In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington DC, June 2004.

Fast Contour Matching Using Approximate Earth Mover's Distance

Kristen Grauman and Trevor Darrell
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA, 02139

Abstract

Weighted graph matching is a good way to align a pair of shapes represented by a set of descriptive local features; the set of correspondences produced by the minimum cost matching between two shapes' features often reveals how similar the shapes are. However, due to the complexity of computing the exact minimum cost matching, previous algorithms could only run efficiently when using a limited number of features per shape, and could not scale to perform retrievals from large databases. We present a contour matching algorithm that quickly computes the minimum weight matching between sets of descriptive local features using a recently introduced low-distortion embedding of the Earth Mover's Distance (EMD) into a normed space. Given a novel embedded contour, the nearest neighbors in a database of embedded contours are retrieved in sublinear time via approximate nearest neighbors search with Locality-Sensitive Hashing (LSH). We demonstrate our shape matching method on a database of 136,500 images of human figures. Our method achieves a speedup of four orders of magnitude over the exact method, at the cost of only a 4% reduction in accuracy.

1. Introduction

The minimum cost of matching features from one shape to the features of another often reveals how similar the two shapes are. The cost of matching two features may be defined as how dissimilar they are in spatial location, appearance, curvature, or orientation; the minimal weight matching is the correspondence field between the two sets of features that requires the least summed cost. A number of successful shape matching algorithms and distance measures require the computation of minimal cost correspondences between sets of features on two shapes, e.g., [2, 17, 8, 6, 3].

Unfortunately, computing the optimal matching for a single shape comparison has a complexity that is superpolynomial in the number of features. The complexity is of course magnified when one wishes to search for similar shapes ("neighbors") in a large database: a linear scan of the database would require computing a comparison of superpolynomial complexity for each database member against the query shape. Hierarchical search methods, pruning, or the triangle inequality may be employed, yet query times are still linear in the size of the database in the worst case, and individual comparisons maintain their high complexity regardless.

To address the computational complexity of current correspondence-based shape matching algorithms, we propose a contour matching algorithm that incorporates recently developed approximation techniques and enables fast shape-based similarity retrieval from large databases. We treat contour matching as a graph matching problem, and use the Earth Mover's Distance (EMD) – the minimum cost that is necessary to transform one weighted point set into another – as a metric of similarity. We embed the minimum weight matching of contour features into L_1 via the EMD embedding of [11], and then employ approximate nearest neighbor (NN) search to retrieve the shapes that are most similar to a novel query. The embedding step alone reduces the complexity of computing a low-cost correspondence field between two shapes from superpolynomial in the number of features to O($nd \log \Delta$), where n is the number of features, d is their dimension, and Δ is the diameter of the feature space (i.e., the greatest inter-feature distance).

In this work we also introduce the idea of a low-dimensional shape descriptor manifold. Using many examples of high-dimensional local features taken from shapes in an image database, we construct a subspace that captures much of the descriptive power of the rich features, yet allows us to represent them compactly. We build such a subspace over the "shape context" feature of [2], which consists of local histograms of edge points, and successfully use it within the proposed approximate EMD shape matching method.

We demonstrate our fast contour matching method on a database of 136,500 human figure images (real and synthetic examples). We report on the relative complexities (query time and space requirements) of approximate versus exact EMD for shape matching. In addition, we study empirically how much retrieval quality for our approximate method differs from its exact-solution counterpart (optimal graph matching); matching quality is quantified based on its performance as a k-NN classifier for 3-D pose. With our method it is feasible to quickly retrieve similar shapes from large databases – an ability which has applications in various example-based vision systems – and our technique eliminates the constraint on input feature set size from which other contour matching techniques suffer.

2. Related Work

In this section we review relevant related work on current shape matching techniques requiring optimal correspondences between features, the use of EMD as a similarity measure, and the embedding of EMD into a normed space and fast approximate similarity search. For additional information about various distance metrics for shape matching and their computational complexities, please refer to [18].

A number of shape matching techniques require optimal correspondences between feature sets at some stage. The authors of [2] obtain least cost correspondences with an augmenting path algorithm in order to estimate an aligning transform between two shapes. They achieve impressive shape matching results with their method, but they note that the run-time does not scale well with the representation size due to the cubic complexity of solving correspondences. The authors of [3] characterize local shape topologies with points and tangent lines and use a combinatorial geometric hashing method to compute correspondence between these "order structures" of two shapes. In [17], a polynomial time method is given where the shock graphs of 2-D contours are compared by performing a series of edit operations, and the optimal alignment of shock edges is found using dynamic programming. In [8], a graduated assignment graph matching method is developed for matching image boundary features that operates in time polynomial in the size of the feature sets.

