Non-Parametric and Light-Field Deformable Models

C. Mario Christoudias, Louis-Philippe Morency, and Trevor Darrell

Computer Science and Artificial Intelligence Laboratory

Massachusetts Institute of Technology

Cambridge, Massachusetts, USA

Abstract

Statistical shape-and-texture appearance models use image morphing to define a rich, compact representation of object appearance. They are useful in a variety of applications including object recognition, tracking and segmentation. These techniques, however, have been limited to objects with Lambertian surface reflectance, simple geometry and topology. In this work we present new shape-and-texture appearance models that overcome these limitations. In the first part of our work we develop a 4D shape-and-texture appearance model, built using light-fields. This model is capable of representing objects with complex surface reflectance and geometry. We demonstrate our light-field appearance model using 50 light-fields of the human head captured from a real-time camera array. Next, we present a non-parametric appearance model of the shape and texture of objects whose appearance manifolds exhibit a varying topology, e.g. have holes. We demonstrate this model using 2D mouth images of speaking people. In our experiments we evaluate the performance of each method and provide a comparison with conventional, linear single- and multi-view deformable models.

Key words: image morphing, principle components analysis, shape-and-texture

appearance model, deformable model, active appearance model, multidimensional morphable model, light-field, nearest-neighbor, non-Lambertian surface reflectance, nonlinear appearance manifold

1 Introduction

Computer vision techniques aim to infer salient semantic information from low-level, visual signals. Object recognition, detection and tracking are common visual tasks that people perform everyday to navigate and understand the world. An object's appearance is governed by many internal and external factors that can make the object difficult to recognize/detect. Object appearance models, such as [1–6], learn the complex appearance of an object from examples. These models define a knowledge-base of object appearance and can be used to recognize an object imaged under an arbitrary configuration or imaging condition.

Statistical shape-and-texture appearance models, also called deformable models, provide a powerful framework for learning object appearance [1,7]. These techniques exploit image morphing to define a rich, compact representation of object appearance. With these methods object appearance is decomposed into its shape and texture components. Intuitively the data variation in the decoupled shape and texture vector spaces is potentially simpler than that in the original image space. As a consequence fewer examples are needed to construct the model. Camera geometry or 3D structure can be used to further decouple and simplify the model [8].

In this paper we describe two complementary extensions to the shape-and-

texture appearance modeling framework. First, we define a deformable model over light-fields, to model pose and non-Lambertian effects. Second, we extend the deformable model to objects whose appearance manifolds are nonlinear, e.g. have holes. Both of these techniques have been discussed in previous publications [9,10]. In this work we place each technique under a unifying framework, offer a more detailed description of each technique, and provide additional results and discussion.

Linear methods such as [2,11] independently model the shape and texture vector spaces of an object with Principle Component Analysis (PCA). If object pose is also to be encoded in the model, these techniques may perform poorly, since the appearance variation of an object imaged under variable pose is in general nonlinear. Nonlinear methods can be used to define 2D appearance models that capture pose variation [3,12,13]. Pose variation, however, is easily represented in 3D, where object pose is kept as an external parameter to the model, as was done by Blanz and Vetter [8]. In their work, objects are represented using simply textured, detailed polygonal meshes. These meshes can be expensive to acquire and this approach has difficulty in representing complex lighting or objects whose surfaces exhibit a non-Lambertian surface reflectance.

The first contribution we present is a 4D shape-and-texture appearance model that can easily model objects with non-Lambertian surface reflectance and complex geometry. Using our approach, each prototype is imaged using a light-field and the view-based 2D shape of each object is computed (see Figure 1). We pursue a method to match a light-field deformable model to monocular images using a rendered Jacobian function. With this method, a full light-field of an object is recovered from a single 2D image which can then be used to

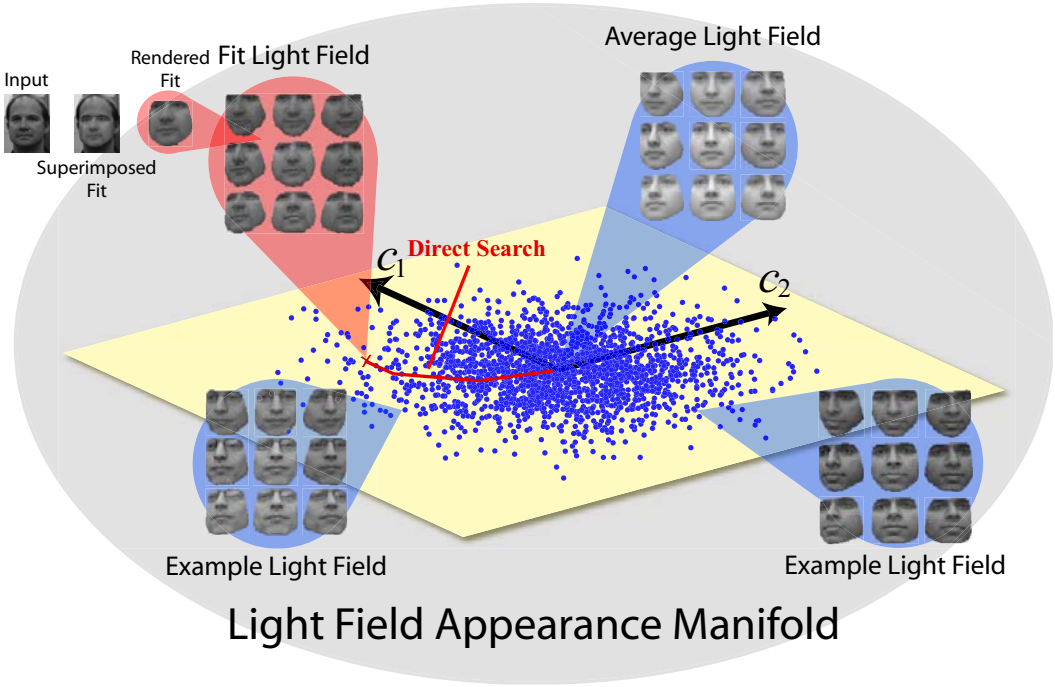


Fig. 1. Light-field appearance manifold of the human face. Provided an image of a previously unseen object with unknown pose, our algorithm matches the object to a point on the manifold that best approximates the object's appearance in the input view. A full light-field of the object is recovered.

render the object under previously unseen views.

There are many objects that, even in the absence of pose change, can exhibit nonlinear shape and texture variation, for which the conventional shape and texture mappings using PCA may poorly approximate the true space, using a light-field or monocular deformable model. This is especially true of biological objects that can deform quite drastically, such as a hand or mouth, or whose texture can drastically vary across different examples (e.g., cats, dogs). The appearance space of such objects has a varying topology; i.e., the object appearance manifold consists of multiple parts or holes. The shape and texture spaces of complex objects can also have varying dimensionality across different parts of the space. For example, an open mouth may have shape features associated with the teeth that are absent from a closed mouth.

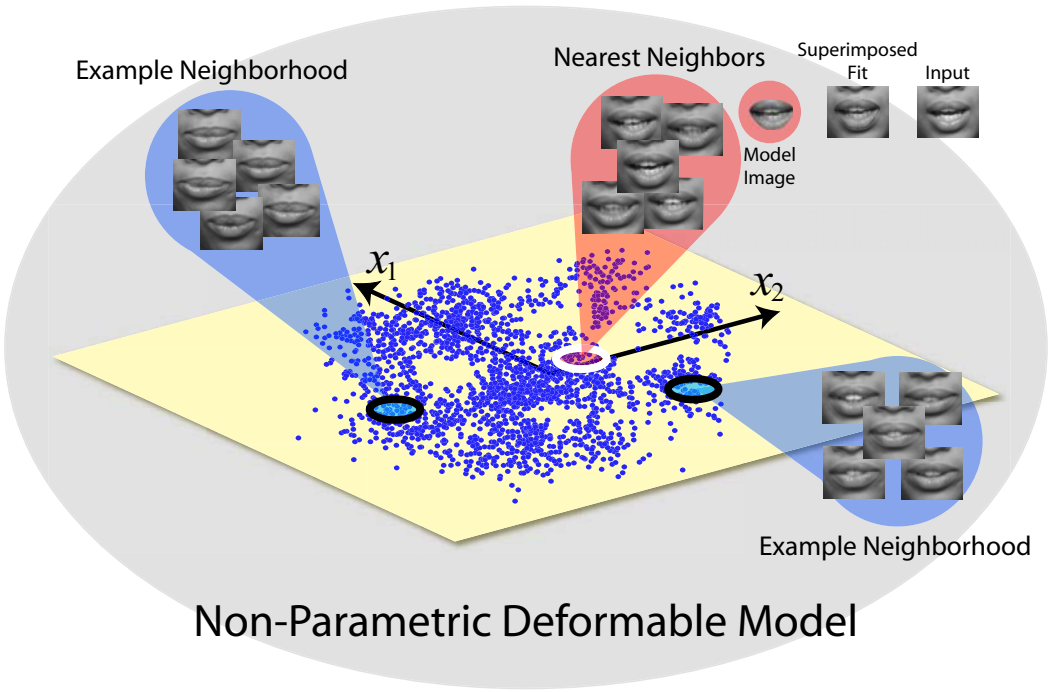


Fig. 2. Non-parametric deformable model of the mouth. The mouth appearance manifold has a varying topology, i.e. it has separate regions and holes, each region having the same parts of the mouth visible. The non-parametric deformable model analyzes a novel input by morphing between a local neighborhood of examples computed with nearest-neighbor.

The second contribution is a non-parametric, example-based technique for modeling shape-and-texture appearance manifolds with varying topology and dimensionality applicable to light-field or monocular deformable models (see Figure 2). With this method, we compute a morph between a neighborhood of examples on the manifold found with nearest-neighbor, using a convex (or bounded) combination of the neighborhood's shape and texture to match the input image. The non-parametric deformable model generalizes well to complex manifolds and, unlike a parametric method, it makes no assumptions about the global structure of the manifold.

We evaluate the above algorithms on two distinct data sets: a light-field head data set and a monocular mouth data set. In our first experiment, we construct a linear, light-field appearance manifold of the human head using a

light-field database of 50 subjects. Each light-field was captured online using a 6x8 camera array [14]. We show light-fields synthesized from 2D images of subjects outside of the database, captured with unknown pose. A comparison to a complementary approach, the view-based AAM [13], is also performed. Our second experiment demonstrates a shape-and-texture appearance manifold of the mouth represented with the non-parametric deformable model outlined above. For this experiment we use speaking person video sequences of five subjects from the AVTIMIT database [15]. We compare the non-parametric technique to a conventional linear shape-and-texture model and a Gaussian mixture deformable model and show that the non-parametric deformable model outperforms these methods.

In the following section we discuss related work. In Section 3 we provide a formal description of light-field appearance manifolds and describe our direct search matching algorithm. The non-parametric shape-and-texture model is outlined in Section 4. We outline our experimental setup and discuss results in Section 5. Finally, in Section 6 we give a concluding summary and remarks.

2 Related Work

Linear models of shape and texture have been widely applied to the modeling, tracking and recognition of objects [7,11,4]. Provided a set of example images, linear shape-and-texture appearance models decompose each image into a shape and texture representation and then model the variation of the data in these spaces using PCA. The shape of an object describes the object's geometry and is typically defined by a set of feature points that outline the object contours. The texture is the "shape free" representation of the object

and is obtained by warping each image to a reference coordinate frame that is usually defined by the average shape computed from the training images.

The Active Appearance Model (AAM) [2] and Multidimensional Morphable Model (MMM) [11] are probably the most well known linear shape-and-texture appearance models. By decomposing appearance into separate shape and texture spaces they achieve a compact, expressive model of appearance, more powerful than pure intensity models defined with PCA (e.g. Eigenfaces [6]). In these methods small amounts of pose change are typically modeled implicitly as part of shape variation on the linear manifold. For representing objects with large amounts of rotation, nonlinear models have been proposed separately for shape [12] and appearance [3]. An alternative approach to capturing pose variation is to use an explicit multi-view representation which builds a PCA model at several viewpoints. This approach has been used for pure intensity models [16] as well as shape and texture models [13]. A model of inter-view variation can be recovered using the approach of Cootes et. al [13], and missing views could be reconstructed. However, in this approach pose change is encoded as shape and intensity variation, in contrast to 3D approaches where pose is an external parameter. Additionally, views are relatively sparse, and individual features are not matched across views.

