Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2004-075
AIM-2004-027

November 22, 2004

Efficient Image Matching with Distributions of Local Invariant Features

Kristen Grauman and Trevor Darrell

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Abstract

Sets of local features that are invariant to common image transformations are an effective representation to use when comparing images; current methods typically judge feature sets' similarity via a voting scheme (which ignores co-occurrence statistics) or by comparing histograms over a set of prototypes (which must be found by clustering). We present a method for efficiently comparing images based on their discrete distributions (bags) of distinctive local invariant features, without clustering descriptors. Similarity between images is measured with an approximation of the Earth Mover's Distance (EMD), which quickly computes the minimal-cost correspondence between two bags of features. Each image's feature distribution is mapped into a normed space with a low-distortion embedding of EMD. Examples most similar to a novel query image are retrieved in time sublinear in the number of examples via approximate nearest neighbor search in the embedded space. We also show how the feature representation may be extended to encode the distribution of geometric constraints between the invariant features appearing in each image. We evaluate our technique with scene recognition and texture classification tasks.

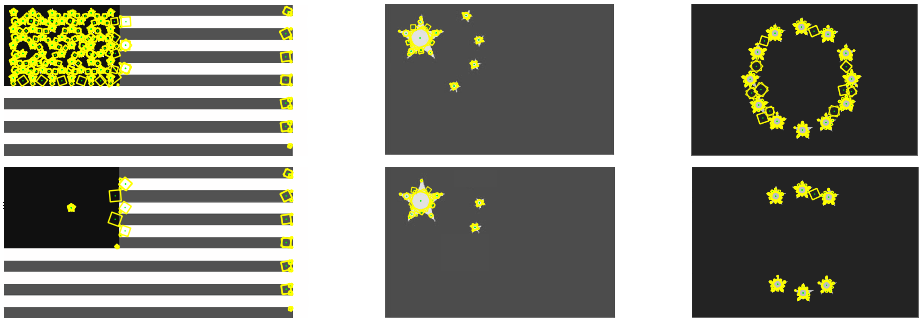


Figure 1: Images with detected SIFT features. Under voting-based matching schemes, the two versions of each flag are indistinguishable, since a query image with fewer stars will vote equally for a flag with fewer stars as it will for the flag with all the stars. Under our distribution-based similarity measure, the flags with different numbers of stars are considered distinct without any additional geometric verification.

1. Introduction

Image matching, or comparing images in order to obtain a measure of their similarity, is an important computer vision problem with a variety of applications, such as content-based image retrieval, object and scene recognition, texture classification, and video data mining. The task of identifying similar objects and scenes within a database of images remains challenging due to viewpoint or lighting changes, deformations, and partial occlusions that may exist across different examples. Global image statistics such as color histograms or responses to filter banks have limited utility in these real-world scenarios, and often cannot give adequate descriptions of an image's local structures and discriminating features. Instead, researchers have recently turned to representations based on local features that can be reliably detected and are invariant to the transformations likely to occur across images (i.e., photometric or various geometric transformations).

A number of recent matching techniques extract the invariant local features for all images, and then use voting to rank the database images in similarity: the query image's features vote independently for features from the database images (where votes go to the most similar feature under some distance, e.g., L_2), possibly followed by a verification step to account for spatial or geometric relationships between the features (e.g., [10, 9, 17, 15]). When sufficiently salient features are present in an image, matching methods based on the independent voting scheme will successfully identify good matches in the database. However, using a query image's features to independently index into the database ignores possibly useful information that is inherent in the co-occurrence of a set of distinctive features, and it fails to distinguish between instances where an object has varying numbers of similar features (see Figure 1). Thus, the voting-based methods must filter through the initial feature matchings and follow up with geometric verification steps.

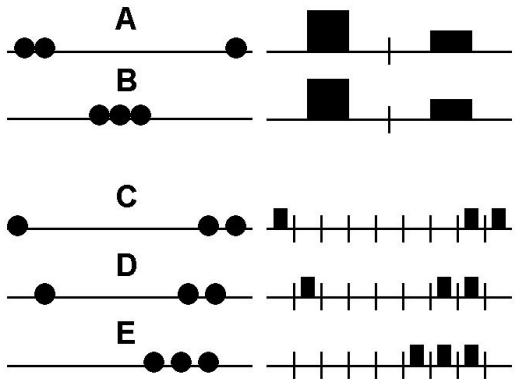


Figure 2: Measuring distances on distributions of quantized features with a bin-by-bin similarity measure is sensitive to bin size. If bins are too wide, discriminative ability is lost (Point sets A and B produce the same prototype frequencies in spite of their distinct distributions). If the bins are too narrow, features that are very similar are placed in separate bins where they cannot be matched (D and E are considered closer than C and D, which are perceptually more similarly distributed). Cross-bin measures such as EMD are an effective way to avoid the bin sensitivity issue, since features are matched based on their similarity to one another, not their assigned bin placement.

