Learning Object-Independent Modes of Variation with Feature Flow Fields

Erik G. Miller, Kinh Tieu, Chris Stauffer
MIT Artificial Intelligence Lab
Cambridge, MA 02139
{*emiller,tieu,stauffer*}*@ai.mit.edu*

Abstract

We present a unifying framework in which *object-independent* modes of variation are learned from continuous-time data such as video sequences. These modes of variation can be used as *generators* to produce a manifold of images of a new object from a single example of that object.

We develop the framework in the context of a well-known example: analyzing the modes of spatial deformations of a scene under camera movement. Our method learns a close approximation to the standard affine deformations that are expected from the geometry of the situation, and does so in a completely unsupervised (i.e. ignorant of the geometry of the situation) fashion. We stress that it is learning a *parameterization*, not just the parameter values, of the data. We then demonstrate how we have used the same framework to derive a novel data-driven model of joint color change in images due to common lighting variations. The model is superior to previous models of color change in describing non-linear color changes due to lighting.

Acknowledgements

Support for this work included ONR grant #'s N00014-00-1-0089 and N00014-00-1-0907 and Project Oxygen at MIT. We thank Professor Tommi Jaakkola for reviewing the memo and providing suggestions.

1 Introduction

Human beings have the remarkable ability, given a single image of an object, to understand[1] how it will appear under changes in current viewing parameters such as (camera) position and lighting. How is it possible for us to have such good models of so many different types of objects? One could adopt the view that we must learn a model of variation for each type of object in the world. However, this is contradicted by the fact that people are able to understand how an object will change appearance even if they have seen only a single example of that object.

Another possibility is that we have highly generic models of images that apply to all objects. For example, some general inference can be done for the images of a new object by considering the statistics of "natural images". However, such generic class-independent models of image data tend to be very weak in their predictive power since images are so highly variable.

To address this problem, we propose developing object-independent models of *image change*, rather than models of images themselves. Although images of different objects may vary greatly, by using the right representation, we can represent changes in those images equivalently. In addition, we can develop a probabilistic model of these changes in the common representation. Subsequently, to create an image model for a new object (from, say, a single example), we can use a single image and the global model of change to estimate a manifold of possible images of the object. A primary goal of this paper is to unify under one mathematical framework the modeling of feature changes for very different types of features.

We should also point out that while a single "mode of variation" often will not serve to describe changes over an entire image, a combination of several such modes often will do the job. In fact, if we segment an image according to the change structure, we have good reason to believe that this segmentation will be meaningfully related to the physical composition of the scene. These issues will be discussed near the end of the memo.

In Section 2, we define the *feature flow field*, a vector field which maps one image into another by mapping the value of the feature at each pixel in the source image to a new feature value. The new image is then constructed from the new feature values. In Section 3, we focus on one well-known type of feature flow fields: optical flow fields. We describe how image deformation fields commonly used for modeling image variability (that is, affine optical flow fields) can be recovered by performing simple clustering algorithms or dimensionality reduction techniques on noisy observations of empirical optical flow fields. In Section 4, we show the generality of feature flow methods by developing a novel model of joint color change in images due to common lighting changes. We demonstrate how to generate a manifold of images of an object given a basis of principal flows and a single example of the object. In Section 5, we propose that while a single flow field may not describe changes well for a complex object, we can adopt more complex models such as mixtures of feature flow fields based on the same principles. In fact, we will describe how variations in feature flow fields can in principle be used to discover structure in scenes.

[1]We take as evidence of this "understanding" the ability of humans to recognize an object under lighting condition A when they have seen the object only under lighting condition B.

2 The feature flow field

In the following, let $\mathbf{f}(p) \in \mathbb{R}^D$ be the vector valued feature of a pixel p in an image. Further assume that each component of $\mathbf{f}(p)$ takes values from a set $\mathcal{S}$. Let $\mathcal{F} = \{s \in \mathcal{S}^D\}$ be the set of all possible feature vector values. We will consider two types of features in this paper. In the first application, we will use *coordinate features*, in which $\mathbf{f}(p)$ will represent the coordinates of some object in the image. In the second application, the features will be *color features*, in which $\mathbf{f}(p)$ will represent the integer RGB color values of the pixel p.

