Palette Power: Enabling Visual Search through Colors

Anurag Bhardwaj
eBay Research Labs
anbhardwaj@ebay.com

Atish Das Sarma
eBay Research Labs
adassarma@ebay.com

Wei Di
eBay Research Labs
wedi@ebay.com

Raffay Hamid
eBay Research Labs
rhamid@ebay.com

Robinson Piramuthu
eBay Research Labs
rpiramuthu@ebay.com

Neel Sundaresan
eBay Research Labs
nsundaresan@ebay.com

ABSTRACT

With the explosion of mobile devices with cameras, online search has moved beyond text to other modalities like images, voice, and writing. For many applications like Fashion, image-based search offers a compelling interface as compared to text forms by better capturing the visual attributes. In this paper we present a simple and fast search algorithm that uses color as the main feature for building visual search. We show that low level cues such as color can be used to quantify image similarity and also to discriminate among products with different visual appearances. We demonstrate the effectiveness of our approach through a mobile shopping application[1]. Our approach outperforms several other state-of-the-art image retrieval algorithms for large scale image data.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Search Engine, Image Search, Visual Search, e-Commerce

1. INTRODUCTION

Recent advances in mobile devices, especially smartphones and tablets, has redefined the dynamics of commerce. The ability to shop anywhere and anytime has allowed users to bridge the gap between offline and online stores. It has also led to newer shopping trends where users browse for goods in offline stores and use online stores to find the best deals [23, 1]. Recent statistics suggest that among the top online shopping categories, 58% of smartphone users research electronic goods in-store but purchase them online [1, 6]. A similar trend has been observed in other categories such as shoes (41%) and apparel (39%). However, searching for a product among the massive collection of items remains a major bottleneck for this shopping experience. Although recent work in text retrieval has addressed some of these issues, categories such as fashion continue to present a huge challenge. This is particularly contributed by the following:

- Most of the items in such categories lack useful product specifications that can be used for indexing.
- The notion of relevance in these categories is mostly visual which exposes the limitation of textual queries.
- Mobile shopping experience requires a fast and low memory solution both for indexing and search.

Many product categories sold online can be described exactly by a limited set of well-defined attributes. For instance, digital cameras can be described accurately by their model name, frame size, and pixel count. On the other hand, it is difficult to describe women's dresses in terms of such textual attributes. Although basic visual properties, such as color, can be specified by the sellers, a cursory data analysis shows that a large fraction of fashion items listed online lack this information. For example, Figure 1 shows the color and pattern distribution of men's neck tie collections on an e-commerce website where this metadata is un-specified for around 50% of the items. Also, for some of the visually complex patterns (such as 19.8% items tagged as multi-color), there is not much textual information about the color content of the items. We believe this reinforces our aforementioned observation regarding the difficulty of using text to describe items that are fundamentally defined by their visual attributes, and highlights the importance of using their image information to index them more accurately.

Consider the history of web search as well. While text based features played a crucial role in the early days, in the form of tf-idf scores, now increasing importance is associated with several other kinds of signals and features that are more semantic and perceptual in nature. In similar vain, image search is also going through a transition where it becomes more important than ever to really understand the perception of users and try to capture the same in any visual task such as image matching, search, or retrieval. As is to be expected, one of the first things that attracts a viewer's attention is the color distribution. Features such as patterns, textures, or styles are only secondary in nature. To this end, the goal of our work is to really focus on color as the primary feature and see how far we can take image search.

In a way, the goal of this work is to see how far something extremely simple can be taken, rather than try to benefit from more complex features. Surprisingly, simplicity actually goes a long way, reinforcing our hypothesis that in the context of images, perceptual features should be the starting point, rather than just a component. To this end, our paper focuses on the image search and retrieval application and is constrained to only capturing an image's

[1]eBay Fashion App available at https://itunes.apple.com/us/app/ebay-fashion/id378358380?mt=8 and eBay image swatch is the feature indexing millions of real world fashion images

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

features in color, and specifically a bag-of-color - i.e. just the color distribution. A dual goal, however, is for us to build an application that actually is practical and usable at a large scale; to this end, the concern that a lot of images, specifically for shopping, are taken with low-end cameras on cell phones is an additional difficulty.

Using image information to index items for a mobile shopping experience is challenging for two main reasons. The quality of these images can have a lot of variance, especially since most of them are captured by amateur users. Secondly, being an online experience, the indexing and retrieval system must be computationally very fast, with a low memory foot-print. These constraints limit the types of information that can be extracted from product images, and the algorithms that can be used to analyze this information. In this paper we address some of these issues by describing a simple and fast visual search algorithm that demonstrates the power of low-level features, specially color, for online shopping of fashion related items.

This paper presents our techniques that go into the fully developed and live system at eBay being used on mobile devices under eBay Fashion. The app is live and functional and is accessible to everyone in the US and can be used to search and shop for fashion clothing inventory on eBay. The app can be found on the Apple app store through *eBay Fashion App*[2]; the feature indexing real world fashion images is called *eBay Image Swatch*; it indexes and searches a large number of images, in the order of millions. The process starts with a user taking a picture (from their cell phone camera), or using a stored image on the phone, of any color combination desired (and typically of a dress, or tshirt or piece of clothing); this is then searched for using our algorithm, on the millions of images indexed by our system, and the *best matches* are returned.

The remainder of the paper is organized as follows: Section 2 discusses related work on image search. Section 3 describes the main insights drawn from our data and the high-level algorithm for our color based search engine. Section 4 highlights the distance measure used in the algorithm, while Section 5 talks about the indexing scheme used for scalability. Section 6 discusses in detail our evaluation and experimental validation. Finally, Section 7 concludes the paper and discusses future work.

