Nonlinear Image Interpolation using Manifold Learning

Christoph Bregler
Computer Science Division
University of California
Berkeley, CA 94720
bregler@cs.berkeley.edu

Stephen M. Omohundro*
Int. Computer Science Institute
1947 Center Street Suite 600
Berkeley, CA 94704
om@research.nj.nec.com

Abstract

The problem of interpolating between specified images in an image sequence is a simple, but important task in model-based vision. We describe an approach based on the abstract task of "manifold learning" and present results on both synthetic and real image sequences. This problem arose in the development of a combined lip-reading and speech recognition system.

1 Introduction

Perception may be viewed as the task of combining impoverished sensory input with stored world knowledge to predict aspects of the state of the world which are not directly sensed. In this paper we consider the task of *image interpolation* by which we mean hypothesizing the structure of images which occurred between given images in a temporal sequence. This task arose during the development of a combined lip-reading and speech recognition system [3], because the time windows for auditory and visual information are different (30 frames per second for the camera vs. 100 feature vectors per second for the acoustic information). It is an excellent visual test domain in general, however, because it is easy to generate large amounts of test and training data and the performance measure is largely "theory independent". The test consists of simply presenting two frames from a movie and comparing the

*New address: NEC Research Institute, Inc., 4 Independence Way, Princeton, NJ 08540

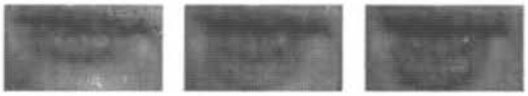


Figure 1: Linear interpolated lips.

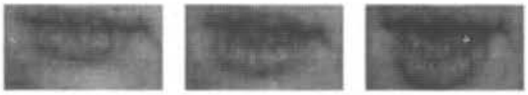


Figure 2: Desired interpolation.

hypothesized intermediate frames to the actual ones. It is easy to use footage of a particular visual domain as training data in the same way.

Most current approaches to model-based vision require hand-constructed CAD-like models. We are developing an alternative approach in which the vision system builds up visual models automatically by learning from examples. One of the central components of this kind of learning is the abstract problem of inducing a smooth nonlinear constraint manifold from a set of examples from the manifold. We call this "manifold learning" and have developed several approaches closely related to neural networks for doing it [2]. In this paper we apply manifold learning to the image interpolation problem and numerically compare the results of this "nonlinear" process with simple linear interpolation. We find that the approach works well when the underlying model space is low-dimensional. In more complex examples, manifold learning cannot be directly applied to images but still is a central component in a more complex system (not discussed here).

We present several approaches to using manifold learning for this task. We compare the performance of these approaches to that of simple linear interpolation. Figure 1 shows the results of linear interpolation of lip images from the lip-reading system. Even in the short period of 33 milliseconds *linear* interpolation can produce an unnatural lip image. The problem is that linear interpolation of two images just averages the two pictures. The interpolated image in Fig. 1 has two lower lip parts instead of just one. The desired interpolated image is shown in Fig. 2, and consists of a single lower lip positioned at a location between the lower lip positions in the two input pictures.

Our interpolation technique is *nonlinear*, and is constrained to produce only images from an abstract manifold in "lip space" induced by learning. Section 2 describes the procedure, Section 4 introduces the interpolation technique based on the induced manifold, and Sections 5 and 6 describe our experiments on artificial and natural images.

2 Manifold Learning

Each $n * m$ graylevel image may be thought of as a point in an $n * m$-dimensional space. A sequence of lip-images produced by a speaker uttering a sentence lie on a

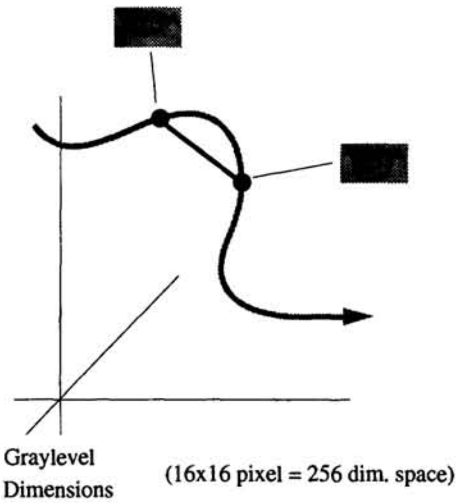


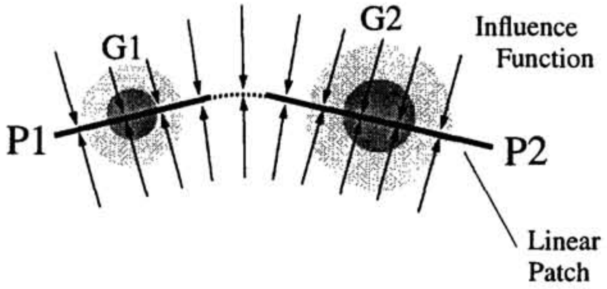
Figure 3: Linear vs nonlinear interpolation.