Suppose we are given two N-pixel images I_1 and I_2 of the same scene taken under two different parameter settings, represented by θ_1 and θ_2. We assume the images are consecutive images from a video sequence, and that the parameter values vary smoothly through time. We assume that we have a method of putting the images in correspondence[2]. Each pair of corresponding image pixels p_1^k and $p_2^k, 1 \leq k \leq N$, in the two images can be interpreted as a mapping $\mathbf{f}(p_1^k) \mapsto \mathbf{f}(p_2^k)$. That is, it tells us how a particular pixel's feature changed from image I_1 to image I_2. This single feature mapping is conveniently represented simply by the vector difference between the two pixel features:

$$\mathbf{d}(p_1^k, p_2^k) = \mathbf{f}(p_2^k) - \mathbf{f}(p_1^k). \tag{1}$$

By computing N of these vector differences (one for each pair of pixels) and placing each vector difference at the point $\mathbf{f}(p_1^k)$ in the feature space $\mathcal{F}$, we have created a vector field that is defined at all points in $\mathcal{F}$ for which there are feature values in image I_1.

That is, we are defining a vector field Φ' over $\mathcal{F}$ via

$$\Phi'(\mathbf{f}(p_1^k)) = \mathbf{d}(p_1^k, p_2^k), \qquad 1 \leq k \leq N. \tag{2}$$

This can be visualized as a collection of N arrows in feature space, each arrow going from a source feature to a destination feature based on the parameter change $\theta_1 \mapsto \theta_2$. We call this vector field Φ' a *partially observed feature flow.* The "partially observed" indicates that the vector field is only defined at the particular feature points that happen to be observed in image I_1.

To obtain a *full feature flow*, i.e. a vector field Φ defined at all points in $\mathcal{F}$, from a partially observed feature flow Φ', we must address two issues. First, there may be points in $\mathcal{F}$ at which no vector difference is defined. Second, there may be multiple pixels of a particular feature value in image I_1 that correspond to different feature values in image I_2. We propose the following radial basis function interpolation scheme, which defines the flow at a feature point $\mathbf{f}^*$ by computing a weighted proximity-based average of observed "flow vectors":

$$\Phi(\mathbf{f}^*) = \frac{\sum_{k=1}^{N} e^{-\|\mathbf{f}(p_1^k) - \mathbf{f}^*\|^2/2\sigma^2} \Phi'(\mathbf{f}(p_1^k))}{\sum_{k=1}^{N} e^{-\|\mathbf{f}(p_1^k) - \mathbf{f}^*\|^2/2\sigma^2}}. \tag{3}$$

This defines a feature flow vector at every point in $\mathcal{F}$. Note that the Euclidean distance function used is defined in *feature space*, not necessarily in the space defined by the [x,y] coordinates of the image. σ^2 is a variance term which controls the mixing of observed flow vectors to form the interpolated flow vector. As $\sigma^2 \to 0$, the interpolation scheme degenerates to a nearest-neighbor scheme, and as $\sigma^2 \to \infty$, all flow vectors get set to the average observed flow vector. The value of σ may need to be adjusted for the type of feature used. Note that feature flows are defined

[2]Since the image sequences are assumed to be continuous in time, this will usually be fairly easy to do.

so that a feature point with only a single nearby neighbor will inherit a flow vector that is nearly *parallel* to its neighbor.

We have thus outlined a procedure for using a pair of corresponding images $\mathcal{I} = (I_1, I_2)$ to generate a full feature flow field. We will write for brevity $\Phi = \Phi(\mathcal{I})$ to designate the flow generated from the image pair $\mathcal{I}$. We now apply this framework to real learning problems by using position and color features.

3 Optical flow fields

To apply the method to variations in images due to deformations, we need to define a *position feature*. Note that we will not explicitly favor any type of deformations over any others, except that our interpolation routine introduces a small amount of smoothing. In particular, the deformations are not parameterized except by their coordinate values.

For a pair of consecutive images I_1, I_2, we define the feature value of pixel p under image I as follows. Let $O(p)$ be the "object" that occurs at pixel p in image I. Let $\mathbf{f}(p)$ be the coordinate of that object O as depicted in image I_1. Under this definition, $\mathbf{f}(p_2^k) - \mathbf{f}(p_1^k)$ will be the image translation necessary to bring the object O in image I_1 in correspondence the same object in image I_2. Thus, adopting this type of coordinate feature, a feature flow field is just an optical flow field that puts the two images in correspondence.