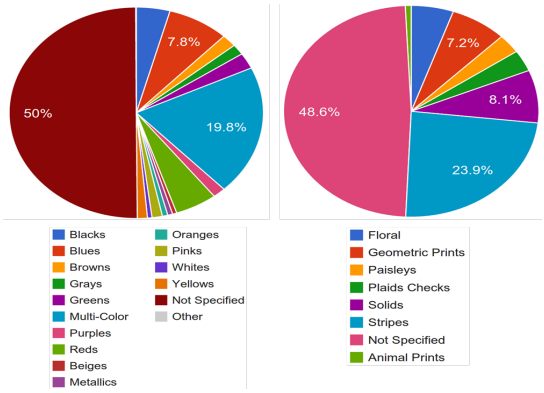


Figure 1: *(L) Color and (R) pattern distribution of men's neck-tie; Nearly 50% of items do not have these attributes specified for textual indexing.*

2. RELATED WORK

The task of visual search has been studied in great detail in Computer Vision community. Most of the existing techniques can be broadly categorized into "Feature Based Approaches" and "Bag-of-Words Based Approaches". Feature Based approaches employ traditional paradigm of extracting low-level image features and use techniques such as histogram distance, euclidean distance etc. to compute image similarity. The most commonly used features are:

- Color Features: Most systems using color features utilize color spaces such as Hue-Saturation-Value (HSV) [24] and generate a color-histogram representation of the image. To effectively match histograms, "cross-talk" between color bins is reduced by using weighted distance based metrics [10]. This technique can be further extended by modeling spatial correlation of color pixels in form of Spatial Chromatic Histogram (SCH) [4]. To address varying image illumination in real-world images, color constancy based algorithms are used [7].
- Texture Features: Haralick et al. [11] proposed elementary texture features in form of gray level co-occurence matrix (GLCM) for extracting second-order statistics from image such as energy, entropy, contrast and homogeneity. Finer features modeling human perception include coarseness, contrast, directionality, line-likeness, regularity and roughness [25] . Statistical texture features in form of Gabor Wavelets [19] are also used to enable filtering in spatial as well as frequency domain.
- Shape Features: One of the earliest systems using shape features for image retrieval include IBM's QBIC system [20] that use shape area, eccentricity and major axis orientation. Higher level shape invariants such as angle between two color edges and the cross-ratio between four color edges are also used [8]. However, most of these features have been shown to perform poorly as opposed to using only color and texture information. Shape Contexts [3] overcome some of these limitations by measuring shape similarity through point correspondences recovered in uniform log-polar space.

Bag-of-Word (BoW) Based Approaches: This technique refers to representing an image by an orderless collection of local features [9]. A basic BoW method clusters local features into different vocabulary groups to generate a "CodeBook". Each image is represented over this codebook using a histogram of vocabulary counts. Finally, histogram distances are used to compute similarity score between two images. Several different types of local features that utilize this particular image representation include Scale Invariant Feature Transform (SIFT) [18], Speeded Up Robust Features (SURF) [2] or Histogram of Oriented Gradients (HOG) [5].

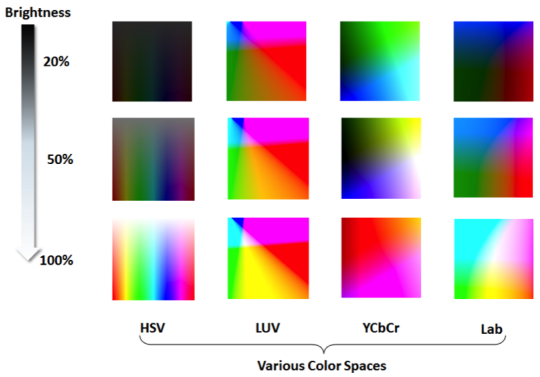


Figure 2: *Sample color spaces; Comparing the shape of decision regions for different colors, only HSV color space has a uniform color distribution which is best suited for uniform sampling.*

[2]Available at https://itunes.apple.com/us/app/ebay-fashion/id378358380?mt=8

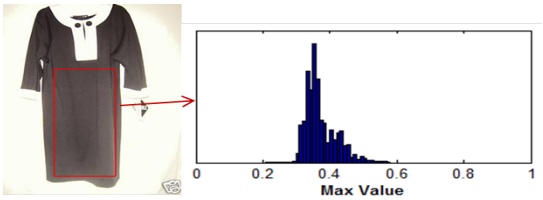


Figure 3: *Lighting Variation induces high variance in RGB values even for single color (shown in red box), thus the histogram of maximum value of R, G, B shows a wide variation, instead of having a single peak.*

Spatial Pyramid matching techniques [17] have also been proposed that define a fixed hierarchy of rectangular windows for capturing "perceptually salient features" of the image. However, one of the major drawbacks of these approaches is the long training time to learn a visual vocabulary. Since online products usually have short life-span and continuous high volume of inventory updates, such long training times cannot be afforded.

Such advances in image search techniques have also enabled several consumer facing applications. Some of the most prominent technologies known today are Google Goggles [13], Bing Visual Search [15]. However, these technologies provide generic image search capabilities which may not be useful for a specific domain such as fashion. There are also customized vertical based image search techniques for fashion such as GILT Groupe's app for iPhone [14] but they only use dominant color information for matching and hence cannot effectively support multi-colored query images. The objective of this work is to thoroughly analyze one specific vertical that is fashion and design a search algorithm geared towards solving fashion-specific image matching problems.

3. INSIGHTS FROM DATA

Designing a visual or image search engine involves multiple steps such as finding salient regions in images (i.e. object localization), computing representative visual properties (feature extraction) and searching the entire repository for similar images (matching). A number of factors affect the performance of system including color distribution, texture, pattern, and shape information that may reveal the clothing style. Each of these phases present a new challenge in terms of scalability and efficiency. To understand the bottleneck of each component and find a possible solution, we choose images of the fashion product family from one of the largest e-commerce marketplaces, and analyze them for possible solutions. Following sub-sections describe insights learned from this data analysis.

3.1 Spatial Prior for Object Localization

Online user-generated image content for commerce often vary in quality. In a peer-to-peer commerce marketplace, casual sellers may not be motivated or skilled to take professional quality pictures as opposed to professional sellers dealing with large sales volume. Most of such low-quality images suffer from poor lighting, low contrast, and cluttered background that makes the task of image matching more difficult (Figure 4). Therefore finding the relevant region in the image (object localization) becomes a very important task. One approach would be to apply a State-of-the-art image segmentation algorithm to remove the background. However, typical image segmentation algorithms operate at pixel levels and may take several seconds to process a good resolution query image which is unacceptable in our constrained settings.

