Multi-Modal Image Retrieval for Complex Queries using Small Codes*

Behjat Siddiquie[1], Brandyn White[2], Abhishek Sharma[2], Larry S. Davis[2]
[1]SRI International, 201 Washington Road, Princeton, NJ, USA
[2]Dept. of Computer Science, University of Maryland, College Park, MD, USA
behjat.siddiquie@sri.com, {bwhite, bhokaal, lsd}@cs.umd.edu

ABSTRACT

We propose a unified framework for image retrieval capable of handling complex and descriptive queries of multiple modalities in a scalable manner. A novel aspect of our approach is that it supports query specification in terms of objects, attributes and spatial relationships, thereby allowing for substantially more complex and descriptive queries. We allow these complex queries to be specified in three different modalities - images, sketches and structured textual descriptions. Furthermore, we propose a unique multi-modal hashing algorithm capable of mapping queries of different modalities to the same binary representation, enabling efficient and scalable image retrieval based on multi-modal queries. Extensive experimental evaluation shows that our approach outperforms the state-of-the-art image retrieval and hashing techniques on the MSRC and SUN09 datasets by about 100%, while the performance on a dataset of 1M images, from Flickr, demonstrates its scalability.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models; H.2.8 [**Database Applications**]: Image databases

General Terms

Algorithms, Design

Keywords

Image Retrieval, Multimedia, Image Search, Hashing, Multi-modal, Semantic Retrieval

1. INTRODUCTION

The amount of visual data such as images and videos available over the web has increased exponentially over the last few years and there is a need for developing techniques that are capable of efficiently organizing, searching and exploiting these massive collections. In order to effectively do so, a system, apart from being able to answer simple classification based questions such as whether a specific object is present(or absent) in an image, should also be capable of searching and organizing images based on more complex descriptive questions. To this end, there have been major advances in the field of image retrieval in the last few years. For example, image retrieval has progressed from retrieving images based on single label queries [1], [7] to multi-label queries [9] [10], [16], [30] and structured queries [19].

In this work, our goal is to enable a user to search for images based on complex and descriptive queries that consist of *objects*, *attributes* - that describe properties of objects; and *relationships* - that specify the relative configuration between pairs of objects. For example, we would like to search for images based on a query like *"red car to the left of a yellow car"*. Unlike, current retrieval approaches which can deal with only single label or multi-label queries, our work allows users to search for images/scenes based on very specific and complex properties. To the best of our knowledge, none of the existing image retrieval approaches can handle such complex and descriptive queries in a scalable manner.

In image retrieval, queries are typically specified using an image, a sketch or a textual description and almost all current image retrieval approaches fall into one of these three categories. We integrate these approaches by proposing a joint framework that allows the queries to be specified in any of these three modalities - i.e. images, sketches or text

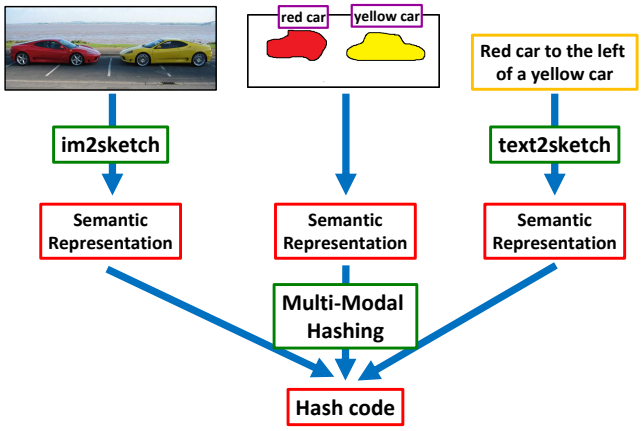


Figure 1: **Overview:** Our proposed multi-modal image retrieval framework. We convert queries of different modalities (text, sketch and images) into a common semantic representation. The semantic representations are then mapped to compact binary codes using a novel multi-modal hashing approach.

*The first two authors contributed equally.

ICMR'14, Apr 01-04 2014, Glasgow, United Kingdom

(Fig. 1). In the case of image based queries, the user provides an image as a query and the goal is to retrieve images that are semantically similar. The query image implicitly encodes the objects present in the image, their attributes and the relationships between them. Unlike several other retrieval approaches [32], [23], [35], [13], we focus on semantic similarity rather than visual similarity. In a sketch based query, the user draws a very rough sketch (a set of regions) and explicitly labels each region with object names and/or attributes, while the locations and spatial relationships are implicitly encoded. Finally in text based queries, the objects, attributes and the relationships are explicitly specified by the user in a pre-defined structured format. However, building a large scale joint retrieval framework for multiple query modalities necessitates the ability to perform an efficient nearest neighbor search from a query of each modality to the elements in the database. Unfortunately, none of the existing hashing approaches can be used for this purpose. We accomplish this, by proposing a novel Multi-Modal hashing approach capable of hashing queries and database elements of different modalities to the same hash code. In our case the modalities correspond to queries in the form of images, sketches and text (Fig. 1). Our multi-modal hashing approach consists of a Partial Least Squares (PLS) based framework [27], which maps queries from multiple modalities to points in a common linear subspace which are then converted into compact binary strings by a learned similarity preserving mapping. This enables scalable and efficient image retrieval from queries based on multiple modalities. See Fig. 1 for an overview.

