MATRIX FACTORIZATION-BASED CLUSTERING OF IMAGE FEATURES FOR BANDWIDTH-CONSTRAINED INFORMATION RETRIEVAL

*Jacob Chakareski, Immanuel Manohar**

University of Alabama, Tuscaloosa, AL 35487.

Shantanu Rane

Xerox PARC, Palo Alto, CA 94304.

ABSTRACT

We consider the problem of accurately and efficiently querying a remote server to retrieve information about images captured by a mobile device. In addition to reduced transmission overhead and computational complexity, the retrieval protocol should be robust to variations in the image acquisition process, such as translation, rotation, scaling, and sensor-related differences. We propose to extract scale-invariant image features and then perform clustering to reduce the number of features needed for image matching. Principal Component Analysis (PCA) and Non-negative Matrix Factorization (NMF) are investigated as candidate clustering approaches. The image matching complexity at the database server is quadratic in the (small) number of clusters, not in the (very large) number of image features. We employ an image-dependent information content metric to approximate the model order, i.e., the number of clusters, needed for accurate matching, which is preferable to setting the model order using trial and error. We show how to combine the hypotheses provided by PCA and NMF factor loadings, thereby obtaining more accurate retrieval than using either approach alone. In experiments on a database of urban images, we obtain a top-1 retrieval accuracy of 89% and a top-3 accuracy of 92.5%.

Index Terms— Clustering, non-negative matrix factorization, principal component analysis, information retrieval.

1. INTRODUCTION

Image-based information retrieval is becoming an important component of several technologies, including car navigation, video surveillance, mobile augmented reality and many more. A camera, usually installed on a mobile device, captures an image of the scene of interest and sends the image, or features extracted from the image to a remote server. The server compares the received features against its own database of images. Depending upon the application, the server sends back to the mobile device: matching images of the scene, or scene metadata, or control instructions to be executed at the mobile device. While accurate information retrieval is important, designing such systems involves negotiating an appropriate tradeoff amongst several other variables. The transmission overhead, computational complexity at the mobile device and at the database server, and robustness to variations in the image acquisition process, are some of the important considerations that need to be balanced. Figure 1 shows an augmented reality use-case in which the query image might not accurately match any of the images in the database owing to scaling variations, shifts, rotations and occlusions in the scene.

To achieve robustness to variations in the image acquisition process, it is customary to use a feature space that is invariant to these variations. The most popular feature set is the Scale Invariant
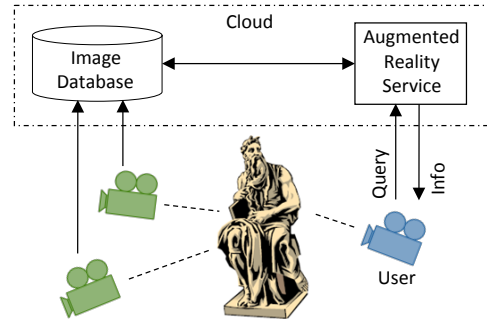

Fig. 1. Image-based information retrieval for augmented reality.

Feature Transform (SIFT) [1], which produces descriptors that are robust to rotation and uniform scaling, and partially invariant to affine distortion and illumination effects. While we use SIFT in our study, the underlying concepts should extend to other features based on "keypoints", i.e., salient locations in the image. These include SURF [2], HoG [3], CHoG [4], BRISK [5], FREAK [6] and others. To find an image in the server's database that matches the image captured by the mobile device, it is necessary to compare features from the query image against those from the database images.

One way to reduce the complexity of the matching process is to reduce the dimensionality of the feature vectors without compromising their matching ability. This is the approach followed in PCA-SIFT which obtains SIFT key points and then employs principal component analysis to reduce the dimensionality of the image patch around the key point [7]. Another class of methods to reduce the dimensionality of the image features is inspired by Locality Sensitive Hashing [8]. These methods involve computing one-bit random projections of the image features, and then matching the images in the subspace of random projections [9]. These methods were later generalized in [10], in which a quantized version of the Johnson-Lindenstrauss transform, was shown to improve matching performance by trading off the quantization step-size against the dimensionality of the projected features.