Other matching approaches have taken feature co-occurrences into account by using vector quantization to represent each image by its frequency of prototypical feature occurrences, then comparing the weighted histograms with a bin-by-bin distance measure [16]. However, while mapping detected features to a set of global prototypes may greatly aid in efficiency for matching the distribution of features, such approaches assume that the space of features that will be encountered in novel images is known a priori when generating the prototypes, and they face the standard difficulty of properly choosing the number of cluster centers to use. Moreover, it has been shown that bin-by-bin measures (e.g., L_p distance, normalized scalar product) are less robust than cross-bin measures (e.g., the Earth Mover's Distance (EMD), which allows features from different bins to be matched) for capturing perceptual dissimilarity between distributions [13] (see Figure 2). Methods that cluster features on a per-example basis still must choose a quantization level and risk losing discrimination power when that level is too coarse [13, 8].

To address these issues, we propose a matching technique that compares images on the basis of their actual distributions of local invariant features (see Figure 3). We also show how spatial neighborhood constraints may be incorporated directly into the matching process by augmenting features with invariant descriptions of their geometric relationship with other features in the image. We measure similarity between two discrete feature distributions [1] with an approximation of EMD—the measure of the amount of work necessary to transform one weighted point set into another. To match efficiently, we use a low-distortion

[1] We use the words "bag" or "discrete distribution" interchangeably to refer to an unordered collection of features that may contain duplications.

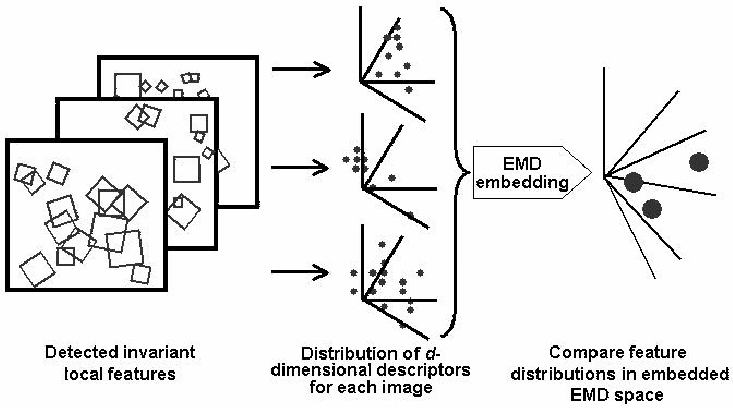


Figure 3: Comparing local invariant feature distributions in embedded EMD space.

embedding of EMD into a normed space which reduces a complex, correspondence-based distance to a simple, efficiently computable norm over very sparse vectors. The EMD embedding also enables the use of approximate nearest neighbor (NN) search techniques that guarantee query times that are sublinear in the number of examples to be searched [4, 2].

We demonstrate our method in the contexts of scene recognition and texture classification. We show the advantage of using the joint statistics when matching with local features as opposed to matching each feature independently under a voting scheme, and we investigate the benefits of matching the actual detected features as opposed to vector-quantized versions of them. We also show the impact of including higher-order invariants (spatial neighborhood or geometric constraints) at the time of initial matching.

2. Related Work

In this section, we review related work regarding image matching techniques based on local invariant features, the use of EMD in vision for matching tasks, and the use of approximate EMD for similarity search.

Our method compares images based on the EMD between their distributions (or bags) of local invariant features augmented with geometric constraints. Recently a number of authors have used local image descriptors extracted at stable, invariant interest points to judge image similarity or to localize an object within an image. In [10], a voting-based indexing method is given: each scale-invariant interest point in a query image votes for images in the database containing an interest point within

a thresholded distance from itself. Similarly, the authors of [17] use affine moment invariants to independently cast votes for similar images in the database. The method for matching scenes given in [15] first uses voting to identify candidate matches, then applies a series of steps to verify geometric consistency within larger neighborhoods. In [9], an object's keypoints are matched independently via a thresholded approximate similarity search to all of the keypoints extracted from the database images; clusters of three matches that agree in pose indicate the object's presence in a database image.

The authors of [16] apply text retrieval techniques to image matching; vector quantization (VQ) is applied to affine invariant regions collected from video data, and each image is represented by a fixed-length vector of weighted frequencies of the pre-established feature prototypes. Then, images in the database are ranked in similarity to a user-segmented query region based on their frequency vectors' normalized scalar product. In [8], textures are recognized based on their histogram of prototypical affine-invariant features, as determined by an exhaustive NN search with exact EMD. The authors of [7] cluster invariant descriptors with EM and assign class labels to descriptors in novel texture images, which are refined with a relaxation step that uses neighborhood co-occurrence statistics from the training set.