Deformable models with 3D shape features have the advantage that viewpoint change can be explicitly optimized while matching or rendering the model. Blanz and Vetter [8] showed how a morphable model could be created from 3D range scans of human heads. This approach represented objects as simply textured 3D shapes, and relied on high-resolution range scanners to construct a model; non-Lambertian and dynamic effects are difficult to capture using this framework. With some manual intervention, 3D models can be learned

directly from monocular video [17,18]; an automatic method for computing a 3D morphable model from video was described by Brand [19]. These methods all used textured polygonal mesh models for representing and rendering shape.

Multi-view 2D [13] and textured polygonal 3D [8,17,18] appearance models cannot model objects with complex surface reflectance. Image-based models have become popular in computer graphics recently and can capture these phenomena; with an image-based model, 3D object appearance is captured in a set of sampled views or ray bundles. Light-field [20] and lumigraph [21] rendering techniques create new images by resampling the set of stored rays that represent an object. Most recently the unstructured lumigraph was proposed by Buehler et. al. [22], and generalized the light-field/lumigraph representation to handle arbitrary camera placement and geometric proxies.

Recently, Gross et. al. [23] have proposed *eigen light-fields*, a PCA-based appearance model built using light-fields. They extend the approach of Turk and Pentland [6] to light-fields and define a robust pose-invariant face recognition algorithm using the resulting model. A method to morph two light-fields was presented in [24]; this algorithm extended the classic Beier and Neely algorithm [25] to work directly on the sampled light-field representation and to account for self-occlusion across views. Features were manually defined, and only a morph between two (synthetically rendered) light-fields was shown in their work.

In this paper we first develop the concept of a *light-field appearance manifold*, in which 3 or more light-fields are "vectorized" (in the sense of [1]) and placed in correspondence. We construct a light-field appearance manifold of facial appearance from real images, and show how that model can be automatically

matched to single static intensity images with non-Lambertian effects (e.g. glasses). Our model differs from the multi-view appearance models of Cootes et. al. [13,26] in that we build a 4D representation of appearance with light-fields. With our method, model coefficients between views are explicitly linked and we do not model any pose variation within the deformable model at a single view. We are therefore able to model self-occlusion due to pose change, and complex lighting effects better than a view-based AAM. We support this claim in our experimental results section. Our model is more similar to the Coupled-View AAM of Cootes et. al. Like our model, the Coupled-View AAM explicitly links the coefficients between views. This model however, has no formal mechanism for combining the discrete image samples to synthesize in-between object poses that is provided by the use of light-fields and light-field rendering [20,21].

As we also show in this paper, conventional shape-and-texture appearance models, such as the AAM and MMM, are unable to faithfully represent the appearance of complex objects with nonlinear appearance manifolds, such as mouths, whose manifolds may have parts or holes. Many nonlinear models have been defined separately for shape and appearance [12,27,3]. Romdhani et. al. [12] use Kernel PCA to define a nonlinear shape model for representing shape across object pose. Cootes et. al. [27] show how a Gaussian mixture model can be used to construct a nonlinear active shape model that restricts its search to valid shapes on the object shape manifold, thus avoiding erroneous matches. In their work, Cootes et. al. extend this idea to define a nonlinear model of shape and appearance called the view-based AAM [13]. The view-based AAM defines a piecewise linear representation of the shape-and-texture appearance manifold in a very similar fashion to the Gaussian mixture model described in

Section 5. The key differences between the Gaussian mixture model and the method described in [13] is that the Gaussian mixture deformable model of Section 5 automatically learns the different regions of the manifold from the data and is not restricted to learning mixture components that vary across pose alone.

A nearest neighbor algorithm is explored by Grauman et. al. [28]. In her work she defines an active shape model across body poses. Several authors have developed example-based models of object appearance, including the metric mixtures approach of Toyama and Blake [29], however, these methods do not exploit shape and texture decomposition. Similarly, Murase and Nayar [3] present a manifold learning algorithm that maps out the space of images of an object imaged across different poses. To the author's knowledge this is the first work that explores example-based techniques for modeling shape-and-texture appearance manifolds.

In the learning literature a number of manifold learning methods have been proposed, two of which are Isometric Feature Mapping (ISOMAP) [30] and Local Linear Embedding (LLE) [31]. ISOMAP uses geodesic distance, or distance along the manifold, to compute the coordinates of each data point on the manifold, while LLE uses local geometry to compute these coordinates—it finds a mapping such that the local geometry of points in the high-dimensional input space is preserved on the manifold. Both ISOMAP and LLE are non-parametric manifold learning methods that function over k-size neighborhoods in the original high-dimensional input space. Our non-parametric deformable model technique can be used to improve the mappings found with these techniques when used to compute appearance manifolds, i.e., when the input space is over images. In specific, image morphing can be used to get a better ap-

proximation of the local geometry or distances used by both of these methods to perform manifold learning. We believe this to be an interesting application of our model and leave this for future work.

3 Light-Field Appearance Manifolds

In this section we discuss how to build an appearance model that represents object appearance in 4D using light-fields. With our model, each point on the shape-and-texture appearance manifold maps to a light-field of an object (see Figure 1). Pose is kept as an external parameter to the model and the resulting appearance manifold for simple objects, such as the human head, is well approximated using a linear model. Light-fields are purely image-based and do not use any scene geometry to model the appearance of an object. Unlike the view-based 2D models of [13] and the 3D models of [8], our model easily represents object classes with complex surfaces and geometry. One of the main ideas behind this section is that pose variation is easily handled in 4D with light-fields, and thus more efficient models of appearance can be derived than with 2D approaches. In the next section we discuss nonlinear techniques for modeling the shape-and-texture appearance manifold of complex objects. For simplicity, we present a linear model in this section, however, the nonlinear techniques of Section 4 are directly applicable.

In the following sub-sections we define the concepts of shape and texture in the context of light-fields and show how to build a generative model of appearance over these vector spaces. To match the model, we extend the direct search algorithm of [2] to function over the space of light-fields. We show how to match a light-field or 2D image of an object to a point on the manifold. When

matching to an image, we automatically estimate its pose by searching over the views of the model light-field. In turn, the model fit can be used to synthesize the object under unseen views. In Section 3.1 we provide formal definitions of light-field shape and texture in the context of both geometric and optical flow based shape features. We then describe the light-field appearance manifold in Section 3.2. The direct search algorithm employed to match the model is explained in Section 3.3.

3.1 Light-Field Shape and Texture

In this section we provide a formal description of the shape and texture of a set of light-field prototypes that define the appearance manifold of an object class. Let $L(u, v, s, t)$ be a light-field consisting of a set of sample views of the scene, parameterized by view indices (u, v) and scene radiance indices (s, t), and let $L_1, ..., L_n$ be a set of prototype light-fields with shape $X'_1, ..., X'_n$. Below we define light-field shape both in the context of 2D feature-points and 4D vector fields. We first discuss the feature point based shape representation and then the vector field based shape representation that can be computed automatically with optical flow.

For most image-based rendering techniques, X'_i is a set of 3D feature points which outline the shape of the imaged object. With a light-field, no 3D shape information is needed to render a novel view of the object. It is therefore sufficient to represent the shape of each light-field as the set of 2D feature points, which are the projections of the 3D features into each view. More

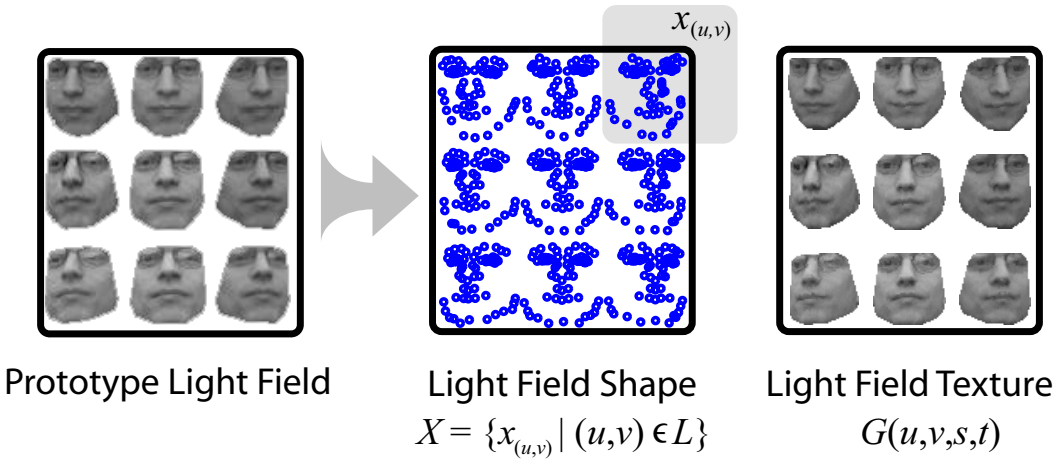


Fig. 3. Example shape and texture of a prototype light-field. Light-field shape is defined by the set of view-based 2D shapes, $x_{(u,v)}$. The texture of a prototype light-field is its "shape-free" equivalent obtained by warping the input light-field to the reference shape.

formally, we define the shape, X, of a light-field L as

$$X = \{x_{(u,v)} \big| (u,v) \in L\} \tag{1}$$

where $x_{(u,v)}$ is the shape in a view (u,v) of L. If the camera array is strongly calibrated, it is sufficient to find correspondences in two views and re-project to the remaining views. With only weak calibration and the assumption of a densely sampled array, feature points may be specified in selected views of the light-field and tracked into all other views using optical flow [32]. An example shape feature vector is displayed for a prototype light-field of the human head in Figure 3.

Once shape is defined for each prototype light-field, to increase model efficiency Procrustes analysis [33] is performed to place the shape of each object into a common coordinate frame. Effectively, Procrustes analysis applies a rigid body transformation to the shape of each light-field such that each object is aligned to the same approximate 3D pose. From the set of normalized shapes

X_i of each prototype, the reference shape X_{ref} is computed as

$$X_{ref} = \mathbf{M}_\alpha \bar{X} \tag{2}$$

where $\bar{X}$ is the mean aligned shape and $\mathbf{M}_\alpha$ is a matrix which scales and translates the mean shape such that it is expressed in pixel coordinates (i.e. with respect to the height and width of the discrete views of the light-field). The matrix $\mathbf{M}_\alpha$ constrains the shape in each view of the reference light-field to be within the height and width of the view.

As in [1], the texture of a prototype light-field is its "shape free" equivalent. It is found by warping each light-field to the reference shape X_{ref}. As will be shown in the next section, this allows for the definition of a texture vector space that is decoupled from shape variation. Specifically, the texture of a light-field L is defined as

$$G'(u, v, s, t) = L(D(u, v, s, t)) = L \circ D(u, v, s, t) \tag{3}$$

where D is the mapping,

$$D : \mathcal{R}^4 \longrightarrow \mathcal{R}^4 \tag{4}$$

that specifies for each ray in L_{ref} a corresponding ray in the prototype light-field L and is computed using the shape of L and X_{ref}. Note Equation (3) implements the light-field warping operation [24]. As in the 2D deformable models of [2], the texture of each prototype, G'_i, is normalized to be under the same global illumination. This results in normalized light-field texture vectors G_i.

