Washington University Technical Report: WU-CS-01-43

Embedding Images in non-Flat Spaces

Robert Pless and Ian Simon

December 2001

Abstract

Multi-dimensional scaling is an analysis tool which transforms pairwise distances between points to an embedding of points in space which are consistent with those distances. Two recent techniques in statistical pattern recognition, locally linear embedding (LLE) and Isomap, give a mechanism for finding the structure underlying point sets for which comparisons or distances are only meaningful between nearby points. We give a direct method to extend the embedding algorithm to new topologies, finding the optimal embedding of points whose geodesic distance on a surface matches the given pairwise distance measurements. Surfaces considered include spheres, cylinders, tori, and their higher dimensional corollaries. We give examples of sets of images that come from spaces with these topologies. Using these embedding techniques, we compute pose estimates for thousands of images of an object without knowing the object model or finding corresponding points.

1 Introduction

This paper considers the problem of analysing thousands of images of a single object. With very low resolution images taken from unknown viewpoints, standard computer vision algorithms do not have a good handle to begin the image understanding process. Instead of corresponding points, one can start by considering image similarity measures. The question then becomes how to find the structure underlying a set of images from pairwise comparisons. Here we focus on data sets taken of a single object in different lighting and pose conditions. An example is given in Figure 1 — given a large set of unsorted images of a particular object (left), find an organization for these images without any a-priori knowledge of an object model (right).

One popular tool that could give an approach to this problem is called multi-dimensional scaling (MDS) [2]. This is a method for creating an embedding of a point set that respects the set of all pairwise distances. Naively applying this to image data requires a meaningful measure of distance between all pairs of

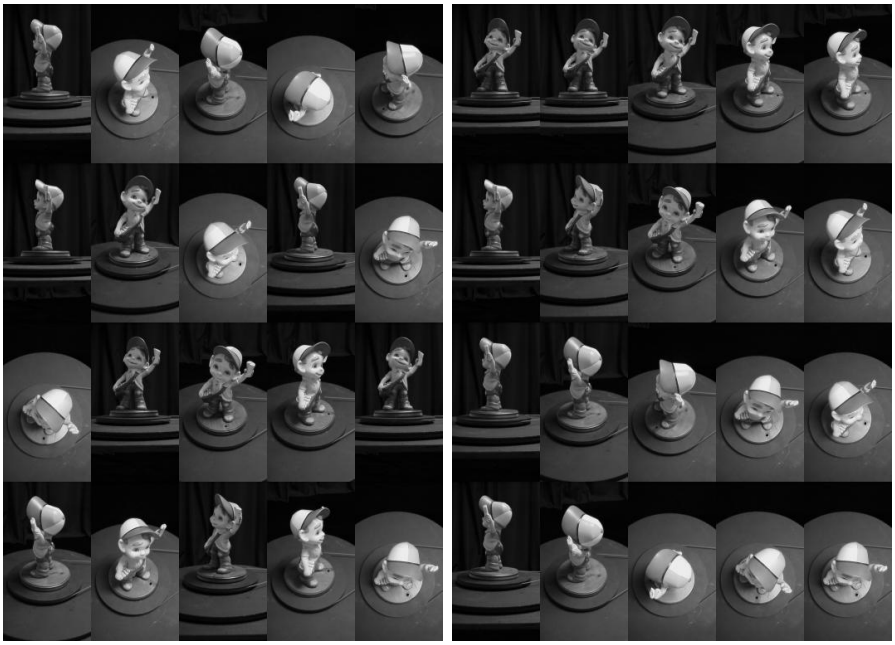

Figure 1: We consider the problem of organizing an unordered set of small images (left). Using Isomap or LLE, it is possible to automatically organize the pictures into a low dimensional parameter space, in this case a 2 dimensional space (right). The subject of this paper is to modify or extend these techniques in order to extract metric properties of camera angle and object pose, and to extend MDS techniques to allow direct embedding on spheres, cylinders, and tori.

images. For images taken from very similar viewpoints, almost any distance metric between images will be small. For images taken from dissimilar viewpoints, almost any image distance metric is likely to be uncorrelated with the actual distance between camera viewpoints. Therefore embedding the images in a parameter space using MDS directly is not satisfactory, because not all pairwise image comparisons are meaningful.