Our work differs from the voting-based techniques in that we do not match features within an image independently, but instead consider the joint statistics of the invariant features as a whole when matching. A naive exhaustive search for NN features makes the voting technique computationally prohibitive; even if an approximate NN technique is used to find close features for voting, our method requires fewer distances to be computed. Unlike the methods of [16] and [8], where VQ is applied to features to obtain a frequency vector, our method represents an image with its actual distribution of features. Furthermore, our method incorporates higher-order features directly into the distribution matching, whereas other techniques attempt pose verification or spatial consistency tests in a separate, secondary processing stage [9, 16, 7, 14, 15].

To incorporate local geometric constraints, we augment each feature with relative information about the geometry of other features in the image. In the geometric hashing approach of [6], all triples of interest points are used to map the remaining points to affine planar coordinates, and then pairs of frames giving the same coordinates to the points are sought. Instead of voting on point tuples, however, our method matches distributions of feature tuples.

EMD was first used in vision in [12] to measure the distance between intensity images. More recently EMD has been used for color- or texture-based similarity in [13], and for comparing vector-quantized signatures of affine invariant features in texture images [8]. Exact EMD is computed with linear programming, and its complexity is exponential in the number of points per set in the worst case. An embedding of EMD into L_1 and the use of Locality-Sensitive Hashing (LSH) for

approximate NN was shown for the purpose of color histogram-based image retrieval in [4], and the embedding was used for matching shapes based on contour features in [3].

The main contributions of this paper are (a) an efficient image matching algorithm that compares raw distributions of invariant appearance features and exploits an EMD embedding and approximate NN search, (b) a study of the tradeoffs between voting schemes that index with features independently and the use of joint statistics of local features, and (c) a feature representation suitable for distribution-based matching that does not require clustering and incorporates higher-order spatial constraints.

3. Approach

We have developed an efficient image matching technique that compares images in terms of their raw distributions of local invariant appearance features annotated with spatial constraints using approximate EMD. In this section we will describe the representations we use, the mechanism by which we efficiently compare them, and the way in which our method seamlessly incorporates higher-order geometric constraints.

3.1. Matching with Distributions of Local Invariant Features

Image features that are stable across varying scales, rotations, illuminations, or viewpoints are desirable for recognition and indexing tasks, since the same object is likely to repeat these invariant features in varying real-world imaging conditions. An interest operator is generally applied to the image to detect stable or distinctive points, and then a local descriptor is extracted from the patch or ellipse around each interest point. In this work, we employ the Scale Invariant Feature Transform (SIFT) interest operator of [9]–which has been shown to be resistant to common image deformations–and the low-dimensional local gradient-based descriptor called PCA-SIFT given in [5]. These features are scale and rotation invariant, and partially invariant to illumination and camera viewpoint. We represent each grayscale image $\mathbf{I}_i$ by the bag $\mathbf{B}_i$ of PCA-SIFT descriptors extracted from its interest points $\mathbf{p}_j$: $\mathbf{B}_i = \{\mathbf{s}^{\mathbf{p}_1}, \ldots, \mathbf{s}^{\mathbf{p}_{n_i}}\}$, where each $\mathbf{s}^j$ is a d-dimensional descriptor extracted from one of the n_i interest points in image $\mathbf{I}_i$. Other interest operators or descriptors are of course possible.

EMD provides an effective way for us to compare images based on these discrete distributions. For a metric space (X,D)

and two n-element sets $\mathbf{B}_p, \mathbf{B}_q \subset X$, the EMD is the minimum cost of a matching π between $\mathbf{B}_p$ and $\mathbf{B}_q$:

$$EMD(\mathbf{B}_p, \mathbf{B}_q) = \min_{\pi:\mathbf{B}_p \to \mathbf{B}_q} \sum_{\mathbf{s} \in \mathbf{B}_p} D(\mathbf{s}, \pi(\mathbf{s})). \quad (1)$$

Comparing bags of local features with EMD is essentially measuring how much effort would be required to transform one bag into the other. The measure of this effort is based on establishing the correspondence between two images' unordered descriptive local features that results in the lowest possible overall matching cost, where matching cost is defined by a ground distance D between two local features (e.g., the L_2 norm). EMD performs partial matching in the case that the two sets have unequal total weight; the distance is then the minimum work needed to cover the mass in the lower-weight set with the mass in the higher-weight one.

Since an object or scene will exhibit a large number of the same local invariant features across varying viewpoints and illuminations, this is a useful way to judge the overall similarity of images for the purpose of scene or texture recognition. In addition, when we incorporate higher-order geometric constraints over the detected features (described below), it is possible to more strictly judge similarity of the distributions based on the features' relative spatial layout.

However, exact EMD is exponential in the number of features per point set. Since we can expect to detect on the order of thousands of invariant features in a textured image of moderate resolution, this is a critical issue. Previously, researchers applying EMD have mapped raw image features to prototypes or cluster centers in order to get around EMD's computational burden; the input to EMD is then a set of prototypes weighted by their frequency in the image [13, 8]. However, by replacing input features with prototypes, such approaches discard some discriminating content present in the unique detected features, and they require some means of choosing the appropriate number of clusters (histogram bins) to impose (see Figure 2).

