Modeling Object Appearance using Context-Conditioned Component Analysis

Daniyar Turmukhambetov[1] Neill D.F. Campbell[1,2]
Simon J.D. Prince[1,3] Jan Kautz[1,4]
[1]University College London, UK [2]University of Bath, UK [3]Anthropics Technology, UK [4]NVIDIA, USA

Abstract

*Subspace models have been very successful at modeling the appearance of structured image datasets when the visual objects have been aligned in the images (*e.g.*, faces). Even with extensions that allow for global transformations or dense warps of the image, the set of visual objects whose appearance may be modeled by such methods is limited. They are unable to account for visual objects where occlusion leads to changing visibility of different object parts (without a strict layered structure) and where a one-to-one mapping between parts is not preserved. For example bunches of bananas contain different numbers of bananas but each individual banana shares an appearance subspace.*

In this work we remove the image space alignment limitations of existing subspace models by conditioning the models on a shape dependent context that allows for the complex, non-linear structure of the appearance of the visual object to be captured and shared. This allows us to exploit the advantages of subspace appearance models with non-rigid, deformable objects whilst also dealing with complex occlusions and varying numbers of parts. We demonstrate the effectiveness of our new model with examples of structured inpainting and appearance transfer.

1. Introduction

Subspace models are commonly used to learn a parametric model of a visual object appearance from a large dataset of images. The parametric model captures low-dimensional representation of the appearance of the visual object as a linear combination of "components". Such representations have been successfully used to model the appearance of the visual object in a range of computer vision and graphics applications such as face detection, identification, image hallucination, image synthesis, *etc*.

These parametric models yield good results when images are preprocessed so that features of the visual objects are *aligned* throughout the dataset. This is applicable to visual objects that have fixed structure and do not have significant deformations or occlusions, for example frontal images of faces, medical scans of organs, *etc*. Some works address a more challenging case of visual objects that exhibit deformations, by jointly learning spatial transformations and appearance representations (*e.g.*, unaligned images of faces, images of mushrooms, medical scans of hands). The models rely on estimating dense mappings between the images of the dataset, usually to a common template.

However, visual objects with varying structure (Figure 1) cannot be readily expressed using previous methods. By fixed structure, we mean that images consist of a *fixed set* of regions that have certain associated textures. For example, side views of cars in general have *two* regions corresponding to two wheels and these regions are always present in the image. On the other hand, an example of a visual object with a varying structure would be a facade of a building with a *varying number* of windows. In this scenario, the texture of window regions can be shared across images, but each image may have a different number of regions corresponding to windows. Furthermore, datasets of visual objects that have occlusions or significant deformations which can cause self-occlusions or 2D topology changes also remain a challenging input, *e.g.*, animals under different poses. We note that, due to the diverse structure of such images, methods that estimate a warping or dense mapping between images, such as SIFT flow [19], are unlikely to succeed as some of their assumptions about the images are not met; *e.g.*, smoothness of the deformation field and the presence of occlusions.

Recently, image datasets of various objects have been provided with high-quality segmentation maps and per-pixel semantic and part labels; such labeled datasets are becoming more prevalent [11] and are being used to tackle a variety of visual challenges. We exploit such additional information by formulating a generative parametric statistical model that explains a visual object's appearance *conditioned* on the additional information, such as the segmentation map, but in a more complex way than just warping. So, *we assume that the additional information of each image is known* during training, as well as during sampling.

Existing methods for statistical shape modeling, for example the recent ShapeBM model [10], produce generative