In this paper, we consider a different approach to dimensionality reduction than the ones discussed above[1]. Concretely, rather than reducing the dimension of each individual feature vector, we seek to reduce the total number of feature vectors per image. This is driven by the observation that, for good retrieval performance, a large number of keypoint-based features have to be extracted and

---

[1]We are interested in techniques that do not require training of a "good" feature set. Training-based methods [11–14] are very accurate, but they can also become cumbersome when the database keeps growing. When new landmarks, products, etc. are added, feature statistics are altered and the trained feature set must be updated.

used for matching. For the image sizes encountered today, this number can easily range from a few hundred to a few thousand descriptors. We compare two matrix factorization-based approaches to reduce the number of descriptors, viz., Principal Component Analysis (PCA) and a sparse Non-negative Matrix Factorization (NMF) approach developed for video querying [15]. Our choice is motivated by the fact that PCA and NMF factor loadings are closely related to cluster centroids that would be obtained if k-means clustering was applied to the image features [16, 17]. In doing so, we encounter a new problem: How many PCA or NMF factor loadings – equivalently, how many clusters – are enough for good matching performance? The answer to this question is image-dependent, and must be known to the mobile device. We employ an estimate of information content, previously used in econometric studies [18] to approximate the number of PCA and NMF factor loadings that will be used for matching. This estimate turns out to be significantly smaller than the number of keypoint-based features extracted per image, and incurs little or no penalty in matching performance.

An additional challenge presents itself at the database server: How does one match the basis vectors received from the mobile device against the set of basis vectors extracted from each database images. A natural solution is to correlate individual PCA or NMF factor loadings of the query image with those of the database images. While this approach works well for video querying [15], it might not provide good matching performance when the objects in the server's database have been photographed from vastly different viewpoints. We investigate a second approach which is based on the angle between the subspaces spanned by the PCA or NMF factor loadings of the query image and those of the database images. We report our findings for both matching criteria, evaluated on a database of urban images [19]. Furthermore, we show how to improve the accuracy of image-based information retrieval by combining the hypotheses obtained from the PCA and NMF bases.

The remainder of this paper is organized as follows: In Section 2, we fix notation and provide a brief description of the main feature clustering approaches investigated in this paper. In Section 3, the proposed image retrieval algorithm is described. The computational complexity of our approach is investigated in Section 4. Our experimental results obtained using a database of urban images, are detailed in Section 5.

2. BACKGROUND AND NOTATION

We now briefly describe the building blocks of our image retrieval scheme and also set the notation that will be used throughout the paper. The main building blocks include the feature extraction and feature clustering schemes based on PCA and NMF.

2.1. Feature extraction

As described earlier, the mobile device extracts keypoint-based features from the captured image. Similarly, the database server extracts keypoint-based features from each of its images. Let the total number of keypoints in a given image be N and let $\mathbf{d}_i \in \mathbb{R}^T$ be the descriptor, or feature vector, corresponding to the ith keypoint. The descriptors can then be stacked to form a $T \times N$ matrix $\mathbf{M} \triangleq [\mathbf{d}_1, \mathbf{d}_2, \ldots \mathbf{d}_N]$.

For concreteness, we focus on SIFT features henceforth, though other feature spaces are also usable within our framework. We frame the problem of computing a compact descriptor for an image as that of finding a low-dimensional representation of the matrix $\mathbf{M}$. Below, we discuss two ways for constructing such a representation.

2.2. PCA-based feature clustering

Using PCA, the matrix $\mathbf{M}$, can be written as $\mathbf{M} = \mathbf{HF} + \mathbf{E}$, where $\mathbf{H} \in \mathbb{R}^{T \times k}$ is called the factor loading matrix, $\mathbf{F} \in \mathbb{R}^{k \times N}$ is a matrix whose columns serve as PCA factors and $\mathbf{E} \in \mathbb{R}^{T \times N}$ is a noise matrix with covariance $\sigma^2 \mathbf{I}$ [18]. The pair $(\mathbf{H}, \mathbf{F})$ is not unique and can be replaced by the pair $(\mathbf{HQ}, \mathbf{Q}^{-1}\mathbf{F})$ for any non-singular matrix $\mathbf{Q}$ but the space spanned, given by $\mathcal{R}\{\mathbf{H}\}$ remains constant. To determine $\mathbf{H}$ and $\mathbf{F}$, we compute the Singular Value Decomposition of $\mathbf{M}$, given by $\mathbf{U\Sigma V}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{\Sigma}$ is a $T \times N$ matrix containing singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_T$ on its diagonal and zeros everywhere else. If the model order, or the dimension of the space spanned by $\mathbf{H}$ is known to be k, then $\mathbf{H}$ is obtained simply by taking the first k columns of $\mathbf{U}$, which are the singular vectors corresponding to the k highest singular values. In practice k is unknown and we will estimate it as described in the next section.