Instead, we use the low-distortion EMD embedding given in [4] to embed the problem of correspondence between sets of local features into L_1. The embedding f maps the unordered point sets into the normed space L_1, such that the L_1 distance between the resulting embedded vectors is comparable to the EMD between the unordered sets themselves:

$$\frac{1}{C} EMD(\mathbf{B}_p, \mathbf{B}_q) \leq ||f(\mathbf{B}_p) - f(\mathbf{B}_q)||_{L_1} \leq EMD(\mathbf{B}_p, \mathbf{B}_q), \quad (2)$$

where C is the distortion factor bounded by $O(\log(\Delta))$ for a space of underlying diameter Δ. Informally, the embedding computes and concatenates several weighted, randomly translated histograms of decreasing resolution for a given point set

Procedure 1 To prepare an image dataset for matching:

Given: A dataset of N images $\{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$, random LSH functions $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_t]$, and randomly translated EMD embedding grids $\mathbf{G}_l$, each with side lengths 2^l, $l = -1, \ldots, \log(\Delta)$, and neighborhood radius r:

1: **for all** $i = 1, \ldots, N$ **do**
2: Detect in $\mathbf{I}_i$ distinct stable image points $\{\mathbf{p}_1, \ldots, \mathbf{p}_{n_i}\}$ with interest operator.
3: Extract a descriptor $\mathbf{s}_j$ with desired invariance from image patch centered at each $\mathbf{p}_j$ to form unordered bag of features $\mathbf{B}_i = \{\mathbf{s}^{\mathbf{p}_1}, \ldots, \mathbf{s}^{\mathbf{p}_{n_i}}\}$.
4: Apply EMD embedding:
$f(\mathbf{B}_i) = \left[\frac{1}{2}\mathbf{G}_{-1}(\mathbf{B}_i), \ldots, 2^l\mathbf{G}_l(\mathbf{B}_i), \ldots, \Delta\mathbf{G}_{\log(\Delta)}(\mathbf{B}_i)\right]$, producing one sparse vector.
5: Insert vector $f(\mathbf{B}_i)$ into hash tables $\mathbf{H}$, and record its hash buckets $[b_1, \ldots, b_t]$.
6: **end for**

Procedure 2 To find similar images among the prepared dataset images:

Given: Image $\mathbf{I}_q$ with bag of features $\mathbf{B}_q$ and embedding $f(\mathbf{B}_q)$:

1: Hash into $\mathbf{H}$ with $f(\mathbf{B}_q)$, yielding hash bucket indices $[b_1, \ldots, b_t]$
2: **for all** $k = 1, \ldots, t$ **do**
3: Compute L_1 distance between $f(\mathbf{B}_q)$ and the W database embeddings $\{f(\mathbf{B}_1), \ldots, f(\mathbf{B})^W\}_k$ that share bucket b_k, $W << N$.
4: **end for**
5: Sort $\cup_{k=1}^{t}\left(\{f(\mathbf{B}^1), \ldots, f(\mathbf{B})^W\}_k\right)$ according to their L_1 distance to $f(\mathbf{B}_q)$ to obtain a ranked image list that includes r-neighbors, $[\mathbf{I}^1, \ldots, \mathbf{I}^W]$.

(see [4]). Once feature sets are mapped to a normed space, it is then possible to apply approximate NN techniques (e.g., LSH [2]) which greatly improve the efficiency of similarity search over large databases, making it possible to find similar examples by computing distances between an input and only a small portion of the database. See Procedures 1 and 2 for an outline of the matching process.[2]

Unlike voting-based matching schemes, where each salient feature of an image is considered independently when matching a query to database items, we consider the distribution of invariant features present in an image collectively. This lets us avoid having to set thresholds to determine whether a single-feature match is strong enough to qualify as a good match; the best match for a query image is simply the database image with the most similar joint distribution of features. We have also found that the information offered by the joint statistics of the feature appearances and geometric relationships can capture similarities between images that may be overlooked when voting on a per-feature basis (see Section 4).

3.2. Spatial Constraints

Note that the features we are using (SIFT, PCA-SIFT) do not individually contain explicit spatial information. Nonetheless, in many natural images it is common for a large portion of the detected stable features to overlap (see Figure 4). Thus each

[2]Note that when the dataset is small enough, it is possible to forgo the LSH step and exhaustively compute L_1 distances between a query's embedding and all embedded database examples. "Small enough" can be defined by the point where the overhead involved in hashing for LSH outweighs the gains in query search time it provides.

Figure 4: *Left*: Natural images often contain many partially overlapping distinctive invariant features, which causes their distribution of features to inherently encode information about the spatial configurations of features within local neighborhoods. Detected SIFT features are shown here. *Right*: For non-overlapping patches, features at the nearest interest points are used to encode spatial constraints.

feature descriptor that has some overlap does encode a spatial constraint in terms of the nearby features that occur with it.

