Spherical Embedding of Inlier Silhouette Dissimilarities

Etai Littwin, Hadar Averbuch-Elor, Daniel Cohen-Or
Tel-Aviv University

Abstract

In this paper, we introduce a spherical embedding technique to position a given set of silhouettes of an object as observed from a set of cameras arbitrarily positioned around the object. Our technique estimates dissimilarities among the silhouettes and embeds them directly in the rotation space SO(3). The embedding is obtained by an optimization scheme applied over the rotations represented with exponential maps. Since the measure for inter-silhouette dissimilarities contains many outliers, our key idea is to perform the embedding by only using a subset of the estimated dissimilarities. We present a technique that carefully screens for inlier-distances, and the pairwise scaled dissimilarities are embedded in a spherical space, diffeomorphic to SO(3). We show that our method outperforms spherical MDS embedding, demonstrate its performance on various multi-view sets, and highlight its robustness to outliers.

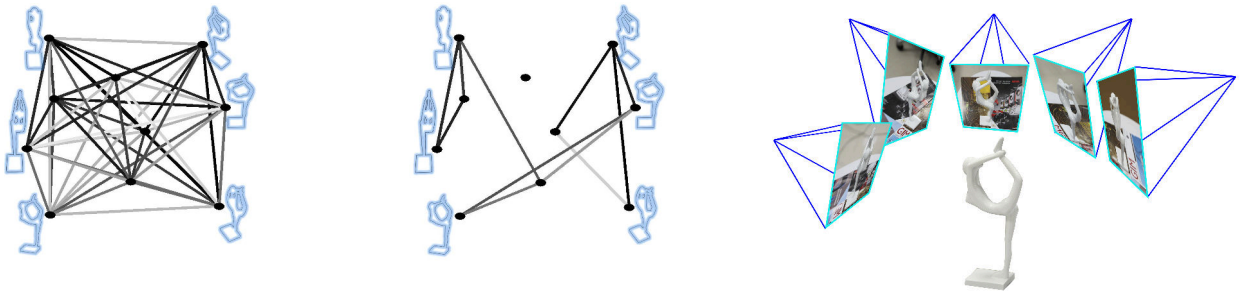
1 Introduction

We consider the following problem: given a set of silhouettes of an object as observed from a set of cameras and an estimated dissimilarity measure among them, we would like to approximate their relative rotations and position them on a sphere (see the illustration in Figure 1c). Similar to previous works (e.g., [17, 8, 13]), we assume that the object silhouettes are the only visual cues provided, and thus traditional structure from motion (SfM) techniques based on common feature correspondence cannot be applied successfully. However, we do not intend to calibrate the cameras from the given silhouettes. The dissimilarity measure among the silhouettes is invariant to scaling and translation and thus we are only interested in the relative rotation among the cameras. Hence, we embed the object's silhouettes in SO(3). Once they are embedded onto the 3D rotation space, the camera positions can be easily approximated over a sphere.

The problem we address is particularly challenging since the similarity estimates are generally unreliable, and directly applying a multi-dimensional scaling (MDS) embedding introduces a significant distortion. Hence, a robust technique is sought, that may ignore portions of the input data. Our technique finds and considers inlier dissimilarities and ignores outlier ones, and then embeds only a subset of the views (see the illustration in Figure 1). The measure that we employ to estimate the dissimilarities between silhouettes, just like most similarity measures, tends to be more reliable for more similar shapes, and completely unreliable for more dissimilar ones. This may suggest that by simply ignoring large dissimilarity measures, a robust embedding can possibly be obtained. However, as we shall show, such a simple approach is not robust enough as some short distance estimates are also erroneous and distort the embedding on the sphere.

The technique that we present is more involved. It carefully defines a graph, that may not necessarily contain all the input silhouettes, nor all their pairwise dissimilarities. The graph is defined by a union of small sub-sampled matrices, each of which is verified to have a plausible embedding in SO(3). The graph of inlier dissimilarities, as a whole, is then embedded into the space of rotations by an optimization that associates relative rotations with the views so that they agree with the dissimilarities defined by the edges of the graph. Our embedding technique employs exponential maps and solves the embedding in the rotation space SO(3), without introducing large dissimilarities to complete the affinity matrix. Our contribution is twofold: First, we present a spherical embedding technique based on exponential maps, and show that it outperforms spherical MDS. Second, we develop an inlier screening technique, and show its robustness to erroneous silhouette dissimilarities.

The rest of the paper is organized as follows: Related works are briefly reviewed in Section 2. We present an overview of our method in Section 3. In Section 4, we review the topology of the SO(3) group and the exponential maps which we apply in our spherical embedding. We introduce our exponential maps embedding, inlier screening techniques, and dissimilarity measure in Sections 5, 6, and 7, respectively. We then show some results and an evaluation of our technique in Section 8 and conclude with a discussion of limitations and future work in Section 9.



(a) (b) (c)

Figure 1: ***Overview of our method.*** *Given a set of silhouettes of an object as observed from a set of cameras arbitrarily positioned around an object, we first (a) compute the full all-pairs dissimilarities. We discover the inlier silhouette dissimilarities using our inlier screening technique and obtain a sparse graph (b). We then perform an optimization to embed the sparse dissimilarities in SO(3). Assuming the contours are associated with photos, then one can place them on a sphere (c).*

2 Related Work

The problem of recovering viewpoints from silhouettes is related to the general problem of structure-from-motion. In the general setting, both the observed structure and the camera parameters are unknown [9]. A large body of research has studied this problem and significant progress has been made. Most methods for camera calibration, or camera motion estimation, rely on feature correspondences. When these correspondences are available, robust solutions, (e.g., [18]), allow for a simultaneous reconstruction of the captured object and the recovery of the latent camera parameters. Other works investigate the complementary setting, i.e., motion and structure estimation when there are no reliable feature correspondences.