There are three main contributions of our work: **1)** We propose an approach for image retrieval based on complex descriptive queries that consist of objects, attributes and relationships. The ability to define a query by employing these constructs gives users more expressive power and enables them to search for very specific images/scenes. **2)** We support query specification in three different modalities - images, sketches and text. Each of these modalities have their own pros and cons - for example, an image query might be the most informative, but the user might not always have a query image; a text based query might not be specific enough, but is easy to compose; a sketch based query might require a special interface. However, when equipped with the ability to search based on any of these modalities a user can choose the one that is the most appropriate for the situation at hand. **3)** Finally, to support querying based on multiple query modalities, we propose a novel multi-modal hashing approach that can map queries of different modalities to the same hash code, enabling efficient and scalable image retrieval based on multi-modal queries. While these three contributions from the key ingredients of our unified and scalable framework for multi-modal image retrieval, they also lay the foundations of a general multimedia retrieval framework with several advantages over existing systems, *e.g.* the ability to query a video database using a complex query comprising multi-modal (audio, video, text) information.

2. RELATED WORK

Image retrieval can be divided into three categories - image based retrieval, text based retrieval and sketch based retrieval - based on the modality of the query. In this work we propose an approach that integrates these methods within a single joint framework. We now briefly describe relevant work in each of these image retrieval categories as well as relate and contrast our proposed approach to them.

In image based retrieval, the user provides a query in the form of an image and the goal is to retrieve similar images from a large database. A popular approach [32], involves utilizing a global image representation such as GIST or Bag-of-Words (BoW). Augmenting a BoW representation by incorporating spatial information has shown to improve retrieval results significantly [23], [35]. Further improvements have been obtained by aggregating local descriptors [13] or by using Fisher kernels [24] as an alternative to BoW. However, a common drawback of these approaches is that, while they perform well at retrieving images that are visually very similar to the query image (e.g. images of the same scene from a slightly different viewpoint), they can often retrieve images that are semantically very different. In contrast, we focus on retrieving images that are semantically similar to the query image. This is facilitated by employing an intermediate representation that encodes the semantic content such as the objects, their attributes and the spatial relationships between them.

Text based image retrieval entails retrieving images that are relevant to a text query, which in its simplest form could be a single word representing an object category. Early work in this area includes [1], [7]. Later work such as [9], [10], [16], [30], allowed for image retrieval based on multi-word queries, where a word could be an object or a concept as in [9], [10] or an attribute as in [16],[30]. Our work further builds upon these methods by providing a user the ability to retrieve images based on significantly more descriptive text based queries that consist of objects, attributes that provide additional descriptions of the objects and relationships between pairs of objects. While recent approaches such as [15], [6], do look at the problem of retrieving a relevant image given a sentence, they primarily focus on the reverse problem - i.e. producing a semantically and syntactically meaningful description of a given image.

Sketch based retrieval involves the user drawing a sketch of a scene and using it to search for images that have similar properties. An advantage of a sketch based query over text based queries is that it implicitly encodes the scale and relative spatial locations of the objects within an image. Initial approaches in sketch based retrieval include [12], [31], where the query was a color based sketch and the aim was to retrieve images that had a similar spatial distribution of colors. In [34], a sketch-like representation of concepts called a *concept map*, is used to search for relevant images. In [2], Cao et al. proposed an efficient approach for real-time image retrieval from a large database. However, their approach primarily relies on contours and hence uses information complementary to our method. In the graphics community, people have looked at the problem of composing (rather than retrieving) an image from multiple images given a sketch [3].

Image retrieval based on each of these modalities have different pros and cons and hence we propose a single framework for image retrieval capable of handling multi-modal queries. While the recently proposed Exemplar-SVM based approach of Shrivastava et al. [29] can match images across domains (images, paintings, sketches), a major drawback of their approach is its scalability - it requires 3 min. to search through a database of 10000 images on a 200-node cluster, making it highly impractical for any kind of online image retrieval. On the other hand, by virtue of representing queries and database elements using compact binary strings, we can search a database of a million images in 0.5 seconds on a single core.

Performing image retrieval in a large scale setting requires scalable approaches for compactly storing the database im-

ages in memory and efficiently searching for images relevant to the query in real-time. For example, storing 50M images using a GIST based representation (960D floats) would require 192GB of memory, but when represented using a 256 bit hash code, the memory requirements drop to a manageable 1.6GB. A popular hashing approach consists of employing locality sensitive hashing (LSH) [5], which uses random projections to map the data into a binary code, while preserving the input-space distances in the Hamming space. Given a query, relevant images can be efficiently retrieved by computing the Hamming distance between the query and database images. Several recent approaches have also attempted to use the underlying data distribution to compute more compact codes [26], [25], [14], [33], [8]. However, these approaches are only applicable to single-modality queries. Hence, we propose a novel multi-modal hashing approach which builds upon [8] and allows multiple representations (modalities) to be mapped to the same binary code. While the work of Kumar and Udupa [17] is similar to ours, as it can handle multi-modal queries, we believe that our approach is superior for two reasons. Firstly, when no prior cross-modal information is available, [17] reduces to a CCA embedding. In contrast, our approach is based on a variant of Partial Least Squares (PLS) that has been shown to be superior to CCA [27] in the presence of noise. Moreover, we also apply the Iterative Quantization technique [8] to the PLS embedding, further improving retrieval performance. Secondly, we evaluate our approach on modalities(image, sketches and text) that are far more diverse compared to the modalities (multilingual corpora) in [17].