Fortunately, two recent papers give tools to allow a MDS like solution for situations when only a local similarity measure is available[3, 7]. Each of these tools takes as input *local* relationships between input data points. Each outputs coordinates for the data points that best satisfy the given relationships. Unlike principal component analysis, these coordinates do not have to correspond to a linear subspace of the space in which the original point set lies — If the point set lies in a low dimensional manifold (like a spiral jelly roll), the coordinates specify point locations within that manifold.

However, both of these techniques require that geodesic distances *within* the manifold are Euclidean, that is, the internal structure of the manifold must be

linear. If this manifold is topologically different than a plane, these techniques fail. Here we provide extensions to the classical MDS algorithm to embed points on a sphere, cylinder and torus.

This is an exploration into the use of image similarity measures for the calculation of metric parameters of interest. This is in contrast to creating a perceptual or intuitive classification or map which typically does not require the embedding to have meaningful coordinates, and differs from classification tasks which assign each image to one of a discrete set of categories. We show two experiments. The first is a pose estimation example which considers 1,800 images of an object and embeds the images into a space parameterized by the angle of elevation of the camera and the angle of rotation of the object. The second is a pose estimation problem with a fixed camera, a rotating object, and a light source which independently rotates around the object. In both cases the images were sub-sampled to be very small (on the order of 32 x 32 pixels). The mean pose angle estimation error for the first experiment was less than 6°, indicating the potential of these techniques for the analysis of very low resolution imagery. This is an important problem in many real world surveillance applications, which often have many low resolution images of a particular object.

Related to this work is [6], who does a similar dimensionality reduction by comparing a large set of images to a set of templates. Comparing the images to the templates avoids the need to compare all pairs of images, and gives a method to find a low dimensional Euclidean embedding of the image set, suitable for content based indexing. However, this still requires that a distance metric be valid for every pair of template and image, instead of a measure that need only be valid for very similar images. Other uses of MDS in the field of vision include [5] who uses the earth movers distance as a similarity measure and gives a perceptual organization of various image classes,

The main contribution of this work is to extend the MDS algorithm to solve for embeddings of points in non-flat spaces — to find point positions on spheres, tori, cylinders, and their higher dimensional counterparts. Since many image sets come from these topological spaces, this expands the applicability of MDS for the vision community. In contrast to many dimensionality reduction techniques which seek to classify or categorize images, these techniques are best suited to the analysis of image sets which are evenly sampled over some parameter space.

2 MDS, LLE and Isomap

This section begins with an overview of the mathematics behind MDS. Modifications of this procedure are necessary to embed in non-flat spaces. All of these procedures solve directly for the embedding and do not iteratively update the point positions. The subsections for spherical, cylindrical, and toroidal MDS are the main theoretical contributions of this work. This section concludes with an overview of LLE and Isomap, two methods that allow the application of

MDS type techniques to situations where not all point to point distances are available. Longer tutorials on these methods and MDS in general give more specific implementation details [4, 7, 2].

2.1 Multi-Dimensional Scaling

The input to the MDS procedure is a distance matrix D, an $n \times n$ matrix of pairwise distances. The following algorithm then computes the embedding.

MDS Algorithm (explanation taken from [7]):

1. Input D, a matrix of pairwise distances, D_{ij} is distance from point i to j, *assumed to be measured in some Euclidean space.*
2. Construct S, the squared distance matrix, ($S_{ij} = D_{ij}^2$)
3. Construct H, the centering matrix, ($H_{ij} = \delta_{ij} - 1/N$, where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.)
4. Define $\tau(D) = -HSH/2$. The centering matrix H is defined to make $\tau(D)$ a dot product form of the distance matrix, i.e. for a matrix X of point coordinates, $X^\top X = \tau(D)$ if and only if $\forall_{ij}(X_i - X_j)^\top (X_i - X_j) = S_{ij}$
5. Solve for positions X such that $X^\top X = \tau(D)$
   - let λ_p be the p-th eigenvalue (in decreasing order) of the matrix $\tau(D)$
   - let v_p^i be the i-th component of λ_p.
   - Set the p-th component of X_i to be $v_p^i \sqrt{\lambda_p}$.
6. Use the first k components of each position vector X_i for the optimal embedding in k-dimensional Euclidean space.