Extracting and matching feature points for textureless or smooth surfaces, for instance, is impractical, and calls for a different approach. The object silhouette is the most reliable image feature in this setting, and many works ([7, 21, 5]), take advantage of this available cue. However, unlike our approach, prior work on camera calibration from silhouettes usually relies on recovering the epipolar tangency points. These are the silhouette points whose tangent is an epipolar line. In our work, we use the object silhouettes merely as means to provide a dissimilarity measure to estimate the distances among the latent cameras.

Shape from silhouette techniques are based on the notion of frontier points [14, 20], i.e., the outermost epipolar tangencies, to reconstruct a static model from silhouettes of uncalibrated cameras. Wong and Cipolla [20] assume that at least three views are taken along a circular path and add additional views incrementally in an optimization scheme that is initialized by the circular motion. Hernandez et al. [10] further generalized the approach by adding more geometric constraints. Using these additional constraints, their technique can handle partial or truncated silhouettes. More recently, Mcilroy et al. [13] proposed a technique for the recovery of the affine projection matrices in a more general setting. However, their method requires a rather good initialization. Our work, in that respect, is complementary and can provide these initial camera positions.

To recover the latent positions, Sinha et al. [17] exercise a RANSAC-based approach by exploring the space of all possible epipole positions in a given video sequence. Thanks to the static camera configuration, the epipolar geometry can be verified and corrected. Furukawa et al. [8] selects sets of frontier point candidates and rejects inconsistent matches. They also assume that the object is captured along a sphere and determine the rotation error. It is important to note, however, that this method, along with the others mentioned above, is sensitive to the accuracy and completeness of the silhouettes. In contrast, our method uses the silhouettes only to estimate the distance between the views and is thus insensitive to partial, truncated or noisy silhouettes.

Generally speaking, all of these techniques are specifically designated for solving the calibration problem from given silhouettes, while our method can use any dissimilarity measure between views to recover the camera rotation parameters. In that sense, our work is closer in spirit to works that embed distances onto spherical structures. Pless and Simon [15] extend the MDS embedding algorithm to spherical manifolds. Begelfor and Werman [2] generalize their approach to manifolds with negative curvature as well. Unlike these spherical MDS techniques, our method recovers the camera rotation parameters directly by applying optimization in the rotation space, and thus, as we shall show, can better handle missing pairs of distances. Furthermore, contrary to these spherical MDS methods, the focus of our work is handling outlier measures as common in dissimilarity estimates among silhouettes.

Recently, Averbuch-Elor and Cohen-Or [1] presented a technique that recovers the ring-ordering of casual images that capture a temporal event. Their technique is also based on a rough dissimilarity measure among the photos. Their method assumes that the topology of the photos is one dimensional. In our work, we extend the problem to a 3D manifold in 4D and embed the photos while considering only inliers.

3 Overview

In this work, we develop a technique to embed dissimilarities onto the space of rotations. For each view and its associated silhouette, we would like to find the rotations R_i relative to some neutral position for each viewpoint i. Let us denote by $D(R_i,R_j)$ the distance between viewpoints i and j. Note that the camera can produce different views while maintaining the same position in 3D space relative to the object due to rotation around its own principal axis. We assume that a significant portion of dissimilarity measures correlate well with the actual $D(R_i,R_j)$, but we would like to tolerate a non-trivial amount of outlier measures.

Given a set of pairwise distances d_{ij} between each pair of viewpoints i and j, we would like to minimize the following expression in the space of rotations:

$$\sum_{ij}(D(R_i,R_j)-d_{ij})^2.$$

The minimization requires the computation of its derivatives with respect to R_i and R_j. Since the first derivatives are not trivial to express with the exponential maps representation, in Section 5 we develop an explicit expression of the first derivatives using the Baker Campbell Hausdorff formula [4].

In Section 6, we develop a method that deals with outlier dissimilarity measures. Figure 2 illustrates the correlation between the ground truth rotational distances among the views and the estimated dissimilarity among their silhouettes. As can be seen, the correlation is not high, and there is a significant amount of noise and outliers. A simple approach which selects only the k-nearest neighbors (KNN) of each point is not robust enough, as demonstrated in Figure 2b. The red dots are the estimates associated with the KNN of the points. Clearly, they include many outliers. Hence, we establish a more elaborate technique that identifies inlier estimates that have a high correlation with the ground truth. These inlier dissimilarities (illustrated with green dots in Figure 2a) reside along the diagonal, and are not necessarily small. The key idea is to search and sample small sub-sampled matrices that embed well onto a hypersphere. We create an aggregate of such sub-sampled matrices that have significant overlap and define a graph where the nodes are a subset of the input points and an edge is defined only for a pair that appears in one of the matrices. We show that the aggregate of sub-sampled matrices embeds well on the unit sphere using our exponential maps embedding technique.