3. APPROACH

3.1 Query Representation

We first define a sketch based query. As illustrated in Fig. 2, a sketch consists of a set of regions drawn by the user, with each region being labeled by an object class. A sketch can be thought of as a dense label map, where the unlabeled portions of the sketch correspond to the background class. Each region can also be labeled by multiple attributes, that could specify its color and texture. We use sketches as our primary form of representation, and convert image and text based queries into sketches. The principal advantage of having a single representation is that it enables us to have a unified framework for multi-modal queries, instead of having to build a separate pipeline for each query modality. We choose a sketch based representation over images or text because a) When compared to images, sketches are more semantically meaningful and encode a query using human describable constructs such as objects and attributes. b) When compared to text based queries, sketches implicitly encode the scales and locations of different objects and the spatial relationships between them.

We convert sketches into a semantic representation that permits easy encoding of the spatial relationships between the objects in an image. The sketches are converted into C_o binary masks representing each object category (whether it appears in the sketch or not) and C_a masks representing each attribute. The binary mask corresponding to the object o_k has value 1 at pixel (i, j) if the sketch contains the corresponding object class at pixel (i, j) and similarly for attributes. These binary masks are then resized to $d \times d$, leading to each sketch being represented by a vector of dimension $(C_o + C_a)d^2$. The conversion of a sketch into the semantic representation is illustrated in Fig. 2. We compare the semantic similarity between two sketches based on the L_2 distance between their corresponding vector representations and based on the Spatial Pyramid Match [20] similarity between the corresponding label maps as done in [32].

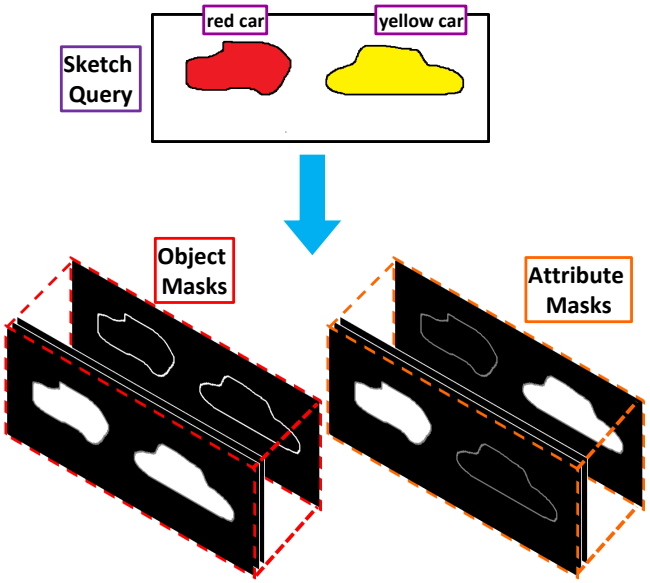


Figure 2: **Semantic Representation:** A sketch based query is converted into the semantic representation, which is a concatenation of the binary mask corresponding to each object and each attribute.

There are two main advantages of the proposed Semantic Representation over raw feature representations such as GIST. Firstly it explicitly encodes the scales and locations of different objects and the spatial relationships between them. Secondly, as validated by our experiments, the Semantic Representation compactly encodes the semantic content within an image. Due to these two properties the Semantic Representation is appropriate for supporting complex semantic queries in a large scale setting. Our proposed semantic representation of objects and attributes bears some resemblance to the "Object Bank" [21] representation of Li et al. However, there is an important difference - while they use sparsity algorithms to tractably exploit the "Object Bank" representation, we instead leverage existing work on efficient hashing approaches to enable application of our representation to large scale problems.

3.2 image2semantic

In order to convert an image into the semantic representation, we semantically segment the image by assigning an object label to each pixel. The segmentation is performed using Semantic Texton Forests (STF) [28]. We choose STFs over other semantic segmentation approaches primarily for their speed. Given a query image, STFs enable fast conversion of the image to the semantic feature representation, which is critical for real-time image retrieval. Training the STF involves learning two levels of randomized decision trees - at the first level a randomized decision tree is learned to cluster image patches into textons, where each leaf node of the tree represents a texton. The second level involves learning multiple decision trees that take into account the layout and the spatial distribution of the textons to assign an object label to each pixel. During the test phase, the image patch surrounding each pixel is simply passed down each tree and the results of multiple trees are averaged to obtain its object label. We direct the reader to [28] for further details of the approach. We also train STF based attribute classifiers

and segment the image based on attributes. By semantically segmenting the image using STFs, we obtain the class and attribute label assignments for each pixel, which we then convert into the semantic representation, as described in Section 3.1.

3.3 text2semantic

We now describe our approach for generating a set of plausible semantic sketches relevant to a text based query. We assume that our text query consists of a set of objects, with each object being described by zero or more attributes and a set of zero or more pairwise relationships between each pair of objects. An example of such a query is *"red car to the left of a yellow car"*. We also assume that the text query has been parsed into its constituent components (see the supplementary material for details).