The key point of this algorithm is the centering matrix, which transforms the distance matrix into dot-product form, after which standard linear algebra tools are appropriate. For distances which arise from measurements in non-Euclidean space, it is necessary to use other methods to compute the dot-product form.

2.2 non-flat Multi-Dimensional Scaling

For many applications, the point of multi-dimensional scaling is to give a visualization of a data set. Since visualizations are projected onto a screen, and eventually, the viewers 2D roughly flat retinal surface, techniques have largely focused on embedding the points in 2D Euclidean space. The applications envisioned here seek to use thousands of images to extract metric information about

a set of images. If these images arise from a data set with a non-Euclidean distance measure, this embedding will be distorted. To avoid this distortion, we must take the given distance measures and embed them directly onto the appropriate surface.

2.2.1 Spherical MDS

The distance between two points on a sphere is the radius of the sphere times the angle between the position vectors of those points. A point set embedded on a 2D sphere has a different set of pairwise distances than a point set embedded in 3D space such that all points happen to lie on a sphere. The following algorithm finds the embedding on a k-dimensional sphere by changing the standard MDS procedure for finding the dot product matrix from the pairwise distances.

Spherical MDS Algorithm

1. Input, set of pairwise distances, D_{ij} *assumed to be measured along great circle of a sphere.*
2. Estimate radius of sphere $r = max_{ij}\frac{D_{ij}}{\pi}$.
3. set $\tau(D) = r^2 cos(\frac{D_{ij}}{r})$
4. solve for position vectors using step 5 from MDS procedure.
5. Choose the first k+1 components of each X_i position vector. Extend each vector to have length r for final embedding on k-dimensional sphere.

2.2.2 Cylindrical MDS

A cylindrical space is characterized by one dimension which is cyclical and one dimension which is flat. Embedding on a cylinder consists of two phase. It is important to embed the dimension along which there is the most variance first. For a cylinder which is much longer than its radius, the two steps are the following:

Cylindrical MDS Algorithm

1. Input, set of pairwise distances, D_{ij}.
2. Compute, squared pairwise distances, S_{ij}.
3. Use standard MDS to solve for optimal 1D embedding to find X_i for each point.
4. Compute residual distances $D'_{ij} = \sqrt{S_{ij} - (X_i - X_j)^2}$.
5. Use spherical MDS with distances D'_{ij} to find best embedding on a circle parameterized by Θ_i.
6. The pair (X_i, Θ_i) is the position of the point on the cylinder.

For a cylinder whose radius is longer than its length, the circular embedding must be done first, then the residual distances are computed and standard MDS is applied to find the flat X_i coordinate. When one of the two dimensions (circular or flat) does not account for most of the initial distances between points, the algorithm often fails. The most likely explanation is the potential difficulty in the spherical MDS algorithm of finding an estimate of the radius of the cyclic dimension.

2.2.3 Toroidal MDS

A toroidal space is characterized by two cyclical dimensions. A set of images with a toroidal topology is given in Section 4. Embedding on a torus is a two phase process, embed on a circle, compute the residual distances, and embed on a circle again.

Toroidal MDS Algorithm

1. Input, set of pairwise distances, D_{ij}.
2. Compute, squared pairwise distances, S_{ij}.
3. Use spherical MDS with distances D_{ij} to find best embedding on a circle parameterized by Θ_i.
4. Compute residual distances $D'_{ij} = \sqrt{S_{ij} - r^2(\Theta_i - \Theta_j)}$.
5. Use spherical MDS with distances D'_{ij} to find best embedding on a circle parameterized by Φ_i.
6. The pair (Θ_i, Φ_i) is the position of the point on the torus.

2.2.4 Sparse Distance Measurements

MDS, of any form, would be useless for vision applications without the ability to deal with sparse distance measurements. Image similarity measurements are only accurate or meaningful for images that have a high correlation. The following two techniques were recently developed to allow MDS algorithms to work for sparse distance measurements.