The direct optimization of our objective function in SO(3) allows solving a rather sparse set of views, without completing large distances as needed in MDS-based techniques. In Section 8, we show that if the dissimilarities d_{ij} are in full correlation with the ground-truth distances $D(R_i,R_j)$ then our method recovers the rotations accurately. We then show

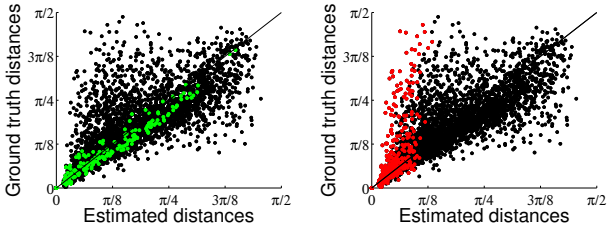


Figure 2: ***Inlier dissimilarities***. *The ground truth distances versus the estimated dissimilarities of the* BULL *set. (a) The green dots illustrate the chosen distances using our inlier embedding technique. Note that it contains large distances as well as small ones, all along the diagonal. (b) Selecting inliers by KNN (the red dots) may include large errors.*

the robustness of our method to erroneous dissimilarities by adding noise to the ground truth data. Moreover, we demonstrate the performance of our method on real data and compare our inlier screening technique with one that uses KNN distances.

4 SO(3) and Exponential Maps

Rotations in 3D space form the special orthogonal Lie group SO(3). Being a Lie group, it is a smooth differentiable manifold that can be studied using differential calculus. Among numerous ways of representing a 3D rotation R_i, a convenient one is defined by the axis of rotation (a pivot) $\vec{n}$ and a rotation angle θ. Given this parametrization, the topology of the SO(3) group can be regarded as a solid ball in R^3 of radius π. More precisely, for every point in this ball there is a 3D rotation, with an axis defined by the normalized vector connecting the point and the ball origin, and a rotation angle defined by the magnitude of the vector. Since rotations through pi around $\vec{n}$ and $-\vec{n}$ represent the same rotation, we identify antipodal points on the surface of the ball. With this identification, we arrive at the topological space homeomorphic to the group.

We can describe a map from this solid ball in R^3 to a a surface of a unit sphere in R^4 using unit quaternions representation: $\vec{q}=[cos(\frac{\theta}{2}),sin(\frac{\theta}{2})\vec{n}]^T$. It is easy to verify that $\vec{q}^T\vec{q}=1$, allowing us to identify the surface of this sphere as a manifold diffeomorphic to the group. Since a topology on the SO(3) Lie group can be derived from the distance measure between its elements, using the diffeomorphism to the 4D sphere we can compute this measure using geodesic distances on the sphere. For more on the SO(3) group, see [6].

Since the SO(3) has a Lie structure, the geodesics are represented using an exponential map, which is a mapping of

antisymmetric matrices to rotation matrices through matrix exponentials. Given $\vec{n}$ and θ, the vector $\vec{m} = \theta\vec{n} = [m_x, m_y, m_z]$ parameterizes a rotation matrix:

$$R_i = e^{[\vec{m}_i]_\times} = e^{H_i},$$

where we use the matrix cross product notation:

$$[\vec{m}_i]_\times = \begin{bmatrix} 0 & m_{iz} & -m_{iy} \\ -m_{iz} & 0 & m_{ix} \\ m_{iy} & -m_{ix} & 0 \end{bmatrix} = H_i.$$

The geodesic distance between R_i and R_j defined as:

$$D(R_i, R_j) = \frac{1}{2}\|log(e^{H_i}e^{-H_j})\|_F, \quad (1)$$

with $\|\circ\|_F$ being the Frobenius norm $\|H\|_F = \sqrt{\sum_{ij}|H_{ij}|^2}$. Note that this metric is proportional to the amount of rotation required to get from one rotation to the next, and it can be shown that it is equivalent to a geodesic distance between two points representing R_i and R_j on the 4D unit sphere diffeomorphic to the rotations group, giving values in the range $[0, \frac{\pi}{2}]$. For more on this metric, see [11].

In order to perform efficient optimizations on SO(3), we need to first compute derivatives which are not trivial to express. In the general case, the matrices H_i and H_j do not commute (i.e., $[H_i, H_j] = H_iH_j - H_jH_i \neq 0$), which prohibits the standard addition of exponents as in the scalar case (i.e., $e^{H_i}e^{-H_j} \neq e^{H_i - H_j}$). Furthermore, computing the derivatives in simple form requires the expansion of $log(e^{H_i}e^{-H_j})$ into a series using the Baker Campbell Hausdorff formula which presents some restrictions on the input as we shall see in Section 5.

5 Embedding with Exponential Maps

As was previously discussed, we aim at minimizing the following function:

$$\sum_{ij}(D(R_i, R_j) - d_{ij})^2,$$

where a set of pairwise distances d_{ij} are given as input.

Following the notation introduced in Section 4, the vector $\vec{m} = [m_x, m_y, m_z] = \theta\vec{n}$, parameterizes a rotation matrix R_i using an exponential map, $R_i = e^{[\vec{m}_i]_\times} = e^{H_i}$, with the distance measure between R_i and R_j defined according to Equation 1.