1-dimensional trajectory in this space (figure 3). If the speaker were to move her lips in all possible ways, the images would define a low-dimensional submanifold (or nonlinear surface) embedded in the high-dimensional space of all possible graylevel images.

If we could compute this nonlinear manifold, we could limit any interpolation algorithm to generate only images contained in it. Images not on the manifold cannot be generated by the speaker under normal circumstances. Figure 3 compares a curve of interpolated images lying on this manifold to straight line interpolation which generally leaves the manifold and enters the domain of images which violate the integrity of the model.

To represent this kind of nonlinear manifold embedded in a high-dimensional feature space, we use a mixture model of local linear patches. Any smooth nonlinear manifold can be approximated arbitrarily well in each local neighborhood by a linear "patch". In our representation, local linear patches are "glued" together with smooth "gating" functions to form a globally defined nonlinear manifold [2]. We use the "nearest-point-query" to define the manifold. Given an arbitrary point near the manifold, this returns the closest point on the manifold. We answer such queries with a weighted sum of the linear projections of the point to each local patch. The weights are defined by an "influence function" associated with each linear patch which we usually define by a Gaussian kernel. The weight for each patch is the value of its influence function at the point divided by the sum of all influence functions ("partition of unity"). Figure 4 illustrates the nearest-point-query. Because Gaussian kernels die off quickly, the effect of distant patches may be ignored, improving computational performance. The linear projections themselves consist of a dot product and so are computationally inexpensive.

For learning, we must fit such a mixture of local patches to the training data. An initial estimate of the patch centers is obtained from k-means clustering. We fit a patch to each local cluster using a local principal components analysis. Fine tuning



$$P(x) = \frac{\sum_i Gi(x) \cdot Pi(x)}{\sum_i Gi(x)}$$

Figure 4: Local linear patches glued together to a nonlinear manifold.

of the model is done using the EM (expectation-maximization) procedure.

This approach is related to the mixture of expert architecture [4], and to the manifold representation in [6]. Our EM implementation is related to [5], which uses a hierarchical gating function and local experts that compute linear mappings from one space to another space. In contrast, our approach uses a "one-level" gating function and local patches that project a space into itself.

3 Linear Preprocessing

Dealing with very high-dimensional domains (e.g. $256 * 256$ graylevel images) requires large memory and computational resources. Much of this computation is not relevant to the task, however. Even if the space of images is nonlinear, the nonlinearity does not necessarily appear in all of the dimensions. Earlier experiments in the lip domain [3] have shown that images projected onto a 10-dimensional linear subspace still accurately represents all possible lip configurations. We therefore first project the high-dimensional images into such a linear subspace and then induce the nonlinear manifold within this lower dimensional linear subspace. This preprocessing is similar to purely linear techniques [7, 10, 9].

4 Constraint Interpolation

Geometrically, linear interpolation between two points in n-space may be thought of as moving along the straight line joining the two points. In our non-linear approach to interpolation, the point moves along a curve joining the two points which lies in the manifold of legal images. We have studied several algorithms for estimating the shortest manifold trajectory connecting two given points. For the performance results, we studied the point which is halfway along the shortest trajectory.

4.1 "Free-Fall"

The computationally simplest approach is to simply project the linearly interpolated point onto the nonlinear manifold. The projection is accurate when the point is close to the manifold. In cases where the linearly interpolated point is far away (i.e. no weight of the partition of unity dominates all the other weights) the closest-point-query does not result in a good interpolant. For a worst case, consider a point in the middle of a circle or sphere. All local patches have same weight and the weighted sum of all projections is the center point itself, which is not a manifold point. Furthermore, near such "singular" points, the final result is sensitive to small perturbations in the initial position.