2.3. NMF-based feature clustering

Using NMF, the matrix $\mathbf{M}$, can be written as $\mathbf{M} = \mathbf{LR}$, where $\mathbf{L} \in \mathbb{R}^{T \times k}$ here corresponds to the factor loading matrix and $\mathbf{R} \in \mathbb{R}^{k \times N}$ is the matrix whose columns serve as NMF factors. We employ the technique used in [15], where NMF was used to cluster video descriptors. SIFT features were extracted from consecutive frames in the video sequence and stacked into a matrix similar to $\mathbf{M}$, containing all non-negative elements. By design, both $\mathbf{L}$ and $\mathbf{R}$ are constrained to have all non-negative elements. There exist many flavors of NMF algorithms [20, 21]. We adopt the sparse NMF algorithm proposed in [15] which constrains each column of $\mathbf{R}$ to belong to the set of standard basis vectors:

$$E_r = \left\{ \mathbf{e}_j \in \mathbb{R}^k : e(j) = 1, \text{and } 0 \text{ otherwise}, j \in \{1, \ldots k\} \right\}.$$

The optimization problem set up to determine $\mathbf{L}$ and $\mathbf{R}$ is given by:

$$(\widehat{\mathbf{L}}, \widehat{\mathbf{R}}) = \min_{\substack{\mathbf{L} \in \mathbb{R}^{T \times k} \\ \mathbf{R} \in \mathbb{R}^{k \times N}}} \frac{1}{2} \|\mathbf{M} - \mathbf{LR}\|_F^2,$$

$$\text{subject to: } \begin{cases} \|\mathbf{L}_i\|_2 = 1, & \forall i \in \{1, \ldots, k\} \\ \|\mathbf{R}_j\|_0 = 1, & \forall j \in \{1, \ldots, N\} \end{cases}.$$

In [15], the model order k was decided *a priori*. While we use the same NMF algorithm, the situation differs in two aspects. Firstly, the procedure is applied to features extracted from a single image, rather than to features extracted from multiple image frames. Secondly, the model order is not determined ahead of time and is allowed to vary depending on the image content. For estimating the model order, we rely on an estimate of information content that is computed from the PCA decomposition of $\mathbf{M}$. Determination of the model order is discussed in the next section.

3. PROPOSED IMAGE-BASED RETRIEVAL SCHEME

A block diagram of our image-based information retrieval scheme is shown in Figure 2. First, a client device captures a query image and extracts keypoint-based image features from it. As explained earlier, the client then compresses the feature space using PCA or sparse NMF. The PCA or NMF factor loadings, representing "compressed" features, are sent to the server where they are matched against the factor loadings extracted from images in the server's database. Information about the top η matching objects is returned to the client.
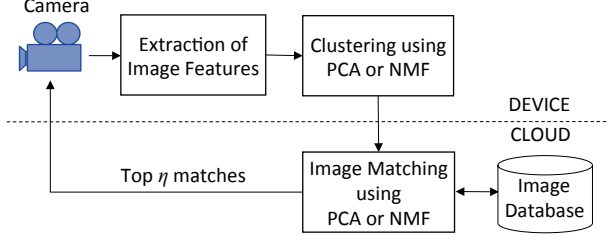
Fig. 2. The database server identifies the object (or relevant information about the object) photographed by the client device by matching a few compact image descriptors derived from a large number of scale-invariant image features.

Below, we first describe the database preparation process that is performed at the server. We also explain how the model order k, i.e., the number of relevant clusters, is chosen for each image at both the client and the server. Finally, we describe how the server determines the object in its database that most closely matches the query image.

3.1. Client-side and server-side processing

Let I_q be the query image captured by the client. As explained in Section 2, image features extracted from the query image are stacked together to obtain a matrix $\mathbf{M}_q$, which is compressed to a $T \times k$ matrix of PCA bases $\mathbf{H}_q$, or to a $T \times k$ matrix of NMF bases $\mathbf{L}_q$. Depending upon which matrix factorization technique is used $\mathbf{H}_q$, or $\mathbf{L}_q$, or both are sent to the server.

Let the server's database consist of K images, given by the set $\{I_1, I_2, \ldots, I_K\}$. Let the corresponding PCA representations be $\mathcal{H} = \{\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_K\}$, and the NMF representations be $\mathcal{L} = \{\mathbf{L}_1, \mathbf{L}_2, \ldots \mathbf{L}_K\}$. To compute the best matching images, according to the PCA representations, we compare $\mathbf{H_q}$ against $\mathcal{H}$. We represent the top η matches obtained using the PCA representation as the set $\mathcal{V}_{PCA} = \{h_1, h_2, ..., h_\eta\}$, where h_1 represents the best match and h_η the worst. If the server has multiple images per object, then the list should contain the η best matching *objects*, rather than the η best matching *images*. To achieve this, we examine the list $\mathcal{V}_{PCA}$, and if it contains more than one image of a given object, we retain only the best matching image and remove the others.