Figure 4: *Sample images with challenging background; Figures (a) and (c) exhibit low contrast between background and foreground as Figure (a) has a red tone in both while figure (c) has a white/cream color for both. Figures (b) and (d) are very cluttered images making it difficult to isolate the foreground object based on color.*

To solve this problem, we take insights from our data and build an offline spatial prior for foreground (dress) and background (clutter) pixels. Firstly, a black-box segmentation algorithm is used (GrabCut [22]) to automatically remove the background for images in the women's dress category. Images with poor confidence for background removal are ignored. The segmentation provides a mask or an outline of foreground pixels. We take the sample mean of RGB pixel values at corresponding mask locations and visualize this for each style of dress in our inventory (Figure 5). As shown, the mask outlines have a close resemblance to the dress style shown on the right (Off-Shoulder, Long Sleeve, Halter, Cap Sleeve, 3/4 Sleeve) illustrating that a large number of dresses respond well to the segmentation algorithm. Further, the mean RGB image in all the styles occupy the center of the image which provides a strong clue as to how users tend to center the dress before they take pictures. We use this insight to design a segmentation mask that samples the center of the image for foreground pixels and assumes pixels falling outside of center-window to be background. One obvious failure case of this technique is shown in figure 6, where the image consists of multiple views of the dress and hence such a sampling would lead to false segmentation. However, since most of these images occur in our online inventory and tend to have simpler background, we can afford to apply black-box segmentation algorithms to find dress regions. Moreover, a smaller portion of such images allows us to keep a good indexing rate.

3.2 Choosing right Color Space

We emphasize the use of color for image retrieval. Therefore the choice of color representation is important for extraction of color distribution. Typically, images can be represented by different color spaces such as HSV, HSL, HSI, Lab, LUV, YCbCr. Each color space has its own set of strengths and weaknesses. These color spaces map pixel values to 2-dimensional chrominance space, and a single luminance/brightness channel that captures most of the lighting variations.

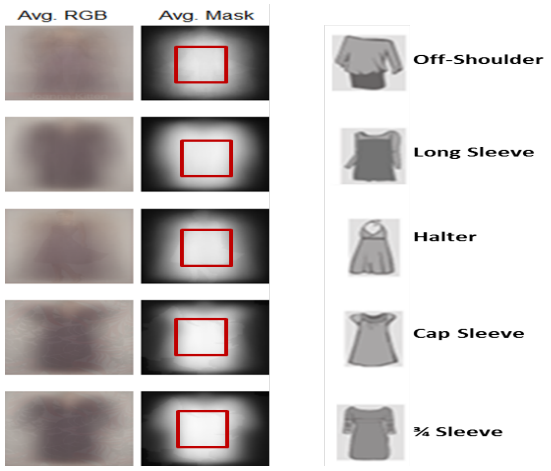


Figure 5: *Spatial prior for various women's dress styles. We segment the image such that a mask or outline is produced for the foreground pixels. We then take the sample mean of RGB pixel values for the mask locations and use it to obtain the dress style from our inventory.*

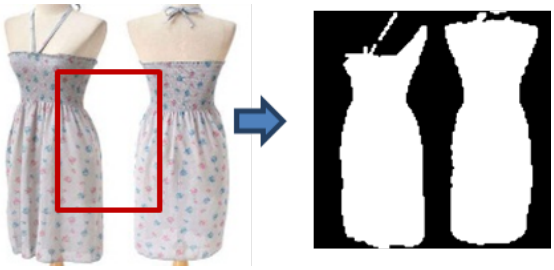


Figure 6: *(Left) Spatial prior failure case, (Right) Addressed using Background Segmentation. The segmentation mask helps sample pixels from the center of the image for foreground extraction.*

Visualization of some of these color spaces are shown in Figure 2. Once color space is chosen, it needs to be sampled in order to get the color histogram. The simplest form of sampling is uniform sampling, forming a rectangular grid, where each axis is divided equally. Each axis may have different number of bins, but along each axis, the bins have equal width. As shown in Figure 2 only HSV color space has a uniform color distribution which is best suited for uniform sampling. Choosing any other color space with such a sampling scheme will lead to color bias. For e.g. Uniform sampling with YCbCr will lead to good matching for red and pink but fewer matches for other colors since these colors dominate the color space. Even in cases with single color, there may be variations due to illumination changes Figure 3 shows an example, whereas even though the sampled region (shown in red) is perceived as a single color, multiple shades appear due to lighting variations (examples: shadows, attenuation of strength of illumination over space). The histogram of maximum value of R, G, B shows a wide variation, instead of having a single peak. These subtle changes can be easily perceived by humans but are difficult for RGB based color models. Using HSV color space provides us with an added advantage as it mirrors the human perception of color, thereby providing a better way to handle illumination variations.

Another important aspect of color matching is color confusion which increases as saturation decreases. To understand this, let us

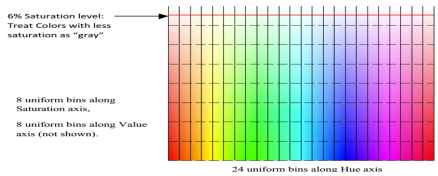


Figure 7: *We use a binning scheme for the Hue-Saturation-Value (HSV) space with 8 bins each along the saturation and value axes and 24 uniform bins along the hue axis.*

