In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2005.

Efficient Image Matching with Distributions of Local Invariant Features

Kristen Grauman and Trevor Darrell
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
{kgrauman, trevor}@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 minimal-cost correspondences 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 evaluate our method with scene, object, and texture recognition tasks.

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 (for example, using a Harris or SIFT [11] interest operator) and are invariant to the transformations likely to occur across images, such as photometric or various geometric transformations.

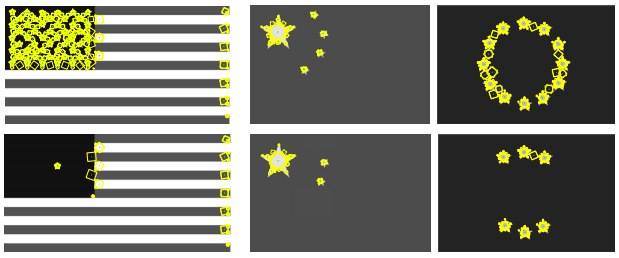

Figure 1. Images with detected features. Under voting-based matching schemes, the two versions of each flag are indistinguishable, since a query image with fewer stars will vote equally for a flag with fewer stars as it will for the flag with all the stars. Under our distribution-based similarity measure,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

A number of recent matching techniques extract 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 [12, 11, 19, 17]. When sufficiently salient features are present in an image, matching methods based on the independent voting scheme may successfully identify good matches in the database. However, using a query image's features to independently index into the database ignores useful information that is captured by the co-occurrence of a set of distinctive features – information that is especially important when categorization of objects or textures is the goal – and it fails to distinguish between images having varying numbers of similar features (see Figure 1).

Other matching approaches have taken feature co-occurrences into account by using vector quantization to represent each image by its frequency of prototypical fea-

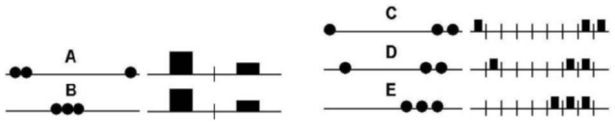

Figure 2. Comparing quantized feature sets with a bin-by-bin similarity measure is sensitive to bin size. If bins are too wide, discriminative ability is lost (Point sets A and B produce the same prototype frequencies in spite of their distinct distributions). If bins are too narrow, features that are 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 avoid the bin sensitivity issue, since features are matched based on their similarity to one another, not their assigned bin placement.

ture occurrences, then comparing the weighted histograms with a bin-by-bin distance measure [18, 2]. However, while mapping detected features to a set of global prototypes may make matching the distribution of features more efficient, 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 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 [16] (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 [16, 10].

To address these issues, we propose a technique that compares images by matching their distributions of local invariant features. 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 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 [7, 4].

We demonstrate our method in three very different contexts: recognition of scenes, objects, and textures. 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.

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

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 of local invariant features. 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 [12], 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 [19] use affine moment invariants to independently cast votes for similar database images. The method for matching scenes given in [17] uses voting to identify candidate matches, then applies a series of steps to verify geometric consistency within larger neighborhoods. In [11], 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 [18] and [2] apply text retrieval techniques to image matching: vector quantization (VQ) is applied to affine invariant regions collected from images, and each image is represented by a fixed-length vector giving the frequencies of the pre-established feature prototypes (also called a "bag-of-words/keypoints"). In [18], images in the database are ranked in similarity to a user-segmented query region based on their frequency vectors' normalized scalar product, while in [2] multi-class classifiers are trained using the frequency histograms as feature vectors. In [10], textures are represented by histograms of prototypical affine-invariant features and then recognized by exhaustive NN search with exact EMD. The authors of [9] 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 [19, 12, 11, 17] 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 [18, 2], and [10], where VQ is applied to features to obtain a frequency vector, our method represents an image with its actual distribution of features.

EMD was first used in vision to measure the distance between intensity images [15]. More recently EMD has been used for global color- or texture-based similarity in [16], and for comparing vector-quantized signatures of affine invariant features in texture images [10]. Exact EMD is computed with linear programming, and its complexity is exponential in the number of points per set for sets of unequal mass, and cubic for sets of equal mass. An embedding of EMD into L_1 and the use of Locality-Sensitive Hashing (LSH) for approximate NN was shown for the purpose of global color histogram-based image retrieval in [7], and the embedding was used for matching shapes based on contour features in [5].

The main contributions of this paper are an efficient image matching algorithm that compares distributions of invariant appearance features and exploits approximate NN search, and a study of the tradeoffs between voting schemes that index with features independently and the use of joint statistics of local features.