The concept of using the Earth Mover's Distance to measure perceptual similarity between images was first explored in [15] for the purpose of measuring distance between gray-scale images. More recently EMD has been utilized for color- or texture-based similarity in [16, 9], and extended to allow unpenalized distribution transformations in [4]. In [12] exact EMD is applied to a database of 1,620 silhouettes whose shock graphs are embedded into a normed space; the method does not use an embedding to approximate the EMD computation itself, and thus may not scale well with input or database size. In [6], a pseudometric derived from EMD that respects the triangle inequality and positivity property is given and applied to measure shape similarity on edges.

In recent work by [1], AdaBoost is used to learn an embedding that maps the Chamfer distance into Euclidean space, and it is applied to edge images of hands in order to retrieve 3-D hand poses from large databases. However, as the authors note, their training algorithm, which requires a large number of exact distance computations, has a running time that thus far prevents their method from embedding more complex distances (such as graph matching or EMD), and retrievals are based on a linear scan of the database.

Our goal is to achieve robust, perceptually meaningful shape matching results as the above methods can, but in a way that scales more reasonably with an arbitrary representation size and allows real-time retrieval from larger databases.

In this work we show how EMD and Locality-Sensitive Hashing (LSH) can be used for contour-based shape retrievals. An embedding of EMD into L_1 and the use of LSH for approximate NN was shown for the purpose of color histogram-based image retrieval in [11]. We utilize the shape context feature (log-polar histograms of edge points) of [2] as a basis for our shape reprentation in this work. While the authors of [14] mention that using approximate NN search algorithms for shape context-based retrieval is a possibility, their system actually utilizes pruning techniques to speed searches. To our knowledge our work is the first to use an EMD embedding for fast contour matching, to employ LSH for contour matching, and to develop a compact shape context subspace feature.

3. Fast Similarity Search with EMD

In this section, for the reader's convenience, we briefly summarize the EMD metric and the randomized algorithms we use in our shape matching method: the approximate similarity search algorithm LSH [7], and the embedding of EMD into a normed space given in [11].

EMD is named for a physical analogy that may be drawn between the process of transforming one weighted point set into another and the process of moving piles of dirt spread around one set of locations to another set of holes in the same space. The points are locations, their weights are the size of the dirt piles and holes, and the ground metric between a pile and hole is the amount of work needed to move a unit of dirt. To use this transformation as a distance measure, i.e., a measure of dissimilarity, one seeks the least cost transformation – the movement of dirt that requires the least amount of work. When the total weight in the two point sets is equal, the solution is a complete one-to-one correspondence, and it is equivalent to the problem of bipartite graph matching. That is, for a metric space (X,D) and two n-element sets $\mathbf{A}, \mathbf{B} \subset X$, the distance is the minimum cost of a perfect matching between $\mathbf{A}$ and $\mathbf{B}$:

$$EMD(A,B) = \min_{\pi:\mathbf{A}\to\mathbf{B}} \sum_{a\in\mathbf{A}} D(a,\pi(a)). \qquad (1)$$

EMD performs partial matching in the case that the two sets have unequal total weights; the distance is then the minimum work needed to cover the mass in the "lighter" set with the mass in the "heavier" one.

Locality-Sensitive Hashing (LSH) indexes a database of examples residing in a normed space by a number of hash tables, such that the probability of collision is high for similar examples and low for dissimilar ones. In particular, LSH guarantees that if for a query point $\mathbf{q}$ there exists a point in the database $\mathbf{p}$ such that $D(\mathbf{p},\mathbf{q}) \leq r$, then (with high probability) a point $\mathbf{p}'$ is returned such that $D(\mathbf{p}',\mathbf{q}) \leq (1+\epsilon)r$. Otherwise, the absence of such a point is reported. The query time for a database of N d-dimensional examples is bounded by $O(dN^{(1/(1+\epsilon))})$. See [7] for details.

The low-distortion embedding of EMD given in [11] provides a way to map weighted point sets $\mathbf{A}$ and $\mathbf{B}$ from the metric space into the normed space L_1, such that the L_1 distance between the resulting embedded vectors is comparable to the EMD distance between $\mathbf{A}$ and $\mathbf{B}$ themselves. Working in a normed space is desirable since it allows the use of fast approximate NN search techniques such as LSH. The general idea of the embedding is to compute and concatenate several weighted histograms of decreasing resolution for a given point set. Formally, given two point sets $\mathbf{A}$ and $\mathbf{B}$, each of cardinality n, and each containing points in $\Re^d$: impose grids G_i, $-1 \leq i \leq \log(\Delta)$, on the space $\Re^d$, with grid G_i having side length 2^i, and Δ equal to the diameter of $\mathbf{A} \cup \mathbf{B}$. Each grid is translated by a vector chosen randomly from $[0,\Delta]^d$. To embed a point set $\mathbf{A}$, a vector $\mathbf{v}_i$ is created for each G_i with one coordinate per grid cell, where each coordinate counts the number of points in the corresponding cell, i.e., each $\mathbf{v}_i$ forms a histogram of $\mathbf{A}$. The embedding of $\mathbf{A}$ is then the concatenated vector of the $\mathbf{v}_i$'s, scaled by the side lengths: $f(\mathbf{A}) = \left[\mathbf{v}_{-1}(\mathbf{A}), \mathbf{v}_0(\mathbf{A}), 2\mathbf{v}_1(\mathbf{A}), \ldots, 2^i\mathbf{v}_i(\mathbf{A}), \ldots\right]$. The distortion of the embedding has an upper bound of $O(\log\Delta)$). [1]

4. Approach

The main contributions of this work are a fast contour matching method that exploits the approximate EMD embedding and NN search algorithms described above, a rich but compact contour feature descriptor manifold that is amenable to approximate EMD, and an extension to the EMD embedding that handles general point sets with a sampling-based approach.