4.2 "Manifold-Walk"

A better approach is to "walk" along the manifold itself rather than relying on the linear interpolant. Each step of the walk is linear and in the direction of the target point but the result is immediately projected onto the manifold. This new point is then moved toward the target point and projected onto the manifold, etc. When the target is finally reached, the arc length of the curve is approximated by the accumulated lengths of the individual steps. The point half way along the curve is chosen as the interpolant. This algorithm is far more robust than the first one, because it only uses local projections, even when the two input points are far from each other. Figure 5b illustrates this algorithm.

4.3 "Manifold-Snake"

This approach combines aspects of the first two algorithms. It begins with the linearly interpolated points and iteratively moves the points toward the manifold. The *Manifold-Snake* is a sequence of n points preferentially distributed along a smooth curve with equal distances between them. An energy function is defined on such sequences of points so that the energy minimum tries to satisfy these constraints (smoothness, equidistance, and nearness to the manifold):

$$E = \sum_i \alpha||v_{i-1} - 2v_i + v_{i+1}||^2 + \beta||v_i - proj(v_i)||^2 \tag{1}$$

E has value 0 if all v_i are evenly distributed on a straight line and also lie on the manifold. In general E can never be 0 if the manifold is nonlinear, but a minimum for E represents an optimizing solution. We begin with a straight line between the two input points and perform gradient descent in E to find this optimizing solution.

5 Synthetic Examples

To quantify the performance of these approaches to interpolation, we generated a database of $16 * 16$ pixel images consisting of rotated bars. The bars were rotated for each image by a specific angle. The images lie on a one-dimensional nonlinear manifold embedded in a 256 dimensional image space. A nonlinear manifold represented by 16 local linear patches was induced from the 256 images. Figure 6a shows

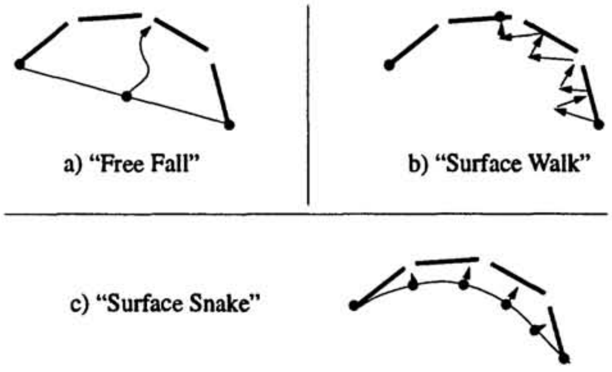


Figure 5: Proposed interpolation algorithms.

Figure 6: a) Linear interpolation, b) nonlinear interpolation.

two bars and their linear interpolation. Figure 6b shows the nonlinear interpolation using the *Manifold-Walk* algorithm.

Figure 7 shows the average pixel mean squared error of linear and nonlinear interpolated bars. The x-axis represents the relative angle between the two input points.

Figure 8 shows some iterations of a *Manifold-Snake* interpolating 7 points along a 1 dimensional manifold embedded in a 2 dimensional space.

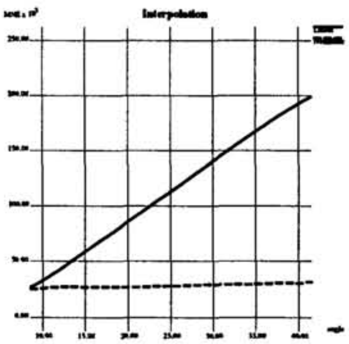


Figure 7: Average pixel mean squared error of linear and nonlinear interpolated bars.

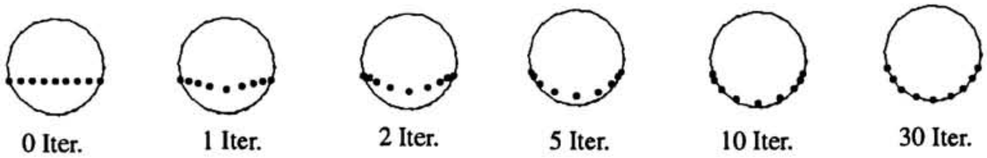


Figure 8: Manifold-Snake iterations on an induced 1 dimensional manifold embedded in 2 dimensions.

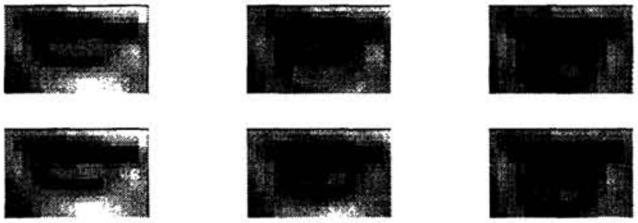


Figure 9: 16x16 images. Top row: linear interpolation. Bottom row: nonlinear "manifold-walk" interpolation.