Corresponding to each object, we generate a large number of candidate bounding boxes. A bounding box X is defined by its scale (s_x, s_y) and location (x, y). For each object o_i that is part of the query, we generate a set of bounding boxes $\mathcal{X}_{o_i}$ by importance sampling the distribution of the object class in the training data and assign each bounding box a probability $P(X|c_i)$ (where c_i is the class of o_i) based on the training distribution. A candidate sketch of the query can be created by simply choosing one bounding box corresponding to each object o_i. However, to create semantically plausible sketches, we use the spatial relationship likelihoods between pairs of object categories, learned from the training data, as well as the specific inter-object relationships provided by the user in the query to generate the set of most likely candidate sketches. We define the likelihood of a sketch as:

$$P(X_{o_1}, X_{o_2}, ..|o_1, o_2, ..) \propto \prod_i P(X_{o_i}|c_i) \prod_{(j,k)} P(X_{o_j} - X_{o_k}|c_j, c_k) \quad (1)$$

where X_{o_i} denotes the bounding box corresponding to object o_i, c_i is the object category of object o_i and $X_{o_j} - X_{o_k}$ represents the difference in the location and scale of the bounding boxes X_{o_j} and X_{o_k}. The first term in the equation represents the likelihood of an object of class c_i having a bounding box X_{o_i}, while the second term restricts the bounding boxes belonging to the pair of classes c_j and c_k from having arbitrary relative locations and scales. The second term is further decomposed into its constituent components (s_x, s_y, x, y) as the joint distribution is very sparse:

$$P(\Delta X_{o_{jk}}|c_j, c_k) = \prod P(\Delta x_{o_{jk}}|c_j, c_k)P(\Delta y_{o_{jk}}|c_j, c_k) \; P(\Delta s_{x_{o_{jk}}}|c_j, c_k)P(\Delta s_{y_{o_{jk}}}|c_j, c_k) \quad (2)$$

where $\Delta X_{o_{jk}}$ represents $X_{o_j} - X_{o_k}$ for brevity, and similarly for the individual components.

The contextual relationship model (Eq. 2) is similar to the one used by [11]. However, unlike [11], where the spatial relationships are binary, we employ a set of discrete bins to capture the degree of separation and relative scales between the objects within an image. We also incorporate information about the spatial relationships between a pair of object classes, contained within the query, into the model. For example, if the query states that object o_j is above object o_k, we can utilize this information to set $P(y_{o_j} - y_{o_k} > 0) = 0$ and then renormalize $P(y_{o_j} - y_{o_k}|c_j, c_k)$, which helps enforce the relationship constraint.

We generate the set of N(=25) most likely candidate sketches based on the likelihood model (Eqn. 2) using the technique proposed by Park and Ramanan [22], which embeds a form of non-maximal suppression within a sequential loopy belief-propagation algorithm and results in a relatively varied, but at the same time highly likely, set of candidate sketches. Note that a sketch is generated based on the likelihoods of the object classes alone, the attributes of each object are then assigned to the corresponding bounding box in the generated sketch. The candidate sketches for some text queries are shown in Fig. 3. Here, the unary and pairwise likelihoods are learned from the SUN09 dataset [4].

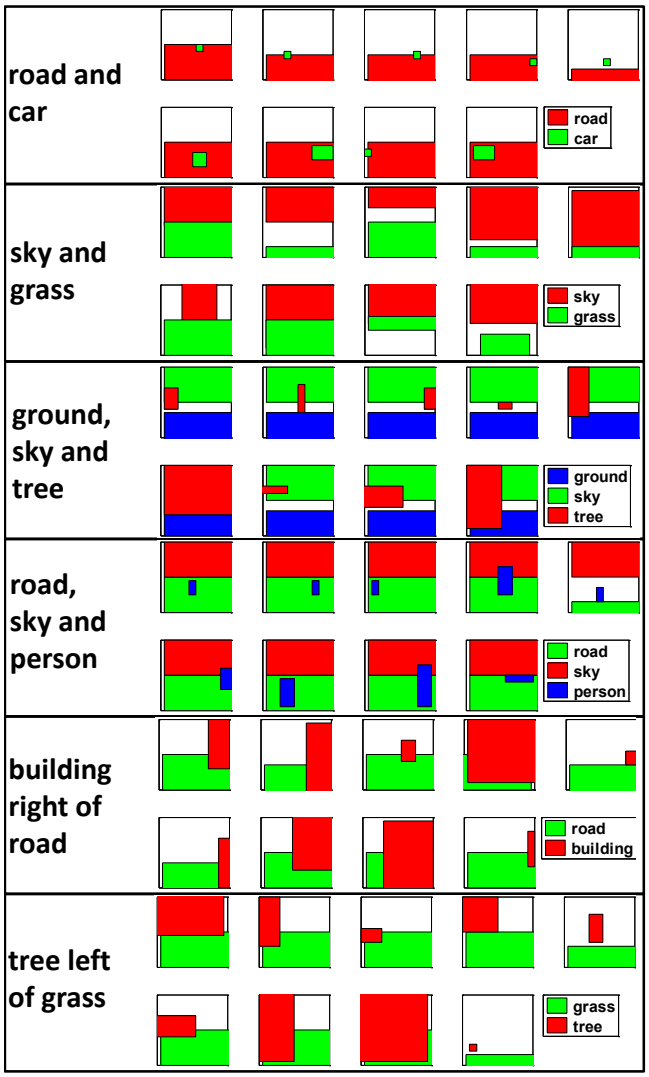


Figure 3: **text2semantic:** The top $k(=9)$ sketches for different text based queries consisting of two or three objects with and without relationship information.