For situations where the detected features are less likely to overlap, we would like to still enforce these spatial constraints. To do this, we augment the local invariant feature representation to include an encoding of the geometry of other interest points in relation to each given feature. The descriptor for each interest point is concatenated with invariant information about the configuration of its spatially nearest interest points in the image. Then, when these higher-order feature distributions are compared under EMD, the low-cost feature matching seeks to satisfy the additional constraints.

There are various possible geometric or neighborhood constraints to include. To designate simple proximity constraints between features, each feature $\mathbf{s}^{\mathbf{p}_j}$ is paired with its nearest-located feature in the image, $\mathbf{s}^{\mathbf{p}'_j}$, to form a new feature, $[\mathbf{s}^{\mathbf{p}_j}, \mathbf{s}^{\mathbf{p}'_j}]$. Additionally, the angle of separation θ between the two features' dominant orientations (as determined by the SIFT operator) can be incorporated to further constrain the orientation relationship between them, forming the new feature $[\mathbf{s}^{\mathbf{p}_j}, \mathbf{s}^{\mathbf{p}'_j}, \theta]$ (see Figure 4). Both result in a similarity-invariant descriptor, since the length ratios of two lines and the angle between two lines are invariant binary relations under similarity transformations [1]. To designate symmetry constraints, each feature is paired with its nearest neighbor in the image in terms of image patch descriptor (i.e., the most similar feature in appearance). The result in this case is an affine-invariant relation, provided the interest point detector and feature descriptor are invariant to affine transformations. Other constraints based on affine or projective invariants are possible but would require higher-order tuples.

4. Results

We have applied our method in two domains where efficient image matching is useful: scene recognition and texture classification.

4.1. Methodology

For each dataset, we use the *normalized average rank* $\bar{R}$ as a measure of matching performance:

$$\bar{R} = \frac{1}{NN_R}\left(\sum_{i=1}^{N_R} R_i - \frac{N_R(N_R-1)}{2}\right), \tag{3}$$

where R_i is the rank at which the ith relevant image is retrieved, N_R is the number of relevant images for a given query, and N is the number of examples in the database. The normalized rank is 0 for perfect performance (i.e., when all relevant images in the database are retrieved as a query's closest nearest neighbors), and it approaches 1 as performance worsens; a random retrieval results in a normalized rank of 0.5 [11]. We also report results in terms of the classification error rates when the k-nearest neighbors (k-NN) are used to vote on the query's label; the normalized rank is more comprehensive, but recognition error is sometimes a more intuitive measure of performance.

For each dataset, we compare our algorithm's performance with two other techniques: a voting technique and a prototypical-feature technique modeled on the "Video Google" method given in [16]. All three methods share the idea of representing images based on their sparse sets of invariant features, but they vary in the way that they judge similarity between the feature sets.

For the voting scheme, each feature in a query image is compared against all of the features extracted from database images, and then that query feature casts a vote for the database image containing the nearest neighbor feature in terms of L_2 distance. The database images are then ranked in similarity to the query based on the number of votes they have received. Note that we used an exact (exhaustive) search to determine each features' nearest neighbor in order to measure the voting technique's performance, but exhaustive search is computationally infeasible in practice. So the voting results should be considered an upper bound; in practice, an approximate-NN technique such as LSH or BBF [9] is used to make voting computationally tractable, but at some cost of matching error.

For the prototypical feature scheme, vector quantization is used to map all image feature descriptors to a discrete set of