Similarly, for the NMF representations, we compare $\mathbf{L_q}$ against $\mathcal{L}$, and represent the top η matches as the set $\mathcal{V}_{NMF} = \{\ell_1, \ell_2, ..., \ell_\eta\}$. The two hypotheses, $\mathcal{V}_{PCA}$ and $\mathcal{V}_{NMF}$ can be combined to get a more refined set of top-η matches. Later, we provide a heuristic algorithm for this purpose.

3.2. Determining the model order

An important consideration in our scheme is the model order, or the number of PCA/NMF factor loadings, or the number of feature clusters. This is the value of k (Section 2) needed for accurate matching. If k is too small, the matching accuracy will suffer. If it is too large, then the transmission and computational overhead of the protocol can become prohibitive. A natural way to ascertain the model order is to examine the singular values of $\mathbf{M}_q$. If the first k singular values are large, while the remaining singular values are small, then it makes sense to truncate the model order to k. This is indeed the case for photographs of many natural and urban scenes. This motivates a technique, originally used for determining the information content in econometric data in [18], which we describe briefly below.

For an image containing T-dimensional features extracted from N key points, the *Information Content* captured in $k < \min(T, N)$ principal components is given by

$$I(k) = \ln\left(V(k, \mathbf{F}_N^{(k)})\right) + k\left(\frac{T+N}{TN}\right) \ln\left(\frac{TN}{T+N}\right)$$

where $V(k, \mathbf{F}_N^{(k)}) = \min_{\hat{\mathbf{H}}} \frac{1}{TN} \sum_{i=1}^{N} \left\|\mathbf{d}_i - \hat{\mathbf{H}}^{(k)} \mathbf{f}_i^{(k)}\right\|_2^2$

Here, $\mathbf{d}_i$ is the ith descriptor in the matrix $\mathbf{M}$ of stacked descriptors, $\hat{\mathbf{H}}^{(k)}$ is the factor loading matrix obtained by assuming model order as k, $\mathbf{f}_i^{(k)}$ is the ith column of the factor matrix, $\mathbf{F}_N^{(k)}$. In the above relations, $\hat{\mathbf{H}}^{(k)}$ and $\mathbf{F}_N^{(k)}$ represent the factor loading matrix and the factor matrix obtained using PCA in subsection 2.2 under the assumption that the model order was k. The estimate of correct model order is given by $k^* = \arg\min_k I(k)$.

The above development allows us to estimate the order for PCA-based feature compression. To estimate the order for NMF-based feature compression, we reason as follows: Suppose, we had performed k-means clustering of the image features, i.e., the columns of $\mathbf{M}$. We know that the subspace spanned by k^* cluster centroids is the same as the subspace spanned by the first $k^* - 1$ columns of the PCA factor loading matrix $\mathbf{H}$ [16]. Furthermore, it has been remarked that NMF factor loadings, i.e., the columns of $\mathbf{L}$ are closely related to the k-means cluster centroids [15, 17]. Because of this relationship between PCA factor loadings, k-means cluster centroids, and NMF factor loadings, we choose the same model order k^* for PCA-based and NMF-based feature compression.

3.3. Comparing the query and database images

For PCA representations, the server's task is to find a member of the set $\mathcal{H}$ that best matches the query factor loading matrix $\mathbf{H}_q$. For NMF representations, it is to find a member of the set $\mathcal{L}$ that best matches the query factor loading matrix $\mathbf{L}_q$. We now consider two matching criteria. Let $\mathbf{A} \in \mathbb{R}^{T \times k_a}$ and $\mathbf{B} \in \mathbb{R}^{T \times k_b}$ be the two matrices to be compared. In our scheme, $\mathbf{A}$ might correspond to either $\mathbf{L}_q$ or $\mathbf{H}_q$ and $\mathbf{B}$ might belong to either $\mathcal{L}$ or $\mathcal{H}$ respectively.

The first metric we consider is the angle between subspaces spanned by $\mathbf{A}$ and $\mathbf{B}$ [22]. Let $\mathbf{P_A}$ and $\mathbf{P_B}$ be the projection matrices for $\mathbf{A}$ and $\mathbf{B}$ respectively, thus $\mathbf{P_A} = \mathbf{A}\left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top$. Then, the angle between the subspaces spanned by $\mathbf{A}$ and $\mathbf{B}$ is given by,

$$\angle(\mathbf{A}, \mathbf{B}) = \cos^{-1}\left(\|\mathbf{P_A}\mathbf{P_B}\|_2\right). \tag{1}$$