3.4 Multi-Modal Hashing

We are given a set of n data points, for which we have two different modalities $\mathcal{X} = \{\mathbf{x}_i\}, \; i = 1 \ldots n, \; \mathbf{x}_i \in \mathcal{R}^{D_x}$ and $\mathcal{Y} = \{\mathbf{y}_i\}, \; i = 1 \ldots n, \; \mathbf{y}_i \in \mathcal{R}^{D_y}$. For example, in our case, $\mathcal{X}$ could consist of the semantic representations computed from the images and $\mathcal{Y}$ could be the representations from the corresponding sketches. In general, we can have more than two modalities. Our goal is to learn projection matrices W_x and W_y that can convert the data into a compact binary code, where the binary code $h_{\mathbf{x_i}}$ for the feature vector $\mathbf{x_i}$ is computed as $h_{\mathbf{x_i}} = \text{sgn}(\mathbf{x_i}W_x)$. Like most other hashing approaches, we want to learn W_x (and

similarly W_y) that assigns the same binary codes $h_{\mathbf{x_i}}$ and $h_{\mathbf{x_j}}$ to data points $\mathbf{x_i}$ and $\mathbf{x_j}$ that are very similar. However, we also have the additional constraint that an image $\mathbf{x_i}$, and a sketch $\mathbf{y_j}$, which are semantically similar, should be mapped to similar binary codes $h_{\mathbf{x_i}}$ and $h_{\mathbf{y_j}}$ by W_x and W_y respectively. Motivated by the approach of [8], we adopt a two stage procedure - the first stage involves projecting different modalities of the data to a common low dimensional linear subspace, while the second stage consists of applying an orthogonal transformation to the linear subspace so as to minimize the quantization error when mapping this linear subspace to a binary code.

We adopt a Partial Least Squares (PLS) based approach to map different modalities of the data into a common latent linear subspace. We employ the PLS variant used in [27], which works by identifying linear projections such that the covariance between the two modalities of the data in the projected space is maximized. Let X be an $(n \times D_x)$ matrix containing one modality of the training data $\mathcal{X}$, and Y be an $(n \times D_y)$ matrix containing the corresponding instances from a different modality of the training data $\mathcal{Y}$. PLS decomposes X and Y such that:

$$\begin{aligned} X &= TP^T + E \\ Y &= UQ^T + F \\ U &= TD + H \end{aligned} \tag{3}$$

where T and U are $(n \times p)$ matrices containing the p extracted latent vectors, the $(D_x \times p)$ matrix P and the $(D_y \times p)$ matrix Q represent the loadings and the $(n \times D_x)$ matrix E, the $(n \times D_y)$ matrix F and the $(n \times p)$ matrix H are the residuals. D is a $p \times p$ matrix that relates the latent scores of X and Y. The PLS method iteratively constructs projection vectors $W_x = \{w_{x1}, w_{x2}, \ldots, w_{xp}\}$ and $W_y = \{w_{y1}, w_{y2}, \ldots, w_{yp}\}$ in a greedy manner. Each stage of the iterative process, involves computing:

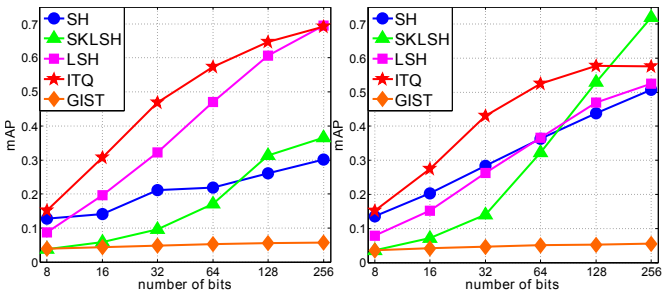
$$[\text{cov}(t_i, u_i)]^2 = \max_{|w_{xi}|=1, |w_{yi}|=1} [\text{cov}(Xw_{xi}, Yw_{yi})]^2 \tag{4}$$

where t_i and u_i are the i*th* columns of the matrices T and U respectively, and $\text{cov}(t_i, u_i)$ is the sample covariance between latent vectors t_i and u_i. This process is repeated until the desired number of latent vectors p, have been determined. One can alternatively use CCA instead of PLS, however we found that PLS outperformed CCA, a conclusion also supported by [27].

PLS produces the projection matrices W_x and W_y that project different modalities of the data into a common orthogonal basis. The first few principal directions computed by PLS contain most of the covariance, hence encoding each direction with a single bit distorts the Hamming distance, resulting in a poor retrieval performance. In [8], the authors show that this problem can be overcome by computing a rotated projection matrix $\tilde{W}_x = W_x R$, where R is a randomly generated $(p \times p)$ orthogonal rotation matrix. Doing so distributes the information content in each direction in a more balanced manner, leading to the Hamming distance in the binary space better approximating the Euclidean distance in the joint subspace induced by PLS. They also propose a more principled and effective approach called Iterative Quantization (ITQ), which involves an iterative optimization procedure to compute the optimal rotation matrix R, that minimizes the quantization error $\mathcal{Q}$, given by:

$$\mathcal{Q}(H, R) = ||H - XW_xR||_F^2 \tag{5}$$

where H is the $(n \times p)$ binary code matrix representing X and $||.||_F$ represents the Frobenius norm. Further details of the optimization procedure can be found in [8]. The effectiveness of the iterative quantization procedure for improving hashing efficiency by minimizing the quantization error has been demonstrated in [8]. Hence, we employ ITQ to modify the joint linear subspace for the multiple modalities produced by PLS and learn more efficient binary codes. The final projection matrices are given by $\tilde{W}_x = W_x R$ and $\tilde{W}_y = W_y R$, where R is obtained from (5).