Above we presented a feature-point based light-field shape representation that

Algorithm 1 Compute Average Light-Field

Let $L_1, ..., L_n$ be a set of prototype light-fields.
Select an arbitrary light-field L_i as the reference light-field L_{ref}
repeat
  for all L_i **do**
    Compute correspondence fields X_i between L_{ref} and L_i using optical flow.
    Backwards warp each view of L_i onto L_{ref} using X_i.
  end for
  Compute the average over all X_i and G_i.
  Forward warp each view of $T_{average}$ using $X_{average}$ to create $L_{average}$.
  Convergence test: is $L_{average} - L_{ref} < limit$?
  Copy $L_{average}$ to L_{ref}
until convergence

is acquired using a semi-automatic process. The shape of a light-field can also be computed using optical flow. With this technique, the shape of a light-field is defined directly as the 4D deformation field which places the light-field in correspondence with the model reference light-field:

$$X_i = D_i(u, v, s, t), \tag{5}$$

where D_i is defined by the mapping (4) and specifies for each ray in the reference light-field L_{ref} a corresponding ray in the prototype light-field L. The shape X_i of each prototype light-field, defined using Equation (5), is computed by applying optical flow between the views of each prototype light-field and that of the reference light-field. As in the MMM [11] the reference object is chosen to be the average object, since by definition its difference in shape and texture is minimal between each of the light-field prototypes and therefore it is the preferred reference light-field. Using optical flow, the average light-field is computed via the bootstrapping algorithm outlined in [34]. This algorithm is presented as Algorithm 1. For efficiency we applied the algorithm independently to each view of the prototype set.

Using definition (5), light-field texture is computed as,

$$G_i(u, v, s, t) = L \circ X_i(u, v, s, t). \tag{6}$$

We will use the above definitions of light-field shape and texture to define a light-field appearance manifold in the following section.

3.2 Light-Field Appearance Manifolds

As illustrated in the previous section, once a reference is defined, each prototype light-field may be described in terms of its shape and texture. The combination of shape and texture form an appearance manifold: given a set of light-fields of the same object class, the combination of their texture warped by a combination of their shape describes a new object whose shape and texture are well approximated by that of the prototype light-fields. Compact and efficient linear models of shape and texture variation may be obtained using PCA, as shown in [2], [11]; or a nonlinear method such as the method described in Section 4 can be used. For the remainder of this section we use a linear PCA model.

Given the set of prototype light-fields $L_1, ..., L_n$, each having shape X_i and texture G_i, PCA is applied independently to the shape and texture vectors to give

$$\begin{aligned} X &= \bar{X} + \mathbf{P}_s \mathbf{b}_s \\ G &= \bar{G} + \mathbf{P}_g \mathbf{b}_g \end{aligned} \tag{7}$$

Using Equation (7), the shape and texture of each model light-field is described

by its corresponding shape and texture parameters $\mathbf{b}_s$ and $\mathbf{b}_g$. As there may exist a correlation between shape and texture, a more compact model of shape and texture variation is obtained by performing PCA on the concatenated shape and texture parameter vectors of each prototype light-field. This results in a combined shape-and-texture PCA space:

$$\begin{aligned} X &= \bar{X} + \mathbf{Q}_s \mathbf{c} \\ G &= \bar{G} + \mathbf{Q}_g \mathbf{c} \end{aligned} \tag{8}$$

where as in [2],

$$\begin{aligned} \mathbf{Q}_s &= \mathbf{P}_s \mathbf{W}_s^{-1} \mathbf{P}_{cs} \\ \mathbf{Q}_g &= \mathbf{P}_g \mathbf{P}_{cg} \end{aligned} \tag{9}$$

and $\mathbf{W}_s$ is a matrix that commensurates the variation in shape and texture when performing the combined shape-and-texture PCA. In our experiments we use $\mathbf{W}_s = r\mathbf{I}$ where $r = \sqrt{\sigma_s^2/\sigma_g^2}$. Here σ_s^2 and σ_g^2 represent the total variance of the normalized shape and texture vectors.

Equation (8) maps each model light-field to a vector $\mathbf{c}$ in the combined shape-and-texture PCA space. To generalize the model to allow for arbitrary 3D pose and global illumination, Equation (8) may be re-defined as follows,

$$\begin{aligned} X_m &= S_t(\bar{X} + \mathbf{Q}_s \mathbf{c}) \\ G_m &= T_u(\bar{G} + \mathbf{Q}_g \mathbf{c}) \end{aligned} \tag{10}$$

where S_t is a function that applies a rigid body transformation to the model shape according to a pose parameter vector $\mathbf{t}$, T_u is a function which scales

and shifts the model texture using an illumination parameter vector $\mathbf{u}$, and the parameter vectors $\mathbf{t}$ and $\mathbf{u}$ are as defined in [2]. Note, the reference light-field has parameters $\mathbf{c} = 0$, $\mathbf{t} = \alpha$ and $\mathbf{u} = 0$, where α is a pose vector that is equivalent to the matrix $\mathbf{M}_\alpha$ in Equation (2).

The set of all light-fields of an object define a manifold of object appearance in light-field space. With our model, the light-field appearance manifold of an object class is modeled as,

$$L_{model} = G_m \circ D_m \tag{11}$$

where L_{model} is a model light-field that maps to a point on the appearance manifold and, as in Equation 4, D_m is a 4D deformation field which maps each ray in the model light-field to a ray in the reference light-field. Using feature-point based shape, D_m is computed using the shape of the model light-field, X_m and the reference light-field, X_{ref}. When X_m is defined using shape derived from optical flow, we set $D_m = X_m$ and we re-define Equation (11) as

$$L_{model} = G_m \circ_f X_m \tag{12}$$

where $\circ_f$ denotes the forward warping operation.

In the remaining section we present a direct search algorithm that optimizes the model over the combined shape-texture space and describe how the light-field appearance manifold can be automatically optimized over images with unknown pose.

3.3 Model Matching

In this section, we show how to generalize the matching technique of [2] to light-fields. We first illustrate how to match a light-field and then discuss the more interesting task of fitting a model light-field to a single 2D image.

A novel light-field, L_s, is matched to a point $\tilde{\mathbf{c}}$ on the shape-and-texture appearance manifold by minimizing the following nonlinear objective function:

$$E(\mathbf{p}) = |G_m - G_s|^2 \tag{13}$$

where $\mathbf{p}^T = (\mathbf{c}^T|\mathbf{t}^T|\mathbf{u}^T)$ are the parameters of the model, G_m is the model texture and G_s is the normalized texture of L_s assuming it has shape X_m. G_s is computed by warping L_s from X_m to the reference shape X_{ref}. The model shape and texture are computed at $\mathbf{p}$ using Equation (10). For simplicity, we assume that the object imaged by L_s has the same approximate 3D pose as the training light-fields.

The direct search gradient descent algorithm of [2] is easily extendible to a light-field deformable model. In [2] a linear relationship for the change in image intensity with respect to the change in model parameters was derived via a first order Taylor expansion of the residual function $\mathbf{r}(\mathbf{p}) = G_m - G_s = \delta\mathbf{g}$. In particular, given a point $\mathbf{p}$ on the manifold, the parameter gradient that minimizes the objective function (13) was computed as, $\delta\mathbf{p} = -\mathbf{R}\delta\mathbf{g}$, where the matrix $\mathbf{R}$ is the pseudo-inverse of the Jacobian, $\mathbf{J} = \frac{\partial \mathbf{r}}{\partial \mathbf{p}}$, derived from the Taylor expansion of the residual function.

In a 2D deformable model the columns of the Jacobian are intensity gradient

images which model how image intensity changes with respect to each model parameter. Analogously, the Jacobian of a light-field deformable model represents the change in light-field intensity with respect to the change in model parameters, each of its columns representing light-field intensity gradients that describe the intensity change across all the views of a light-field. As in a 2D AAM, the Jacobian is learned via numerical differentiation.

A more interesting extension of the AAM framework arises when performing direct search to match a light-field deformable model to a single 2D image; with a light-field the Jacobian matrix is rendered based on pose. A novel image I_s is matched to a point on the light-field appearance manifold by minimizing the objective,

$$E(\mathbf{p}, \epsilon) = |F(G_m, \epsilon) - g_s|^2 \tag{14}$$

where ϵ is the camera pose of I_s, F is a function that renders the pose ϵ of the model texture [20,22] and g_s is the texture of I_s assuming it has shape x_m. g_s is computed by warping I_s from x_m to the reference shape x_{ref}. Both 2D shapes are obtained by rendering X_m and X_{ref} into view ϵ using,

$$x = F_x(X, \epsilon) \tag{15}$$

where F_x is a variant of the light-field rendering function F: it renders shape in view ϵ via a linear interpolation of the 2D shape features defined in each view of X.

Overall, the objective function in Equation (14) compares the novel 2D image to the corresponding view in L_{model}. Minimizing this objective function fits a model light-field, L_{model}, that best approximates I in view ϵ. An efficient way to optimize Equation (14) is by defining a two step iteration process, in which

the pose ϵ is optimized independently of the model parameters $\mathbf{p}$. We estimate ϵ using the pose estimation algorithm described below. The pose parameter $\mathbf{t}$ can be used to further refine this pose estimate during matching.

Once ϵ is approximated, direct search may be employed to match I to a point on the shape-and-texture appearance manifold. As previously discussed, each column of the Jacobian, $\mathbf{J}$ of a light-field deformable model is a light-field intensity gradient. To approximate the intensity gradient in view ϵ of the target image I, light-field rendering is applied to each column of $\mathbf{J}$. This yields a "rendered" Jacobian matrix, $\mathbf{J}_\epsilon$, specified as,

$$\mathbf{J}_\epsilon^i = F(\mathbf{J}^i, \epsilon),\ i = 1, ..., m \tag{16}$$

where $\mathbf{J}^i$ represents column i of the matrix $\mathbf{J}$ and m is the number of columns in $\mathbf{J}$. Note similar to the model and image textures of Equation (13) the columns of $\mathbf{J}_\epsilon$ have shape x_{ref} defined above.

Using $\mathbf{J}_\epsilon$, we optimize Equation (14) using the direct search algorithm presented in Algorithm 2. An important step of the algorithm, is in the application of the pose parameter vector $\mathbf{t}$. In this step, the global affine warp S_t is applied to the rendered model image and not to the model light-field (step 3 of the *Residual* function in Algorithm 2). This is because rotating, scaling, and/or translating the images of L_{model} according to S_t may violate the light-field construction when matching to an image. To see this, consider manipulating a single-slab light-field. Applying S_t to this light-field effectively rotates or displaces the focal plane (st-plane) of the light slab (note, scaling the images correlates to widening the gap between the camera and focal planes of the light slab). Clearly, moving the focal plane of the light-field will alter where

the scene rays will intersect it. If the imaged object is planar then the scene rays will follow the motion of the focal plane. For non-planar objects this is not necessarily the case, however.

Algorithm 2 Direct Search Algorithm for Matching an Image

Let I_s be an input image with estimated pose ϵ, X_m, G_m the model shape and texture vectors of the light-field appearance manifold, and $\mathbf{J}$ the model Jacobian.

Compute $\mathbf{J}_\epsilon$ using Equation (16).
Set $\mathbf{p} = \mathbf{p}_0$
Evaluate $\delta\mathbf{g} = Residual(I_s, \mathbf{p}, \epsilon)$
repeat
  Compute error $E_0 = |\delta\mathbf{g}|^2$
  Evaluate $\delta\mathbf{p} = -\mathbf{R}\delta\mathbf{g}$
  Update parameters, $\mathbf{p}_1 = \mathbf{p} + \delta\mathbf{p}$
  Evaluate $\delta\mathbf{g} = Residual(I_s, \mathbf{p}_1, \epsilon)$
  Compute error at new $\mathbf{p}$ value: $E = |\delta\mathbf{g}|^2$
  if $|E - E_0| \geq 0$ **then**
    Try different step sizes: $k = 1.5, 0.5, 0.25, ...$
    Set $\mathbf{p}_1 = \mathbf{p} + k\delta\mathbf{p}$
    Evaluate $\delta\mathbf{g} = Residual(I_s, \mathbf{p}_1, \epsilon)$
    Compute error $E = |\delta\mathbf{g}|^2$
  end if
  if $|E - E_0| < 0$ **then**
    Set $\mathbf{p} = \mathbf{p}_1$
  end if
until $|E - E_0| \geq 0$

function $\delta\mathbf{g} = Residual(I_s, \mathbf{p}, \epsilon)$
$x_{ref} = F_x(X_{ref}, \epsilon)$
$x_s = F_x(X_m(\mathbf{b}_s, t_0), \epsilon)$
$x_s = S_t(x_s)$
$\mathbf{g}_s = Whiten(I_s \circ W(x_s, x_{ref}))$
$\mathbf{g}_m = Whiten(F(G_m(\mathbf{b}_g), \epsilon))$
$\delta\mathbf{g} = \mathbf{g}_m - \mathbf{g}_s$

function $\mathbf{g}_w = Whiten(\mathbf{g})$
$\mathbf{g}_w = \mathbf{g} - mean(\mathbf{g})$
$\mathbf{g}_w = \mathbf{g}_w / var(\mathbf{g}_w)$

By applying the affine warp on the rendered model image the model light-field remains in the coordinate frame of the reference light-field, while still affording

the model affine flexibility in the coordinate frame of the input image. Another benefit of the above matching algorithm is that it avoids the need to optimize over z, the depth of the focal plane of the unstructured lumigraph, during matching. The scaling performed by S_t when applied to the model light-field effectively changes this value and thus z would need to be optimized over as well when performing the match. By keeping the model light-field in the coordinate frame of the reference light-field, this need is eliminated and we let $z = z_0$, the depth of the average light-field.