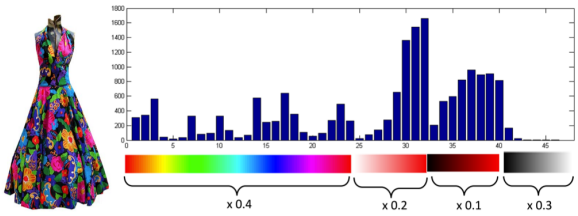


Figure 8: *Illustration of stacked 1D histogram for multi-colored dress: We see that the dress results in higher concentration in certain bins along the axis and this is then used for indexing and matching.*

look at the mathematical formulae for hue, saturation and value channels of HSV space as shown below:

$$
\begin{aligned}
V &= max(R,G,B) \\
S &= \begin{cases} \frac{V-min(R,G,B)}{V}, & if\ V > 0 \\ 0, & Otherwise \end{cases} \\
H &= \begin{cases} \frac{60(G-B)}{S}, & if\ V = R \\ 120 + \frac{60(B-R)}{S}, & if\ V = G \\ 240 + \frac{60(R-G)}{S}, & if\ V = B \end{cases}
\end{aligned} \tag{1}
$$

This formula lets us visualize HSV space in an alternative form as shown in Figure 7. Notice that for small values of V, all three values R, G, B will be similar. The same is true for small values of saturation, since maximum value and minimum values of R, G, and B will be similar. This means that, due to noise, each of R, G, and B may be dominant spuriously. Thus, hue will have discontinuous values (note the conditional assignment for hue). In other words, hue is not reliable when saturation is low. When saturation is low, the color will lack vividness and look grayish. To overcome this issue, we consider all pixels with saturation less than 6% as "gray" pixels and bin them separately along with rest of the H, S,V components. Figure 8 shows an example of a stacked 1D histogram containing all the bins along with their respective weights.

4. COLOR BASED DISTANCE MEASURE

Given an RGB swatch, we first convert it into HSV space and build a color histogram in each dimension. Since sampling in full 3D space as cross-product will create a sparse histogram matrix, we treat each channel (Hue, Saturation, Value) separately. Uniformly spaced bins along Hue(nH), Saturation(nS) and Value(nV) axes are stacked to form 1D histograms. This produces relatively dense and more reliable histogram, which is much smaller than the full 3D sample scheme ($nH + nS + nV$ vs. $nH * nS * nV$). This has an added advantage of having smaller memory requirement and faster matching. Weights that emphasize the Hue channel are ap-

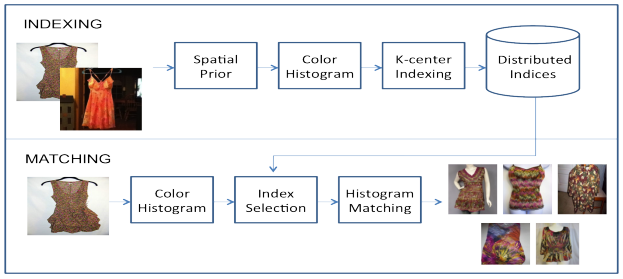


Figure 9: *The figure shows our overall system overview. The indexing proceeds by first incorporating a spatial prior on the image and then obtaining a color histogram. This is then indexed using the k-center scheme and stored as distributed indices. During the matching phase, the query image's color histogram is extracted in a similar manner and the index selection is performed by reading into the distributed indices. From the searched results, the images that match best on the histogram spectrum are returned.*

plied. For example, in our fashion inventory, 24, 8, 8 and 8 are chosen as the number of bins for H, S, V and Gray channels respectively. Their respective weights are set to 0.4, 0.2, 0.1, 0.3. Finally, we normalize the entire bin such that the ratio of color pixels to gray pixels is encoded in the stacked histogram. A similar process is repeated for an input query image. To compute similarities between query and candidate histograms, Hellinger distance is utilized (Equation 2).

$$S(H_1, H_2) = \sqrt{1 - \sum_{i=1}^{N} \sqrt{(\sum H_{1i} \sum H_{2i})}} \qquad (2)$$

where N is the number of bins, H_1 and H_2 represent the query image histogram and index image histogram respectively. Figure 9 outlines the overall system architecture of our proposed method.

Given the scale of images we deal with, and the fact that image query search results need to be returned in the order of milliseconds, it is impractical to assume the backend algorithm can perform a flat search at runtime. With tens of millions of images to compare the query image against, it is necessary to design efficient retrieval techniques for online processing.

5. INDEXING SCHEME TAILORED FOR IMAGE SEARCH

Indexing schemes have been heavily researched in a variety of different contexts such as traditional search engines. The scale at which we wish to admit image searches has grown rapidly in the past couple of years. Notice, in our context both the query and the result sets are images. We contrast the vocabulary and intent in a few kinds of search tasks:

- Keyword-Document search: This is the most widely researched area with indexing schemes such as inverted index forming the basis of several industry standards. Notice that the indexing scheme heavily relies on the fact that the query is only a few keywords.
- Document-Document search: While in the case of retrieving similar documents as the query, the vocabulary of both query and results is text, and they are of similar sizes, the challenge here is in reconciling a good *distance* function on the space of all documents.
- Keyword-Image search: The dictionary of query and results are different here, and so any distance function imposed on the query and the result space is over dissimilar types. An example of such a setting is Google Image Search.
- Image-Image search: The challenges here are different from any of the previous settings and unfortunately it is unclear how any of them can be adapted to our context. However, the main observation that can be exploited for images or swatched images is that the query space and the document space look similar (unlike text search) and therefore can leverage the same distance function for indexing as well as retrieval. The challenges therefore lie in designing a distance function and efficient search/retrieval scheme based on the suitably chosen feature spaces.

Parallel this with traditional and very well-studied text search, where efficient techniques have been developed for retrieving relevant documents corresponding to keyword queries. Unfortunately, though, the same techniques do not directly adapt to image search given that their indexing schemes are largely text-based. In the following section, we describe a methodology that scales well for image search and retrieval.

We now present a backend clustering and indexing scheme that admits efficient image similarity retrieval for online queries. This helps us overcome the naive linear scan look-up and obtain a tunable parameter that can trade-off time complexity without much loss in retrieval accuracy.