3. Matching with Distributions of Features

We have developed an efficient image matching technique that compares images in terms of their raw distributions of local invariant appearance features using approximate EMD. In this section we will describe the representations we use, and the mechanism by which we efficiently compare them.

Image features that are stable across varying scales, rotations, illuminations, or viewpoints are desirable for recognition and indexing tasks, since objects are 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.

We represent each grayscale image $\mathbf{I}_i$ by the bag $\mathbf{B}_i$ of the local descriptors extracted from its interest points: $\mathbf{B}_i = \{\mathbf{s}^{\mathbf{p}_1}, \ldots, \mathbf{s}^{\mathbf{p}_{n_i}}\}$, where each $\mathbf{s}^j$ is a d-dimensional descriptor extracted from one of the n_i interest points $\mathbf{p}_1, \ldots, \mathbf{p}_{n_i}$ in image $\mathbf{I}_i$.

In this work, we experiment with two types of interest operators: the Harris-Affine detector described in [13], which detects points that are invariant to scale and affine transformations, and the Scale Invariant Feature Transform (SIFT) interest operator of [11], which detects points that

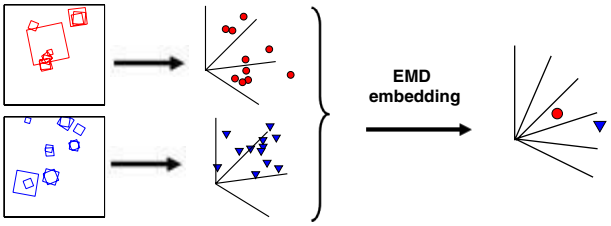

Figure 3. Comparing local invariant feature distributions in embedded EMD space. Patches are extracted from interest points in each image (left), producing a distribution of patch descriptors (points in $\Re^d$) for each image (center). Each distribution is mapped to a single point in a space where L_1 distance approximates the EMD between the original feature sets (right).

are invariant to scaling and rotation and has been shown in practice to be resistant to common image transformations. We employ the low-dimensional gradient-based descriptor called PCA-SIFT [8] as a descriptor for patches extracted at these interest points. Other interest operators or descriptors are of course possible.

EMD provides an effective way for us to compare images based on their discrete distributions of local features. For a metric space (X,$\mathcal{D}$) and two equal-mass sets $\mathbf{B}_p, \mathbf{B}_q \subset X$, the EMD is the minimum possible cost of π, a matching between $\mathbf{B}_p$ and $\mathbf{B}_q$:

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

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 $\mathcal{D}$ between two local features (e.g., the L_2 norm). 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 content-based retrieval and recognition.

However, the complexity of finding the optimal correspondences between two equal-mass sets under exact EMD is cubic in the number of features per 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 [16, 10]. 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 can use a low-distortion EMD embedding to reduce the problem of correspondence between sets of local features to an L_1 distance (see Figure 3). Previously, an EMD embedding was developed for global color histogram matching [7] and sets of local shape features [5]. The embedding f maps an unordered point set to a single (high-dimensional) point in the normed space L_1, such that the L_1 distance between the embedded vectors is comparable to the EMD between the original 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 $O(\log(D))$ for a feature space of underlying diameter D. The embedding computes and concatenates several weighted, randomly translated histograms of decreasing resolution for a given point set [7]. We normalize the weight given to each feature in a set to produce equal-mass sets, in order to allow each set to vary in cardinality (but see [6] for an extension that allows unequal-mass sets and partial matchings).

Once feature sets are mapped to a normed space, it is then possible to apply approximate NN techniques (e.g., LSH [4]) 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. 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 as the point where the overhead involved in hashing for LSH outweighs its query time speedups. See Procedures 1 and 2 for an outline of the matching process.

Unlike voting-based matching schemes, where each salient feature of an image is considered independently when matching a query to database items, we consider an image's distribution of invariant features 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. The information offered by the joint statistics of the feature appearances can capture similarities between images that may be overlooked when voting on a per-feature basis. In fact, we have found that a distribution-based approach is more effective than voting for tasks more general than matching to the same instance of an object, such as object categorization or texture recognition (see Section 4).