Note, when fitting the model to an object light-field, we can safely apply S_t to the images of the model light-field. This is because the set of allowable affine transformations is constrained by the 3D pose of the input light-field. Matching an image is more ambiguous, and can result in transformations S_t that when applied to the images of the light-field violate its construction.

To estimate ϵ during matching, we first use gradient descent on the views of the light-field to obtain a coarse estimate of the object's pose. Provided an input image I_s, we obtain an initial estimate of the object's pose, ϵ_0, by performing cross-correlation between the image and each view of the average light-field. We then match the image to this view and each of its eight-connected neighbors. We move to the neighbor with smallest fit error and iterate until the central view has the smallest fitting error of its neighbors. To avoid local minima we randomly perturb the fit upon convergence. Final convergence is declared when the algorithm converges to the same discrete pose twice.

Once convergence is declared at a discrete pose of the model light-field, gradient descent can once again be applied to obtain a refined pose estimate. In our implementation, we efficiently estimate the object's pose, ϵ, by fitting a

quadratic to the fit error of the eight-connected neighborhood centered about the computed discrete pose. The pose, ϵ, is set to the minimum of the fit quadratic.

4 Nonlinear Appearance Manifolds

In this section we present a nonlinear technique for modeling shape-and-texture appearance manifolds. Here we focus on 2D appearance models, however, this technique is directly applicable to the 4D appearance model of the previous section.

The images of a complex object such as a mouth generally belong to a nonlinear appearance manifold with parts or holes as demonstrated by Figure 4. This figure illustrates the shape and texture of example mouth images taken from the AVTIMIT database [15]. The average image and shape are displayed along with example textures and shapes of selected prototype images.

Consider modeling the mouth appearance using a linear model such as an AAM. Figure 4 demonstrates the difficulty with modeling the mouth using a linear method. In particular, notice the stretched region in the texture of a closed mouth and that the inside of the mouth is lost in the texture of an open mouth. These artifacts cripple the computed model; in general, linear methods have difficulty modeling the full range of mouth appearance. Such artifacts are a result of the varying topology of the appearance manifold of this object—some features (or surfaces) are visible in certain mouth images but not in others (e.g., teeth). Intuitively, this is seen by the fact that there exist sets of mouth configurations for which the same parts of the mouth are

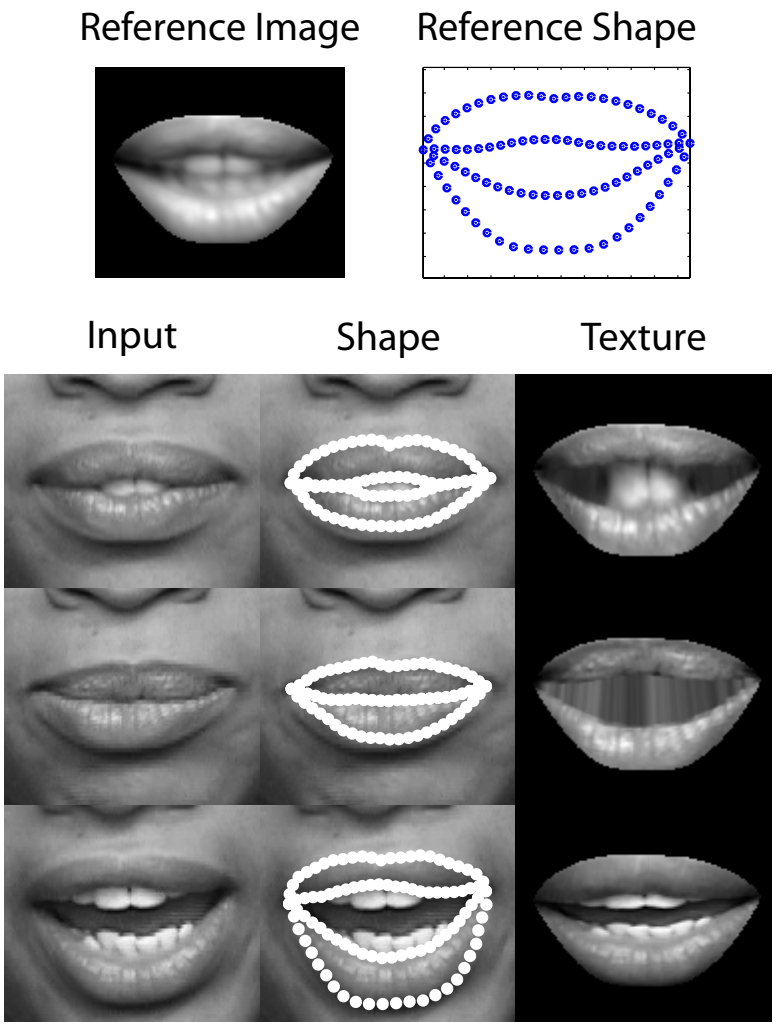


Fig. 4. Linear models compute a texture space by warping each example to a single reference frame. Note the stretched region present in the closed mouth textures and that the inside of the mouth is lost in the texture of the open mouth.

visible in each set.

In addition to varying topology, the shape-and-texture spaces of nonrigid object classes can have varying dimensionality across examples when different sets of landmarks are used. Once again, consider the mouth images of Figure 4. The presence of teeth in the open mouth introduces new shape features that are absent from the image of the closed mouth. Allowing for varying shape dimensionality results in a more expressive and accurate model of appearance.

Below we present a non-parametric deformable model for modeling shape-

Fig. 5. First five nearest neighbors computed with our algorithm on a database of 100 mouth images.

and-texture appearance manifolds. Unlike a parametric approach, this model assumes nothing of the global structure of the appearance manifold. Instead, it looks at local neighborhoods on the manifold that are assumed to belong to the same region of the topology computed with nearest-neighbor. As we show below, the nearest-neighbor model is easily extendible for use with shapes of variable dimensionality.

4.1 Non-Parametric Deformable Model

The nearest-neighbor model provides an implicit representation of the manifold. Specifically, this model focuses on local neighborhoods of the manifold defined by k examples. In this region it is assumed that the same parts of

the nonrigid object are visible. Given the local neighborhood, the shape and texture of a new input is found by taking bounded combinations of the shape and texture of the k neighborhood examples. Therefore, given a new image, we wish to find a local neighborhood observing the above properties, whose shape and texture best explain the input.

We use nearest-neighbor search to find a set of examples on the manifold whose appearance most closely approximate that of the input. Given a novel input, $\mathbf{x}_s$, we compare it to each image, $\mathbf{x}_i$, of the prototype set to compute its k nearest neighbors. Although we use an exhaustive search there exist fast methods for computing approximate nearest neighbors [35] that we leave for future work. In our algorithm, we compute proximity using Euclidean distance in pixel space. We compute the distance,

$$d(\mathbf{x}_s, \mathbf{x}) = \|\mathbf{x}_s - \mathbf{x}\|^2, \tag{17}$$

between $\mathbf{x}_s$ and each prototype image and retain the k examples having smallest distance. Figure 5 displays the results of this nearest-neighbor algorithm on a database of 100 images of a single subject's mouth taken from the AV-TIMIT database. The nearest neighbors of a novel input appear to form a local neighborhood in image space.

The shape and texture of an input image are computed by taking a convex combination of the shape and texture of its k nearest neighbors. Let $\mathbf{x}_j$ and $\mathbf{s}_j$, $j = 1, ..., k$ be the k nearest neighbors of the input and their associated shapes. The texture of each example is computed as

$$\mathbf{t}_j = \mathbf{x}_j \circ W(\mathbf{s}_j, \mathbf{s}_{ref}),\ j = 1, ..., k, \tag{18}$$

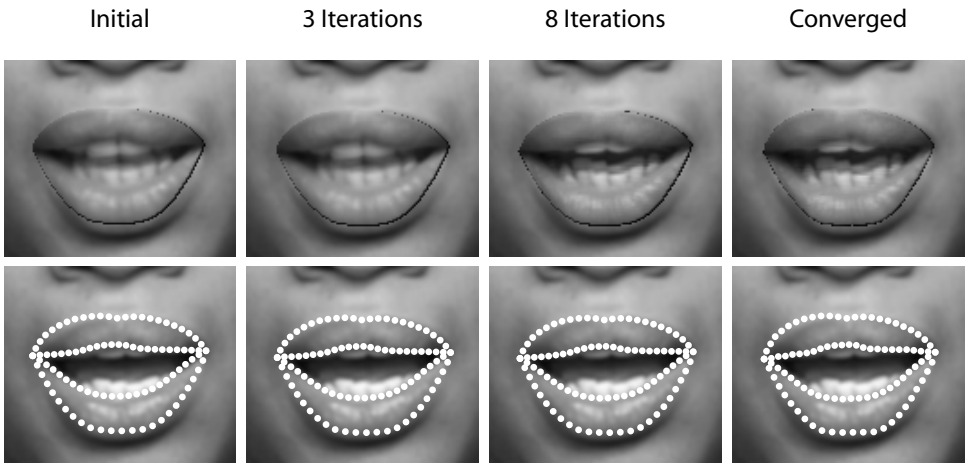


Fig. 6. A convincing reconstruction of the shape and texture of an input mouth image is computed in a few iterations using the gradient descent algorithm of the nearest neighbor model.

where $\circ$ denotes the warping function, $W()$ is a function that computes the piecewise affine correspondence between two images given their shape [2], and $\mathbf{s}_{ref}$ is the reference shape of the local neighborhood defined to be the mean of the example shapes,

$$\mathbf{s}_{ref} = \frac{1}{k}\sum_{j}\mathbf{s}_j. \tag{19}$$

Given the k nearest neighbors of the input, we search over bounded combinations of their shape and texture that best match the input by minimizing the following error objective function,

$$E(\mathbf{x}_s, \mathbf{b}, \mathbf{c}, \mathbf{t}) = \|\mathbf{x}_s \circ W(\mathbf{s}_m(\mathbf{c}, \mathbf{t}), \mathbf{s}_{\text{ref}}) - \mathbf{t}_m(\mathbf{b})\|^2, \tag{20}$$

where

$$\mathbf{t}_m(\mathbf{b}) = \textstyle\sum_j b_j \mathbf{t}_j,$$

$$\mathbf{s}_m(\mathbf{c}, \mathbf{t}) = S_t(\textstyle\sum_j c_j \mathbf{s}_j),$$

S_t is a function that applies a rigid body transformation to the model shape according to a pose parameter vector $\mathbf{t}$ and b_j, c_j take values on the closed interval $[\alpha, \beta]$. Note that α and β restrict the search to a bounded region of

Fig. 7. Variable size shape representation used by the nearest-neighbor model. Each example image is labeled with varying feature sets according to what parts of the mouth are visible. Three examples are shown: (left) with only lip features, (middle) with lip and top teeth features, (right) with lip, top and bottom teeth features.

the manifold containing the k nearest-neighbor examples. If $\alpha = 0$ and $\beta = 1$ then the search is restricted to the convex hull of the example shape-and-texture vectors. This restriction results in a compact representation of the manifold and assures that we match an input to a point on the manifold.