We begin by describing the simple clustering based approach that helps significantly speed up query run time complexity for image retrieval. The algorithm is based on computing and storing a backend k-center clustering, and then at query time. At run time, the query's distance is computed to each of the k centers of the k clusters. Subsequently, the query is compared with all points in the cluster corresponding to the nearest cluster center, and the top matches are returned. While this approach has potentially several benefits, for the purposes of this paper, the focus is entirely on obtaining nearly results as good as the naive approach, but with a significantly enhanced time complexity. Below we describe the notion of k-center clustering objective, a 2-approximation algorithm (which is folklore) and present the proof and time complexity for completeness. This is followed by experimental validation of this method for our context.

k-center objective. The goal of the k-center clustering algorithm, given a set S of n points in a metric space, is identify a set of k centers (and an allocation of each of the remaining points to its nearest center) in order to minimize the maximum diameter of the clusters. Specifically the diameter of a cluster is measured as the maximum inter-point distance, over all points in a cluster.

Let us use $d(a, b)$ to denote the distance between any two points a and b. Mathematically stated, the goal is to find a set C of centers, with $|C| = k$, in order to minimize $\max_{c \in C} \max_{s_1, s_2 \in c} d(s_1, s_2)$. Here we slightly abuse notation to use c as a set as well (including all points in the cluster corresponding to this center). Optimizing this objective is a well-known NP-hard problem [12] but there is a very simple *greedy* algorithm that achieves a constant factor approximation and runs in $O(nk)$ time.

Algorithm description (GREEDY-ALGORITHM). The algorithm proceeds by greedily picking k centers as follows. The first center is picked arbitrarily, call this c_1. The second center, c_2, is picked as the farthest point from the remaining $n - 1$ points. After, i centers have been picked, the next center c_{i+1} is picked as the point (from among the remaining $(n-i)$ points) as the one farthest from the already picked centers - here the distance of a point to a set of points (or a set of centers) is measured as the distance to the nearest of the points. Therefore, let C_i denote the set of first i centers picked,

the next center c_{i+1} is picked as $\arg\max_{c_{i+1} \in S} d(c_{i+1}, C_i) = \arg\max_{c_{i+1} \in S} \max_{j=1}^{i} d(c_{i+1}, c_j)$. All the k centers are picked in this manner. Finally, the allocation of all n points to their respective centers is done by picking the nearest center independently for each point.

THEOREM 5.1 ([12]). GREEDY-ALGORITHM *achieves a 2-factor approximation for the k-center objective.*

PROOF. Let D be the objective cost generated by the greedy algorithm. I.e. $D = \max_{c \in C} \max_{s_1, s_2 \in c} d(s_1, s_2)$. Now consider the optimal set of cluster centers say C^*, and let the associated objective cost be D^*. Consider a hypothetical scenario where the algorithm generated $(k+1)$ center points instead of k by the same greedy process, and call the last center c_{k+1}. Then by pigeonhole principle, at least two of $c_1, c_2, ..., c_k, c_{k+1}$ would fall in the same cluster in optimal clustering C^*. Let these two centers be c_i and c_j. Therefore, the optimal cost D^* is at least $d(c_i, c_j)$. However, note that the distance from any point in S to its nearest center in C is at most $d(c_{k+1}, C)$ because of the greedy process, which is at most $d(c_i, c_j)$ since c_{k+1} was the last hypothetical center to be picked. Further, by triangle inequality (given it is a metric space), the diameter of any cluster in C is at most twice $d(c_i, c_j)$. Combining these two bounds, it follows that $D \leq 2 * D^*$ which completes the proof. □

Retrieval. Once the k-center clustering has been pre-computed, given a query image, the retrieval algorithm is really simple: rather than compute the distance of the query to all n points, we compute the distance to each of the k centers. Subsequently, we pick the nearest center, and only consider points allocated to this center. The distance of the query is then computed to each point corresponding to this center's cluster, and the top results are returned.

Time Complexity. The time complexity of the clustering phase is $O(nk^2)$. This is easy to see as to choose i-th center, one considers the distance of each of the $(n-i+1)$ remaining points to the $(i-1)$ already picked centers. Therefore, the time complexity is of the order of $n + 2n + 3n + ... + kn$ which is $O(nk^2)$.

The time complexity of the retrieval phase varies depending on the size and distribution of all clusters, but is expected to be of the order of $O(k + \frac{n}{k})$. This is because the distance of the query point is first computed with each of the k centers. Then, assuming a roughly equal distribution of points to centers, the query point is subsequently compared with $O(\frac{n}{k})$ points. Notice that for $k \approx \sqrt{n}$, the query complexity is only $O(\sqrt{n})$. This is a substantial improvement over the naive query complexity of $O(n)$.

As a comment, the above approach is intuitive and may work well even when using a distance that is not necessarily a metric. The theoretical approximation guarantees however hold only for a metric space. Also one can incorporate several additional heuristics to improve performance, such as comparing with points in a small constant number of clusters rather than just one cluster (depending on the center distances). Further, it is possible to explore other clustering algorithms such as k-means. We pick k-center for its simplicity and as it scales efficiently to a large number of points, and present experimental results in the next section as a validation of its performance.

6. EXPERIMENTS AND DISCUSSION

6.1 Experiments - Fashion Dataset

One of the main motivations of our work is to improve the search experience for online fashion shopping which is predominantly visual in nature. For our experiments, we created a dataset of nearly 1 million images by taking a set of random snapshots from a large e-commerce website. In this data collection, we focus only on 6 major categories for women's clothing: Dresses, Tops & Blouses, Coats & Jackets, Skirts, Sweaters, and T-Shirts. For evaluating the system performance, we developed it as a mobile application and deployed it for 15 users who scanned 1600 query images over a period of 30 days. Since annotating such a large number of queries, each for 1 million images was infeasible, we randomly sampled 40 query images for our evaluation. 3 human judges were shown the top-10 matches for every query image and were asked to rate it on a scale of 0-4 (0 marking non-relevant matches). Finally, judgements from all human judges was collated and average rating per query-match pair was recorded as the final relevance.