4.1. Matching Contours with Approximate EMD

Recall our motivation to design an efficient means of calculating the least cost correspondences between two shape feature sets: such correspondences are required by a number of effective shape matching algorithms, but typically optimal solutions make large per-object feature set sizes or large database retrieval problems impractical for these algorithms. To address these issues, we embed the problem of correspondence between sets of local shape features into L_1, and use the approximate solution to match the shapes.

Our method proceeds as follows: features are extracted from a database of shape images, and each image's features are treated as a uniformly weighted point set. Using the L_1 embedding of EMD over the point sets, one sparse vector is produced for each input shape. Next, a set of random LSH hash functions are generated, and all of the embedded database vectors are entered into the hash tables. Both the database embedding and hash table construction is performed offline. Then, given a novel shape, the embedding for its features is computed using the same random grid translations used for the database embedding. Finally, shapes similar to the novel query are retrieved from the database by computing the L_1 distance between the query's embedding and only those vectors in the union of the hash buckets that are indexed by the query's embedding.

The embedded vector resulting from an input point set is high-dimensional, but very sparse; only $O(n\log(\Delta))$ entries are non-zero. The time required to embed one point set is $O(nd\log(\Delta))$. Thus the computational cost of obtaining the near-optimal feature correspondences for our shape matching method will be $O(nd\log(\Delta)) + O(n\log(\Delta)) = O(nd\log(\Delta))$, the cost of embedding two point sets, plus an L_1 distance on the sparse vectors. [2] The exact methods typically used in shape matching to solve for correspondences (such as the transportation simplex algorithm for linear programming, or the Hungarian method for bipartite graph matching) require time cubic or exponential in n.

Probably the most direct application of EMD for 2-D

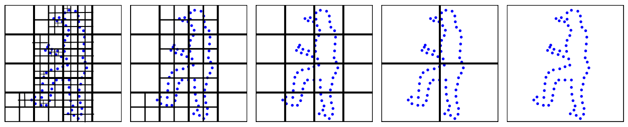
Figure 1: Imposing a hierarchy of grids on a set of contour points to get its embedding. Embedding shape features residing in higher dimensions is an analogous process of imposing hyperplane grids on the space.

[1]The distortion C of a metric embedding f describes how much information loss the embedding induces: $\frac{1}{C}EMD(\mathbf{A},\mathbf{B}) \leq ||f(\mathbf{A}) - f(\mathbf{B})||_{L_1} \leq EMD(\mathbf{A},\mathbf{B})$.

[2]L_1 distances are computed in time proportional to the number of non-zero entries in the sparse vectors as follows: a sparse vector data structure is composed of a vector of non-zero indices plus a vector of values at those indices. At the time of embedding, the vectors are sorted according to their indices (an offline computational cost). Then computing the L_1 distance is a matter of running two pointers through the vectors to sum the difference of their values where they have the same index, or add the value of one vector where the index is not shared.