Note that the type of local features we are using may not individually contain explicit spatial information. However, our method allows spatial constraints among the local features to be enforced by augmenting the 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 constraints to include. To designate simple proximity constraints between features, each feature $\mathbf{s}^{\mathbf{p}_j}$ is paired with its m nearest-located features in the image to produce m new features of the form $[\mathbf{s}^{\mathbf{p}_j}, \mathbf{s}^{\mathbf{p}_i}]$, where $\mathbf{p}_i$ is the ith closest interest point to $\mathbf{p}_j$, for $1 \leq i \leq m$. Additionally, the angle of separation θ_i between two features' dominant orientations can be incorporated to further constrain their geometric relationship, creating features of the form $[\mathbf{s}^{\mathbf{p}_j}, \mathbf{s}^{\mathbf{p}_i}, \theta_i]$. 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 [3]. Other constraints based on affine or projective invariants are possible but would require larger tuples.

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 $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_t]$, randomly translated embedding grids $\mathbf{G}_l$, each with side lengths 2^l, $l = -1, \ldots, \log(D)$, 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 $\mathbf{s}_j$ having the desired invariances from image patch centered at each $\mathbf{p}_j$ to form bag of features $\mathbf{B}_i = \{\mathbf{s}^{\mathbf{p}_1}, \ldots, \mathbf{s}^{\mathbf{p}_{n_i}}\}$.
4: Weight each feature $\mathbf{s}^{\mathbf{p}_j}$ by $\frac{1}{n_i}$ and apply EMD embedding: $f(\mathbf{B}_i) = \left[\frac{1}{2}\mathbf{G}_{-1}(\mathbf{B}_i), \ldots, 2^l\mathbf{G}_l(\mathbf{B}_i), \ldots, D\mathbf{G}_{\log(D)}(\mathbf{B}_i)\right]$, 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 $\mathbf{B}_q$, embedding $f(\mathbf{B}_q)$, and random LSH functions $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_t]$:

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

4. Results

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

4.1. Methodology

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

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

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 nearest neighbors), and it approaches 1 as performance worsens; a random retrieval results in a normalized rank of 0.5 [14]. 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 class label; we (arbitrarily) set $k = 3$ in all experiments. 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 [18]. 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 when measuring the voting technique's performance, we used an exact (exhaustive) search to determine each feature's nearest neighbor, 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 [11] 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 descriptors to a discrete set of prototypes, which are found by running k-means on a set of examples containing images from each class. 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. Our implementation is modeled on the video data mining method in [18]; we omit the "stop-list" and temporal feature tracking steps since we are matching static, non-sequential images in addition to video frames in these experiments.

To extract the SIFT, PCA-SIFT, and Harris-Affine features in these experiments, we used the code that the authors of [11, 8, 13] have provided online. We used 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 (100, 400, and 700 clusters for the scenes, objects, and textures, respectively); 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. The SIFT interest operator was used to detect keypoints, and PCA-SIFT descriptors formed the feature sets.

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.

The left column of Figure 4 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-labeled 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 labeled data.

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. This experiment indicates 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