The non commutative nature of the rotations group does not allow the trivial simplification of the logarithm in Equation 1. Indeed, the computation of Equation 1 and its derivatives requires the expansion of the matrix logarithm into an infinite power series of its arguments. An alternative and a more convenient way of expressing $log(e^{H_i}e^{-H_j})$ as an infinite series is by the rotations group Lie algebra terms, or an infinite sum of nested commutators. To express Equation 1 in terms of $\vec{m}_i$ for viewpoint i and $\vec{m}_j$ for viewpoint j, we use the Baker Campbell Hausdorff (BCH) formula that states that for square matrices H_i and H_j:

$$log(e^{H_i}e^{-H_j}) = H_i - H_j - \frac{1}{2}[H_i, H_j] - \frac{1}{12}[H_i, [H_i, H_j]] + \frac{1}{12}[H_j, [H_i, H_j]]...$$

where $[H_i, H_j] = H_iH_j - H_jH_i$ is the commutator between H_i and H_j.

In the case of the SO(3) group, the BCH series is decreasing rapidly, allowing us to neglect high order terms while remaining relatively accurate (see [4] for more details). Taking the leading terms above we get:

$$D(R_i, R_j) \approx \frac{1}{2}\|H_i - H_j - \frac{1}{2}[H_i, H_j] - \frac{1}{12}[H_i, [H_i, H_j]] + \frac{1}{12}[H_j, [H_i, H_j]]\|_F.$$

We can now formulate an optimization problem that is not strictly convex in the rotation parameters $\vec{m}_i$, but will converge to the correct solution up to a 4D rotation and a reflection regardless of the initial guess, provided that the convergence criteria of the BCH formula (described at the end of this section) is met.

We can express the following terms as:

$$[H_i, H_j] = [\vec{m}_i]_\times[\vec{m}_j]_\times - [\vec{m}_j]_\times[\vec{m}_i]_\times = [\vec{m}_i \times \vec{m}_j]_\times,$$
$$[H_i, [H_i, H_j]] = [\vec{m}_i \times (\vec{m}_i \times \vec{m}_j)]_\times,$$
$$[H_j, [H_i, H_j]] = [\vec{m}_j \times \vec{m}_i \times \vec{m}_j]_\times,$$

and can therefore approximate Equation 1 as follows:

$$\frac{1}{2}\|[\vec{m}_i - \vec{m}_j - \frac{1}{2}\vec{m}_i \times \vec{m}_j - \frac{1}{12}\vec{m}_i \times (\vec{m}_i \times \vec{m}_j) + \frac{1}{12}\vec{m}_j \times \vec{m}_i \times \vec{m}_j]_\times\|_F.$$

Let us denote with $\vec{V}_{ij}$ the expression:

$$\vec{m}_i - \vec{m}_j - \frac{1}{2}\vec{m}_i \times \vec{m}_j - \frac{1}{12}\vec{m}_i \times (\vec{m}_i \times \vec{m}_j) + \frac{1}{12}\vec{m}_j \times \vec{m}_i \times \vec{m}_j,$$

to obtain:

$$D(R_i, R_j) \approx \frac{1}{2}\|[\vec{V}_{ij}]_\times\|_F. \quad (2)$$

To express Equation 2 as an L2 norm of some vector, we use the following equality that holds for any vector $\vec{v}$:

$$\|[\vec{v}]_\times\|_F = \sqrt{2}\|\vec{v}\|_2$$

Therefore:

$$D(R_i, R_j) \approx \frac{1}{\sqrt{2}}\|\vec{V}_{ij}\|_2 = \sqrt{(\vec{V}_{ij}{}^T\vec{V}_{ij})/2}.$$

We can then define the objective function as follows:

$$obj = \sum_{ij} (\sqrt{\vec{V}_{ij}{}^T \vec{V}_{ij}} - d_{ij})^2. \quad (3)$$

Taking its derivative with respect to $\vec{m}_i$ yields:

$$\frac{\partial obj}{\partial \vec{m}_i} = 2 \sum_{ij} (\sqrt{\vec{V}_{ij}{}^T \vec{V}_{ij}} - d_{ij}) \left(\frac{\vec{V}_{ij}}{\sqrt{\vec{V}_{ij}{}^T \vec{V}_{ij}}} \right)^T \left(\frac{\partial \vec{V}_{ij}}{\partial \vec{m}_i} \right)^T.$$

To express $\frac{\partial \vec{V}_{ij}}{\partial \vec{m}_i}$, we observe that:

$$\frac{\partial (\vec{m}_i \times \vec{m}_j)}{\partial \vec{m}_i} = [\vec{m}_j]_\times,$$

$$\frac{\partial (\vec{m}_j \times \vec{m}_i \times \vec{m}_j)}{\partial \vec{m}_i} = [\vec{m}_j]_\times^2,$$

$$\frac{\partial (\vec{m}_i \times (\vec{m}_i \times \vec{m}_j))}{\partial \vec{m}_i} = -[\vec{m}_i]_\times [\vec{m}_j]_\times + [\vec{m}_j \times \vec{m}_i]_\times,$$

and thus $\frac{\partial \vec{V}_{ij}}{\partial \vec{m}_i} =$

$$I - \frac{1}{2}[\vec{m}_j]_\times + \frac{1}{12}[\vec{m}_j]_\times^2 + \frac{1}{12}[\vec{m}_i]_\times [\vec{m}_j]_\times - \frac{1}{12}[\vec{m}_j \times \vec{m}_i]_\times$$

We can now explicitly express the derivatives $\frac{\partial obj}{\partial \vec{m}_i}$ with respect to each rotation parameters $\vec{m}_i$.