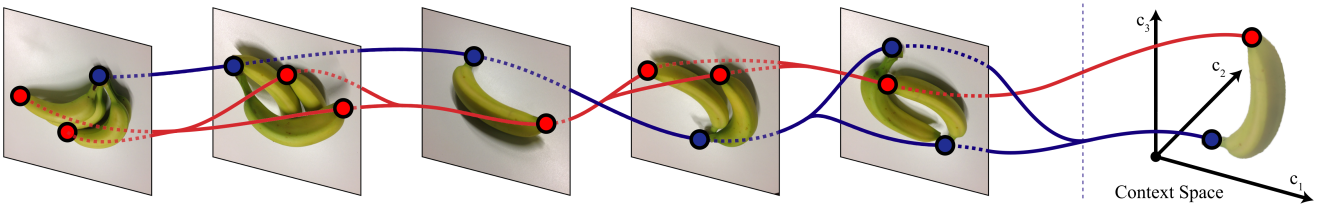


Figure 1. An example of an unstructured dataset: due to multiple instances and occlusions there is no clear way of aligning the images via a global transformation, or indeed estimating a dense warping (one-to-one mapping) between them. However, additional information, such as part labelling or segmentation of the bananas, can be used to *align the data in the context space* (right hand side of the figure); and allow us to learn a subspace model of the appearance of the bananas. In this example, the red and blue dots denote corresponding locations in the context space.

models of shape that could be adapted to include part labels. Alternatively, *discriminative* approaches that learn part labels, for example [25], can be used to produce part segmentations to train our *generative* model of appearance.

Our proposed subspace model allows us to model the appearance of objects with varying structure, such as animals and facades of buildings. We demonstrate the performance of the proposed subspace model on the task of structured inpainting of unobserved test images, and by transferring of the appearance of one instance (*e.g.*, in a specific pose) to others (*e.g.*, different poses).

2. Related work

Aligned Subspace Appearance Models One of the earliest works in learning a statistical model of a visual object from a dataset of images was Eigenfaces [28], by Turk and Pentland, that was used to address the problem of face detection and identification. Their approach expressed each image of a face as a feature vector of pixel intensities. These feature vectors are approximated as a linear combination of a subset of principal components derived from the covariance matrix of the probability distribution over the vector space of the face image pixel intensities.

Many related approaches followed; for example, the Fisherfaces algorithm [1] makes use of linear discriminant analysis (LDA). In this work, the subspace aims to maximise the ratio of inter-individual to intra-individual variance. The null-space LDA approach [4] makes use of the directions without intra-individual variance (these are ignored by Fisherfaces). The Dual-Space LDA approach [29] combined both of these subspace directions. All of these approaches rely on the image datasets to be aligned.

Active Appearance Models The Active Appearance Model (AAM) [7] by Cootes *et al*. is a more sophisticated statistical model that models both the shape and the appearance of a visual object from a dataset that consists of images and corresponding coordinates of a set of landmark points on each image. AAM extends subspace models by incorporating parameterized affine warps of all images to a common template. Edwards *et al*. [8] used AAM to address the problem of face detection and identification. The AAM can be applied to image alignment problems that estimate transformations of images to a common coordinate system or a common template [6, 24].

Jones and Soatto [14] proposed a Layered Active Appearance Model which allows for missing features, occlusion, substantial spatial rearrangement of features, and that provides a more general representation that extends the applicability of the traditional AAM. In LAAM the images are subdivided into "layers" (a set of compact regions determined by a pre-defined group of the landmark points with pixel's local coordinates, pixel intensities and landmark point coordinates) which are modeled with weighted PCA. Jones and Soatto demonstrated the model on a dataset of cars. The dataset consisted of frontal images of cars with and without certain features (such as foglights) and overlapping features (such as license plates). However, LAAM requires the ordering of layers to be fixed, *i.e.*, foglights always occlude bumpers and bumpers never occlude foglights. Such an ordering assumption is not always applicable (*e.g.*, consider the legs of animals). Also, the number of instances of each part must be known beforehand (a fixed number of layers) which is not required for our context based approach. The Morphable Model [2] extended the AAM into 3D for modeling faces which also helps with occlusions; however, it still cannot handle varying numbers of parts and requires alignment to a template.

Combining Parametric and Non-Parametric Models for Appearance Synthesis Liu *et al*. combined a parametric subspace model and a local non-parameteric model in a two step procedure to solve the problem of face hallucination [17, 18]. At the first step, principal components are learned from the dataset to express the relationship between the high-resolution face images and the corresponding blurred and down-sampled lower resolution images. At the second step, the residue between an original high-resolution image and the image reconstructed by using the learned subspace model is modeled by a patch-based non-parametric Markov network, to hallucinate the high-frequency details of the image.