- Isomap
  Input: an $n \times n$ matrix pairwise distances with some (perhaps most) distances unknown.
  Output: Point coordinates such that the pairwise distances are best approximated.
  Method: Define a graph whose vertices are the set of points, and whose edges are the known pairwise distances. Compute all-pairs shortest path distances in this graph, which defines a distance between every pair of nodes. Use MDS to find point coordinates which satisfy these (now complete) distance constraints.

- Locally Linear Embedding (LLE)
  Input: A $n \times n$ weight matrix W which expresses each point as a weighted sum of other points (probably neighbors).
  Output: Point coordinates best fitting the local constraints
  Method: Solve an Eigenvalue problem to find reasonable point coordinates X such that $WX = X$.

Both Isomap and LLE output a set of point coordinates. In the subsequent section, we explore techniques to force those point coordinates to have a meaning in terms of parameters defining the image set..

3 Constrained Embeddings

The matrix of pairwise distances is invariant to rigid transformations of the point coordinates. Extra knowledge is required to transform the embedded point set into one that expresses metric information. There are two categories of external knowledge that can be brought to bear on the embedding process. The first method is to enforce absolute knowledge of the desired parameter location for one or a small set of the points. The second is to enforce global properties of the embedding, for instance the knowledge that the data set comes from an even sampling of the desired parameterization. The form of global constraints which are appropriate is highly application dependent, and we discuss techniques used in our experiment within the experimental section.

3.1 Local Constraints

The LLE approach to embedding the point set starts with a weight matrix W which which expresses each point as a weighted sum of other points. It then

seeks a set of point coordinates X which respect this weighting:

$$WX = X, \text{ or,}$$
$$(W - I)X = 0$$

Requiring that certain points must be embedded in particular locations requires the solution to a similar problem:

$$(W' - I)X' = C,$$

where X' is the remaining unknown point coordinates for which we are solving, W' is the matrix of relative constraints between these points, and C is a matrix encoding the effect of the location of the fixed points.

Alternatively, the points can be warped after the fact to force the satisfaction of a particular set of constraints. Different warping functions may be required depending upon the number of points whose position is fixed. A general linear transform is an 8 parameter transform allowing any four points to be fixed. The four embedded points, (x, y) and their desired locations (x', y'), define a linear system which can be solved (for unknowns a, b, c, d, e, f, g, h) to define the transformation for each point:

$$(x', y') = (\frac{ax + by + c}{gx + hy + 1}, \frac{dx + ey + f}{gx + hy + 1}).$$

When embedding points into a flat space, this transformation can force the axes of the embedded space to conform to parameters of interest, such as pose angles.

4 Experiment

We present two experiments, each using a large number of images captured from a known space with two parameters. In each case the images were sub-sampled to be very small, both for computational efficiency, and to illustrate that these algorithms work with very low resolution imagery. Both image data sets were captured with the object capture device shown in Figure 2. The first experiment requires the embedding of the images into a flat space parameterized by object pose and camera angle. The second experiment uses images which are parameterized by two cyclic dimensions, the lighting angle and the object pose angle which varied independently through all 360°.

4.1 Flat Embedding

The object capture system captured 1,800 images of the object shown in Figures 1 and 5. These images evenly sample the space of object rotations (every 3 degrees, over one half a rotation) and camera viewing angle (every 3 degrees, from horizontal to vertical). The images were sub-sampled to 32 x 64 pixels, and all pairs of image distances were computed as both the sum of squares of