Algorithm 1 The procedure for embedding sets of local features

```
Given: $N$ weighted point sets $\{\mathbf{A}_1,\ldots,\mathbf{A}_N\}$, where $\mathbf{A}_i = \{(\mathbf{f}_1,w_1)\ldots,(\mathbf{f}_{m_i},w_{m_i})\}$ is a weighted point set composed
    of $m_i$ $d$-dimensional features $\mathbf{F}^i$ with scalar weights $\mathbf{W}^i = [w_1,\ldots,w_{m_i}]$, where $t(\mathbf{A}_i) = \sum_{j=1}^{m_i} w_j$, and $\Delta$ is the di-
    ameter of $\cup_{i=1}^N \mathbf{F}^i$,
 1: Let $t_{max} = \max_i t(\mathbf{A}_i)$
 2: for all $i = 1,\ldots,N$ do
 3:     if $t(\mathbf{A}_i) < t_{max}$ then
 4:         $\mathbf{A}_i \leftarrow \{(\mathbf{f}_1,w_1),\ldots,(\mathbf{f}_{m_i},w_{m_i}),(\mathbf{d}_1,u),\ldots,(\mathbf{d}_q,u)\}$,
            where $u$ is unit weight, $q = t_{max} - t(\mathbf{A}_i)$, and $\mathbf{d}_z$ is a
            random selection from $\{\mathbf{f}_1,\ldots,\mathbf{f}_{m_i}\}$.
 5:     end if
 6: end for
 7: Let $L = \lceil \log\Delta/\log 2 \rceil + 1$.
 8: For $1 \le l \le L$, let each $\mathbf{s}^l = [s_1^l,\ldots,s_d^l]$ be a random vector
    from $[0,2^l]^d$.
 9: for all $\mathbf{A}_i$, $1 \le i \le N$ do
10:     for all $(\mathbf{f}_j = [f_1^j,\ldots,f_d^j], w_j) \in \mathbf{A}_i$ do
11:         for all $\mathbf{s}^l$, $1 \le l \le L$ do
12:             $\mathbf{x}_l^j = [c(s_1^l,f_1^j),\ldots,c(s_d^l,f_d^j)]$,
                where $c(s_k^h,f) = trunc((f - s_k^h)/2^h)$
13:             $\mathbf{v}_l^j = w_j \times 2^l$
14:         end for
15:         $\mathbf{p}_j^i = [(\mathbf{x}_1^j,v_1^j),\ldots,(\mathbf{x}_L^j,v_L^j)]$
16:     end for
17:     $embed(\mathbf{A}_i) = tally(sort([\mathbf{p}_1^i,\ldots,\mathbf{p}_{m_i}^i]))$,
        where pairs $(\mathbf{x},v)$ represent sparse vector entries with index $\mathbf{x}$ and value $v$, $sort()$ returns
        $[(\mathbf{x}_{s_1},v_{s_1}),\ldots,(\mathbf{x}_{s_n},v_{s_n})]$ such that $\mathbf{x}_{s_i} \le_{LEX} \mathbf{x}_{s_{i+1}}$
        (lexicographic ordering of concatenated vector elements)
        and $tally()$ sums values of sparse vector entries with equal
        indices.
18: end for
```

contour matching is to compose point sets from the literal points on the two contours (or some subsets of them) and use the Euclidean distance between two contour points' image coordinates as the ground distance D in (1). For this simple positional feature, examples must be translated and scaled to be on par with some reference shape. To embed a set of 2-D contour points, we impose a hierarchy of grids on the image coordinates themselves, starting with a grid resolution where each image coordinate receives its own cell, and ending with a single cell grid the size of the largest image, $G_{\log\Delta}$ (see Figure 1).

When two weighted point sets have unequal total weights, EMD does not satisfy the triangle inequality or positivity property, and thus is not a true metric. However, it is desirable for robust matching to allow point sets with varying total weights and cardinalities (e.g., different numbers of edge points occur in different images, inputs may originate from images of different resolutions, etc.). While the embedding of EMD into a normed space is defined for

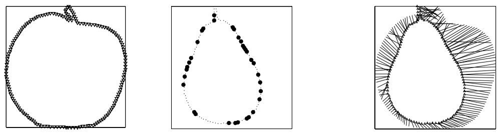
(a) Input shapes (b) EMD flow

Figure 2: To simulate equal total weights for point sets having unequal weights, randomly sampled points from the lighter set are counted multiple times in the embedding grids.

only the metric case of EMD, we preserve the partial matching abilities of EMD in the approximate case by modifying the manner in which the points are embedded. In order to embed two sets of contour features with different total weights, we simulate equal weights by adding the appropriate number of duplications of random points from the lower weight set. This means that multiple features from one set may map to a single feature in the other set. For example, in Figure 2, when points are sampled uniformly from the apple and pear contours, the pear has 33 fewer points than the apple, so 33 points are randomly chosen from its contour to be duplicated (circled points). The EMD flow (b) will then contain many-to-one correspondences. Pseudo-code for the EMD contour matching embedding is given in Algorithm 1.

Once a database of contours is embedded into a normed space, we do fast (time sublinear in the database size) retrievals for a novel embedded query contour via LSH. In addition to the complexity savings for a single shape match described above, the time required for retrieving similar shapes is reduced to $O(sN^{(1/(1+\epsilon))})$, where N is the number of shapes in the database, ϵ is the LSH parameter relating the amount of approximation of the normed distance, and s is the dimension of the sparse embedded contour vectors, s having a space requirement of $O(n\log(\Delta))$. Results showing the quality of the approximate NN contours we retrieve with LSH are reported in Section 5.