The above expression of the derivatives of Equation 3 requires the normal convergence of the BCH series, which has been the topic of considerable research (e.g., [4]). In this work, we assume that our input is restricted to cases where the convergence is guaranteed, namely in the domain $\|H_i\|_f + \|H_j\|_f < \sqrt{2}\pi$, for matrices H_i and H_j. Now, since $\|H_i\|_f = \sqrt{2}\theta_i$, we can easily verify that this criteria is met by ensuring that there exist a row or a column i in the distance matrix where each entry is smaller than $\frac{\pi}{4}$ and set its corresponding rotation parameters $\vec{m}_i = \vec{0}$.

Finally, given a pairwise distances d_{ij}, we can optimize the objective function (Equation 3) using standard optimization methods such as gradient descent and get the rotation parameters $\vec{m}_i$ for each element i.

6 Inlier Embedding

The spherical embedding presented above converges to the correct solution provided that the pairwise similarity measures roughly agree with the amount of rotation the camera undergoes from one view to the other. However, in practice, such a strong correlation between all the similarity estimates and the real corresponding distances is not likely. Generally, the correlation tends to be stronger with short distances, which gives rise to the filtering of large distances and an embedding of the images only based on the short distances. However, as we shall show later with some experiments, the short distances are still unreliable enough to produce adequate results (see Section 8).

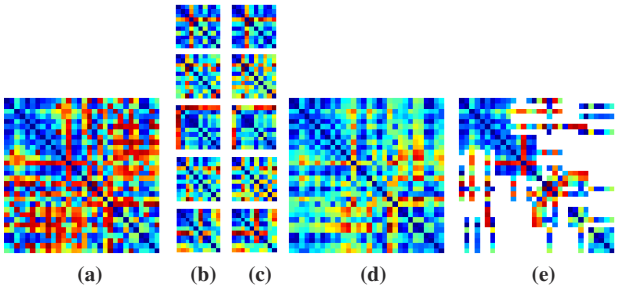


Figure 3: *An illustration of our inlier embedding technique. The matrices illustrated in (a) and (d) are the input (estimated) and ground truth distance matrices, respectively. Large differences between the two result in significant errors in the embedding. (b) The obtained sub-sampled matrices exhibit a much closer similarity to the ground truth, demonstrated in (c). The output graph G is created from the illustrated sparse matrix in (e).*

Distance measures between elements in the SO(3) group correspond to geodesic distances between points on a unit 4D hypersphere. We can therefore search through the NxN distance matrix for $N'xN'$ sub-sampled matrices that can best be embedded in a 4D spherical surface, where $N' << N$. As we shall see, this can be done efficiently without resorting to optimization methods. However, for large N' it is not likely to be able to guess such a sub-sampled matrix given the high amount of noise and outliers in a typical set. If N' is rather small, it is more probable to guess a sub-sampled matrix that can be successfully embedded on a hypersphere. Our approach is, therefore, to fuse many such sub-sampled matrices to obtain one large coherent embedding. The selection of these sub-sampled matrices is the inlier screening process.

Although the union of the embedded points is large, it does not necessarily contain its all-pairs distances, but rather only those that can jointly be embedded. Therefore, an embedding technique which takes as input a set of pairwise distances, such as the one described in the previous section, is needed to perform the coherent embedding. The goal of the screening scheme is, thus, to discover those meaningful (inlier) pairwise distances to be used by the optimization technique. From a full, but noisy, distance matrix (see the illustration in Figure 3d), we seek to obtain a sparse, yet meaningful, inliers distance matrix (such as the one illustrated in Figure 3e).

Given an $N \times N$ distance matrix (or a sub-sampled matrix) M , we would like to measure how well it can be embedded on a unit 4D hypersphere. The geodesic distances be-

tween points on the hypersphere correspond to the inverse cosine of the scalar product between the corresponding vectors. Thus, by performing eigenvalue decomposition on the cosine of M and evaluating its fifth eigenvalue, we can examine whether we can obtain a meaningful embedding in a $4D$ spherical space. Denote by $\lambda_1 ... \lambda_N$ the eigenvalues of $cos(M)$ in descending order, then:

$$E = \lambda_5.$$

E measures how well the matrix M can be embedded in a $4D$ spherical space. A sub-sampled matrix associated with a small E is then considered to consist of inlier distances. Furthermore, to avoid degenerate solutions, we only consider matrices with an L1 norm, normalized according to the matrix size, above a threshold T. All our results were obtained with $T = 0.4$.

Clearly, N' must be larger than four in order for the embedding to be non-trivial. The smaller N' is, the more likely it is to find a sub-sampled matrix of size $N' \times N'$ that can be embedded on a hypersphere while yielding a small error E. Conversely, for large N' it is less likely to find a sub-sampled matrix that can have a plausible embedding on the hypersphere. For efficiency reasons, we prefer to fuse large sub-sampled matrices. In practice, we found that using $N' = 10$ produces plausible results. To find a 10×10 sub-sampled matrix with a small E we use a RANSAC-based approach where we sample thousands of 10×10 sub-sampled matrices, compute their E values, and consider only those with a small E. Denote by Q the union of sub-sampled matrices Q_i that yield a small E.

We then need to generate a larger embedding by fusing together a proper subset $\hat{Q}$ of Q. The fusion of $\hat{Q}$ forms a graph, where the nodes are the union of all points in $\hat{Q}$, and the edges are the given distances between the points. More precisely, a pair of nodes is connected by an edge only if there exists a sub-sampled matrix Q_i in $\hat{Q}$ where both nodes are present. To get a proper embedding, the graph must be connected, or even strongly connected. To this end, the graph is required to have a min-cut of at least five edges.