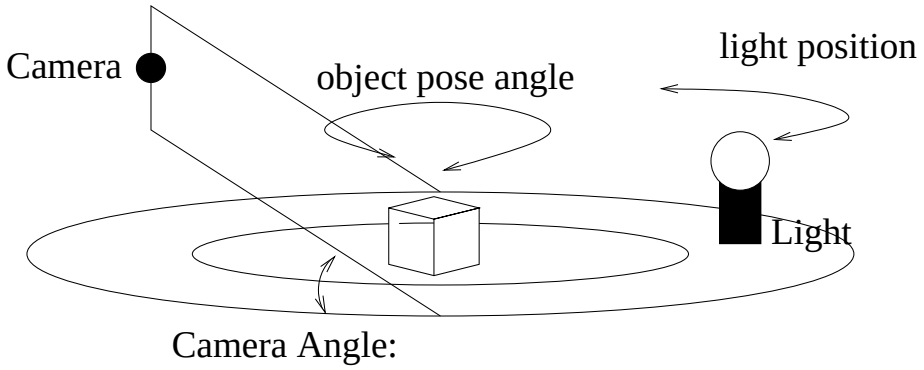

Figure 2: Object capture system, can take pictures of an object from any camera angle and object rotation. For this experiment, we took 1,800 images, sampling ϕ and θ every 3 degrees (data set available upon request, uncompressed pgm format).

differences of normalized pixel intensity, and the sum of squares of differences of normalized edge maps. We found that using the edge image gave qualitatively similar but slightly better and more consistent results that using distances computed from original images. These images were embedded using Isomap and LLE.

For Isomap, the local neighborhood graph of each image consisted of the 8 nearest images, all other distances were initially unknown and defined during the shortest path component of the Isomap procedure. For LLE, each point was expressed as a weighted sum of its neighbors using the following algorithm suggested in [4]. Using all pairwise distance between a point and its eight nearest neighbors, embed these (nine) points using MDS. Then, express the central point as a weighted sum of its neighbors, and use these weights as the input constraints to LLE.

Figure 3 shows the result of the LLE embedding of all 1800 images. The four points corresponding to the (known) extremes of camera angles and object rotations are marked with small circles. In the coordinate system defined by these four points, the points should be arranged as a rectilinear grid. This metric structure underlying the point set is not exhibited by this embedding. Exploring why this fails is a subject of future work — It may be better to compute each image directly as a linear combination of neighbor rather than first computing distances, then locally embedding, then using that local structure. The qualitative structure found is shown in Figure 1 (right); as one moves in a path through neighboring points in this embedding, there is a smooth transition between image viewpoints, but the embedding does not directly capture the parameters of the object pose.

Figure 4 (top) shows results using the standard Isomap procedure, without enforcing external constraints on the coordinates. Choosing four extreme points

Figure 3: Locally Linear Embedding: Each image was expressed as a linear combination of nearby images. The points were embedded with the additional constraint that the four corner images lie at fixed positions at the corner of a square.

(circled), and solving for the general linear transform which forces these four points to have specific coordinates (bottom left) allows one to define meaningful axes to the embedded space. Finally, the a-priori knowledge that the parameter space was evenly sampled gives a global constraint on the embedded point set. The final embedding uses a variant of the thin-plate spline warping technique [1] to enforce that the density of points in every region of the embedded space is approximately constant (bottom right). Evenly sampling this space and choosing the closest image to the sample points gives a graphical depiction of this embedding (Figure 5).

Finally, since this data set was taken in a laboratory setting, the actual pose coordinates are known for each image. Over all 1800 images, the mean error in the embedded θ, ϕ coordinates was: $6.98°, 2.97°$. This numbers should not be compared directly to other pose estimation algorithms, and are extremely good, given that they come from the analysis of 32 x 64 pixel images of an unknown object.

4.2 Embedding on a torus

A second experiment considered a data set where the images are parameterized by two cyclic dimensions. In this case it is not possible to embed these images in a 2d flat parameter space, so neither MDS (augmented with Isomap) nor

Figure 4: At the top is the initial Isomap solution embedding the positions of all 1,800 images. In the bottom left, this embedding is warped by a general linear transform so that the four circles points lie in fixed points in the final embedding. In the bottom right image, a variant of thin plate spline enforces the constraint that the parameter space was evenly sampled. The horizontal axis of this plot corresponds to the camera angle ϕ, and the vertical axis is the object rotation angle θ.

LLE would not be able to find an appropriate embedding. The object capture device captured 3,600 images, evenly sampling the entire $[0, 360°]$ range of object pose angle and light position. Examples of the input image data are shown in Figure 6. Each image was taken from the same camera position. The images were sub-sampled to 32×24 pixels each and the image distance measure was the sum of squared pixel intensity differences. This image distance was computed for all pairs of images.