The second metric we consider is based on the maximum correlation between the columns of $\mathbf{A}$ and $\mathbf{B}$. To obtain this, construct a matrix $\mathbf{S} \triangleq \mathbf{A}^\top \mathbf{B} \in \mathbb{R}^{k_a \times k_b}$. Then obtain the maximum value along each column of $\mathbf{S}$ and write the vector $\mathbf{s}_{max}$ such that $\mathbf{s}_{max} = [s_1, \ldots, s_{k_b}]$, where, s_j is the maximum value in the jth column of $\mathbf{S}$. The correlation score is then given by

$$\text{score}(\mathbf{A}, \mathbf{B}) \triangleq \sum_{l=1}^{k_b} s_l. \tag{2}$$

3.4. Combining the PCA and NMF hypotheses

We now describe how to improve the retrieval accuracy by combining the hypothesis of the matching image obtained via the NMF and PCA-based feature clustering approaches. We choose the NMF retrieval result as the primary hypothesis and the PCA retrieval result

as the secondary hypothesis. Using the notation from Section 3.1, we have $\mathcal{V}_{pri} = \mathcal{V}_{NMF} = \{\ell_1, \ell_2, ..., \ell_\eta\}$ and $\mathcal{V}_{sec} = \mathcal{V}_{PCA} = \{h_1, h_2, ..., h_\eta\}$. To combine the primary and secondary hypothesis, an iterative algorithm is designed. The order of any pair of retrieved objects in the primary list is reversed if both the following conditions are satisfied: (a) The pair of objects appear in the reverse order in the secondary list, and (b) if the gap between the objects in the primary list is exceeded by $0 \leq \alpha \leq \eta$ places in the secondary list. The parameter α serves as a relative weighting factor for the primary and secondary hypotheses. Increasing α favors the primary hypothesis, and setting $\alpha = \eta$ completely ignores the secondary hypothesis. A step-by-step procedure for combining the primary and secondary hypotheses is provided in Algorithm 1.

Algorithm 1: Combining PCA and NMF retrieval hypotheses.

Data: α , $\mathcal{V}_{pri}$, $\mathcal{V}_{sec}$
Result: Resorted $\mathcal{V}_{pri}$
1   $i = 1, j = 1, permutations = 1$;
2   **while** *permutations = 1* **do**
3     $permutations = 0$;
4   **while** $i < \eta/2$ **do**
5   **while** $j <= \eta - i$ **do**
6   Find objects $h_a, h_b \in \mathcal{V}_{sec}$ that correspond to objects ℓ_i and $\ell_{i+j} \in \mathcal{V}_{pri}$;
7   **if** $a + \alpha + j > b$ **then**
8   switch ℓ_i and ℓ_{i+j};
9           $permutations = 1$
10   **end**
11         $j = j + 1$
12   **end**
13       $i = i + 1$
14   **end**
15 **end**

---

4. COMPLEXITY OF SERVER-BASED MATCHING

We assume that NMF-based and PCA-based feature clustering has already been performed for all K images in the server's database. Suppose that there is an average of N keypoint-based features per image. To compute the angle between subspaces, we multiply two $T \times T$ projection matrices, and compute matrix norms. This incurs $O(T^3)$ complexity per image, resulting in a total complexity of $O(KT^3)$. On the other hand, computing the pairwise correlation between T-length columns of factor loading matrices with model order k_p and k_q respectively, incurs $O(Tk_pk_q)$ complexity. Writing $k = \max(k_q, k_p)$, the total complexity for image matching based on pairwise correlations is $O(TKk^2)$. In our experiments, described below, we find that the angle between subspaces metric gives higher accuracy for NMF-based features, while both similarity metrics work for PCA-based features. To reduce the overall complexity, we use the correlation metric for PCA-based features first to obtain η matches, and then use the angle between subspaces metric with NMF-based features only on those η matching images. This brings the total complexity to $O(TKk^2 + \eta T^3)$. Effectively, the combined scheme (described above) amounts to performing PCA-based matching for a given list length η, reordering that list using NMF-based matching, and then *selectively reversing* some of the reordering based on the parameter α. Note that, as η is very small (usually less than 20), we ignore the extra complexity of Algorithm 1.

Scheme	Complexity
Correlation between columns, given by (2)	$O\left(Kk^2\right)$
Angle between subspaces, given by (1)	$O\left(KT^3\right)$
Proposed combined PCA + NMF scheme	$O\left(TKk^2 + \eta T^3\right)$
Using (2) without feature clustering	$O\left(KTN^2\right)$
Methods described in [23], [7], [10]	$O\left(KN^2\right)$

Table 1. Computational complexity of various schemes.