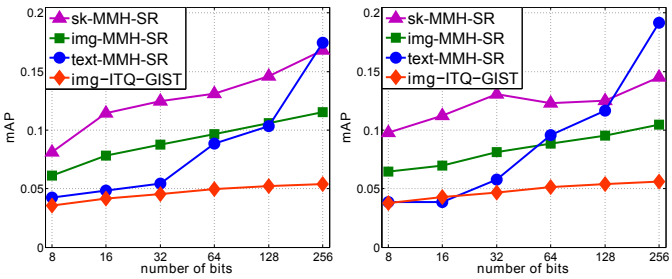
(a) SUN09 - L2 dist. (b) SUN09 - SPM dist.

Figure 4: **Single-Modality Hashing:** Performance of hashing algorithms (SH [33], SKLSH [25], LSH [5], ITQ [8]) using our semantic representation vs GIST.

4. EXPERIMENTS AND RESULTS

4.1 Semantic Representation

We first show that our semantic representation can be efficiently compressed using current hashing techniques. We perform experiments on the standard MSRC dataset and the SUN09 [4] dataset. The results on the MSRC dataset are contained in the supplementary material. The SUN09 dataset consists of 4367 training and 4317 test images and we only use the 21 most frequent object categories. Our evaluation protocol is similar to [8]. The ground truth segmentations of the images are treated as sketches and are converted into the semantic representations. The training images are utilized for learning the parameters of the hashing, image2semantic and text2semantic algorithms. The test images are divided into two equal sets, the semantic representations from the first set are used as queries, while those from the other set are used to form the database against which the queries are performed. For each query the average distance to the k-*th* nearest neighbor, is used as a threshold to determine whether a retrieved image is a true positive. We set $k = 20$ in case of the SUN09 dataset. We use the Euclidean (L2) distance as well as the Spatial Pyramid Match (SPM) distance for evaluation. Note that for this experiment there is just a single mode which is the semantic representation computed from ground truth segmentations, and hence we are able to use standard hashing algorithms. Our aim here is to simply show that the semantic representation, which is a 13125D binary code that implicitly encodes location/scale and relationship information, can be quantized to a small number of bits. The results, shown in Fig. 4, plot the mean Average Precision (mAP) as a function of the number of bits. We can see that the semantic representation can be effectively quantized to 128/256 bits while still having a good retrieval performance, in case of both the SPM and L2 distances. We also compare the performance of our semantic representation against GIST features quantized using ITQ [8]. While this comparison is not completely fair, as the semantic representations have been computed from the ground truth segmentations, the large difference in performance shows that by employing a semantic segmen-



(a) SUN09 - L2 dist. (b) SUN09 - SPM dist.

Figure 5: **Multi-Modal Hashing:** Retrieval performance for image(img-MMH-SR), sketch(sk-MMH-SR) and text(text-MMH-SR) based queries using our Multi-Modal Hashing(MMH) Approach which uses the Semantic Representation(SR) against a GIST feature representation followed by ITQ hashing [8](img-ITQ-GIST).

tation algorithm such as [28], we should be able to perform much better than GIST, which we show next.

4.2 Multi-Modal Hashing

We now evaluate our multi-modal hashing algorithm. We follow the same protocol that was used in the previous experiment. However, we now have three query modalities - *images*, *sketches* and *text*. The true positives are determined based on the distances between the ground truth semantic representations of the queries and the database images.

We evaluate three scenarios 1) **Sketch Queries:** Here the ground truth segmentations of the test images are used as sketch based queries, while the training images segmented using STF [28] are used as the database. We would like to point out that the ground truth segmentations of the SUN09 dataset are quite coarse with large background regions and therefore they approximate hand drawn sketches to some extent. 2) **Image Queries:** Here the semantic representations computed from the test(training) images after segmentation by STF [28] are used as queries(database elements) and no ground truth information is used. 3) **Text Queries:** Here, our queries consist of 93 different textual descriptions comprising two or three different objects, with and without relationship information. These text queries are converted into sketches using the "text2semantic' technique(Fig. 3). The baseline consists of a GIST based feature representation, the most widely used feature representation in large scale image retrieval [25], [33],[8], followed by ITQ hashing [8].

The results for the SUN09 dataset (5a, 5b), demonstrate that we are able to improve substantially over GIST, with the mAP score of our approach for image based queries, using 256 bit hash codes, being 113% and 89% better than that of GIST for L2 and Spatial Pyramid distances respectively. The performance of sketch and text based queries is even better. We chose Semantic Texton Forests [28] primarily for their speed; however using a more accurate (albeit much slower) semantic segmentation technique such as [18] would further improve the performance of our approach.