Visio-lization [21] by Mohammed *et al*. used a similar

framework to address the problem of synthesizing frontal face images. First, a parametric subspace model is sampled to generate a low-resolution novel image which lacks high-frequency details. The second step upsamples the low-resolution image by performing image quilting [9] on overlapping image regions. Image quilting uses patches extracted from the images of the dataset that were conditioned on the low-resolution image.

Jointly Estimating Deformation and Appearance Transformed Component Analysis [13] estimates a global transformation for each image in the dataset to bring the images into alignment as well as finding an appearance subspace. The set of transformations they consider are rigid body motions and therefore not suited to highly deformable objects or occlusions.

Winn and Jojic introduced LOCUS [30], an unsupervised generative model that learns segmentations of visual objects. Most interestingly, they model visual objects with deformation fields and achieve dense registration between different images of the same class despite differences in appearance or pose.

Mobahi *et al*. [20] also learn a model of a visual object from a set of images. They assume that images are generated as a nested composition of color, appearance and shape transforms. By modeling each component as a low-dimensional subspace they can learn a regularized solution of such compositional model from a set of images. Their shape(geometric) transform is jointly learned for all images, and outperforms SIFT flow and robust optical flow for any pair of images from the set.

Both the LOCUS model and the work of Mobahi *et al*. make use of a deformation (or flow) field for dense warping. Such a field cannot encompass the range of transformations that we address with our approach (complex occlusions and varying numbers of instances).

3. Context-Conditioned Component Analysis

We begin by providing an illustrative example that reveals the limitations of existing approaches; we use this example to motivate the use of our method for modeling the appearance of complex visual objects. We then provide a high-level description of our new model that we call Context-Conditioned Component Analysis (C-CCA). We show a general formulation and demonstrate that our model encompasses Probabilistic Principal Component Analysis (PPCA) as a special case.

For simplicity, we will begin by assuming that the images are greyscale, but we relax this to include the case of color images in section 4.6.

3.1. Motivating Example

Consider images of bunches of bananas as shown in Figure 1. This example dataset does not display fixed structure as every image has a different number of bananas that may occlude one another with no consistent ordering (*c.f*. the layer structure limitation of LAAM). Clearly, aligning these images, or estimating a dense warping between them, is not possible; *i.e*., deforming 3 bananas to 1 or vice versa. However, the appearance of the bananas in each image is correlated. We argue that additional information, such as part labelling or segmentation of the bananas, can be used to *align the data in the context space* and allow us to learn the subspace model of the appearance of the bananas. In this example, the red and blue dots denote corresponding locations in the context space (*not* fiducial points as used for AAM).

3.2. Model Description

Consider a set of greyscale images $\{\mathbf{x}_i\}_{i=1}^{I}$ of a given visual class (e.g. horses, facades, cats). Each image is represented as a vector $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots x_{iJ}]^T$ where the element x_{ij} represents the j^{th} pixel from the i^{th} image. Associated with each pixel in each image is a *context vector* $\mathbf{c}_{ij}$. This vector provides some information about the state of the object at that pixel. For example, it might encode the part of the object at that pixel (door, wall of a house), the local shape of the silhouette of the object, or the distance from some pre-determined keypoint. The context vector constrains the intensity values that might be present at a given pixel, but does not entirely determine it.

The model for the j^{th} pixel of the i^{th} image is

$$x_{ij} = \mu\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_\mu\right] + \sum_{f=1}^{F} \phi\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_f\right] h_{if} + \epsilon_{ij} \quad , \qquad (1)$$

where the first term gives the mean value μ_{ij} at the pixel and is a function of the context $\mathbf{c}_{ij}$ at that pixel and a parameter vector $\boldsymbol{\theta}_\mu$. The second term consists of a weighted sum of F function terms $\phi[\cdot,\cdot]$ where the weights $\{h_{if}\}_{f=1}^{F}$ are constant for the whole image and can be considered hidden variables. Each of the function terms maps the pixel context $\mathbf{c}_{ij}$ to a scalar value where the F mappings in the weighted sum are determined by associated parameter variables $\{\boldsymbol{\theta}_f\}_{f=1}^{F}$. Finally, each term ϵ_{ij} is an independent stochastic noise variable distributed as $\text{Norm}_{\epsilon_{ij}}[0, \sigma^2]$.