Retrieval accuracy based on different comparison measures

- NMF, Correlation b/w columns
- PCA, Correlation b/w columns
- NMF, ∠ b/w subspaces
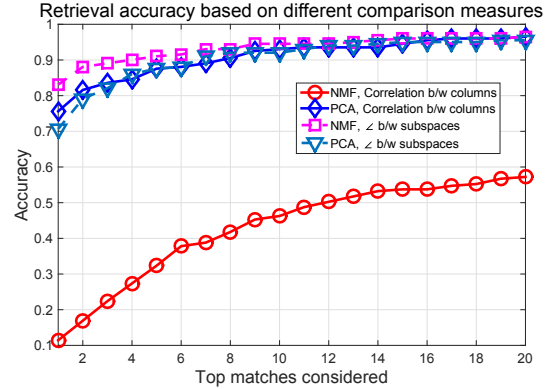- PCA, ∠ b/w subspaces

Fig. 3. PCA-based retrieval has almost the same accuracy with both similarity metrics. For NMF-based retrieval, the angle between subspaces metric is significantly more accurate.

In comparison (See Table 1), approaches that do not reduce the number of features incur much higher matching complexity. For instance, to obtain the top-1 match, the methods using direct matching of SIFT features, such as [10], incur a complexity of $O(KN^2)$. To see why the proposed approach is significantly less complex, recall that $N \gg k$ and $N \gg T$. E.g., in our experiments, $T = 128$ for SIFT vectors, the average value of k is 25, while $N = 2253$.

5. EXPERIMENTS

We use the Zurich Buildings Database (ZuBuD), [19, 24], which consists of images of 201 buildings, each captured from 5 different viewpoints. The first image of each building was selected as the query image while the remaining 4 were regarded as database images. The accuracy of retrieval, defined as the probability of a correct match, is used as a performance metric. SIFT descriptors – 128 dimensions per descriptor – were extracted from each image using the algorithm proposed in [1] with the default recommended parameters. The factor loading matrix, denoted as $\mathbf{H}$, in the case of PCA, and $\mathbf{L}$, in the case of NMF, was constructed for each image based on its SIFT descriptors, as described in Section 2.2 and Section 2.3, respectively. The $\mathbf{H}_q$ and $\mathbf{L}_q$ for a query image are then matched against the corresponding matrices in the database collections $\mathcal{H}$ and $\mathcal{L}$. The two similarity measures described in Section 3.3 are used to determine the correct match. The resulting average matching accuracy for all 201 objects is shown in Figure 3 with each similarity criterion.

It can be seen from Figure 3 that for PCA-based matching, using $\mathbf{H}_q$ against $\mathcal{H}$, both metrics work well. The accuracy is slightly better when correlation amongst the columns is used as a similarity metric. However, this metric gives poor accuracy for NMF-based matching, using $\mathbf{L}_q$ against $\mathcal{L}$. This is in contrast to the superior
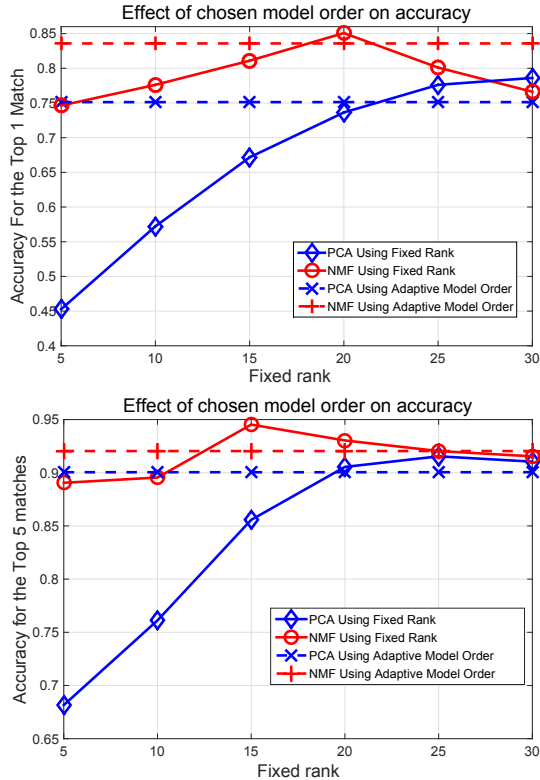
Fig. 4. Estimating the model order according to Section 3.2, incurs little or no penalty with respect to the best possible model order.

Fig. 5. Matching accuracy versus quantization rate (top 20 matches).