4.3 Multi-Modal Hashing vs Single-Modal Hashing

Our proposed framework involves converting each query modality to a common semantic representation and hence multi-modal hashing might seem unnecessary. However, the semantic representations obtained from the three modalities differ significantly and hence we expect multi-modal hashing to substantially improve hashing performance. We tested this hypothesis by comparing our Multi-Modal Hashing technique against a Single-Modal Hashing approach (ITQ)[8], on the SUN09 dataset using the L2 distance. The results in Table1 show the retrieval performance (mAP) of Multi-Modal and Single-Modal Hashing approaches for different query modalities using 256 bit codes. A more comprehensive evaluation against other Single-Modal Hashing techniques is provided in the supplementary material. The results clearly demonstrate that the Multi-Modal Hashing significantly improves retrieval performance over Single-Modal hashing in each query modality. The reason is that the Semantic Representations (SR) computed from images or text queries are quite noisy due to the errors introduced through "text2semantic" and "image2semantic". While the original semantic content of different modalities is correlated, the noise between them is independent. Consequently the PLS based approach, which maximizes the covariance between different modalities in the projected subspace, disregards the noise leading to superior performance compared to single-modal hashing approaches. Finally, the results demonstrate that even in the case of a single query modality (images) our proposed Semantic Representation (SR) provides a superior representation to GIST, which is currently the most widely used feature representation in large scale image retrieval tasks.

Table 1: Multi-Modal Hashing vs Single-Modal Hashing - Retrieval performance (mAP).

	Image	Sketch	Text	Image (GIST)
MVH (our)	**0.115**	**0.168**	**0.175**	-
SVH-ITQ [8]	0.084	0.154	0.138	0.054

4.4 Text Queries

Our approach for generating sketches from text images enables us to accommodate text based queries within our multi-modal framework. However, a more natural retrieval approach based on text queries alone, would involve utilizing the semantic segmentation to first identify images that contain the query objects and then filtering these images by verifying whether or not the given objects satisfy the relationships specified in the query. We compare the retrieval accuracy using text based queries of our method (using 256 bits) against such a verification based approach, for 93 different text queries. The verification based approach is applied to the segmentations obtained from STF [28]. The results (Fig. 6) show that the retrieval performance of our approach is close to the performance of the verification based approach, despite the fact that our approach loses information during hashing. Additionally, the verification based approach uses the uncompressed segmentation mask, which occupies at least two orders of magnitude more memory than our hash code. These results demonstrate the effectiveness of our sketch generation algorithm, showing that the generated sketches are relevant as well as diverse. Furthermore, these results also show that our text based retrieval approach is competitive with a verification based approach, while also being much more compact.

An alternative representation for text based retrieval would involve representing an image by a list of bounding box coordinates and the object class id of each detected object. However, this representation would still require 800 bits for an image containing 5 objects, compared to 128/256 bits required by our approach. Moreover during retrieval, such a representation would require complicated graph match-

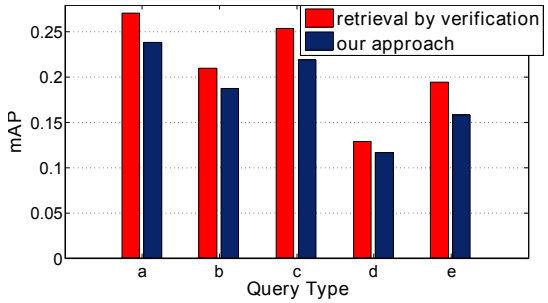


Figure 6: **Text based retrieval:** A comparison of our approach against a verification based approach for text queries. The query types are a) 2 object queries w/o relationship information b) 3 object queries w/o relationship information c) All queries w/o relationship information d) Queries with relationship information e) All queries.

ing algorithms to compute spatial layout similarity between query and database images, which, in our framework, can be accomplished by simply computing the Hamming distance.

4.5 Large Scale Dataset - 1 Million Images

To evaluate the efficiency, scalability and accuracy of our approach on a large scale, we downloaded a set of one million images from Flickr. Using this set of 1M images as the database, we perform image, sketch and text based queries. For image and sketch based queries, we utilize 200 images (image queries) and their corresponding ground truth segmentations (sketch queries) randomly selected from the test images of the SUN09 dataset. For text queries, we use the five most probable sketches generated for each of the 93 text queries that we have used in the previous experiments, resulting in a total of 465 text based queries. In case of image based queries, we use GIST followed by ITQ hashing [8], as a baseline for comparison. We use 256 bit hash codes for each case and evaluate these approaches based on the precision@K. Given a query q (image/text/sketch containing n object classes), we define precision@K $= \sum_i^K score(\text{img}_i)/K$, where $score(\text{img}_i)$ denotes the fraction of the n query object classes contained in img_i, the i-th retrieved image for query q. The advantage of the precision@K metric is that one needs to annotate only the top K images retrieved for each query, which is a small fraction of the entire database. The precision scores are then averaged over all the queries of each type. The parameters for the hashing algorithm as well as the segmentation model are learned from the training images of the SUN09 dataset.

The results are shown in Fig. 7. In case of image based retrieval, our approach performs on par with GIST. Here, the evaluation is based only on the presence or absence of query objects in the retrieved images, disregarding their spatial configuration, due to lack of finer annotations on this dataset. Since our approach explicitly encodes the spatial information, we expect it to substantially outperform GIST when using metrics such as the L2 or the SPM distance, as was the case on the MSRC and SUN09 datasets. We can also observe that sketch queries significantly outperform image queries. This is due to the errors introduced in the segmentation("image2semantic") algorithm, which lead to a distorted semantic representation of an image query. The retrieval performance in the case of text based queries is also comparable to that of GIST, demonstrating that our approach can perform at least as well as GIST for each modality in a large scale setting. Fig. 8 shows some qualitative results.