We minimize the objective function (20) using gradient descent. Figure 6 displays an example match using the above algorithm. The algorithm is able to generate a convincing reconstruction of the mouth from the shape and texture of the nearest-neighbor examples.

It is straightforward to extend the nearest-neighbor model to handle varying shape dimensionality. With this representation a shape vector, $\mathbf{s}^M$, is defined as

$$\mathbf{s}^M = \langle x_1, x_2, ..., x_M, y_1, y_2, ..., y_M \rangle \,. \tag{21}$$

In the above representation, each shape has dimensionality $2M$. This shape representation is illustrated by Figure 7. In the nearest-neighbor model we associate each prototype image with a shape vector that has dimensionality according to what is visible in the image. When computing nearest neighbors, we intersect the shapes of the neighborhood examples and use the shape features common to all examples to match the novel input. To compute the shape of the input, we keep the shape features that appear in a majority of

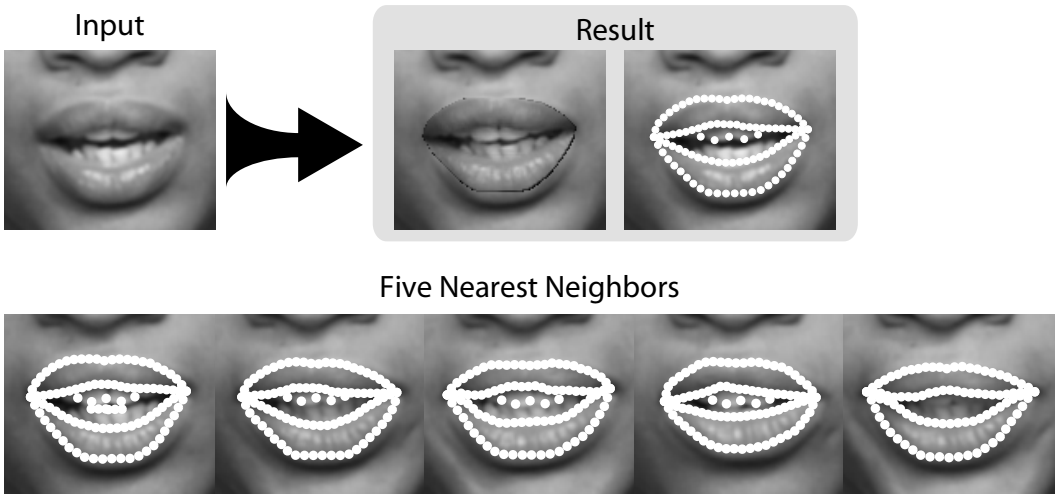


Fig. 8. Shape intersection algorithm used by the nearest-neighbor model. To compute the shape of the input, the shape of the nearest neighbors is intersected and the shape features that appear in a majority of the examples are used.

its nearest-neighbor examples. This process is illustrated by Figure 8. The use of multiple shape dimensionality results in a more expressive and accurate appearance model.

5 Experiments

In this section we evaluate the light-field deformable model and the non-parametric deformable model of this paper. We begin by describing our implementation and experimental setup. We then demonstrate a light-field head appearance manifold of the human head. In this experiment we compare the view-based AAM to our method and display full light-fields synthesized from 2D images of novel subjects with unknown pose. Next, we demonstrate the non-parametric shape-and-texture appearance model using speaking mouth video sequences of five subjects taken from the AVTIMIT database. We perform a qualitative and quantitative evaluation of the non-parametric model and compare it against a baseline linear method and a parametric deformable

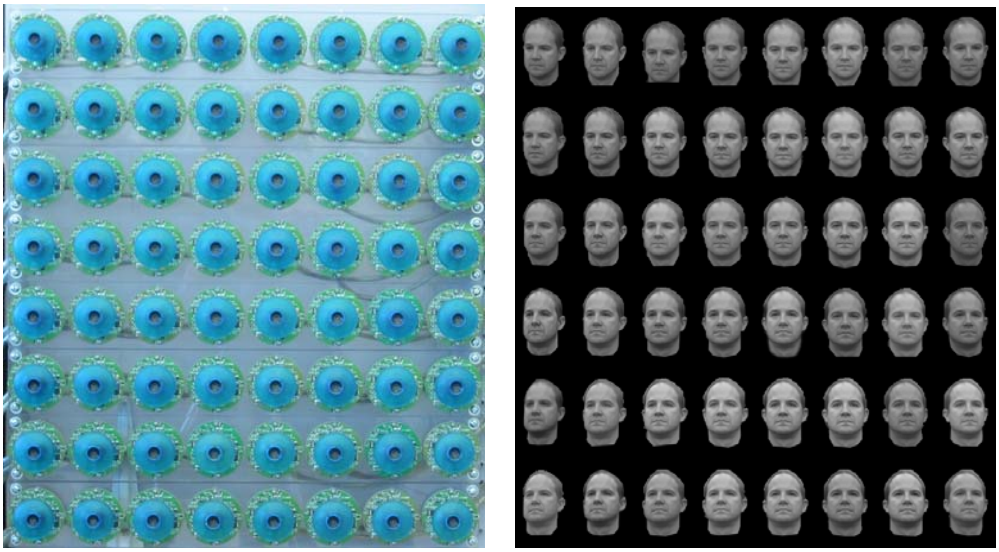
model.

5.1 Experimental Setup

We built a light-field appearance manifold of the human head by capturing light-fields of 50 subjects using a real-time light-field camera array [14]. We collected 48 views (6×8) of each individual and manually segmented the head from each light-field (Figure 9). Our head database, displayed in Figure 10, consists of 37 males and 13 females of various races. Of these people, 7 are bearded and 17 are wearing glasses. The images in each view of the prototype light-fields have resolution 320 x 240. Within each image, the head spans a region of approximately 80 x 120 pixels. The field of view captured by the camera array is approximately 25 degrees horizontally and 20 degrees vertically. To perform feature tracking, as described in Section 3, we used a multi-resolution Lukas-Kanade optical flow algorithm [**?**], with 4 pyramid levels and Laplacian smoothing [1]. When matching our model to an image we assume that the object's image location is approximately known. In the case of a head model, such information can be readily obtained from a face detector [36].

For comparison, we built a view-based AAM using the views of the light-field camera array [13]. In both the definition of the view-based and light-field active appearance models the parameter perturbations displayed in Table 1 were used to numerically compute the Jacobian matrix. To avoid over-fitting to noise, shape-and-texture PCA vectors having low variance were discarded from each model, the remaining PCA vectors modeling 90% of the total model

[1] We acknowledge Tony Ezzat for the Lukas-Kanade optical flow implementation.



(a) (b)

Fig. 9. (a) Light-field camera array. [14] (b) example 6×8 head light-field taken from our light-field head database.

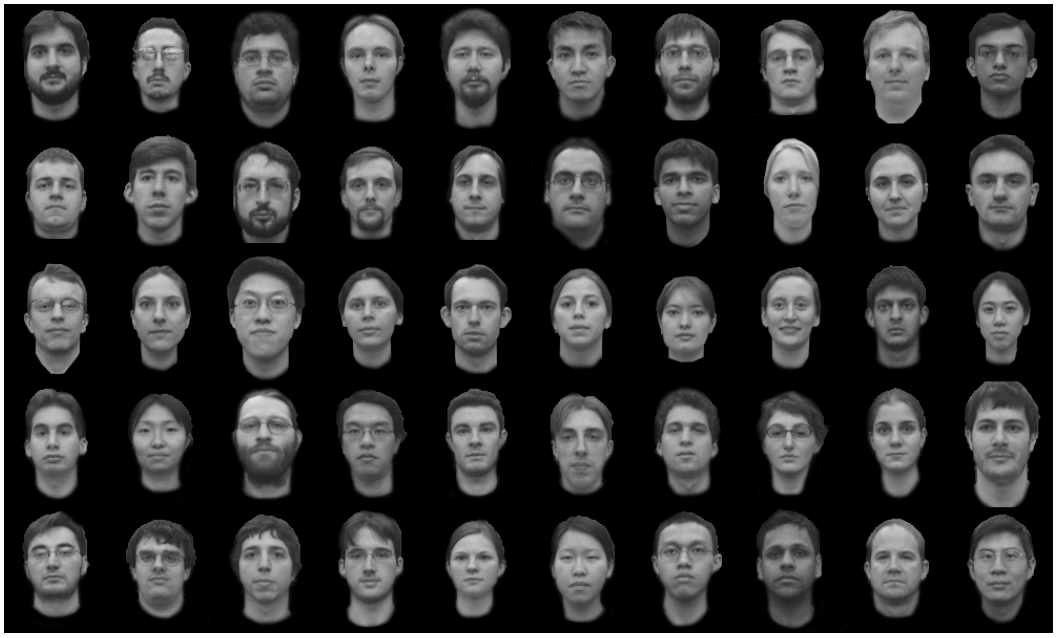


Fig. 10. Head database of 50 subjects. A single, frontal view of the 6×8 light-field of each subject is displayed.

variance.

We implemented the view-based AAM and light-field appearance models in MATLAB. To perform light-field rendering we use the unstructured lumigraph

algorithm described in [22]. This algorithm has two parameters: l for the number of source views used to render the scene and z_0 the approximate depth of the focal plane of the light-field. In our experiments we used a value of $l = 4$ when optimizing both the feature based and the optical flow based models. As discussed in Section 3 the model light-fields are kept in the coordinate frame of the reference light-field upon matching, thus we need only note the approximate depth of the reference light-field to optimize the model. In our experiments we found values of $11 \leq z_0 \leq 12$ to work well for the approximate depth of the reference light-field of both the optical flow and feature based models. Our matching algorithm typically converged between 4 and 15 iterations when matching to an image and between 4 and 10 iterations when matching to a light-field. Each iteration took a few seconds in un-optimized MATLAB. We believe that using a real-time light-field renderer [22] would result in matching times similar to those reported for a 2D AAM [37].

To evaluate the non-parametric algorithm of Section 4, we used mouth sequences of five subjects taken from the AVTIMIT database [15] (see Figure 11). The sequence of each subject consisted of 8 different utterances and contained on the order of 900 frames. For each subject we randomly hand selected about 100 frames from their first three utterances and manually labeled them with lip shape features [2] . Using the labeled features, we cropped the images of each subject to only contain the mouth. Using this training set we constructed an Active Appearance Model [2] for each subject. The training images of each subject were also labeled with teeth features (see Figure 7) to form shape vectors with variable dimension. These shapes were used by the nearest-neighbor

[2] Subject one's database contained 122 examples. The databases of subjects 2 through 5 had 100 examples.

Variables	Perturbations
x, y	$\pm 5\%$ and $\pm 10\%$ of the height and width of the reference shape
θ	± 5, ± 15 degrees
scale	$\pm 5\%$, $\pm 15\%$
c_{1-k}	± 0.25, ± 0.5 standard deviations

Table 1
Perturbation scheme used to train our linear models. [37]

model discussed in Section 4.

For comparison, we also built a Gaussian mixture deformable model for each subject using a three dimensional subspace of the training image data computed with PCA, retaining 56 % of the total model variance, and with $m = 5$ mixture components. We found these parameters to work well in our experiments. Using a three dimensional subspace also allowed us to visualize our models. To compute the Gaussian mixture, we used the NetLab library [38]. With this model, the local shape-and-texture variation at each mixture component is modeled using a linear deformable model. In particular, at each component, we compute an AAM using the examples that lie under the support of the component's Gaussian. We consider an example to be under the support of a Gaussian if it is less than three standard deviations away from the mean. To analyze a new example image with this model, we independently fit each local AAM to the example and retain the fit with the lowest error.