For now, we remain agnostic about the exact form of the functions $\mu\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_\mu\right]$ and $\phi\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_f\right]$, except to say that they map from the context vector $\mathbf{c}_{ij}$ to a scalar value, and so take the form of regression functions.

We can equivalently write equation 1 in vector form as

$$\mathbf{x}_i = \boldsymbol{\mu}_i + \boldsymbol{\Phi}_i \mathbf{h}_i + \boldsymbol{\epsilon}_i \quad , \qquad (2)$$

where the j^{th} row of the $J \times 1$ vector $\boldsymbol{\mu}_i$ is given by $\mu\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_\mu\right]$ and the j^{th} row and f^{th} column of the $J \times F$ basis matrix $\boldsymbol{\Phi}_i$ is given by the function $\phi\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_f\right]$. We

have similarly vectorized the hidden variables so that $\mathbf{h}_i = [h_{i1}, h_{i2}, \ldots h_{iF}]^T$ and $\boldsymbol{\epsilon}_i = [\epsilon_{i1}, \epsilon_{i2} \ldots \epsilon_{iJ}]^T$.

Equation 2 can be written in probabilistic form as

$$Pr(\mathbf{x}_i|\mathbf{h}_i, \boldsymbol{\theta}_\bullet, \sigma^2) = \text{Norm}_{\mathbf{x}_i}\left[\boldsymbol{\mu}_i + \boldsymbol{\Phi}_i\mathbf{h}_i, \sigma^2\mathbf{I}\right] \quad , \quad (3)$$

where the notation $\boldsymbol{\theta}_\bullet$ is shorthand[1] for all of the unknown function parameters $\boldsymbol{\theta}_\mu$ and $\{\boldsymbol{\theta}_f\}_{f=1}^F$ and $\mathbf{I}$ is the identity matrix. To complete the model, we also define an independent standard Gaussian prior over this hidden variable vector $\mathbf{h}_i$, so

$$Pr(\mathbf{h}_i) = \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \quad . \quad (4)$$

3.3. Relation to Probabilistic PCA

Probabilistic Principal Component Analysis (PPCA) is a special case of the proposed model; for the case of a set of images where the context vectors match at each pixel so that $\mathbf{c}_{1j} = \mathbf{c}_{2j} \ldots = \mathbf{c}_{Ij}$, the mean vectors $\boldsymbol{\mu}_i$ and the basis matrices $\boldsymbol{\Phi}_i$ will be the same for every image i and equation 2 becomes the generative model for PPCA:

$$\mathbf{x}_i = \boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}_i + \boldsymbol{\epsilon}_i \quad . \quad (5)$$

It follows that the proposed model is a generalisation of PPCA where the mean vector $\boldsymbol{\mu}$ and the principal components $\boldsymbol{\Phi}$ depend on the context vectors for the given image $\{\mathbf{c}_{ij}\}_{j=1}^J$. Hence, we create a different but related PPCA model for each image in the set.

4. Learning

In this section we will discuss how we fit our new appearance model to the training image data. We start by outlining the general learning procedure before discussing the specific approach that we choose. The adopted approach requires estimating three unknown variables that we discuss in turn in sections 4.2 to 4.4. For tractability, we choose a special form of the functions $\phi[\cdot,\cdot]$ in section 4.5. Then, we explain how we model color images in section 4.6. Finally, the algorithm listing 1 describes how we initialize the variables and summarizes the learning procedure that we use to fit our model to a collection of training images.

4.1. Learning Approach

We now outline a general learning procedure. We aim to take a set of images $\{\mathbf{x}_i\}_{i=1}^I$ with known context vectors $\{\mathbf{c}_{ij}\}_{i,j=1}^{I,J}$ and estimate the unknown parameters $\boldsymbol{\theta}_\bullet$ and the noise term σ^2 using a maximum likelihood formulation

$$\hat{\boldsymbol{\theta}}_\bullet, \hat{\sigma}^2 = \underset{\boldsymbol{\theta}_\bullet, \sigma^2}{\text{argmax}} \; \log\left[Pr(\mathbf{x}_i|\boldsymbol{\theta}_\bullet, \sigma^2)\right] \quad , \quad (6)$$