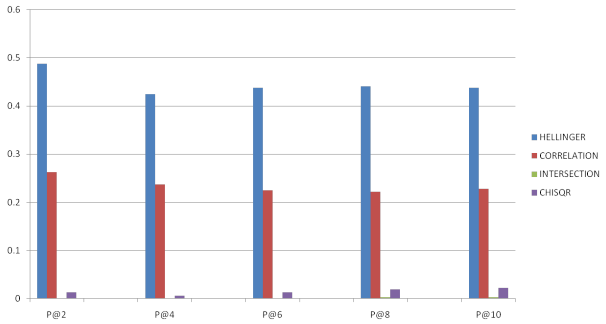


Figure 10: *Precision@k evaluation for different distance metrics at rank k. x-axis shows k varying between 2 and 10 and we see that across all the Hellinger distance metric performs best uniformly across the spectrum.*

Figure 11: *Two query examples where matched pattens found in the Top-4 results. The first image in each row shows the query image and the remaining four images in the row are the returned results from our inventory.*

We use this human judgement to compare different distance metrics for histogram matching. 4 distance metrics: Hellinger, Correlation, Intersection and Chi-Square were evaluated. Figure 10 illustrates the relative performance of each of these metrics and as shown, Hellinger Distance outperforms all other distance metrics which supports our use of this metric in our matching algorithm.

We also show how our proposed algorithm fares in retrieving fashion images containing distinct pattern. Figure 11 shows two such examples of plaid pattern as query and top-4 results returned by our system. As shown, even without any explicit pattern information encoded, our algorithm is able to retrieve matching patterns from the inventory.

6.2 Experiments - k-center Based Indexing

In this section, we describe our experimental evaluation of k-center based indexing technique. For a large scale inventory, flat

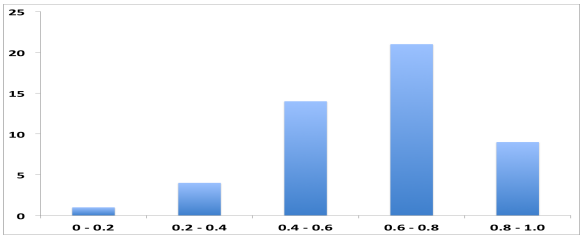


Figure 12: *Plot of % overlap between indexed and non-indexed results for $k = 50$ clusters. The x-axis shows the range of similarity varying in bucket sizes of 0.2 and the y-axis shows the fraction of queries for which the corresponding similarity was observed. As can be seen, the histogram corresponding to the bucket of $[0.6, 0.8]$ has the maximum height suggesting that for many queries, the overlap between the returned top-50 results in either approach (indexing or not indexing) was between 60 and 80 percent. Therefore the indexing scheme is able to obtain almost as good results as the non-indexing approach.*

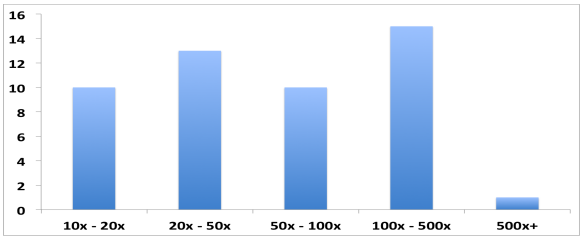


Figure 13: *Plot of speedup distribution using indexed approach for $k = 50$ clusters. The x-axis shows the speedup factor obtained by the indexing scheme compared to the non-indexing scheme. The y-axis shows the fraction of queries that obtained a speedup in the specified range. The factor varies from 10-20 times faster all the way up to 500 times faster. This is the obtained speed up for $k = 50$. Contrasting this with the plot of overlap shown above, we notice that the indexing scheme obtains substantial speed up without much loss in result quality.*

matching across millions of images can be prohibitively expensive. Our results in this section show that the suggested indexing and retrieval approach not only provides substantial speedups over flat matching but also ensures high overlap with results from flat matching approach. Due to space constraints, we only outline a sampling of the plots here.

Figure 12 and 13 show % overlap with flat matching results and speedup against flat matching results respectively when using k-center based indexing strategy for cluster size $k = 50$. As can be seen, our proposed approach obtains a minimum of $10x$ speedup for all queries in our database while maintaining an average overlap of around 60% for top-20 matches.

We also study the effect of varying the cluster size. Figure 14 and 15 illustrate the trade-off that can be obtained between the speedup and overlap ratio as k scales. Specifically, we observe a linear speedup in matching time at almost no performance degradation.

6.3 Experiments - Generic e-commerce Dataset

Though our system is primarily designed for matching soft goods such as fashion, we also experimented to benchmark its generalization to other commerce categories containing rigid goods such as Camera, Toys and Sports. Examples of this dataset is shown in Figure 17. For each query image, approximately 15 true matching images are collected. 5 are transformed versions of the query

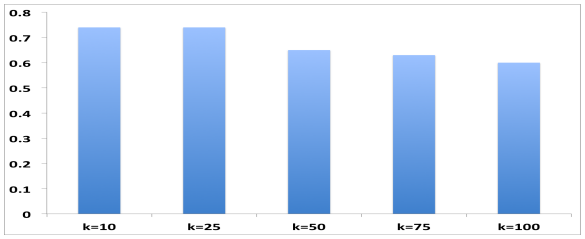


Figure 14: *Plot of median % overlap between indexed and non-indexed results for varying cluster sizes. In this plot we show that even as k is increased, the percentage overlap between the result sets does not degrade much. This suggests that we can obtain higher speedups without little loss in accuracy by tuning k appropriately.*

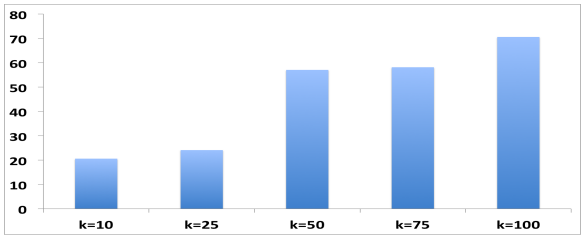


Figure 15: *Plot of median speedup using indexed results for varying cluster sizes. We see that as k is increased, the median speed up also increases steadily compared to the non-indexed approach.*