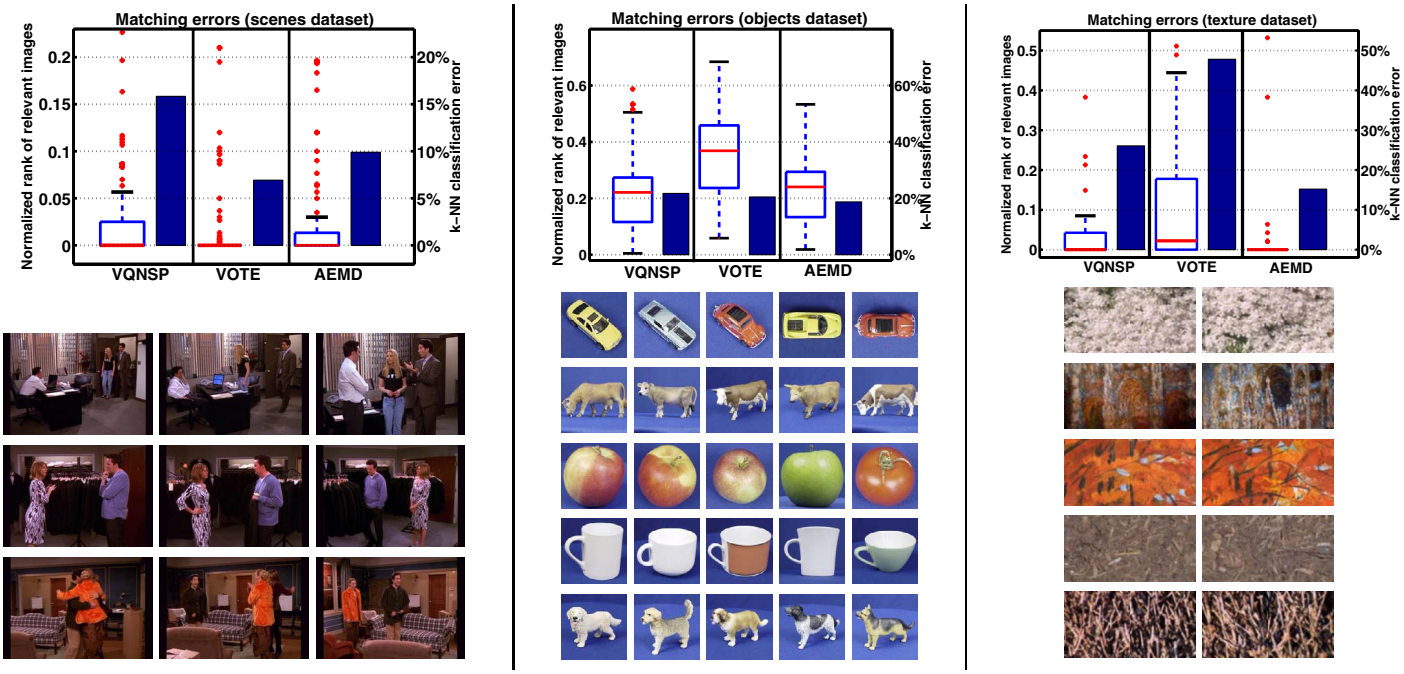

Figure 4. Matching results for three datasets: scenes (left), objects (center), and textures (right). Each plot shows the distribution of normalized ranks of relevant images (boxplot on left for each method, axis on left) and k-NN classification error rates for $k = 3$ (dark bar on right for each method, axis on right). 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. Below each plot are example retrievals obtained with AEMD for each dataset. Query is leftmost image in each row, and subsequent images are nearest neighbors. See text for experimental details.

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; exact voting requires hours for a match that our method performs in under a second (see Section 4.5). 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.

4.3. Object Categorization

We evaluated our method on an object categorization task using the ETH-80 database [1], which contains images of 80 objects from eight different classes in various poses against a simple background. We included five widely separated views of each object in the database, for a total of 400 images. The Harris-Affine detector was used to detect interest points, and PCA-SIFT descriptors were again used to compose the feature sets. The k-NN classification accuracy and the relevant rankings were measured via cross-validation, where all five views of the current test object were held out from the database. Since no instances of the query object are ever present in the database, this is a more challenging task than the previous scene recognition experiment; the goal is to rank all other instances of the object class as a query's nearest neighbors, but the other instances will exhibit intra-class appearance variation.

The center column of Figure 4 shows the matching performance of the three methods for this dataset. AEMD gives the best classification accuracy, meaning that the three nearest neighbors it found for each query were most consistently from the correct class, although the other methods do nearly as well (errors range from 19% to 22%). However, both AEMD and VQNSP assign substantially better relevant rankings than VOTE; under AEMD and VQNSP many of the relevant images from the same object class were usually given high rankings, whereas VOTE could only assign high ranks to a few very similar objects.

We found local gradient-based descriptors to be fairly effective for matching these images; however, this representation does have some limitations that are revealed by this database. Some objects from different classes are similar enough in terms of shape and local corner-based interest points that the PCA-SIFT feature cannot discriminate between them. For instance, an apple query at times retrieves a tomato among its nearest neighbors (see Figure 4, center column, third row of images). Additional features such as color distributions may be necessary to improve performance for any of the methods on this type of data.

Voting does poorly in this experiment because it finds matches based on how well individual features correspond, ignoring the higher-level information captured by the distribution of local features that occur on an object. The appearance variation across different instances of the same object class cause variations in the detected local features, which misguides the independent votes cast by the query object's features. This indicates that generally the set of features an object exhibits is more discriminative than each individual feature considered separately, causing the distribution-based approaches (AEMD and VQNSP) to be more successful. Thus while voting appears to be an effective but expensive strategy when searching for the same instance of an object (as in the scene recognition task), this dataset shows it to be a weaker approach for categorization tasks.

4.4. Texture Classification

Another useful application of image matching is texture classification. There are issues unique to comparing textures as opposed to the scenes and objects compared in the above experiments; in particular, textures are often defined in terms of how local features or structures co-occur. Each instance of a texture is a sample of an underlying, nonuniform pattern. This makes texture matching another 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.

The right column of Figure 4 shows the matching performance of the three methods for this dataset. We randomly selected a set of 25 VisTex examples.[2] 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.

[2]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.