The embedding began using the Isomap procedure. The local neighborhood graph of each image consisted of the 8 nearest images, all other distances were initially unknown and defined during the shortest path component of the Isomap procedure. The image embedding, computed using the toroidal MDS procedure defined in Section 2.2.3, gives an embedding of the images shown in Figure 7 (top). The two axes of this embedding correspond to the object pose angle and the lighting angle. This organization of the image comes directly from the embedding procedure and does not require any transformations as were needed in the flat embedding. The structure of the space of images is illustrated with sample images drawn on a torus in Figure 7 (bottom). The mean error in the embedding of the lighting angle was 2.7°.

The embedding of the pose angle is less accurate. Although the data set evenly samples the set of all lighting angles and object pose angles, it is visible from the embedding that for some lighting angles (positions along the x-axis),

the images do not cover all pose angles. The cause of this is the image set itself, in the input data (Figure 6), for some lighting conditions, one view of the lizard appears very similar to the view after an object rotation of 180°. In this case, the space of images is not cyclic, so it cannot be effectively embedded on the torus. For lighting conditions where this is not the case, the embedding covers all of both dimensions.

The main insight to be gained from this experiment is that analysis of images by similarity measures requires not "images from nearby points in the parameter space must be judged to be similar", but rather "images from far away points in the parameter space must not be judged to be similar".

References

[1] F Bookstein. Principal warps: Thin-plate splines and the decomposition ofdeformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989.

[2] Ingwer Borg and Patrick Groenen. *Modern Multidimensional Scaling: Theory and Applications.* Springer-Verlag, 1997.

[3] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000.

[4] Sam T Roweis and Lawrence K Saul. An introduction to locally linear embedding. http://www.gatsby.ecl.ac.uk/~roweis/lle/papers/lleintro.pdf, 2001.

[5] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A metric for distributions with aplications to image databases. In *Proc. International Conference on Computer Vision*, 1998.

[6] Haim Schweitzer. Template matchine apporach to content based image indexing by low dimensional euclidean embedding. In *Proc. International Conference on Computer Vision*, pages 566–569, 2001.

[7] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A gloval geometric framework for nonlinear dimensionality reduction. *Science*, 2000.