3.1 Structure in optical flow fields

Several authors have used structure in optical flow both to analyze specific classes of images[9] and to study the generic structure of motion images[4]. We include a discussion of optical flow here as an illustration of the generality of the technique rather than as a novel application.

Optical flow can be caused by either scene motion or camera motion. Noting that the motion of a human being is more common than the motion of the scenes they would typically look at, then we might expect certain strong patterns in the optical flow fields. If the distribution of optical flow fields is concentrated enough, we can adopt them as a model of image variation for certain types of images.

3.1.1 Acquiring data for an optical flow model

To test these ideas, we recorded video footage of a static scene as we moved about the scene, panning, tilting, and rotating the camera as we moved. The scene was a typical office lounge scene, with the range of objects varying between about 3 feet and 30 feet from the camera. The camera was moved at a speed so that the video was easily interpretable by people at all points in the video, but otherwise the speed of movement of the camera was not controlled carefully.

After the video was taken, optical flow fields were generated between every two successive frames. A simple template matching optical flow algorithm was used. Since similar movements of the camera produce similar motion fields, by clustering the optical flow fields, we hoped to capture the dominant modes of variation in the image, and hence to automatically capture a useful parameterization for images of objects under camera movement.

A simple variant of the K-means clustering algorithm was used to cluster the optical flow fields. The results are shown in Figure 1. There are clusters clearly corresponding to horizontal translation, vertical translation, rotation, scaling, and

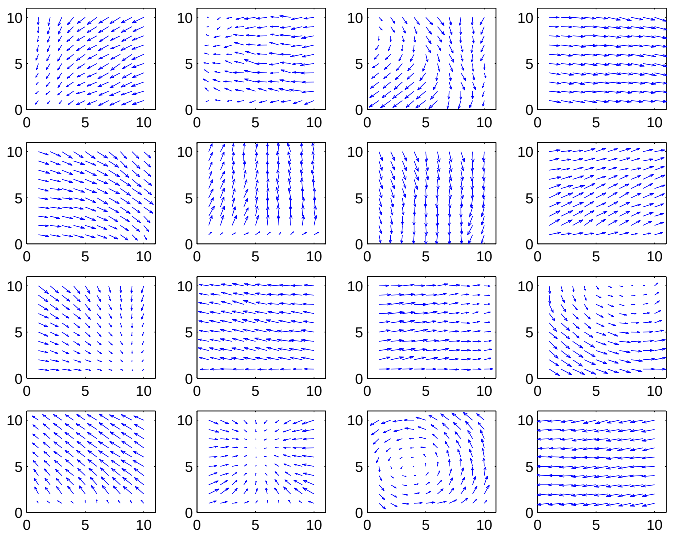

Figure 1: Means of flow field clusters derived from video sequences.

to other combinations of affine flows[3]. Using these flows as a basis, we can generate a manifold of images by applying small amounts of each of these transforms to a particular image. Such techniques have been used to augment data sets as in [6, 8].

Alternatively, such an analysis could be used to develop a model of optical flows under special conditions, such as for analyzing traffic data from a static camera, or in a vehicle-mounted camera, where the common optical flows would likely be very different. In the next section, we discuss a more novel application of the feature flow models.

4 Joint color change

In this section, we adopt as our feature $\mathbf{f}(p)$ the RGB color triple of a pixel. Our feature flow field is now a *color flow field*, the purpose of which is to describe a map from an image I_1 to another image I_2 by describing how each color changes from image to image. Thus, a color flow field is a vector field defined on the 3-D color space. Since it is learned from data, it will be independent of the parameterization of the color space (except for the interpolation procedure). We reported our statistical model of joint color change in previous work[7], but we review the key points here.

4.1 Structure in the joint color change space

Certainly an image feature appearing as one color, say blue, in one image could appear as almost any other color in another image due to a lighting change. Thus the *marginal distribution* of mappings for a particular color, over all possible *photic*[4]

[3]Similar results, albeit with fewer distinct affine flows, were obtained from a principal components analysis.

[4]By *photic*, we mean any parameter that affects the brightness or color of a pixel, such as lighting or gain control, but not parameters that affect the position in which an object appears (geometric parameters).