4.2. Shape Context Manifolds

There are drawbacks to using the simple positional feature for shape matching with approximate EMD. Though straightforward to embed, it can be a brittle feature, and in order to achieve scale or translation invariance this feature demands a pre-processing stage on all shapes (which requires some a priori knowledge about the shapes, and can itself be brittle). Richer shape descriptors can help overcome this, as several authors have noted [13, 2, 3]. Thus, we have experimented with richer shape descriptors with the approximate EMD distance, as we expect to achieve more robust

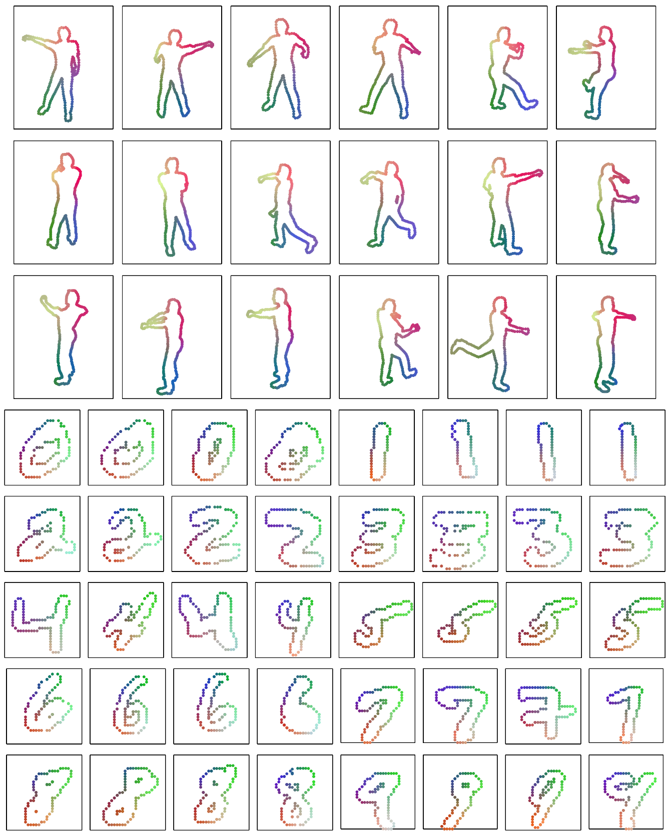
Figure 3: Visualization of feature subspace constructed from shape context histograms for two different datasets. The RGB channels of each point on the contours are colored according to its histogram's 3-D PCA coefficient values. Matching with EMD in this feature space means that contour points of similar color have a low matching cost (require little "work"), while highly contrasting colors incur a high matching cost. (This figure must be viewed in color.)

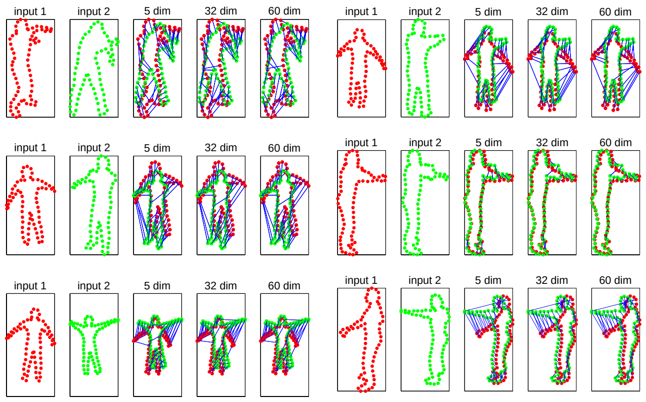

Figure 4: Shape context subspace captures local contour variations in much fewer dimensions. The optimal correspondences for feature sets composed of 5 and 32 shape context PCA coefficients are shown here and compared with the correspondences that result with the raw (d=60) shape context feature. The optimal correspondence found by the raw high-dimensional feature is generally achieved by projections onto a much lower dimensional subspace.

matchings and more meaningful correspondence fields from descriptive feature representations. We employ the shape context local descriptor of [2]. The shape context feature at a single contour point is a log-polar histogram of the coordinates of the rest of the point set, measured using the reference point as the origin. It is inherently translation invariant, and scale and rotation invariance may be added [2].

While matching with the full shape context feature is possible with our method, a low-dimensional feature descriptor is desirable since any constant change in point dimensions changes the constant distortion factor C in the embedding, and also changes the d factor in the complexity of computing the embedding itself. Thus we find a low-dimensional feature subspace based on a large sample of the shape context histograms, and then perform the embedding step in the domain of this subspace. The subspace is constructed from a large sample of features drawn from the database of contours on which we wish to apply our method. All contour features (from the database items and novel queries alike) are then projected onto the subspace, and the approximate EMD embedding is performed in the domain of a small number of their subspace coordinates. We use principal components analysis (PCA) to determine the set of bases that define this "shape context manifold".

We found that a very low-dimensional subspace was able to capture much of the local contour variation in our datasets. Figure 3 gives examples of the shape context subspace for human figures and handwritten digits. In Figure 4 we measure its expressiveness as a function of feature dimension, as compared to a higher-dimensional raw point histogram. In Section 5 we report results using the shape context subspace representation.

5. Results

5.1. Dataset and Representation

We have tested our method on a database of contours from 136,500 images of synthetic human figure contours in random poses that were generated with a computer graphics package called Poser [5]. We query the database with a separate test set of 7,000 synthetic human figure images, and a test set of 1,000 real images from a single human subject in various poses.