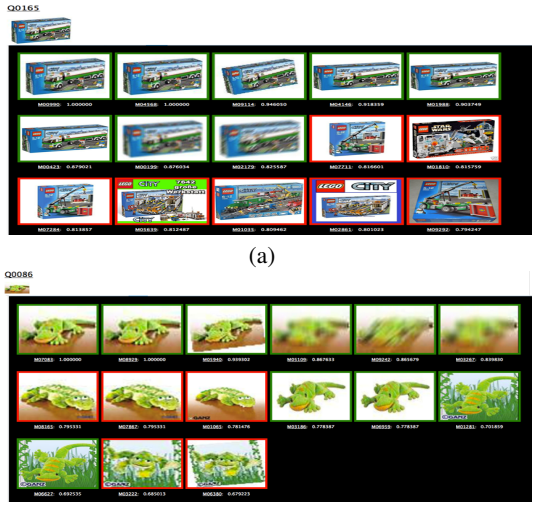
image with randomly chosen transformation parameters for Gaussian Blur, Perspective Distortion, Rotation etc. Figure 18 shows the sizes of each categories of this dataset.

For comparison, we baseline our results with State-of-the-art descriptors SIFT [2] and SURF [18]. Figure 19 shows the performance of each category in the dataset, where our method outperforms both SIFT and SURF for all three categories in terms of Mean Average Precision (mAP) [21].

Figure 16 and Figure 20 show examples of matching results for Toys, Camera and Sports respectively. Figure 16(a) and Figure 20(b) show that our system successfully retrieves images with different transformations, such as shape, view angle and blurring. Besides, as shown in (b) of both figures, due to the center spatial prior, our algorithm successfully avoids the noisy information either from the complex background or from the image headers, which may be commonly seen in online shopping images.

6.4 Experiments - Public Image Search Dataset

We further extend our quantitative evaluation to a more general INRIA Holidays dataset. This dataset [16] consists of images from different categories such as natural scenes, monuments, buildings with varying resolutions and perspective. Each category contains 500 unique images where the first image is used as query and the rest as matching. We compared our results with the state-of-art bag-of-words (BOF) based method [9]. Figure 21 shows the comparison between our proposed algorithm and the BoW baseline algorithm. We show the baseline performance for two different vocabulary sizes: $k = 2000$ (BOF2000) and $k = 20000$ (BOF20000) that were reported in [16]. For the proposed algorithm we discard the spatial prior and use all pixels for computing the histogram since the dataset does not contain any strong spatial prior. We also experiment with different number of bins and set the size of Hue histogram to $nH = 36$. Weights are also adjusted accordingly to be 0.5, 0.25, 0.15, 0.1 for H, S, V for color



(a)

(b)

Figure 16: *Toy examples of retrieved results.*

(a) Toys

(b) Sports

(c) Camera

Figure 17: *Examples of one collection of the online commerce data that contain three different categories.*

pixels and V for gray pixels, respectively. This is done to account for larger color variation obtained by sampling from the whole image instead of center. The mAP result of our algorithm is denoted as color-A in Figure 21. To utilize spatial information, we further divide each image into 5 patches (left, right, top, bottom, center) and compute their individual similarity. These similarities are averaged to generate a final matching score. The mAP result of this extension is denoted as color-B in Figure 21. Note that by only using 60 features(36+8+8+8=60), our algorithm outperforms the baseline result BOF2000. By dividing the entire image into several small patches to include some spatial information, our result (color-B) even outperforms the result by using 200000 visual words (BOF200000).

6.5 Computational Costs

Using 24GB RAM, Intel Xeon E5630, 2.53GHz, the average feature extraction costs 10ms per image, and retrieval about 80ms, (i.e. average time required for matching the a query image with all the images in the 1 million online commerce image dataset). Each feature vector consists of 49 float values totaling to 196 bytes of memory. Matching server for 1 million items takes 190MB of RAM to load feature indices and return query results in real-time.

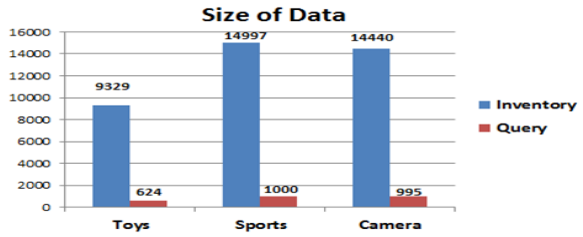


Figure 18: *Data distribution of one collection of the online commerce data that contain three different categories.*

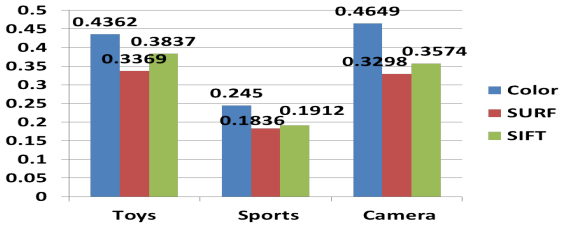


Figure 19: *mAP (Mean Average Precision) comparison between the proposed method with SIFT and SURF.*

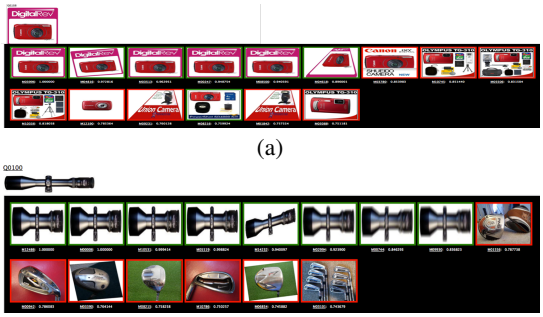
Figure 22 shows the average time used for indexing and matching each image for different categories. Our speed metrics show that our system is at par with the leading online search engines. Furthermore, our low memory footprint indicates the scalability of the system to a larger number of categories.