We build the graph G incrementally by taking sub-sampled matrices one by one and augmenting the graph progressively. The initial graph is constructed from the sub-sampled matrix with the smallest E value. In each step, we augment G with an additional inlier sub-sampled matrix and ensure that it has at least four overlapping points with the points selected so far. We continue until G is sufficiently large or until all the valid sub-sampled matrices in Q are picked. Figure 3 illustrates the entries in the NxN distance matrix that can jointly be embedded on a hypersphere using our technique. All the sub-sampled matrices in $\hat{Q}$ are illustrated in Figure 3b. Lastly, the sparse matrix in Figure 3e consists of the union of the inlier distances, and these define the edges of the graph G.

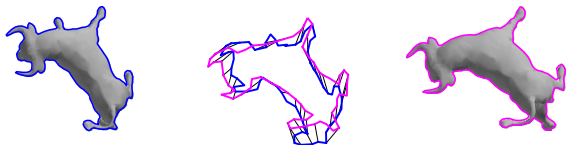


Figure 4: *Contour Dissimilarity Measure. The black lines connecting the blue and pink contours connect between similar points along the contours. The average spatial distance is then computed and later scaled accordingly. The computed distance between the two shapes illustrated above is* 20.6°*, and the true geodesic distance is in fact* 23.5°.

Dataset		Inlier			KNN	
Name	#img	avg	max	#img	avg	max
BULL	160	4.3	7.2	40	5.9	15.2
HORSE	80	5.2	13.6	40	6.5	20.3
AIRPLANE	80	8.3	15.6	20	16.5	29.9
CAMEL	80	6.8	12.3	35	12.5	22.8
DOG	80	6.1	14.6	40	15.8	27.4
LADY	60	7.6	17.3	30	13.2	22.1

Table 1: ***Inlier vs. KNN Screening****. The errors (measured in degrees) obtained on each of our datasets. For our inlier embedding technique, the number of embedded images is displayed alongside the errors.*

7 Contour Dissimilarity Measure

Our algorithm requires a dissimilarity measure that relates to the geodesic distance between the camera positions, so that images captured from similar viewpoints induce small distances. Since we aim at recovering the rotation parameters at SO(3), the measure must be rotation-variant. More precisely, if two images were captured from the same position but contain a rotation between them, we would like the algorithm to recover this rotation as well.

First, we normalize each outer-contour by translating its center of mass to the origin and scaling it so that each contour has a unit average distance to the origin. We then sample and match the normalized contours, namely c_1 and c_2, using the inner-distance shape context method [12]. This technique extends shape context [3] by replacing Euclidean distances with inner-distances. It computes descriptors for each sampled point on the contours. We compute the average spatial distance between a point on c_1 and its most similar point (in descriptor space) on c_2. Finally, we normalize the dissimilarity measures so that the greatest value is exactly $\frac{\pi}{2}$, assuming that there is a pair of views that are $\frac{\pi}{2}$ apart. See Figure 4 for an illustration of the distance between two input shapes.

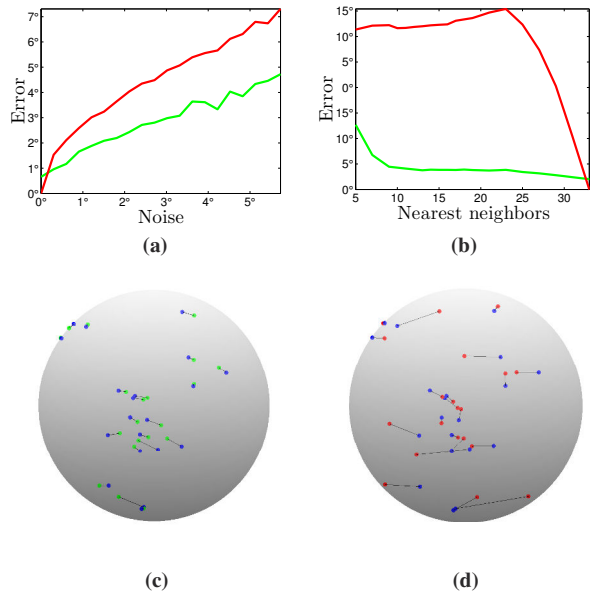


Figure 5: ***Exponential Maps vs. Spherical MDS****. (a) Comparing the errors obtained using exponential maps (green) and MDS (red) as a function of the noise. (c)-(d) The points illustrate the ground-truth locations (blue) versus the computed ones (red or green) in one test case when the noise added has a standard deviation of* 6°*. (b) The average error obtained when only a subset of the pairwise distances is given as input. The x-axis corresponds to the number of nearest neighbors for which the distances are provided (out of a full matrix of size* 35×35*).*

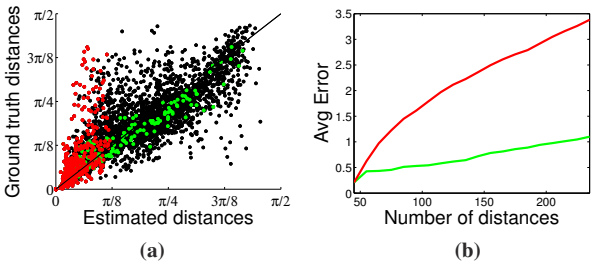


Figure 6: ***Inlier vs. KNN Screening****. Embedding a subset from our* HORSE *set. (a) The inlier distances chosen using our inlier embedding technique (green) and the distances chosen using KNN (red). (b) Evaluating the validity of the distances chosen by both techniques by examining the average distance from the diagonal in (a) as a function of the number of distances chosen for the embedding.*

8 Evaluation

To analyze the performance of our method we performed quantitative and qualitative evaluations. To measure the performance quantitatively we need ground truth camera positions. Thus, we generated six datasets, which contain multiple rendered images of an input model from various viewpoints. We also experimented on real photographs of a physical object, and assessed the resulting embedding qualitatively.