The local AAMs constructed in the Gaussian mixture model and the single AAM models were constructed using the parameters listed in Table 1. In each local AAM, as well as the single AAM, 95 % of the model variance was retained by the combined shape-and-texture space. In our experiments we evaluated the nearest-neighbor algorithm for varying values of k. The value used is made explicit in each experiment. We set $\alpha = 0$ and $\beta = 1$, restricting the solution

Fig. 11. Video sequences of five subjects taken from the AVTIMIT database [15] used to train and test our models. A frame from each sequence is shown.

to the convex hull of the shape and texture of the computed neighborhood examples. We also restricted t such that the model image is rotated at most ± 10 degrees, its size is scaled between [0.75,1.5] and it is translated by at most ± 10 pixels in the horizontal and vertical directions.

The test set of each subject was formed by taking 500 images from the subject's video sequence that are outside the training set. In our experiments, we assume that the location of the mouth is coarsely initialized by an external mouth detector. Both the Gaussian mixture model and the AAM optimize for location during model search and therefore require only approximate initialization of the mouth location. We refine the mouth location estimate in the nearest-neighbor model by finding the nearest neighbor using the input location and then computing a normalized cross correlation between the nearest neighbor and same-sized patches in the input image centered about locations in an 11×11 search window about the initial center. We reset the center of the mouth to the location having the highest correlation score and repeat this process until convergence or the maximum number of iterations is reached. In our experiments, we found this algorithm typically converged in a few iterations.

5.2 Light-Field Head Manifold Results

5.2.1 Comparison to View-Based Linear Model

To compare our method to a view-based AAM we built a single-view 2D AAM and compared it against a feature-based light-field deformable model. Each model was constructed in color using all fifty subjects, and was matched to two people in the training set, imaged from an unknown pose. The resulting fits are displayed in Figure 12. In the figure, the first subject without glasses is modeled equally well using both methods. The second subject is wearing glasses which self-occlude the subject in extreme views of the camera array. These self-occlusions are difficult to model using a view-based AAM, where inter-pose variation is modeled as shape. Note that the view-dependent texturing effects in the person's glasses are preserved by the light-field deformable model, but are lost by the view-based AAM even though the person remains in the model.

The difference in performance between each model is explained by how they model pose variation. The view-based AAM blends the shape and texture of multiple poses at a given local-linear model. Thus, one would expect that inter-pose self-occlusion and view-dependent texture would not be properly modeled using this technique, unless many such local linear models are introduced, rendering the model inefficient. The light-field deformable model represents appearance in 4D, thus the shape and texture of each pose are kept separate and pose is an external parameter of the model. As a result the light-field deformable model can easily handle the view-dependent texture and self-occlusions introduced by the glasses whereas the view-based AAM

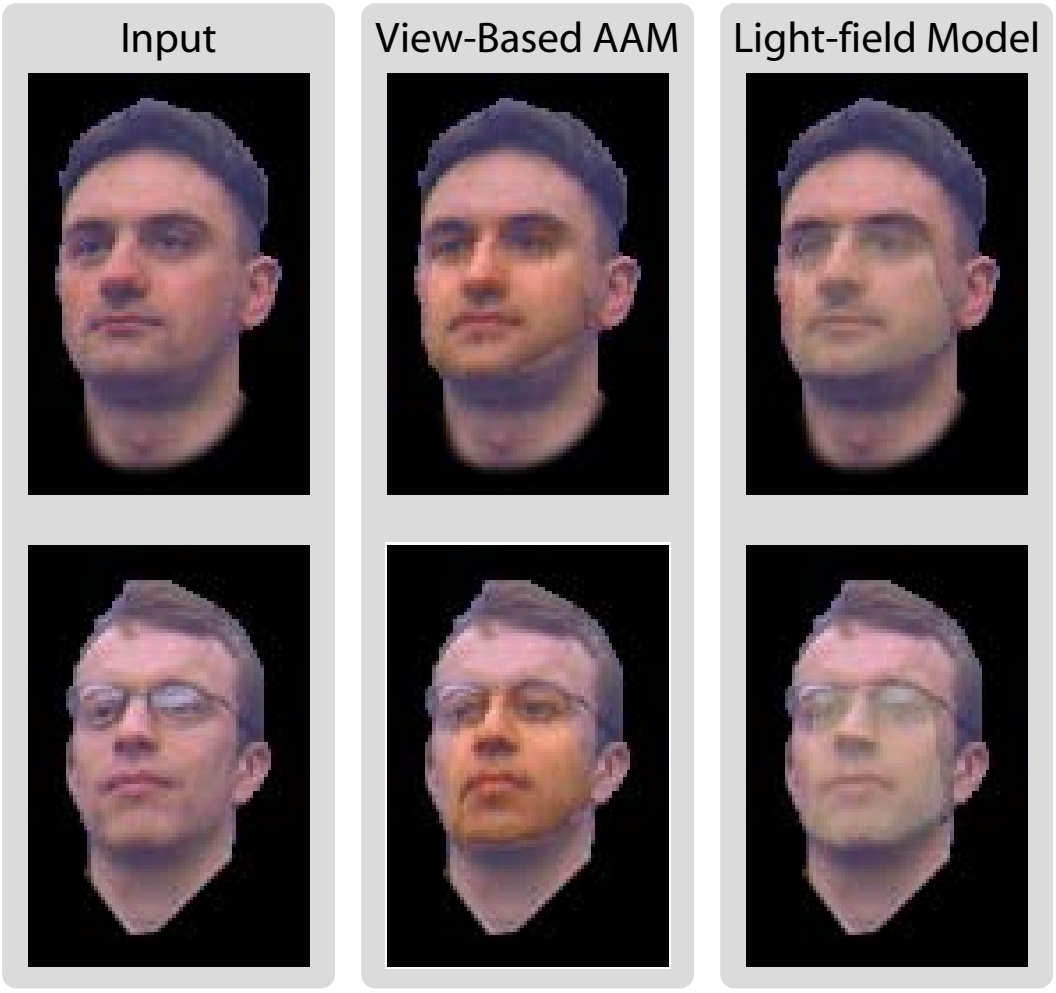


Fig. 12. Comparison of a light-field deformable model to a view-based AAM. The left column shows the input, the middle column the best fit with a 2D AAM, and the right column the light-field fit. The 2D and light-field appearance models both exhibit qualitatively good fits when the surface is approximately smooth and Lambertian. Unlike the light-field deformable model, the 2D model is unable to model the specularity in the glasses of the second subject.

cannot.

Note that the blurring in the light-field result is caused by low image resolution (we are using 80x120 image regions) and calibration error of the light-field dataset. We believe that the collection of a better dataset would amplify the difference between the view-based AAM and light-field model and we leave this as future work. What is important to note is that the white specular regions in the glasses are preserved by our model and are lost by the view-based AAM

even though the subject is in the training dataset.

5.2.2 Light-field Model Synthesis

To demonstrate the light-field synthesis capabilities of a light-field deformable model, we match the model to a single 2D image or light-field of novel subjects using "leave-one-out" experimentation. Figure 13 illustrates light-field synthesis for two people taken out of the model. To conserve space, only selected views of each light-field are displayed. Both fits are shown superimposed onto the corresponding input light-field. Each light-field is also provided for ground truth comparison. As seen from the figure, the input light-fields are well matched and a convincing reconstruction of each person is generated. Specifically, the shape and texture of both individuals is well captured across views.

Figure 14 illustrates our model's ability to generate convincing light-field reconstructions from 2D images. This figure provides two example matches to 2D images with unknown pose. For each match, the person was removed from the model and imaged at a randomly selected pose not present at a discrete view of the light-field model. The synthesized light-field, rendered at the selected pose of each person, is displayed below each input image. The synthesized light-fields are also displayed. Note our method built a light-field with 48 views from a single 2D image.

Figures 13 and 14 display results for both the feature-point based and optical flow based shape features. Comparing the results of these figures one finds that each model performs quite similarly: the synthesized light-fields resulting from each model are approximately the same. Such performance is expected since

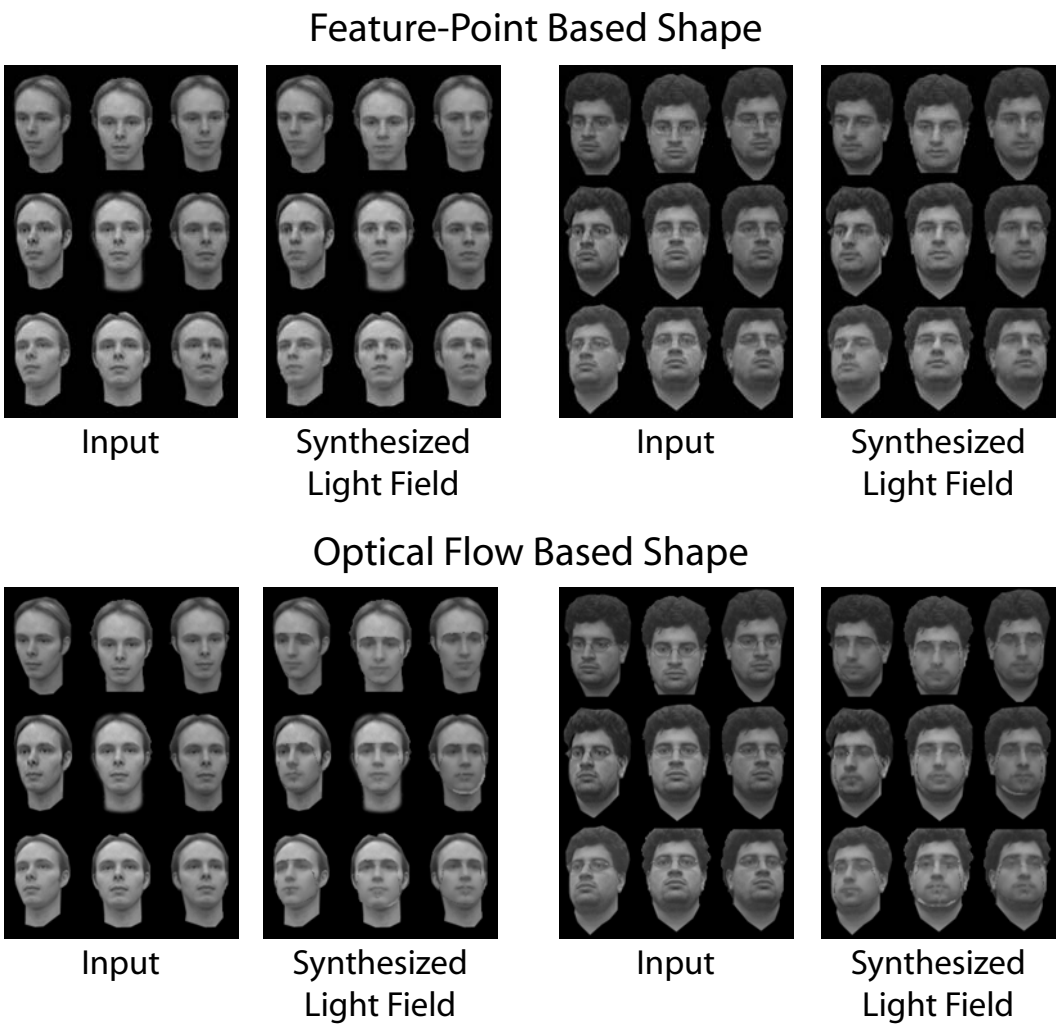


Fig. 13. Feature-point based and optical flow based light-field deformable model optimized over light-fields of two subjects outside of the model database.

each model is trained on the same training set and each model is designed with the same framework using PCA. Close inspection of each figure shows that there are some minor differences between the fit of each algorithm, due to the use of different shape features. For example, the optical flow based model has difficulty about the edges of the face due to ambiguity in the optical flow, however, as illustrated by the figures these errors are minor.