We constructed a set of hash functions for the synthetic image dataset in order to perform LSH approximate-NN retrievals. We determined the LSH parameters (number of

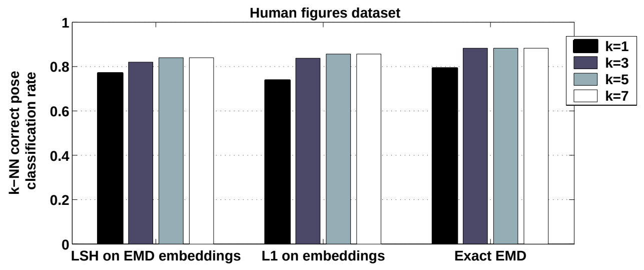

Figure 5: Comparison of the quality of retrievals from the exact versus approximate methods. Quality is measured via k-NN classification task. See text for details.

hash tables and number of bits per hash value) based on the proof in [10] which shows how to select parameters for the case of the Hamming space over a non-binary alphabet, such that the desired level of approximation versus speed tradeoff is achieved. For the complete dataset of 136,500 examples, this meant using 8 tables and 120-bit functions. For all experiments ϵ is set to 1.

We have run a number of experiments with this dataset using shape context subspace features (see Section 4.2). We construct a shape context subspace from 5 x 12 log-polar histograms extracted from the training set; we used a sample of 855,600 histograms. The representation of a novel contour is determined by projecting its shape context histograms onto the low-dimensional subspace. We found that for our dataset, a 2-D projection adequately captured the descriptive power of the shape context feature and resulted in good contour matches. This representation is translation invariant, and the scale of the shape context histograms initially extracted from the data is determined from the mean inter-point distance per shape. Because the coefficients are real-valued, they must be appropriately scaled and discretized before the embedding grids can be imposed. We remap the projection coefficients to positive integers by subtracting the minimum projection value from all examples, then scaling by 100.

5.2. Retrieval Quality

We measure the retrieval quality of our method by comparing the 3-D pose of each query example with the pose of the k-NN's that are retrieved from the database. When the synthetic human figure database is generated, the 3-D pose (a set of 3-D joint positions) is recorded for each example. If the joint positions of a retrieved shape are on average within some threshold of distance, we consider the retrieved shape a good match. We chose a threshold of 10 cm, since this is a distance at which the 3-D poses are perceptually similar.

The chart in Figure 5 quantitatively compares the quality of results obtained with our approximate method with

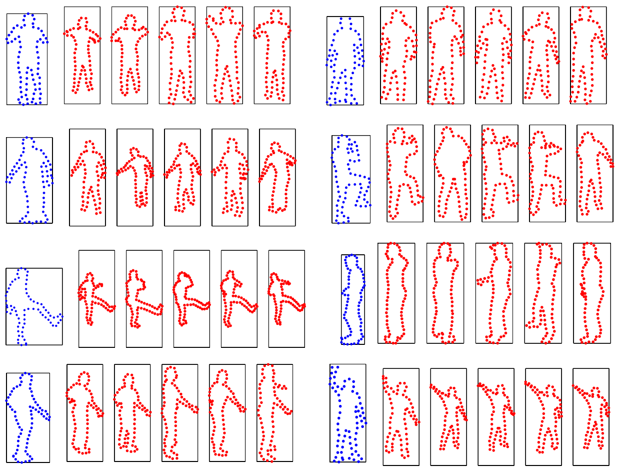

Figure 6: Real image queries: examples of query contours from a real person (left,blue) and 5 NN retrieved from synthetic database of 136,500 images using L_1 on EMD embeddings of shape context subspace features.

those obtained from exact EMD for a database of 10,000 images. [3] In this figure the optimal results were obtained by running the transportation simplex algorithm to compute EMD on full, 60-D shape context features, whereas results for the two approximations (the embedding and LSH) were obtained using only 2-D shape context subspace features. There is a slight decrease in classification performance at each approximation level; however, we found that the practical bound on the distortion introduced by the EMD embedding is significantly (about one order of magnitude) lower than the upper theoretical bound.

We note that in a practical system classification rates could be improved for the approximate methods if a refining step were implemented – for instance a handful of exact computations on the approximate matches. Figure 7 shows some example retrievals using our approximate EMD method with synthetic query images. Examples of the synthetic NN that were retrieved for the images from a real person are shown in Figure 6.

5.3. Empirical Measure of Complexity

As discussed in Section 4, the theoretical computational complexity of retrieving the approximate minimum cost feature correspondences with our method for feature sets of cardinality n and dimension d residing in a space of diameter Δ is $O(nd\log(\Delta))$. The diameters of the spaces in which our point sets reside are on the order of 10^3 up to 10^5, depending on the representation; with n on the order

[3]Due to the complexity of exact EMD, the exact comparisons were necessarily computed with a parallel implementation.