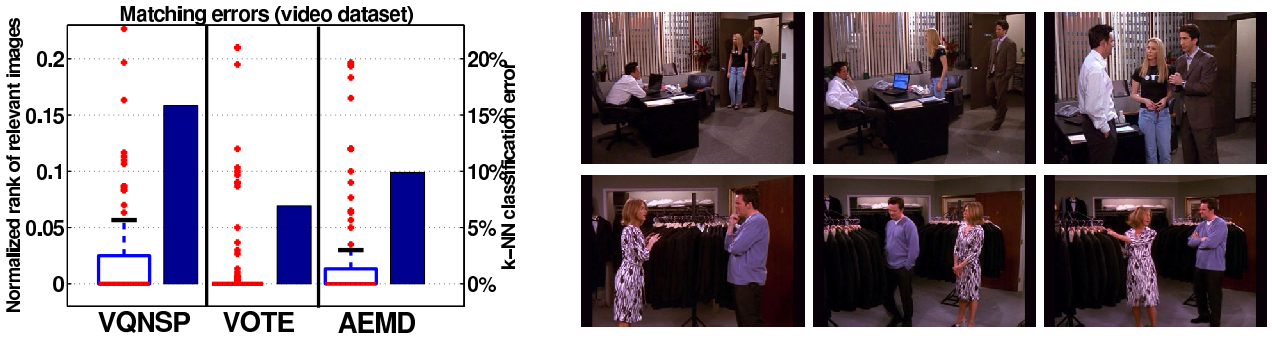


Figure 5: Matching performance comparison. *Left:* Distribution of normalized ranks of relevant images (boxplot on left for each method, axis on left) and k-NN classification error rates (dark bar on right for each method, axis on right) for 100 test examples from the *Friends* episodes dataset with ground truth labels. VQNSP denotes matching with the normalized scalar product applied to vector-quantized features, VOTE denotes voting with each feature independently, and AEMD denotes matching with approximate EMD on raw feature distributions (the proposed method). A normalized rank of zero signifies a perfect ranking, where all relevant images are returned as the closest nearest neighbors. Red line in boxes denotes median value; top and bottom of boxes denote upper and lower quartile values, respectively. Dashed lines show extent of rest of the data, pluses denote outliers. *Right:* Each row is an example of images from the ground truth dataset matched with our method. AEMD does nearly as well as voting with exhaustive NN search, but is much more efficient.

prototypes, which are found by running k-means on a subset of the database that includes images from each label. Each image is represented by a vector giving the frequency of occurrence of each prototype, weighted by the *term frequency - inverse document frequency*. The database images are then ranked in similarity to the query based on the normalized scalar product between their frequency vectors and the query's frequency vector. Our implementation is modeled on the video data mining method in [16]; we omit the "stop-list" and temporal feature tracking steps since we are matching static, nonsequential images in addition to video frames in these experiments.

To extract the SIFT and PCA-SIFT features in these experiments, we used the code that the authors of [9] and [5] have provided online. We use the first eight dimensions of the PCA-SIFT features as input to all methods, and on the order of 10^2 prototypes for the prototypical-feature method; these parameters were optimized for recognition performance on a held out set of examples.

4.2. Scene Recognition

Shot matching is a specific use of scene recognition where the goal is to automatically identify which video frames belong to the same scene in a film. To test our method in this regard, we used a dataset of images from six episodes of the sitcom *Friends*. We extracted one frame for every second of the video (so as to avoid redundancy in the database), for a total of 8,335 images.

With the approximate-NN technique LSH it is only necessary to compute L_1 distances between the query's EMD embedding and a small portion of the database embeddings. In this case, queries on average required only 480 distances to be computed, i.e., on average each image was compared to 5% of the database.

Figure 5 shows the matching performance of our method, voting, and prototype-feature matching on a ground truth subset of the *Friends* dataset containing 100 images that were hand-labelled with scene identity. These 100 images contain frames from 27 different scenes, with about four images from each scene; each image from the same scene is taken from a different camera shot so that the viewpoints and image content (actors' positions, etc.) vary. We used leave-one-out cross validation (LOOCV) for these ground truth tests in order to maximize the use of the labelled data. This dataset required 100 prototypes, and we did not use additional spatial constraints on the features here.

Using the k-NN under each method as a classifier of scene identity, voting classifies 93% correctly, our method classifies 90% correctly, and the VQ approach classifies 84% correctly (k=3). This experiment indicates to us that the salient SIFT features were reliably extracted in each instance of a scene, making it possible for voting to be very successful. This seems reasonable; although the images have some viewpoint variation, due to the nature of the source–a TV sitcom set–they have consistent quality and illumination, and each scene is unique enough that discriminating features have some leverage under voting. However, this voting result did require exhaustive search for NN features, which is computationally prohibitive in practice. We would expect marginally worse performance from voting if an approximate method were used to find NN features, as the reduction in computational complexity does come at the cost of some accuracy. Our method does nearly as well as "perfect" voting, yet is much more efficient.

The relevant rank distribution is wider under the prototype-feature method (VQNSP), indicating that the quantization of the features adversely affects performance for this dataset. In our experiments, we found that VQNSP matching quality was fairly sensitive to the clustering that defined the prototypes (see Figure 6). The number of clusters can be thought of as the "bin size" for this method–more clusters means smaller bins. The quality of the matching varied substantially depending on both the number of clusters, as well as the random starting point of k-means. As shown in Figure 5, our method more consistently ranked the *Friends* images, and it does not require a parameter choice since it matches with the (un-binned) features themselves.

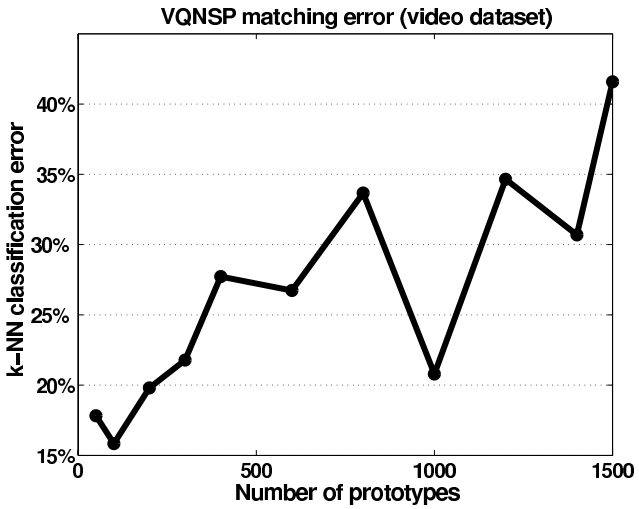


Figure 6: VQNSP matching performance is sensitive to the number of prototypes chosen.