To better quantify the performance of our method and in order to compare our results to an alternative technique, we performed Monte-carlo experiments on N virtual camera positions distributed randomly across a hemisphere. We examined how well our method performs under increasing levels of noise and outliers. For these experiments, we considered the geodesic pairwise distances instead of the contour-based ones. In what follows, we elaborate on each of these experiments.

Exponential Maps vs. Spherical MDS To evaluate the performance of our embedding technique, we examined its robustness to noise and outliers. Furthermore, we compared the results to those obtained by an MDS embedding on a 4D unit sphere (similar to the approach of [15] and [2]). Since the alternate technique provides a point in 4D, to obtain the rotation parameters, we interpreted each point as a unit quaternion.

First, we examined both embeddings when all the camera orientations, and therefore, all the pairwise distances, are provided as input. Note that while the location on the 3D sphere does not reveal the difference in its rotation around its own look-at axis, the computed distance between the rotations of the embedding and ground truth for each point does reveal the overall accuracy in orientation. Figure 5a demonstrates the average error as the noise level increases. At each noise level, we generated multiple input orientations and added Gaussian-distributed noise with a varying standard deviation to the full distance matrix. Recall that our method is inherently an approximation, as we consider only the leading terms of the infinite sum. Thus, our method is not completely accurate when the distances are exact. However, as the noise level increases, the advantage of our method over spherical MDS is clearly evident.

We also examined our approach when only a subset of the distances, the k-nearest neighbors ($K = 10$), is provided. Since MDS assumes that all the pairwise distances are provided, we estimated the missing distances with Isomaps, as suggested also by [2]. Figure 5b illustrates the average error obtained by both techniques as a function of K.

Inlier vs. KNN Screening We performed a quantitative comparison to demonstrate the advantage of our inlier screening versus a KNN screening. In both cases, the data is embedded with our exponential maps embedding. In a KNN screening, we use only the distances between all points and their k-nearest neighbors. In order to demonstrate the quality of our chosen subsets, we measured the

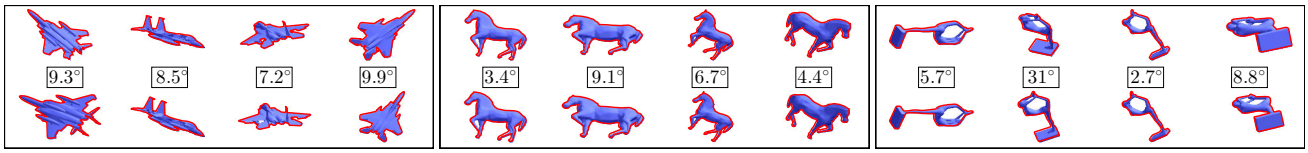


Figure 7: *Four randomly selected views and their rotational errors, belonging our* PLANE,HORSE, *and* LADY *datasets. Illustrated above are the input silhouettes together with the ground-truth silhouettes generated from the specified rotations.*

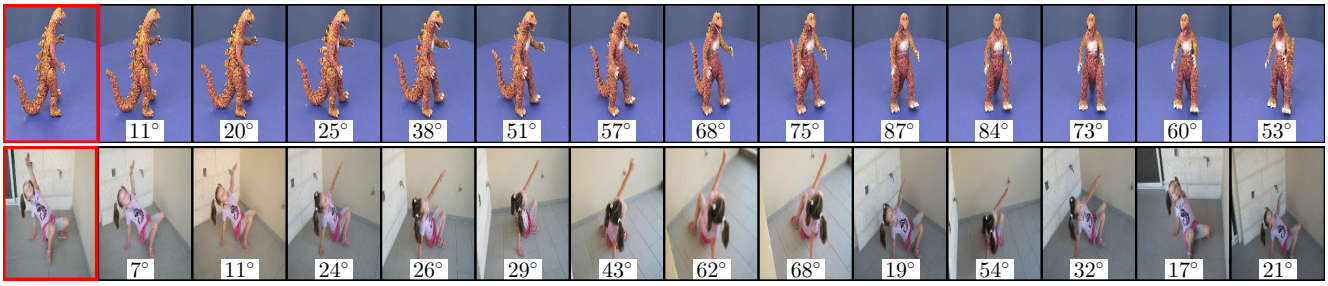


Figure 8: *Experiments on Real Photographs. A reference image (in red) and the distances between the embedded points, measured in degrees, of the remaining set images demonstrated above for the* DINOSAUR *set (top) and the* GIRL *set (bottom).*

average offset (of all the distances chosen) from the diagonal as a function of the number of the chosen distances (see Figure 6b). Table 1 demonstrates the average and the max error for a number of datasets. Note that the advantage of the inlier screening is consistent in all cases.