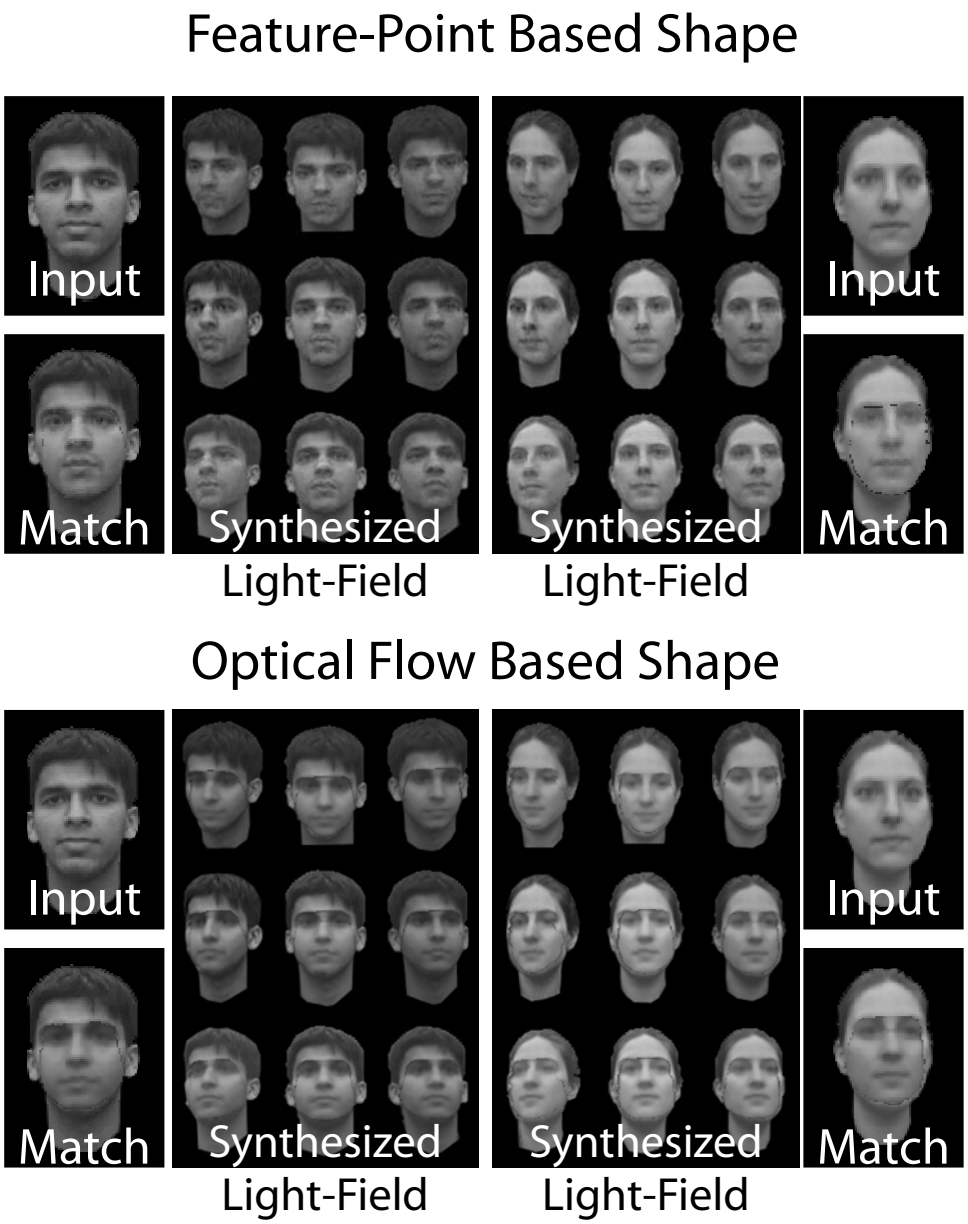


Fig. 14. Feature-point based and optical flow based light-field deformable models optimized over images of objects with unknown pose. The models were optimized over 2 subjects removed from the model database. Our method is able to synthesize convincing light-fields from a single input image.

5.3 Non-Parametric Mouth Manifold Results

A qualitative comparison of the non-parametric deformable model with the AAM and the Gaussian mixture deformable model is provided in Figure 15. In the figure, three images from the 500-image test set of the first four subjects

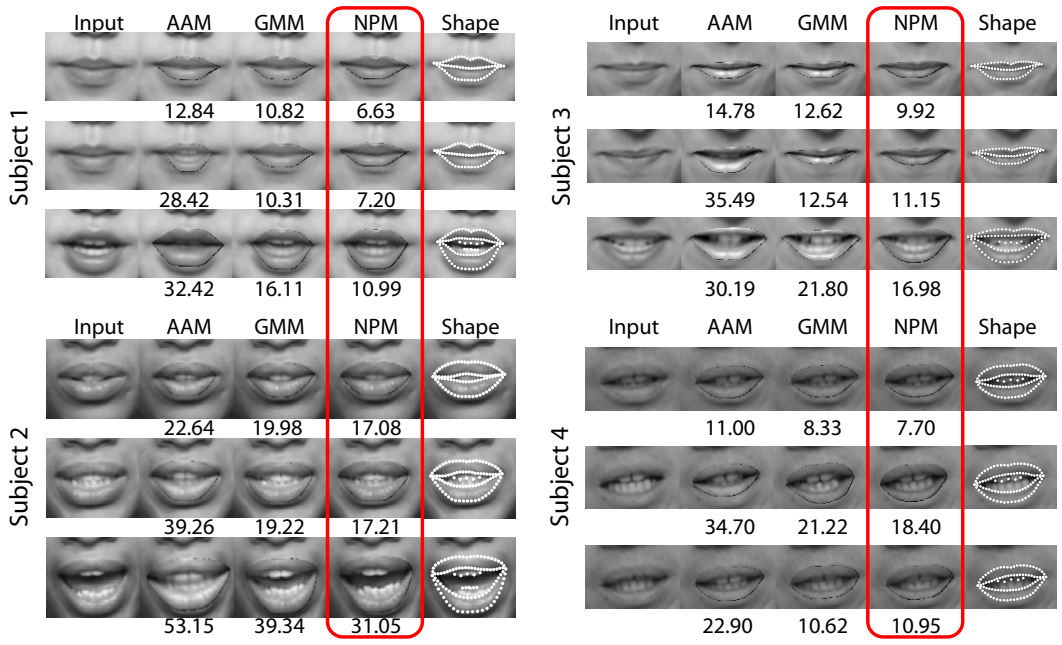


Fig. 15. Qualitative comparison between each nonlinear method and a baseline linear model. The input and synthesized image computed with each model is shown for each example. To conserve space, only the synthesized shape from the nearest-neighbor model is displayed. The AAM has difficulty modeling the full range of mouth appearance. The last two examples of each subject illustrate scenarios where the nonlinear methods outperform the AAM.

are displayed along with the synthesized images generated by each model. To conserve space, only the synthesized shape from the nearest-neighbor model is displayed; the main difference in performance is noted in the synthesized image of each model. The RMS fit error is also provided below each fit. In this experiment, the nearest-neighbor models have $k = 10$. For each subject, the first test image is modeled well using all three models. Comparing the RMS error of each fit, however, the single AAM does the worst and the nearest-neighbor deformable models perform the best. The next two examples of each subject demonstrate scenarios where the nonlinear methods outperform the AAM. This is especially seen in the examples of the first three subjects. The fourth subject under-articulates when he speaks and therefore his mouth appearance varies less than the first three subjects and the difference in performance between the three methods is less drastic on this subject.

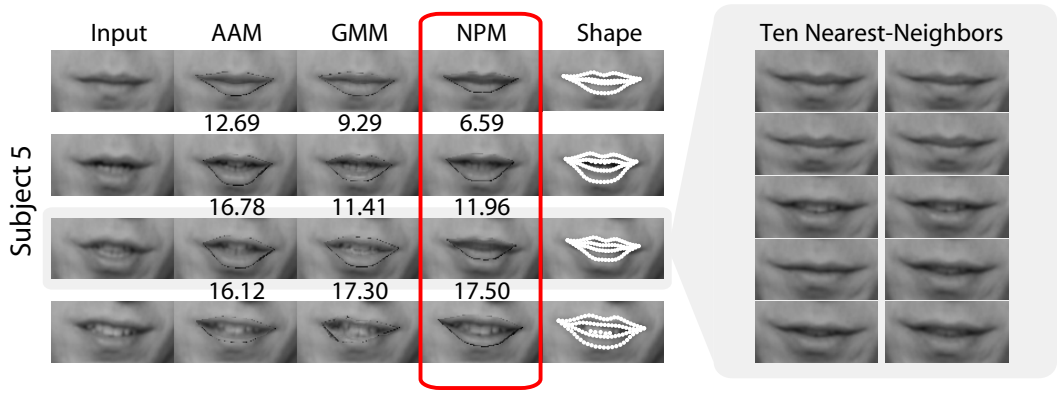


Fig. 16. Qualitative comparison of each method on the fifth subject: (right) input test image displayed along with synthesized image generated from each model and synthesized shape from nearest-neighbor model and (left) first ten nearest-neighbors computed for the third example. The results for the first and second input image display cases where the AAM performs similarly and worse than the nonlinear methods respectively. The last two examples show cases where the nonlinear techniques perform poorly. Note, that unlike the nearest-neighbor model, the Gaussian mixture model can converge to non-mouth images much like the linear AAM.

All three methods had difficulty modeling the fifth subject. Selected test images of this subject are displayed in a similar fashion to Figure 15 in Figure 16. The first test image is an example of where each method performs similarly and the second an example where the nonlinear techniques perform better than the linear AAM. The last two examples are instances of where the nearest-neighbor and Gaussian deformable models perform poorly. Note the AAM also had difficulty modeling these images.

The poor performance of the nearest-neighbor deformable model on the third example of Figure 16 is attributed to error in the nearest-neighbor computation: we currently compute nearest-neighbor using a naive, intensity-based distance metric that is sensitive to differences in rotation, translation and scale. The 10 nearest-neighbors computed for the third test frame are displayed in Figure 16. The nearest-neighbors do not match the input well. In contrast to the other subjects, this subject displayed a fair amount of head motion when he spoke. It is possible that the error in the nearest-neighbor computation is

attributed to the difference in rotation between the input image and training images, although it may also be because the input image is not well represented by the training set. The former case may be corrected by using more intelligent distance metrics which we discuss as part of future work.

Note that, while the nearest-neighbor deformable model degraded gracefully in the presence of error, the Gaussian mixture deformable model failed much in the same way a linear AAM would in the fourth example of Figure 16. This makes sense since it is possible that one or more of the fit mixture components of the Gaussian mixture span portions of the space with holes and the model can converge to non-mouth images. This error in the model may be due to poor model selection, but can also be because the mouth appearance manifold is poorly represented as piecewise linear. The nearest-neighbor deformable model avoids the need to perform model selection and generalizes better to complex manifolds.

A quantitative comparison of each model is provided by Figure 17. In the figure, a Root-Mean-Square error box plot is shown for each approach computed over the 2500-image test set, combined from all five subjects. This result is also summarized by Table 2 which gives the average RMS error rate along with the standard deviation away from the mean for each technique. A pairwise comparison of the error distributions of each technique using statistical t-tests gave p-values of $p<<0.01$ for each distribution pair. Both the Gaussian mixture model and the nearest-neighbor do the same or significantly better than the single AAM throughout the test sequence. The error box plot shows that with $k = 20$ the nearest-neighbor algorithm outperforms each approach on a whole (different values of k are considered next). The noteworthy performance of the nearest-neighbor model is expected since it makes the fewest

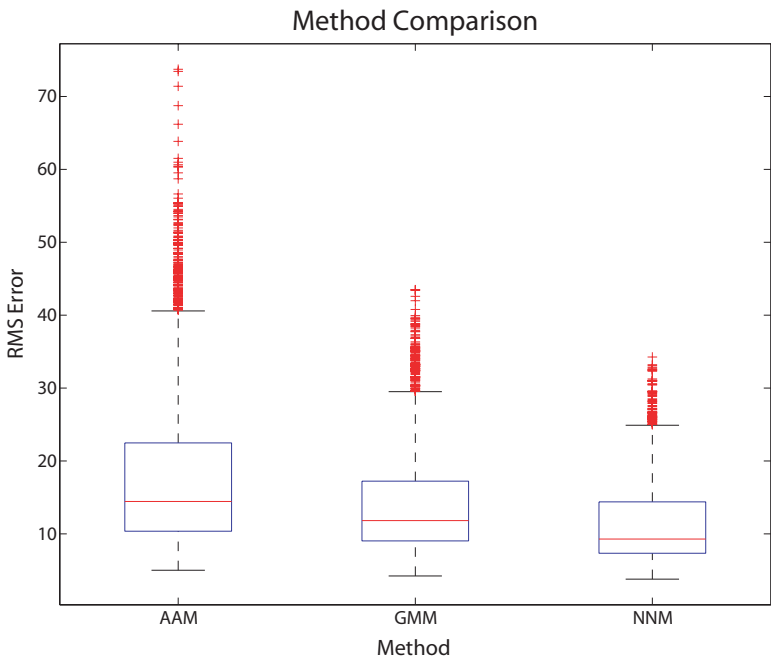


Fig. 17. Quantitative comparison between each method and a baseline linear model (Summarized in Table 2). A box plot of the RMS error of each model evaluated over 2500 test mouth images, combined from all five subjects, is shown. In the plot, the horizontal lines of each box represent the top quartile, median and bottom quartile values, the whiskers give the extent of the rest of the data and the red crosses label data outliers. Of the three methods, the AAM displays the worst performance and the nearest-neighbor model performs the best.

assumptions about the underlying structure of the appearance manifold.