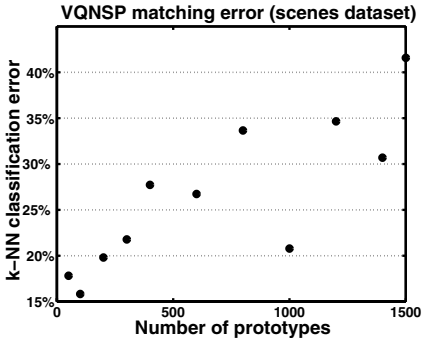

Figure 5. The matching performance of prototype-based methods is sensitive to the number of prototypes chosen.

AEMD results in the best texture classification performance on this dataset. 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 consistently assign low ranks to images of the same texture, resulting in a much wider distribution centered at 0.14. 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. As with the object categorization task, voting suffers because it does not capture co-occurrence information that is critical for texture matching. Based on the texture retrievals obtained by voting, we found that it risks casting excessive votes for images containing a repeated "generic" feature.

4.5. Discussion

In all our experiments, we found that VQNSP matching quality was quite sensitive to the clustering that defined the prototypes (see Figure 5), as was also observed in [2]. 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. Additionally, the type of images used to create the prototypes was critical to VQNSP's matching performance. We found it necessary to build prototypes from images that were very similar to the test examples in order to get achieve VQNSP's best performance (i.e., other scene images from the same video, or other examples from the same objects in the ETH-80 database), which suggests that it may be difficult to use the prototype-based methods to do "general purpose" matching on purely unseen test data. As

shown in Figure 4, compared to VQNSP, our method more consistently ranked both the *Friends* and textures images (and equally ranked the ETH-80 images), and it does not require a parameter choice since it matches the (un-binned) features themselves.

The EMD embedding vector resulting from an input feature set is high-dimensional, but very sparse; only $O(n \log(D))$ entries are nonzero, where n is the number of features in an example, and D is the diameter of the feature space. The time required to embed one d-dimensional point set is $O(nd \log(D))$. Thus, the computational cost of comparing two images' local feature distributions under approximate EMD is $O(nd \log(D)) + O(n \log(D)) = O(nd \log(D))$, the cost of embedding two point sets, plus an L_1 distance on the sparse vectors [7]. LSH reduces the time required to retrieve similar images to $O(sN^{\frac{1}{1+\epsilon}})$, where N is the number of examples in the database, ϵ is the LSH parameter related to the amount of approximation of the neighborhood radius, and s is the number of nonzero entries in the vectors. In our experiments we set ϵ to 1, making the upper bound on the query time $O(ds\sqrt{N})$.

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 $O(dn^2N)$ if exact NN features are found. For the prototype-frequency method, the query time has an upper bound of $O(dpnN)$, where p is the number of prototypes. To give a concrete example, with our implementations and $d = 8$, $N = 8335$, $n = 1000$, $p = 500$, and $D = 316$, a single voting query requires over 2.2 hours, a single query with the prototype-based method requires 1.62 seconds, and our method requires 0.49 seconds.

This work has dealt with matching equal-mass sets of features under an L_1 embedding of the minimal-cost correspondence problem. In recent work we have extended the method to allow partial matchings over unequal-mass sets using multi-resolution histogram intersection, and we show its capacity as a kernel for discriminative classification [6]. Additional examples and information about the methods can be found at http://people.csail.mit.edu/people/kgrauman/match.html.

5. Conclusions

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. The proposed algorithm is efficient, accurate, and does not require choosing a number of clusters. In future work, we intend to evaluate the impact of the proposed local geometry constraints within distribution-based matching.

References

[1] *http://www.vision.ethz.ch/projects/categorization/*.

[2] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual Categorization with Bags of Keypoints. In *Proc. ECCV*, Prague, Czech Republic, May 2004.

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

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

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

[6] K. Grauman and T. Darrell. Pyramid Match Kernels: Discriminative Classification with Sets of Image Features. Technical Report AIM-2005-007, MIT CSAIL, March 2005.

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

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

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

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

[11] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Intl Jrnl of Computer Vision*, 60(2):91–110, Jan 2004.

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

[13] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *Intl Jrnl of Computer Vision*, 1(60):63–86, Oct 2004.

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

[15] S. Peleg, M. Werman, and H. Rom. A Unified Approach to the Change of Resolution: Space and Gray-level. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(7):739–742, July 1989.

[16] Y. Rubner, C. Tomasi, and L. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *Intl Jrnl of Computer Vision*, 40(2):99–121, Nov 2000.

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

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

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