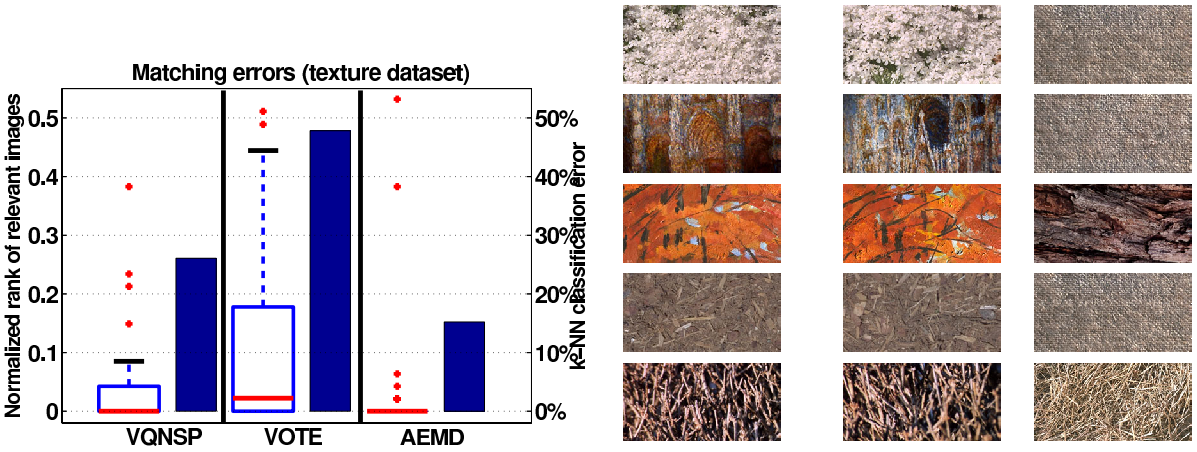
4.3. Texture Classification

Another useful application of image matching is texture classification. There are issues unique to comparing textures as opposed to the scenes compared in the above experiment; in particular, textures are often defined in terms of how local features or structures co-occur. Each instance of a texture is not necessarily from the same scene under different viewing conditions, but rather it is another sample of an underlying, nonuniform pattern. This makes texture matching a domain that is especially amenable to our method since it captures the joint statistics of invariant features. We ran experiments with the publicly available VisTex Reference database, which contains 168 images of textures under "real world" conditions, including both frontal and oblique perspectives and non-studio lighting.

Figure 7 shows the matching performance of the three methods for this dataset. We randomly selected a set of 25 VisTex examples.[3] This dataset required 700 clusters for VQNSP. Each image was split into halves, making 50 images, and again we tested with LOOCV. The goal in this test was for each query to match most closely with the other half of the texture image from which it originated. Note that most of the textures are mainly homogeneous at a high level, but that the two halves of each image still have significant variations, and different structures occur in each half.

While the methods capturing co-occurrence information (our method and VQNSP) assign rankings that are tightly distributed close to zero (the ideal normalized rank), voting fails to assign low ranks consistently to images of the same texture,

[3]The entire VisTex database was not used for the comparative study due to the computation time needed to get exact NN features for voting. Using all the textures, our method achieves a median normalized relevant rank of 0.003.



(a) Comparison of matching errors (b) Query (c) AEMD NN (d) VOTE NN

Figure 7: Matching performance comparison: distributions of the normalized rank of relevant images for three methods (a) and examples of the first NNs retrieved with our method (AEMD) and voting (b,c,d). The chart shows the rank distributions and k-NN classification errors for a ground truth matching test of 50 texture images randomly drawn from the VisTex database. Plotted and labeled as in the previous figure. See text for details. (Textures best viewed in color.)

resulting in a much wider distribution centered at 0.1372. While voting was successful for scene matching where distinct features were repeated, it breaks down for texture matching due to the variation of the features within different samples of the same texture. Voting suffers because it does not capture co-occurrence information that is critical for texture matching, and it risks casting excessive votes for images containing a repeated "generic" feature, as suggested by the example matches shown in Figure 7. AEMD also shows better texture classification performance than VQNSP.

We also evaluated the effect of explicit spatial constraints when matching with AEMD on this dataset. When using the paired features with angles discussed in Section 3.2, we found that overall matching performance on the database remained the same. However, for textures where local features sparsely covered the image and their patches did not have significant overlap, we found that the paired features did lead to improved performance. In 70% of images identified as having few overlapping features, the NN computed with paired features were more accurate. Figure 8 shows one such example: the tile texture is matched correctly under both representations for the first NN, but only the spatially constrained features can identify the higher resolution samples of this texture as matches, due to the "confuser" tile with similar individual boundary features but a different spatial layout.