The poor performance of the single AAM on the above mouth dataset is a direct result of the simplicity of the model. This model assumes a single texture space over the mouth appearance manifold. Since the appearance manifold has varying topology, a global texture space is ill-defined; the appearance variation of the mouth is not well represented using a single reference coordinate frame. This point was demonstrated by Figure 4 in Section 4. Also, the single AAM has no knowledge of the local structure of the manifold and can therefore converge to non-mouth images. Each of these properties contribute to the AAM's poor performance in modeling the appearance of the mouth. The nearest-neighbor and Gaussian mixture deformable models provide shape-and-

Method	AAM	GMM	NPM
Average RMS Error	18.7 ± 11.7	14.2 ± 7.3	11.3 ± 5.7

Table 2
Method Comparison. The average RMS error of each technique is displayed ± the standard deviation. A pair-wise comparison of the error distributions of each technique using statistical t-tests gave p-values of $p<<0.01$ for each distribution pair.

texture mappings that take into account the varying topology of the mouth appearance manifold and therefore are able to more faithfully represent the full range of mouth appearance variation.

Finally, we evaluate the performance of the nearest-neighbor algorithm for different k values. Figure 18 displays an RMS error box plot for the nearest-neighbor model evaluated over the 2500 test frames with different k values. The figure illustrates that the model performs better for increasing values of k. This verifies our intuition that morphing between examples does better than simply taking the nearest neighbor. As the number of examples increases the model is provided with more degrees of freedom and can therefore match the input image more closely.

The model reaches optimal performance around $k = 20$ for this dataset. For larger values of k ($k > 20$) the model begins to degrade in performance. This is expected since for larger values of k the assumptions made by the model may no longer hold and its performance becomes more similar to that of a linear deformable model which performs a global search over the manifold. Note, however, that even when k is set such that it includes all the examples in the training set ($k = 100$), the non-parametric model still performs better than the AAM. This is because the non-parametric model restricts its search to the convex hull of the examples during matching, whereas the AAM does not.

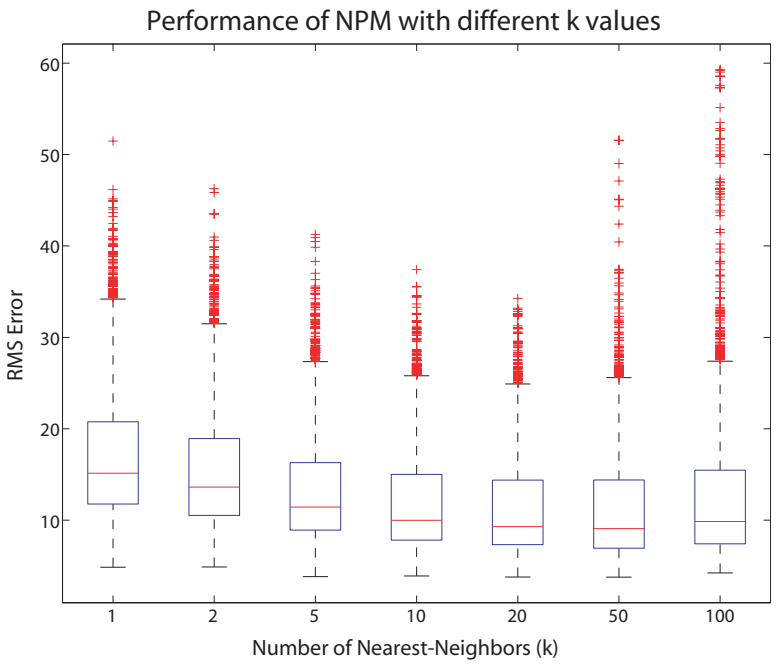


Fig. 18. Quantitative comparison of the nearest-neighbor method for different k. An RMS error box plot for the nearest-neighbor model evaluated over the 2500 test frames is displayed for different k values. The model performs better for increasing values of k. As the number of examples increases the model is provided with more degrees of freedom and can therefore match the input more closely. Too large k degrades model performance. The optimal value of k for this dataset is around $k = 20$.

6 Conclusions

This paper addresses some of the important limitations of contemporary shape-and-texture appearance models. We introduced a novel deformable modeling method based on an image-based rendering technique. Light-field deformable models have the potential to overcome some of the limitations presented by current 2D and 3D appearance models. They can easily model complex scenes, non-Lambertian surfaces, and view variation. We demonstrated the construction of a light-field manifold of the human head using a light-field dataset of 50 subjects and showed how to match the model to a light-field or single 2D image of a novel subject with unknown pose. The experiments performed on this dataset showed some of the advantages of the light-field model, how-

ever, we believe that the collection of a more extensive (both in the number of light-field views and subjects collected) and higher-quality dataset would better demonstrate the advantages of our approach.

We have presented a non-parametric technique for modeling the shape-and-texture appearance manifolds of complex objects whose appearance manifold has a varying topology consisting of parts or holes. This model generalizes well to complex manifolds, offers a compact representation of the manifold and allows for varying feature sets. In particular, with this technique a new input is analyzed by morphing a local set of examples that belong to a convex or bounded region of the manifold.

We evaluated the performance of the non-parametric deformable model using the AVTIMIT database, where we built a shape-and-texture appearance model of the mouth. We compared this approach to a baseline linear model and Gaussian mixture deformable model. We demonstrated that linear models poorly represent the appearance of complex objects such as mouths and that the nonlinear techniques of this paper are able to define a more convincing shape-and-texture mouth appearance model by taking into account the varying topology of the mouth appearance manifold. Of the three methods the linear deformable model performed the worst and the non-parametric deformable model performed the best. The noteworthy performance of the non-parametric model is expected since it makes the fewest assumptions about the underlying structure of the object appearance manifold.

There are many interesting avenues of future work. A clear next step of this work would be to demonstrate a nonlinear light-field deformable model that benefits from the strengths of each approach described in this paper. Sep-

arately, interesting topics of future work include the investigation of other image-based rendering techniques for constructing deformable models that require less imagery than light-fields and the use of BRDF models for representing objects under varying illumination. The development of a person-independent mouth deformable model would also be an exciting extension to this work. Possible improvements to the non-parametric deformable model include the use of Locality Sensitive Hashing [35] as an alternative, more efficient method for computing nearest neighbors and the consideration of different distance metrics that are less sensitive to lighting, location, orientation and scale.

References

[1] D. Beymer and T. Poggio. Face recognition from one example view. Technical Report AIM-1536, September 1995.

[2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Lecture Notes in Computer Science*, 1407:484–98, 1998.

[3] H. Murase and S. Nayar. Visual learning and recognition of 3-d objects from appearance. *IJCV*, 14(1):5–24, 1995.

[4] S. Sclaroff and J. Isidoro. Active blobs. In *ICCV*, Mumbai,India, 1998.

[5] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *International Conference on Computer Vision*, 2003.

[6] M. Turk and A. Pentland. Eigen faces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.

[7] G. Edwards, C. Taylor, and T. Cootes. Interpreting face images using active

appearance models. In *3rd International Conference on Automatic Face and Gesture Recognition*, pages 300–305, 1998.

[8] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, Los Angeles, 1999.

[9] C. M. Christoudias, L-P. Morency, and T. Darrell. Light field appearance manifolds. In *Proceedings of European Confernce on Computer Vision*, volume 4, pages 481–493, Prague, Czech Republic, May 2004.

[10] C. M. Christoudias and T. Darrell. On modelling nonlinear shape-and-texture appearance manifolds. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, June 2005.

[11] M. J. Jones and T. Poggio. Multidimensional morphable models. In *ICCV*, pages 683–688, 1998.

[12] S. Romdhani, S. Gong, and A. Psarrou. A multi-view nonlinear active shape model using kernel pca. In *British Machine Vision Conference*, pages 483–492, 1999.

[13] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor. View-based active appearance models. *Image and Vision Computing*, 20:657–664, 2002.

[14] J. C. Yang, M. Everett, C. Buehler, and L. McMillan. A real-time distributed light field camera. In *Eurographics Workshop on Rendering*, pages 1–10, 2002.

[15] T. J. Hazen, K. Saenko C. H. La, and J. Glass. A segment-bsaed audio-visual speech recognizer: Data collection, development, and initial experiments. In *Proc. ICMI*, 2005.

[16] B. Moghaddam and A. Pentland. Probabilistic visual learning for object recognition. *IEEE PAMI*, 19(7):696–710, 1997.

[17] P. Fua and C. Miccio. From regular images to animated heads: a least squares approach. In *ECCV*, pages 188–202, Springer,Berlin, 1999.

[18] F. H. Pighin, R. Szeliski, and D. Salesin. Resynthesizing facial animation through 3d model-based tracking. In *ICCV*, pages 143–150, 1999.

[19] M. E. Brand. Morphable 3D models from video. In *CVPR*, May 2001.

[20] M. Levoy and P. Hanrahan. Light field rendering. *Computer Graphics*, 30:31–42, 1996.

[21] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. *Computer Graphics*, 30(Annual Conference Series):43–54, 1996.

[22] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, and M. F. Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, pages 425–432, 2001.

[23] R. Gross, I. Matthews, and S. Baker. Appearance-based face recognition and light fields. *IEEE PAMI*, 26(4), April 2004.

[24] Z. Zhang, L. Wang, B. Guo, and H. Shum. Feature-based light field morphing. In *Conference on Computer graphics and interactive techniques*, pages 457–464. ACM Press, 2002.

[25] T. Beier and S. Neely. Feature-based image metamorphosis. In *Computer Graphics (Proceedings of SIGGRAPH 92).*

[26] T.F. Cootes, G.V. Wheeler, K.N. Walker, and C.J. Taylor. Coupled-view active appearance models. In *Proc. British Machine Vision Conference*, volume 1, pages 52–61, 2000.

[27] T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. In *Image and Vision Computing*, volume 17, pages 567–574, 1999.

[28] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In *CVPR*, June 2004.

[29] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *International Journal of Computer Vision*, 48(1):9–19, 2002.

[30] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

[31] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[32] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt. Performance of optical flow techniques. *CVPR*, 92:236–242.

[33] C. Goodall. Procustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society*, 53(2):285–339, 1991.

[34] T. Vetter, M. J. Jones, and T. Poggio. A bootstrapping algorithm for learning linear models of object classes. Technical Report AIM-1600, 1997.

[35] A. Gionis, P. Indyk, and R. Motwani. Similiarity search in high dimensions via hashing. In *25th International Conference on Very Large Data Bases*, pages 518–529, 1999.

[36] M. Jones and P. Viola. Fast multi-view face detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.

[37] M. B. Stegmann. Analysis and segmentation of face images using point annotations and linear subspace techniques. Technical report, Technical University of Denmark, DTU, August 2002.

[38] I. T. Nabney. *NETLAB Algorithms for Pattern Recognition*. Springer, 2004.