Fig. 6. Varying α changes the relative weights of the primary (NMF) and secondary (PCA) hypotheses, altering overall performance.

performance observed in [15]. We conjecture that this difference in performance is due to the difference in the type of data. In [15], the descriptors are constructed out of successive frames of a video sequence, while here, the descriptors are constructed out of vastly differing viewpoints of the same scene. As shown in Figure 3, the NMF-based matching gives much higher accuracy when the similarity metric used is the angle between the subspaces. Measuring the angle between subspaces spanned by the NMF factor loadings of the query and database images, is akin to measuring the discrepancy between the subspaces spanned by the centroids of the query features and the database features. In all subsequent experiments, the comparison of $\mathbf{L}_q$ with $\mathcal{L}$ is carried out using the angle between subspaces and the comparison of $\mathbf{H}_q$ with $\mathcal{H}$ is carried out by evaluating the maximum correlation between the columns of the matrices.

The effectiveness of the model order estimation from Section 3.2 is examined in Figure 4. Here, the rank of $\mathbf{H}$ and $\mathbf{L}$ for both query and server images was fixed at varying levels (x-axis). The retrieval accuracy was compared against that obtained using the estimated model order (horizontal lines). Using the estimated model order incurs little or no performance penalty relative to the fixed order schemes. Evidently, the method of Section 3.2 provides a reasonable estimate of the model order, and consequently, the amount of query information sent to the server. The average model order, i.e., the average number of descriptors, for the ZuBuD images was 24.5980.

Next, we examine the impact of quantizing the elements of $\mathbf{H}$ and $\mathbf{L}$. Quantization limits the amount of query information uploaded to the server and reduces the memory required to store the image descriptors at the server. In Figure 5, we examine the matching accuracy for different levels of fixed-rate quantization. Using
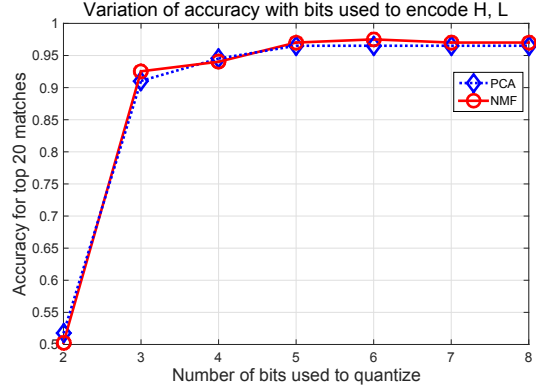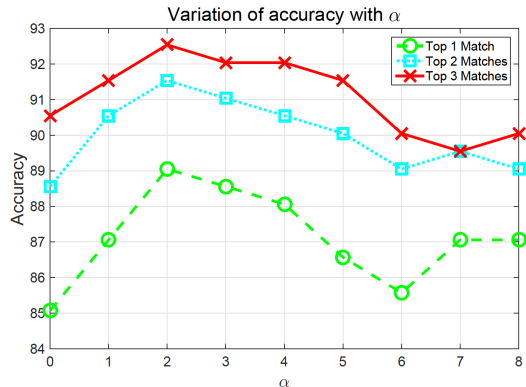
more than 5 bits to quantize the entries in $\mathbf{H}_q, \mathbf{L}_q, \mathcal{H}$ and $\mathcal{L}$ does not lead to further gains in matching accuracy. Thus, we employ 5-bit quantization to encode $\mathbf{H}_q, \mathbf{L}_q, \mathcal{H}$ and $\mathcal{L}$ in subsequent experiments. At this quantization level, the average size of the payload (per image) sent from the client to the server was 3.84 kilobytes for $\mathbf{H}_q$ and $\mathbf{L}_q$ combined. In comparison, the method of [10] requires only 2.5 kilobytes per image, but incurs a server-based complexity quadratic in the number of SIFT features, as described in Section 4.

Using NMF and PCA individually leads to matching accuracies of 83% and 75% respectively, for the top 1 match. Now, we choose $\eta = 20$ and set NMF as the primary hypothesis, and PCA as the secondary hypothesis. The accuracy of the combined scheme is plotted in Figure 6 for the top 1, 2, and 3 matches, as a function of the parameter α. Recall that α serves to refine the primary hypothesis using the secondary hypothesis in Algorithm 1. We find that $\alpha = 2$ leads to the best performance, though retrieval accuracy does not decrease monotonically for $\alpha > 2$. Finally, with this combination of $\alpha = 2$, 5-bit quantization, and the above similarity metrics, we examine the matching accuracy of all feature clustering techniques for the top-1 to top-20 matches in Figure 7. The combined scheme outperforms the two individual approaches, leading to a top 1, top 2 and top 3 matching accuracy of 89.05%, 91.54% and 92.54% respectively.
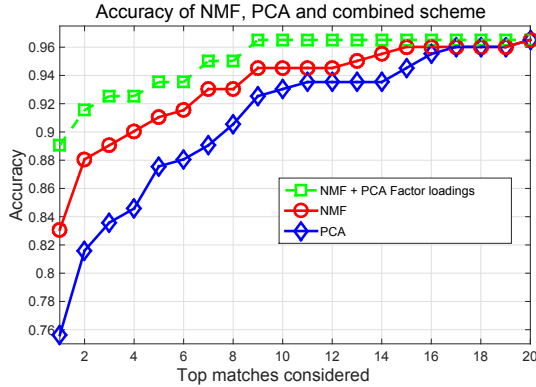
Fig. 7. The combined scheme gives more accurate retrieval than the separate PCA-based or sparse NMF-based approaches.