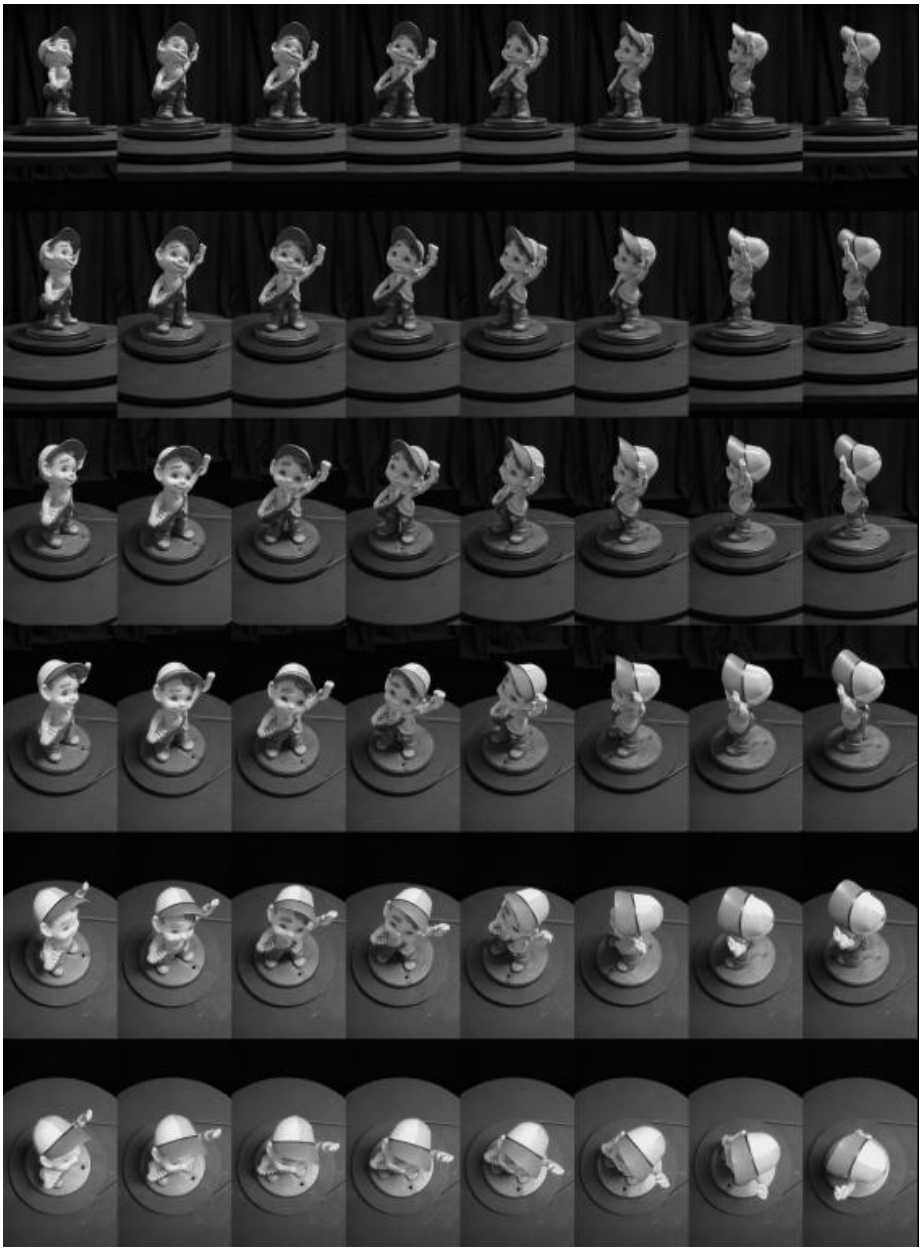

Figure 5: Images from an even sampling of the final embedded space shown in Figure 4, (bottom right). The mean embedding error over all 1800 images is only 6.98° for the rotation, and 2.97° for the camera elevation.
.