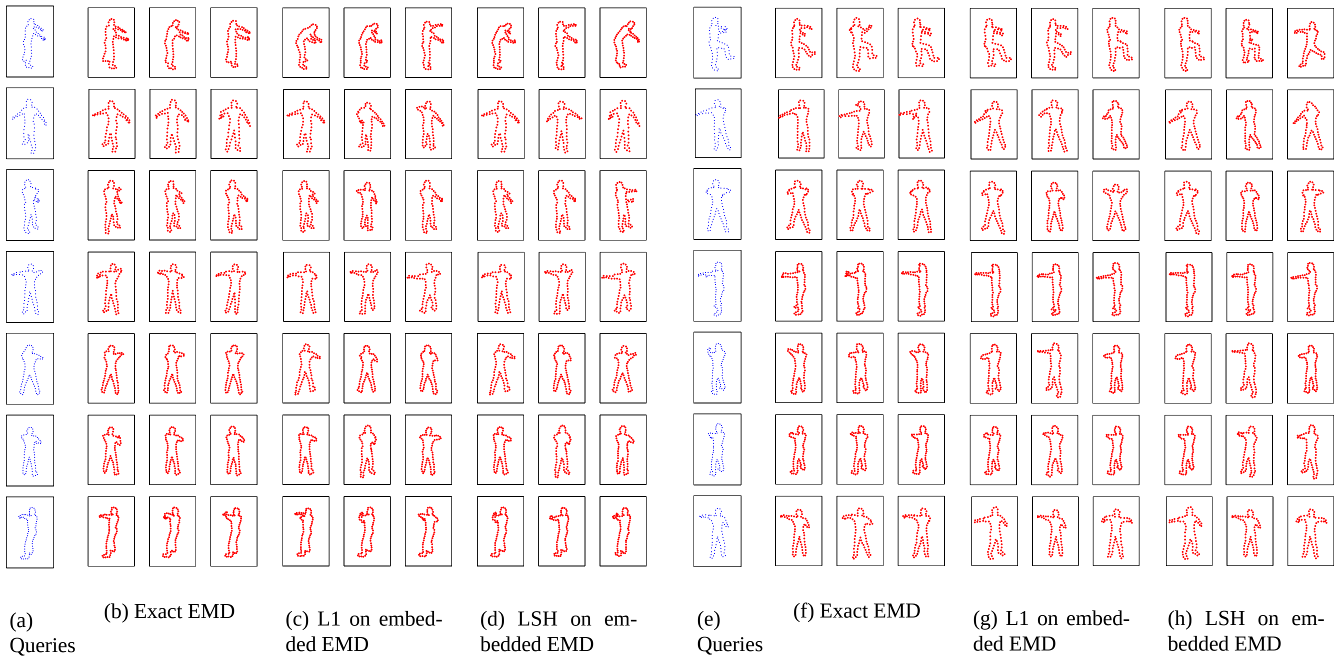

Figure 7: Approximate EMD retrieves shapes very similar to those retrieved by the optimal matching. Figure shows examples of 3 NN (left to right in rank order) with embedded EMD contour matching (c,g) and embedded EMD contour matching with LSH (d,h), compared to NN under exact EMD contour matching (b,f). Examples shown were chosen randomly from 7,000 test set results, and NN were retrieved from a database of 136,500 examples. Columns (c,d,g,h) use embedded 2-D shape context subspace feature; (b,f) are from exact EMD applied to full 60-D shape context feature. Note that the embedded match results are qualitatively similar, yet several orders of magnitude faster to compute.

of 10^2, $d = 2$, theoretically this means that a single embedding and L_1 distance cost requires on the order of 10^3 operations. This is the cost of embedding two point sets, plus performing an L_1 distance on the very sparse vectors. In practice, for $n = 200$ an unoptimized C implementation of our method takes about 0.005 seconds to perform a single matching with exact L_1 this way (less than 0.005 seconds to compute the two embeddings, plus 7.9×10^-5 seconds to compute the L_1 distance). In comparison, to compute a single exact EMD distance using a C implementation of the simplex algorithm required on average 0.9 seconds for data of the same dimensions. In accordance with the upper bound on the embedding's space requirements, the number of non-zero entries in the sparse embedded vectors was on average 350 for the histogram representation.

Figure 8 gives a summary of the empirical run-time behavior of the embedding. Our experiments confirm that the run-time for embedding point sets increases only linearly with the size of the input's dimension or cardinality. This means that our method scales well to handle inputs with large representation sizes.

The larger payoff for using the approximate embedding, however, comes when we use LSH to query a large database with an embedded point set. The input must be compared against only a small fraction of the database's embedded vectors – those in the union of the hash buckets that are indexed by the query's embedding. On average, in our experiments LSH needed to compute only 1,915 L_1 distances per query, (less than 2% of the database). The median query time for the complete 136,500 item Poser database was only 1.56 seconds. In comparison, a single query with the exact EMD method would require 34 hours (performing a worst-case linear scan of the database). Figure 9 shows how query time for the human figure dataset varies with the database size.