6. CONCLUSIONS

PCA and sparse NMF were explored for feature clustering, i.e., reduction of the number of scale-invariant features extracted for content-based image retrieval. A measure of information content, previously used in econometrics, was employed to estimate the number of descriptors to be sent by the client device. For a database of urban images, combining the PCA and NMF approaches provides a top-3 matching accuracy of 92.54%, which is competitive with previous work, but incurs significantly lower matching complexity due to feature clustering. Compared to pairwise matching based on thousands of native SIFT features per image, our method needs only about 25 PCA and NMF factor loadings per image. In ongoing work, we are extending this approach to other feature spaces (e.g., BRISK, FREAK, etc.), studying new theoretically motivated ways to estimate the model order, and examining new applications based on image classification.

7. REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. European Conf. Computer Vision (ECCV)*, pp. 404–417, Springer, 2006.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, IEEE, 2005.

[4] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, Y. Reznik, R. Grzeszczuk, and B. Girod, "Compressed histogram of gradients: A low-bitrate descriptor," *Intl. Journal of Computer Vision*, vol. 96, pp. 384–399, 2012.

[5] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. IEEE Intl. Conf. Computer Vision (ICCV)*, pp. 2548–2555, 2011.

[6] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *CVPR*, pp. 510–517, 2012.

[7] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *CVPR*, vol. 2, pp. II–506, 2004.

[8] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proc. ACM Symposium on Theory of Computing*, pp. 604–613, 1998.

[9] C. Yeo, P. Ahammad, and K. Ramchandran, "Coding of image feature descriptors for distributed rate-efficient visual correspondences," *Intl. Journal of Computer Vision*, vol. 94, pp. 267–281, 2011.

[10] M. Li, S. Rane, and P. Boufounos, "Quantized embeddings of scale-invariant image features for mobile augmented reality," in *Proc. IEEE Multimedia Signal Processing Workshop (MMSP)*, pp. 1–6, 2012.

[11] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *CVPR*, pp. 1 –8, 2008.

[12] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in Neural Information Processing Systems*, pp. 1753–1760, 2009.

[13] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local images descriptors into compact codes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[14] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "LDA-Hash: Improved matching with smaller descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, pp. 66 –78, Jan. 2012.

[15] H. Mansour, S. Rane, P. Boufounos, and A. Vetro, "Video querying via compact descriptors of visually salient objects," in *Proc. IEEE Intl. Conf. Image Processing (ICIP)*, pp. 2789–2793, 2014.

[16] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proc. Intl. Conf. Machine Learning (ICML)*, pp. 29–37, ACM, 2004.

[17] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proc. Intl. Conf. Knowledge Discovery and Data Mining (SIGKDD)*, pp. 126–135, ACM, 2006.

[18] J. Bai and S. Ng, "Determining the number of factors in approximate factor models," *Econometrica*, vol. 70, no. 1, pp. 191–221, 2002.

[19] H. Shao, T. Svoboda, and L. Van Gool, "Zubud-Zurich Buildings Database for image-based recognition," *Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, Tech. Rep*, vol. 260, p. 20, 2003.

[20] N. Gillis, "The why and how of nonnegative matrix factorization," *Regularization, Optimization, Kernels, and Support Vector Machines*, pp. 257–282, 2014.

[21] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research.*, vol. 5, pp. 1457–1469, 2004.

[22] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), 2000.

[23] V. Chandrasekhar, M. Makar, G. Takacs, D. Chen, S. S. Tsai, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, and B. Girod, "Survey of SIFT compression schemes," in *Proc. Intl. Workshop Mobile Multimedia Processing*, pp. 35–40, 2010.

[24] H. Shao, T. Svoboda, T. Tuytelaars, and L. Van Gool, "HPAT indexing for fast object/scene recognition based on local appearance," in *Image and Video Retrieval: Proc. Intl. Conf CIVR*, pp. 71–80, Springer, 2003.