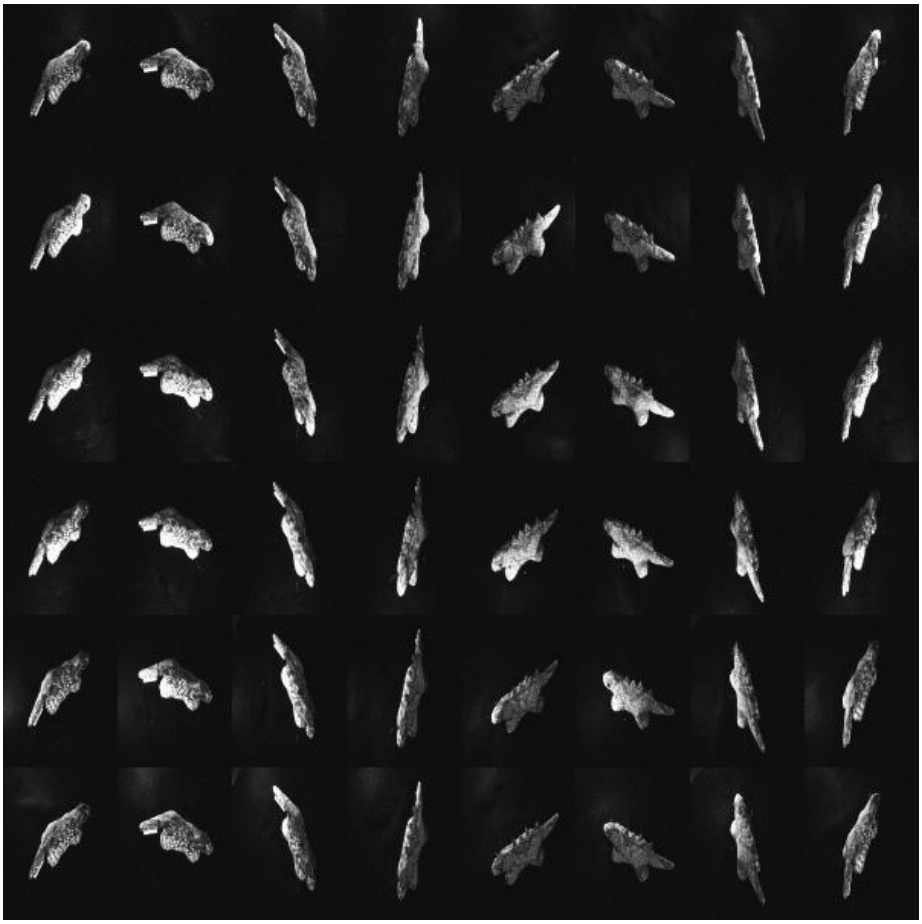

Figure 6: Example images from the data set used for the second experiment. A small object was imaged for different pose angles and lighting conditions. 3600 images sample the space evenly.

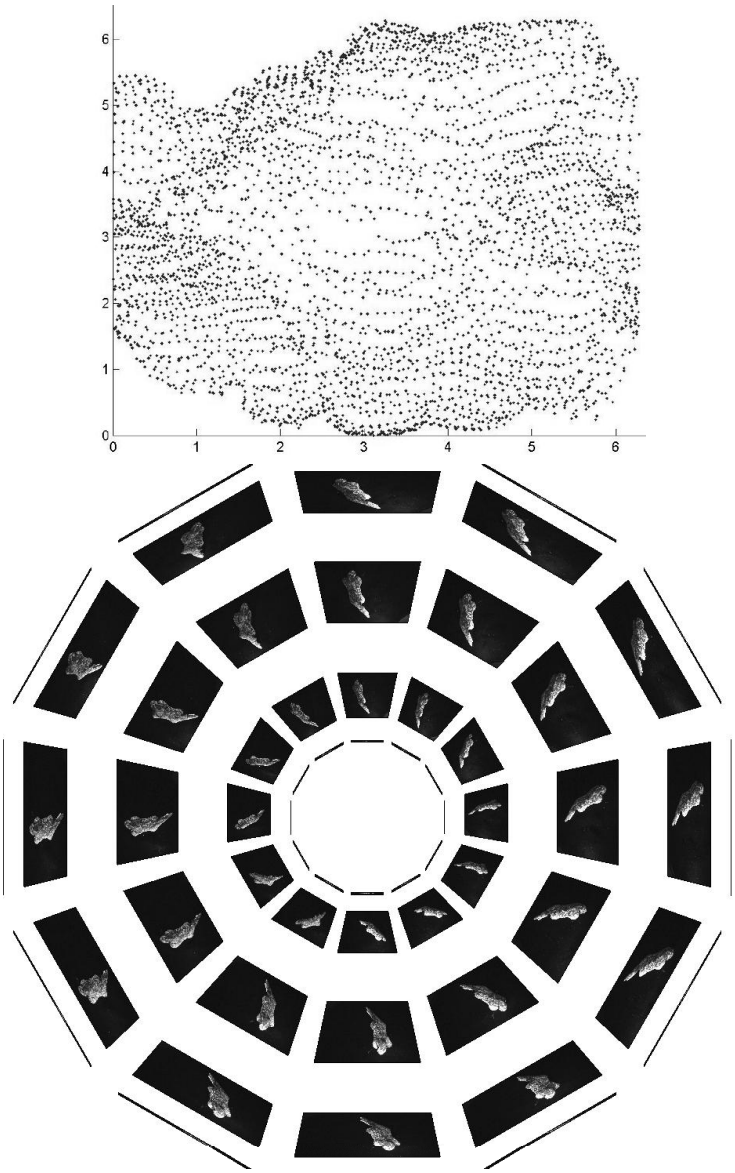

Figure 7: (top) embedded points from the lizard data sequence. the x-axis corresponds to the lighting angle, the y-axis corresponds to the object pose angle. This is a toroidal embedding, along each axis points with coordinate $0 + \epsilon$ are very close to points with coordinate $2\pi - \epsilon$. (bottom) Illustration of the images embedded on the torus.

.