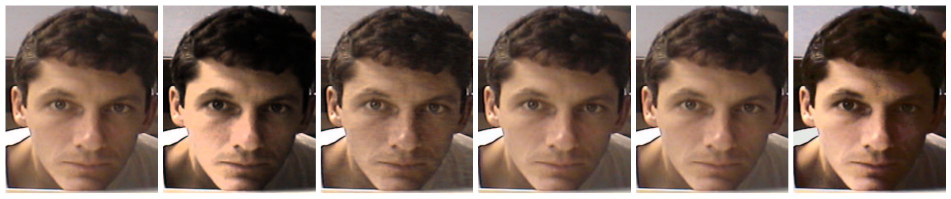

Figure 2: Image **b** is the result of applying a non-linear operator to the colors in image **a**. **c-f** are attempts to match **b** using **a** and four different algorithms. Our algorithm (image **f**) was the only one to capture the non-linearity.

parameter changes, is very broadly distributed. However, when color mappings are considered jointly, i.e. as color flows, we hypothesize that the space of possible mappings is much more compact. We test this hypothesis by statistically modeling the space of joint color maps, i.e. the space of color flows.

In learning color flows from real data, many common color flows can be anticipated. To name a few examples, flows which make most colors a little darker, lighter, or redder would certainly be expected. These types of flows can be well modeled with simple global linear operators acting on each color vector[1, 2, 5]. That is, we can define a 3x3 matrix $\mathbf{A}$ that maps a color $\mathbf{c_1}$ in the image I_1 to a color $\mathbf{c_2}$ in the image I_2 via

$$\mathbf{c_2} = \mathbf{A}\mathbf{c_1}. \tag{4}$$

Such linear maps work well for many types of common photic parameter changes. However, there are many effects which these simple maps cannot model. Perhaps the most significant is the combination of a large brightness change coupled with a non-linear gain-control adjustment or brightness re-normalization by the camera. Such photic changes will tend to leave the bright and dim parts of the image alone, while spreading the central colors of color space toward the margins. These types of changes cannot be captured well by the simple linear operator described above, but can be captured by modeling the space of color flows.

A pair of images exhibiting a non-linear color flow is shown in Figures 2**a** and **b**. Figure 2**a** shows the original image and **b** shows the same image after a non-linear color change (the contrast has been enhanced). Notice that the brighter areas of the original image get brighter and the darker portions get darker. This effect cannot be modeled using a scheme such as that given in Equation 4. The non-linear color flow allows us to recognize that images **a** and **b** may be of the same object, i.e. to "match" the images.

4.2 Color flow PCA

Our aim was to capture the structure in color flow space by observing real-world data in an unsupervised fashion. To do this, we gathered data by observing a large colored poster under standard office lighting conditions. It is important to note that a variety of non-linear normalization mechanisms built into the camera were operating during this process.

Our goal was to capture as many common lighting conditions as possible. We did not use unusual lighting conditions such as specially colored lights. Although a few images that were captured probably contained strong shadows, most of the captured images were shadow-free. Smooth lighting gradients across the poster were not explicitly avoided or created in our acquisition process. A total of 1646 raw images of the poster were obtained in this manner. We then chose a set of 800 image pairs $\mathcal{I}^j = (I_1^j, I_2^j), 1 \leq j \leq 800$, by randomly[5] and independently selecting individual images from the set of raw images. Each image pair was then used to estimate a full color flow $\Phi(\mathcal{I}^j)$ as described in Equation 3.

Note that since a color flow Φ can be represented as a collection of $3P$ coordinates, it can be thought of as a point in $\mathbb{R}^{3P}$. Here P is the number of distinct RGB colors at which we compute a flow vector, and each flow vector requires 3 coordinates: dr, dg, and db, to represent the change in each color component. In our experiments we used $P = 16^3 = 4096$ distinct RGB colors (equally spaced in RGB space), so a full color flow was represented by a vector of $3 * 4096 = 12288$ components.

Given a large number of color flows (or points in $\mathbb{R}^{3P}$), there are many possible choices for modeling their distribution. We chose to use Principal Components Analysis since 1) the flows are well represented (in the mean-squared-error sense) by a small number of principal components (see [7] for details), and 2) finding the optimal description of a difference image in terms of color flows was computationally efficient using this representation.

4.3 Synthesizing novel images

There are many potential applications of these color flow models described in [7], but perhaps the most fundamental is in generating synthetic images of a new object. Figure 3 shows the result of the application of various amounts of the first three principal color flows to a novel image. Recall that the color flow model was based on the observation of an entirely different image, and thus, we have effectively created a simple model of the manifold of the new object from a single example.