4.6 System Requirements

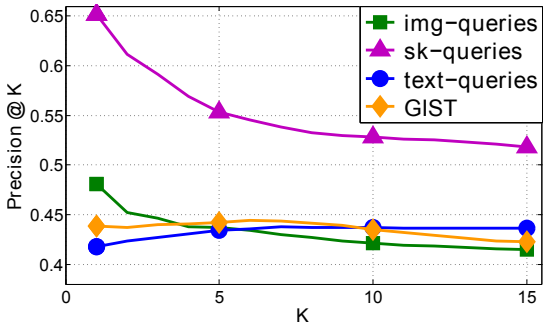


Figure 7: Retrieval performance on the Flickr dataset.

In our setup the overall response time is the sum of computing the Semantic Representation(SR) and using the SR to search the database. In an image query, conversion to the SR involves semantic segmentation of the image [28] which takes about 0.2s on a single core. In a text query, the time required to generate the candidate semantic sketches for 2 and 3 object queries, using [22], is about 0.4s. Since the SR is very similar to a sketch based query, the time required to convert a sketch into the semantic representation is negligible. Once the SR has been computed, the time required to convert it into a 256 bit hash code and search for similar instances in a database of size 1 Million is about 0.5s. Hence the total response time for generating the search results given an image, sketch or text query is less than a second. Also note that these timings are on a single core and we can use a cluster to significantly scale up the dataset while still maintaining sub-second response times.

The memory required for storing 1M images using 256 bit hash codes is about 32MB. Finally, the time required for building the database index for 1M images is about 10 hours on a single core. Note that this Performed offline and does not impact user experience and moreover it can be significantly sped up on a cluster. The supplementary material is available at `http://icmr0118.s3.amazonaws.com/0118_supplementary_material.pdf`.

5. CONCLUSION

We have presented a framework for image retrieval based on complex multi-modal queries. Our framework supports query specification using semantic constructs such as objects, attributes and relationships. Furthermore, our framework allows for queries to be specified in the form of an image, sketch or a text. The effectiveness of our approach has been demonstrated on three different datasets of varying difficulty, including a large scale dataset of 1M images.

6. REFERENCES

[1] T. Berg and D. Forsyth. Animals on the web. *CVPR*, 2006.
[2] Y. Cao, W. Changhu, Z. Liqing, and L. Zhang. Edgel inverted index for large-scale sketch-based image search. *CVPR*, 2011.
[3] T. Chen, M. Cheng, P. Tan, A. Shamir, and S. Hu. Sketch2photo: Internet image montage. *SIGGRAPH ASIA*, 2009.
[4] M. Choi, J. Lim, A. Torralba, and A. Willsky. Exploiting hierarchical context on a large database of object categories. *CVPR*, 2010.
[5] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. *SOCG*, 2004.
[6] A. Farhadi and et. al. Every picture tells a story: Generating sentences for images. *ECCV*, 2010.
[7] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman.

Learning object categories using google's image search. *ICCV*, 2005.
[8] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. *CVPR*, 2011.
[9] D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *PAMI*, 2008.
[10] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. *ICCV*, 2009.
[11] A. Gupta and L. S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. *CVPR*, 2008.
[12] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multi resolution image querying. *ACM SIGGRAPH*, 1995.
[13] H. Jegou, M. Douze, and C. Schmid. Aggregating local descriptors into a compact image representation. *CVPR*, 2010.
[14] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. *ICCV*, 2009.
[15] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating simple image descriptions. *CVPR*, 2011.
[16] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. *ECCV*, 2008.
[17] S. Kumar and R. Udupa. Learning hash functions for cross-view similarity search. *IJCAI*, 2011.
[18] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Graph cut based inference with co-occurrence statistics. *ECCV*, 2010.
[19] T. Lan, Y. W. W. Yang, and G. Mori. Image retrieval with structured object queries using latent ranking svm. *ECCV*, 2012.
[20] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006.
[21] L.-J. Li, H. Su, E. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. *NIPS*, 2010.
[22] D. Park and D. Ramanan. N-best maximal decoders for part models. *ICCV*, 2011.
[23] M. Perd'och, O. Chum, and J. Matas. Effcient representation of local geometry for large scale object retrieval. *CVPR*, 2009.
[24] F. Perronnin and et. al. Large-scale image retrieval with compressed fisher vectors. *CVPR*, 2010.
[25] M. Raginsky and S. Lazebnik. Locality sensitive binary codes from shift-invariant kernels. *NIPS*, 2009.
[26] R. Salakhutdinov and G. Hinton. Semantic hashing. *SIGIR*, 2007.
[27] A. Sharma and D. W. Jacobs. Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. *CVPR*, 2011.
[28] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. *CVPR*, 2008.
[29] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. *SIGGRAPH Asia*, 2011.
[30] B. Siddiquie, R. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. *CVPR*, 2011.
[31] J. R. Smith and S.-F. Chang. A fully automated content-based image query system. *ACM Multimedia*, 1996.
[32] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large databases for recognition. *CVPR*, 2008.
[33] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. *NIPS*, 2008.
[34] H. Xu, J. Wang, X.-S. Hua, and S. Li. Image search by concept map. *SIGIR*, 2010.
[35] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. *CVPR*, 2011.

Figure 8: Qualitative Results on the Flickr dataset.