7. CONCLUSIONS

In this paper, we presented a color based visual search engine for fashion. Text-based search has been extensively studied in the literature over the last two decades. As the web evolves from text to more sophisticated content including billions of images, it is crucial to re-evaluate and design new systems robust enough to deal with modern applications such as image search. Take for example the application of searching for clothing items in an online search engine. The task at hand is of a visual nature where users would like or even expect engines to be able to return results from their inventory very similar to a dress that they have a picture to. This happens often when someone takes a picture of a dress from a cell phone, either of another person, or in a store, and would like to be able to look up prices or stock availability from online stores immediately in seconds. We design a massive system specific to this application and introduce techniques of computer vision that may go beyond this domain. Our system is live and open to use by anyone under the eBay Fashion App and leverages the eBay inventory using techniques presented in this paper.

Our algorithm is motivated from insights into real-world fashion data that enables us to design simple yet efficient object localization and color matching technique. Our main findings from this work are as follows:

- Spatial priors can be extremely useful in localizing objects in the images taken by amateur sellers. This is because intuitively most of the sellers tend to place the product at a certain location in the image that be learned from the data.
- By using only color information, we can also get a reasonable handle on the textures and patterns of fashion items without explicitly modeling them.



(a) (b)

Figure 20: *Camera and Sports examples of retrieval result for online commerce dataset. The top results are for a query image camera and one can notice several similar camera images are returned. The bottom results are for a sports equipment.*

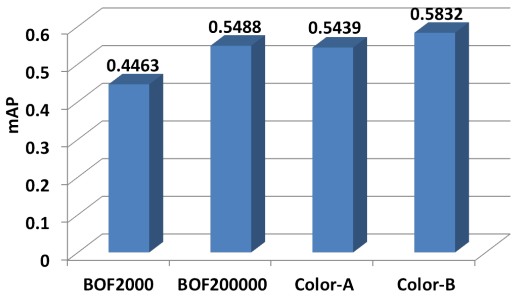


Figure 21: *mAP (Mean Average Precision) comparison for the proposed algorithm and BOF baseline algorithm on INRIA holidays data.*

- It is important to separate the chromatic (color) content of the image from its achromatic (grayscale) content. This is because most of the product images uploaded by amateur sellers have low saturation which makes it difficult to differentiate among them by only using their color content.
- We present an indexing scheme that is particularly suited for image based query search and is able to exploit the similar query and result spaces of images or swatched images, unlike traditional text search engines. This helps scale our techniques to millions of images and obtain efficient yet accurate searches.

Using these insights as designing principals of our visual search engine, we compare the results of our system to other approaches. We also illustrate successful matching results for distinctive clothing patterns. Furthermore, we baseline our performance with state-of-art image retrieval methods such as SIFT, SURF and Bag-of-Words techniques and illustrate our low computational cost advantages.

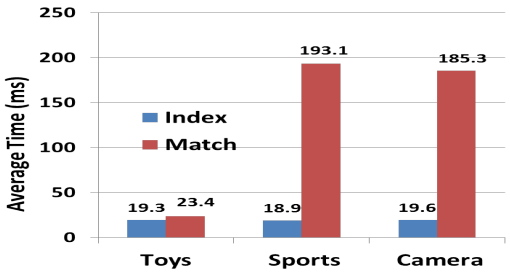


Figure 22: *Averaged computing time for each image used for online commerce dataset. Our algorithm is computationally light and requires small memory.*

Our future work focuses on extending the proposed method to incorporate non-uniform bins and explore tf-idf based vector space models for matching. It would also be very interesting to design and implement image based search applications for other genres beyond fashion and understand the technical challenges specific to these domains.

8. REFERENCES

[1] Anonymous. 10 more online shopping trends, November 2011. http://www.searchandise.net/blog/bid/47488/10-More-Online-Shopping-Trends.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110:346–359, June 2008.

[3] S. J. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. Technical report, 2002.

[4] L. Cinque, S. Levialdi, A. Pellicano, and K. A. Olsen. Color-based image retrieval using spatial-chromatic histograms. In *Proceedings of the 1999 IEEE International Conference on Multimedia Computing and Systems - Volume 02*, pages 969–, 1999.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. CVPR '05, pages 886–893, 2005.

[6] A. Eisner. Study finds: Retailers are not providing smartphone-equipped shoppers what they need, October 2011. http://www.retrevo.com/content/blog/2011/10/retailers-not-providing-smartphone-equipped-shoppers-what-they-need.

[7] B. V. Funt and G. D. Finlayson. Color constant color indexing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17:522–529, May 1995.

[8] T. Gevers and A. W. M. Smeulders. Pictoseek: combining color and shape invariant features for image retrieval. *IEEE Trans. Image. Processing*, page 2000, 102-119.

[9] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *J. Mach. Learn. Res.*, 8:725–760, May 2007.

[10] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17:729–736, July 1995.

[11] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, may 1979.

[12] D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. In *Mathematics of Operations Research*, Vol. 10, No. 2, pages 180–184, 1985.

[13] G. Inc. Google goggles. http://www.google.com/mobile/goggles.

[14] G. G. Inc. Gilt for iphone. http://www.gilt.com/apps/iphone.

[15] M. Inc. Bing mobile. http://m.bing.com/.

[16] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, pages 304–317, 2008.

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06.

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November 2004.

[19] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18:837–842, August 1996.

[20] C. W. Niblack, R. J. Barber, W. R. Equitz, M. D. Flickner, D. Glasman, D. Petkovic, and P. C. Yanker. The qbic project: Querying image by content using color, texture, and shape. 1908:173–187, feb 1993.

[21] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[22] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *ACM Trans. Graph.*, pages 309–314. ACM, August 2004.

[23] B. Siwicki. Many consumers are looking in stores but buying online, survey finds, October 2011. http://www.internetretailer.com/2011/10/26/many-consumers-are-looking-stores-buying-online-survey.

[24] J. R. Smith and S.-F. Chang. Single color extraction and image query. In *Proceedings of the 1995 International Conference on Image Processing - Volume 3*, ICIP '95, page 3528, 1995.

[25] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception vl , smc-8. *IEEE*, 75(6):460–473, 1978.