6. Conclusions and Future Work

We have presented a new fast contour matching algorithm that utilizes an approximation to EMD to judge similarity between sets of local shape descriptors. Our technique enables fast shape-based similarity retrieval from large databases, and its run-time is only linearly dependent on the number of feature points used to represent a shape. We have also constructed a rich but compact contour feature manifold based on shape contexts for approximate EMD.

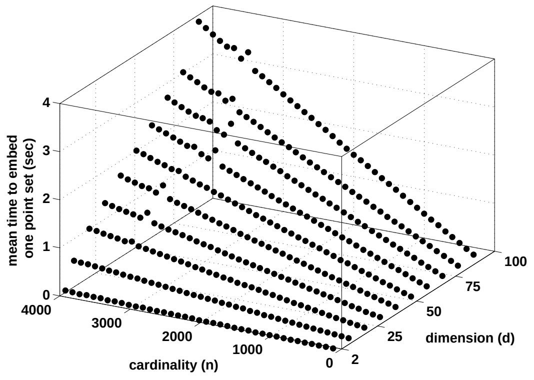

Figure 8: Mean embedding time per point set for 500 point sets with varying dimensions and cardinalities.

In the future we intend to experiment with approximate EMD and different shape representations. We will also explore alternative means of compactly representing inherently continuous features within the discrete embedding framework, such as vector quantization or multidimensional scaling. We are also interested in investigating ways to improve the efficacy of the NN hashing process in this context.

Acknowledgments

We would like to thank Piotr Indyk for useful discussions and for suggesting duplicating points to deal with unequal cardinalities, Greg Shakhnarovich for generating the database of synthetic human figure images, and the anonymous CVPR reviewers for their comments.

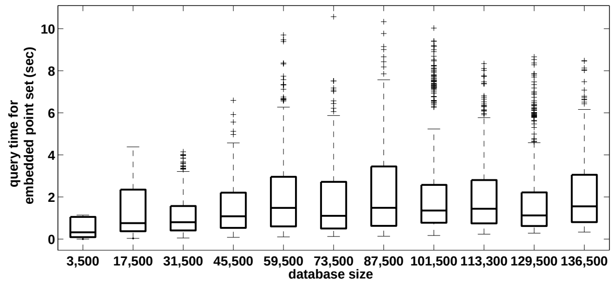

Figure 9: Query time distributions for embedded point sets for increasing database sizes. Test set is composed of 1,000 examples from the human figure database, d = 2, n = 200. Lines in center of boxes denote median value; top and bottom of boxes denote upper and lower quartile values, respectively. Dashed lines show extent of rest of the data, pluses denote outliers. The median query time for the 136,500 item database is only 1.56 seconds; exact EMD could require over a day to perform the same query.

References

[1] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: A Method for Efficient Approximate Similarity Rankings. In *CVPR*, Washington, D.C., 2004.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *TPAMI*, 24(24):509–522, 2002.

[3] S. Carlsson. Order Structure, Correspondence and Shape Based Categories. In *Intl Wkshp on Shape Contour and Grouping*, Sicily, May 1998.

[4] S. Cohen and L. Guibas. The Earth Mover's Distance under Transformation Sets. In *ICCV*, Corfu, Greece, Sept 1999.

[5] Egisys Co. Curious Labs. Poser 5 : The Ultimate 3D Character Solution. 2002.

[6] P. Giannopoulos and R. Veltkamp. A Pseudo-metric for Weighted Point Sets. In *ECCV*, Copenhagen, May 2002.

[7] 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.

[8] S. Gold and A. Rangarajan. A Graduated Assignment Algorithm for Graph Matching. *TPAMI*, 18(4):377–388, 1996.

[9] H. Greenspan, G. Dvir, and Y. Rubner. Region Correspondence for Image Matching via EMD Flow. In *IEEE Wkshp on Content-based Access of Image and Video Libraries*, 2000.

[10] P. Indyk. *High-Dimensional Computational Geometry*. PhD thesis, Stanford University, 2000.

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

[12] Y. Keselman, A. Shokoufandeh, M. F. Demirci, and S. Dickinson. Many-to-Many Graph Matching via Metric Embedding. In *CVPR*, Madison, WI, 2003.

[13] F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and Efficient Shape Indexing through Curvature Scale Space. In *BMCV*, Edinburgh, UK, 1996.

[14] G. Mori, S. Belongie, and J. Malik. Shape Contexts Enable Efficient Retrieval of Similar Shapes. In *CVPR*, Lihue, HI, Dec 2001.

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

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

[17] T. Sebastian, P. Klein, and B. Kimia. Recognition of Shapes by Editing Shock Graphs. In *ICCV*, Canada, Dec 2001.

[18] R. Veltkamp and M. Hagedoorn. State-of-the-Art in Shape Matching. In *Tech Report UU-CS-1999-27*, Utrecht University, 1999.