6 Natural Lip Images

We experimented with two databases of natural lip images taken from two different subjects.

Figure 9 shows a case of linear interpolated and nonlinear interpolated $16 * 16$ pixel lip images using the *Manifold-Walk* algorithm. The manifold consists of 16 4-dimensional local linear patches. It was induced from a training set of 1931 lip images recorded with a 30 frames per second camera from a subject uttering various sentences. The nonlinear interpolated image is much closer to a realistic lip configuration than the linear interpolated image.

Figure 10 shows a case of linear interpolated and nonlinear interpolated $45 * 72$ pixel lip images using the *Manifold-Snake* algorithm. The images were recorded with a high-speed 100 frames per second camera[1]. Because of the much higher dimensionality of the images, we projected the images into a 16 dimensional linear subspace. Embedded in this subspace we induced a nonlinear manifold consisting of 16 4-dimensional local linear patches, using a training set of 2560 images. The linearly interpolated lip image shows upper and lower teeth, but with smaller contrast, because it is the average image of the open mouth and closed mouth. The nonlinearly interpolated lip images show only the upper teeth and the lips half way closed, which is closer to the real lip configuration.

7 Discussion

We have shown how induced nonlinear manifolds can be used to constrain the interpolation of graylevel images. Several interpolation algorithms were proposed

[1] The images were recorded in the UCSD Perceptual Science Lab by Michael Cohen

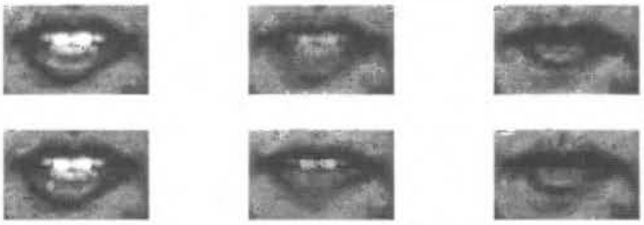


Figure 10: 45x72 images projected into a 16 dimensional subspace. Top row: linear interpolation. Bottom row: nonlinear "manifold-snake" interpolation.

and experimental studies have shown that constrained nonlinear interpolation works well both in artificial domains and natural lip images.

Among various other nonlinear image interpolation techniques, the work of [1] using a Gaussian Radial Basis Function network is most closely related to our approach. Their approach is based on feature locations found by pixelwise correspondence, where our approach directly interpolates graylevel images.

Another related approach is presented in [8]. Their images are also first projected into a linear subspace and then modelled by a nonlinear surface but they require their training examples to lie on a grid in parameter space so that they can use spline methods.

References

[1] D. Beymer, A. Shahsua, and T. Poggio *Example Based Image Analysis and Synthesis* M.I.T. A.I. Memo No. 1431, Nov. 1993.

[2] C. Bregler and S. Omohundro, *Surface Learning with Applications to Lip-Reading,* in Advances in Neural Information Processing Systems 6, Morgan Kaufmann, 1994.

[3] C. Bregler and Y. Konig, *"Eigenlips" for Robust Speech Recognition* in Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Adelaide, Australia, 1994.

[4] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton, *Adaptive mixtures of local experts* in Neural Compuation, 3, 79-87.

[5] M. I. Jordan and R. A. Jacobs, *Hierarchical Mixtures of Experts and the EM Algorithm* Neural Computation, Vol. 6, Issue 2, March 1994.

[6] N. Kambhatla and T.K. Leen, *Fast Non-Linear Dimension Reduction* in Advances in Neural Information Processing Systems 6, Morgan Kaufmann, 1994.

[7] M. Kirby, F. Weisser, and G. Dangelmayr, *A Model Problem in Represetation of Digital Image Sequences,* in Pattern Recgonition, Vol 26, No. 1, 1993.

[8] H. Murase, and S. K. Nayar *Learning and Recognition of 3-D Objects from Brightness Images* Proc. AAAI, Washington D.C., 1993.

[9] P. Simard, Y. Le Cun, J. Denker *Efficient Pattern Recognition Using a New Transformation Distance* Advances in Neural Information Processing Systems 5, Morgan Kaufman, 1993.

[10] M. Turk and A. Pentland, *Eigenfaces for Recognition* Journal of Cognitive Neuroscience, Volume 3, Number 1, MIT 1991.