(a) Query

(b) First three NN, raw (top) and paired (bottom)

Figure 8: An example where explicit spatial constraints via feature tuples with angles (bottom row) disambiguates textures that are not clear when matching with single features alone (top row).

4.4. Complexity

The EMD embedding vector resulting from an input feature set is high-dimensional, but very sparse; only $O(n\log(\Delta))$ entries are nonzero, where n is the number of features in an example, and Δ is the diameter of the feature space. The time required to embed one d-dimensional point set is $O(nd\log(\Delta))$. Thus, the computational cost of comparing two images' local feature distributions under approximate EMD is $O(nd\log(\Delta))+O(n\log(\Delta)) = O(nd\log(\Delta))$, the cost of embedding two point sets, plus an L_1 distance on the sparse vectors [4]. LSH reduces the time required to retrieve similar images to $O(sN^{(1/(1+\epsilon))})$, where N is the number of examples in the database, ϵ is the LSH parameter related to the amount of approximation of the normed distance between neighbors, and s is the number of nonzero entries in the sparse embedded vectors. In our experiments we set ϵ to 1, making the upper bound on the query time $O(sN^{\frac{1}{2}})$.

In comparison, to process one query, a voting scheme must perform n retrievals from a database with on the order of $N \times n$ items in order to match each of its d-dimensional features to the database, making a single query cost $O(dNn^2)$ if exact NN features are found. Even if an approximate similarity search technique is used for voting, query time still increases with both N and n. For VQ prototype-frequency methods, the query time has an upper bound of $O(kN)$, where k is the number of prototypes.

5. Conclusions and Future Work

We have developed an image matching method that offers a means of efficiently matching distributions of local invariant features, and we have demonstrated its advantages over voting and prototype-histogram techniques. We have also provided

a way to extend the feature representation to incorporate higher-order spatial constraints in distribution-based matching. The proposed algorithm is efficient, accurate, and does not require choosing a number of clusters.

In future work, we intend to explore ways to extend the method to better handle occlusion and clutter, to make use of low-level segmentation when extracting the feature distributions, and to evaluate the effect of the proposed local geometry constraints on a database that specifically has sparse, non-overlapping features.

References

[1] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*, chapter 18. Prentice Hill, Upper Saddle River, New Jersey, 2003.

[2] A. Gionis, P. Indyk, and R. Motwani. Similarity Search in High Dimensions via Hashing. In *Proceedings of the 25th Intl Conf. on Very Large Data Bases*, 1999.

[3] K. Grauman and T. Darrell. Fast Contour Matching Using Approximate Earth Mover's Distance. In *CVPR*, Washington D.C., June 2004.

[4] P. Indyk and N. Thaper. Fast Image Retrieval via Embeddings. In *3rd Intl Wkshp on Statistical and Computational Theories of Vision*, Nice, France, Oct 2003.

[5] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *CVPR*, Washington, D.C., June 2004.

[6] Y. Lamdan and H. Wolfson. Geometric Hashing: A General and Efficient Model-Based Recognition Scheme. In *ICCV*, Tarpon Springs, FL, 1988.

[7] S. Lazebnik, C. Schmid, and J. Ponce. Affine-Invariant Local Descriptors and Neighborhood Statistics for Texture Recognition. In *ICCV*, Nice, France, Oct 2003.

[8] S. Lazebnik, C. Schmid, and J. Ponce. A Sparse Texture Representation Using Affine-Invariant Regions. In *CVPR*, Madison, WI, June 2003.

[9] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, Jan 2004.

[10] K. Mikolajczyk and C. Schmid. Indexing Based on Scale Invariant Interest Points. In *ICCV*, Vancouver, Canada, July 2001.

[11] H. Muller, W. Muller, S. Marchand-Maillet, and T. Pun. Performance Evaluation in Content-Based Image Retrieval: Overview and Proposals. *Pattern Recognition Letters*, 22(5):593–601, 2001.

[12] S. Peleg, M. Werman, and H. Rom. A Unified Approach to the Change of Resolution: Space and Gray-level. *TPAMI*, 11(7):739–742, July 1989.

[13] Y. Rubner, C. Tomasi, and L. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *IJCV*, 40(2):99–121, 2000.

[14] C. Schmid. Constructing Models for Content-Based Image Retrieval. In *CVPR*, Kauai, HI, Dec 2001.

[15] F. Shaffalitzky and A. Zisserman. Automated Scene Matching in Movies. In *Proceedings, Challenge of Image and Video Retrieval*, London, U.K., July 2002.

[16] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *ICCV*, Nice, France, Oct 2003.

[17] T. Tuytelaars and L. Van Gool. Content-based Image Retrieval based on Local Affinely Invariant Regions. In *3rd Intl Conference on Visual Information Systems*, Amsterdam, the Netherlands, June 1999.