5 Mixtures of feature flow fields

Until now we have discussed modeling change over an image with a single flow field, be it an optical flow field or color flow field. However, it is clear that in many instances the change in a scene might be some combination of several flow fields. For example, if several objects are moving in a scene simultaneously, then a single *statistically common* optical flow field will not tend to describe the change in the scene. We must describe the scene as a combination or mixture of flows. This type of work has been done for optical flow fields by Jepson and Black, for example [3].

The same need for mixtures of flows occurs in modeling color changes in images. If a scene consists of a single flat object with distant lighting, then single color flows may be able to explain common lighting changes for that scene. However, for a scene with nearby lighting, a scene composed of many surfaces, or a scene with curved surfaces, we can only expect color flows to be locally consistent.

In [7], we show how multiple lighting changes in a single scene can be modeled by performing *patchwise* color flows between two images, enforcing consistent flows only over small regions (see figure 4). This type of approach allows explanation of complex lighting affects while maintaining a limited capacity for the model, since the model is still limited in the joint color mappings it can select.

[5]Non-adjacent pairs of images were used since most adjacent pairs of images exhibit extremely small color flows. While it is possible if we had enough data to capture many lighting changes by using adjacent pairs of images, by using distal pairs of images, we can study common lighting changes with a much small amount of data.

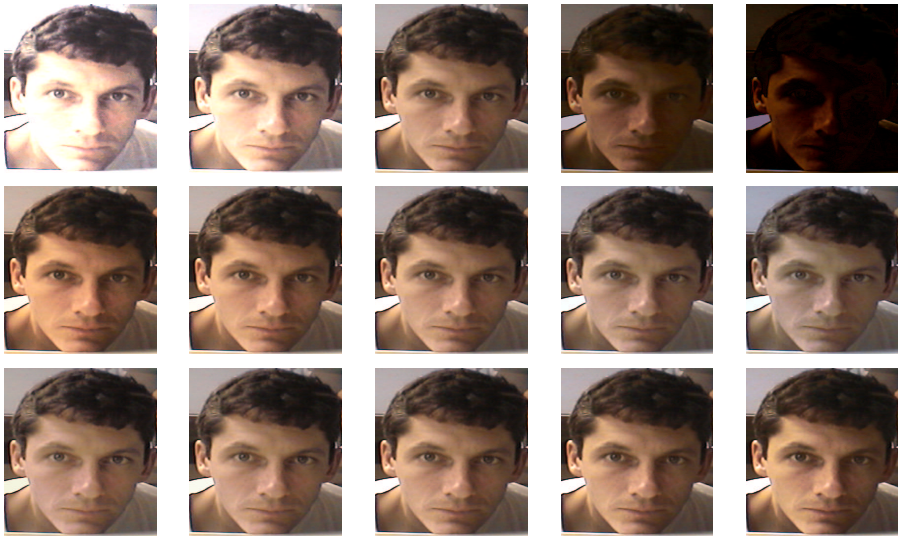

Figure 3: Effects of the first 3 principal color flows.

For curved surfaces, we can expect the color flow to change from point to point, but as long as the surface is smooth, then we would expect the color flows to change smoothly as well. Also, for flat surfaces, if a light source is nearby, then the angle of incidence at each point on the surface may vary. Thus, if the light changes position, the effect on the angle of incidence will be different at each point, and thus we could expect the color flow to be different at each point as well. However, we still expect smooth changes in the color flow coefficients. And we should still expect that two images of the same scene can be "flowed" to each other using locally constant flows, or at least slowly varying flows (linear or quadratic).

Sharp changes in the color flow coefficients would tend to be an indication of sharp changes in object gradient, or perhaps an occlusion or discontinuity, either in the lighting (i.e. a shadow) or due to the overlap of various surfaces from the point of view of the camera. Hence the *non-constancy* of flows, in principle, can actually tell us a lot about scene composition. We are currently investigating these ideas.

6 Conclusions

We have presented a framework in which the difference in continuously observable feature values across an image can be used to model the changes in those images. This method is applicable whenever feature values can be observed continuously. In particular, we are currently evaluating whether this technique can be applied to understanding common mappings in spectrograms to model allowable variations in speech and other auditory data.

References

[1] G. Buchsbaum. A spatial processor model for object color perception. *J. Franklin Inst.*, 310, 1980.

