. Vlachos, M., Vagena, Z., Yu, P.S., Athitsos, V.: Rotation invariant indexing of shapes and line drawings. In: Proceedings of the 14th ACM international conference on Information and knowledge management. pp. 131–138. ACM (2005) 8

43. Yarlagadda, P., Ommer, B.: From meaningful contours to discriminative object shape. In: European Conference on Computer Vision, pp. 766–779. Springer (2012) 2
44. Zhou, R., Chen, L., Zhang, L.: Sketch-based image retrieval on a large scale database. In: Proceedings of the 20th ACM international conference on Multimedia. pp. 973–976. ACM (2012) 2, 13
45. Zhu, Q., Song, G., Shi, J.: Untangling cycles for contour grouping. In: Computer Vision, 2007. IEEE 11th International Conference on. pp. 1–8. IEEE (2007) 3