Experiments We generated a number of datasets of varying sizes and examined the results obtained by our algorithm. Table 1 summarizes the input set sizes, the number of embedded images, and the average and max errors for each of the sets. To visually assess the typical quality of the results, we selected a random subset per set and displayed the input silhouettes together with the ground-truth silhouettes generated from the specified rotations (see Figure 7 and more results in the Supplementary Material). Together with each pair, we display the rotational error between them, measured in degrees. Note that since the output of our algorithm is the relative rotation associated with each viewpoint, we used Kabsch's algorithm to obtain the optimal alignment between the algorithm's output and the ground truth.

We also qualitatively examined the results of our algorithm on real photos. Figure 8 illustrates the results obtained on the publicly available DINOSAUR set [19], and our GIRL set. The contours were extracted using the GrabCut method [16]. In each row, the leftmost image is the reference image. The approximated relative rotation to the reference image is displayed below the other images. As the figure demonstrates, the results are qualitatively plausible.

9 Summary and Future Work

In this work, we presented a robust technique to embed dissimilarities to space of rotations. The key is to perform an inlier screening process prior to the embedding. We then embedded the points with an optimization scheme that is applied directly over the rotations, represented with exponential maps.

We compared our embedding method to spherical MDS. Our evaluation confirms that under noise or given only a partial distance matrix, our method outperforms the alternatives. We examined our technique on various sets and were able to recover a subset of the dissimilarities almost faultlessly. Nevertheless, a noisy and erroneous dissimilarity matrix will ultimately yield a distorted embedding. In that respect, our method relies on a having a significant set of inliers, or a large enough set of dissimilarity estimates that correlate with the rotational distances. Moreover, our method provides only an approximation to the camera positions, one that can provide a good initial guess for camera calibration techniques which typically require a warm start.

In the future, we would like to explore other measures of dissimilarities among views. We would also like to try to embed the subset of cameras that were screened out. The idea is to rely on the inlier embedding to carve out an approximated 3D shape. Then, by projecting the carved model back to the sphere, we hope to identify the peculiar silhouettes that were associated with the outliers. We also believe that our inlier embedding technique can be beneficial for various applications which require MDS embedding, not necessarily on a sphere. In general, we believe that robust techniques that eliminate outliers explicitly are becoming increasingly important in computer vision applications where dissimilarity estimations are necessarily used.

Acknowledgements

We thank Thomas Lewiner for his deep and insightful comments on exponential maps. This work was supported by the Israel Science Foundation and The Yitzhak and Chaya Weinstein Research Institute for Signal Processing.

References

[1] H. Averbuch-Elor and D. Cohen-Or. Ringit: Ring-ordering casual photos of a temporal event. *ACM Transactions on Graphics*, 35(1):to appear, 2015. 3

[2] E. Begelfor and M. Werman. The world is not always flot or learning curved manifolds. *School of Engineering and Computer Science, Hebrew University of Jerusalem., Tech. Rep*, 2005. 3, 7, 8

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002. 6

[4] S. Blanes and F. Casas. On the convergence and optimization of the baker–campbell–hausdorff formula. *Linear algebra and its applications*, 378:135–158, 2004. 3, 4, 5

[5] A. Bottino and A. Laurentini. Introducing a new problem: Shape-from-silhouette when the relative positions of the viewpoints is unknown. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(11), 2003. 2

[6] C. Chevalley. *Theory of Lie groups*, volume 1. Princeton University Press, 1999. 4

[7] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *International journal of computer vision*, 9(2):83–112, 1992. 2

[8] Y. Furukawa, A. Sethi, J. Ponce, and D. Kriegman. Structure and motion from images of smooth textureless objects. In *Computer Vision-ECCV 2004*. Springer, 2004. 1, 2

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 2

[10] C. Hernández, F. Schmitt, and R. Cipolla. Silhouette coherence for camera calibration under circular motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):343–349, 2007. 2

[11] D. Q. Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009. 4

[12] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE Trans. Pat. Ana. & Mach. Int.*, 29(2):286–299, 2007. 6

[13] P. McIlroy and T. Drummond. Reconstruction from uncalibrated affine silhouettes. In *BMVC*. Citeseer, 2009. 1, 2

[14] P. R. S. Mendonca, K.-Y. Wong, and R. Cippolla. Epipolar geometry from profiles under circular motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):604–616, 2001. 2

[15] R. Pless and I. Simon. Embedding images in non-flat spaces. In *In: Proc. of the International Conference on Imaging Science, Systems, and Technology.(2002*. Citeseer, 2001. 3, 7

[16] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. (SIGGRAPH)*, 23(3):309–314, 2004. 8

[17] S. N. Sinha, M. Pollefeys, and L. McMillan. Camera network calibration from dynamic silhouettes. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–195. IEEE, 2004. 1, 2

[18] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph. (SIGGRAPH)*, 25(3):835–846, 2006. 2

[19] Visual Geometry Group, University of Oxford. Multi-view and Oxford Colleges building reconstruction. [Online]. Available: http://www.robots.ox.ac.uk/%7Evgg/data/data-mview.html. 8

[20] K.-Y. Wong and R. Cipolla. Reconstruction of sculpture from its profiles with unknown camera positions. *Image Processing, IEEE Transactions on*, 13(3):381–389, 2004. 2

[21] J. Y. Zheng. Acquiring 3-d models from sequences of contours. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(2):163–178, 1994. 2