[2] D. A. Forsyth. A novel algorithm for color constancy. *Int. J. Comp. Vis.*, 5(1), 1990.

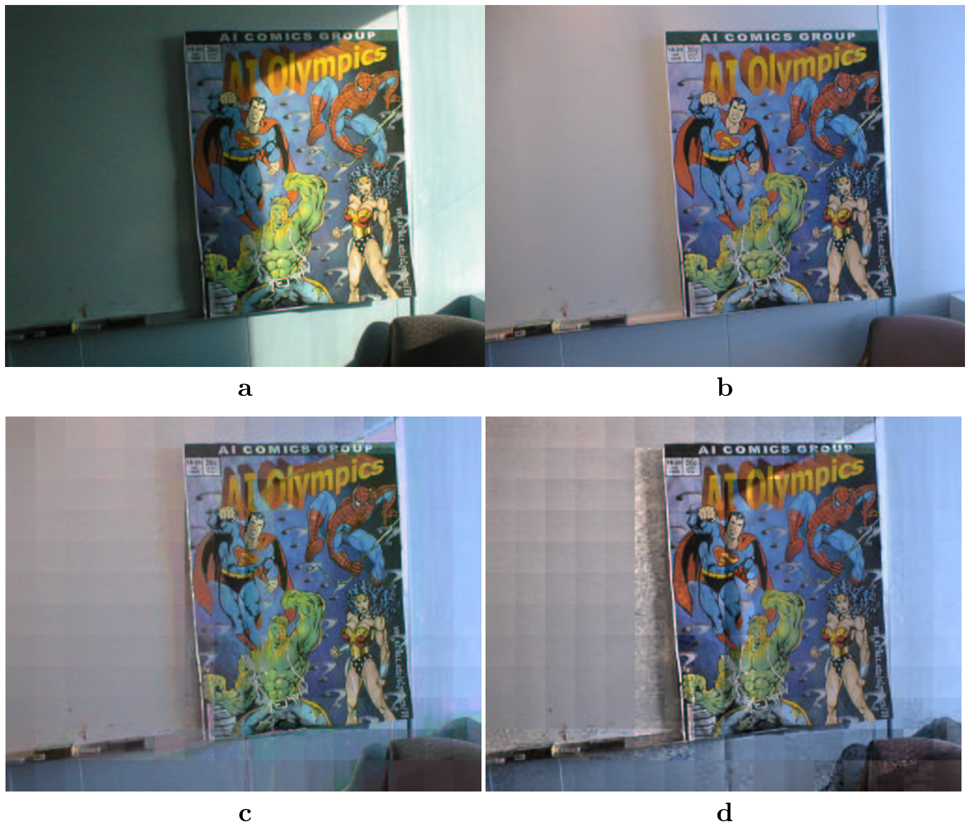

Figure 4: **a.** Image with strong shadow. **b.** The same image under more uniform lighting conditions. **c.** Color flow from **a** to **b** using *patchwise* eigenflows. By clustering flow coefficients, we hope to be able to define the shadow boundary. **d.** Flow from **a** to **b** using the patchwise version of a competitive color mapping model (see [7] for details). Notice the strong artifacts in the competitive color mapping scheme.

[3] Allan Jepson and Michael J. Black. Mixture models for optical flow. In *IEEE Comp. Vis. Patt. Recog. Conf.*, pages 760–761, 1993.

[4] Jenn-Jier James Lien. *Automatic Recognition of Facial Expressions Using Hidden Markov Models and Estimation of Expression Intensity.* PhD thesis, Carnegie Mellon University, 1998.

[5] J. J. McCann, J. A. Hall, and E. H. Land. Color mondrian experiments: The study of average spectral distributions. *J. Opt. Soc. Amer.*, A(67), 1977.

[6] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities on transforms. In *IEEE Comp. Vis. Patt. Recog. Conf.*, 2000.

[7] E. Miller and K. Tieu. Color eigenflows: Statistical modeling of joint color changes. In *Int. Conf. Comp. Vis.*, 2001.

[8] P. Simard, Y. LeCun, and J. Denker. Efficient pattern recognition using a new transformation distance. 5:51–58, 1993.

[9] T. Vetter, M. Jones, and T. Poggio. A bootstrapping algorithm for learning linear models of object classes. In *IEEE Comp. Vis. Patt. Recog. Conf.*, pages 40–46, 1997.