[1] Throughout we use the symbol $\bullet$ to denote the unravel and concatenate operation similar to the `Matlab` colon operator '`:`', *e.g.*, `Vector = Matrix(:);`.

where the log likelihood in this equation is obtained by marginalising out the hidden variables so that

$$Pr(\mathbf{x}_i|\boldsymbol{\theta}_\bullet, \sigma^2) = \int Pr(\mathbf{x}_i|\mathbf{h}_i, \boldsymbol{\theta}_\bullet, \sigma^2)Pr(\mathbf{h}_i)d\mathbf{h}_i \; . \quad (7)$$

Alternative approaches to learning these parameters include

1. Integrating out the hidden variables $\mathbf{h}_i$ in closed form and maximize the criteria in equation 6.
2. Using the EM algorithm to alternately (i) estimate a probability distribution over the hidden variables to determine a bound on the likelihood (E-Step) and then (ii) optimise this bound with respect to the parameters (M-Step).
3. We could replace the integration in equation 7 with a maximization over the hidden variables and alternately estimate the hidden variables and the unknown parameters. This can be shown to be a special case of the generalized EM algorithm.

In this paper, we adopt the third of these three options for simplicity.

4.2. Estimation of hidden variables

For the fixed parameters $\boldsymbol{\theta}_\bullet, \sigma^2$ we estimate each of the I hidden variables $\mathbf{h}_i$ as

$$\hat{\mathbf{h}}_i = \underset{\mathbf{h}_i}{\text{argmax}} \; \log\left[Pr(\mathbf{x}_i|\mathbf{h}_i, \boldsymbol{\theta}_\bullet, \sigma^2)Pr(\mathbf{h}_i)\right] \quad . \quad (8)$$

This maximization has a known closed form solution, which can be found by noting that the likelihood term $Pr(\mathbf{x}_i|\mathbf{h}_i, \boldsymbol{\theta}_\bullet, \sigma^2)$is a normal distribution with a mean that is a linear function of $\mathbf{h}_i$ (see equation 2) and can be rewritten as a constant with respect to $\mathbf{h}_i$ times a normal distribution in $\mathbf{h}_i$. What remains is a product of two Gaussians both in $\mathbf{h}_i$. The product of two Gaussians is also Gaussian and the optimal parameter value will be at the mean of this new distribution and is given by

$$\hat{\mathbf{h}}_i = (\boldsymbol{\Phi}_i^T\boldsymbol{\Phi}_i + \sigma^2\mathbf{I})^{-1}\boldsymbol{\Phi}_i^T(\mathbf{x}_i - \boldsymbol{\mu}_i) \quad . \quad (9)$$

More details can be found in section 7.6.2 of [22].

4.3. Estimation of noise

The noise parameter can also be estimated in closed form and is determined by the difference between the model predictions and the observed data:

$$\begin{aligned} \hat{\sigma}^2 &= \underset{\sigma^2}{\text{argmax}} \sum_{i=1}^{I} \log\left[Pr(\mathbf{x}_i|\mathbf{h}_i, \boldsymbol{\theta}_\bullet, \sigma^2)Pr(\mathbf{h}_i)\right] \quad (10) \\ &= \frac{1}{IJ}\sum_{i=1}^{I}(\mathbf{x}_i - \boldsymbol{\mu}_i - \boldsymbol{\Phi}_i\mathbf{h}_i)^T(\mathbf{x}_i - \boldsymbol{\mu}_i - \boldsymbol{\Phi}_i\mathbf{h}_i) \; . \end{aligned}$$

4.4. Estimation of function parameters

For fixed hidden variables $\mathbf{h}_i$ and noise σ^2 we estimate the function parameters using maximum likelihood:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_\bullet &= \underset{\boldsymbol{\theta}_\bullet}{\operatorname{argmax}} \sum_{i=1}^{I} \log\left[Pr(\mathbf{x}_i|\mathbf{h}_i, \boldsymbol{\theta}_\bullet, \sigma^2) Pr(\mathbf{h}_i)\right] \\ &= \underset{\boldsymbol{\theta}_\bullet}{\operatorname{argmax}} \sum_{i=1}^{I} \log\left[Pr(\mathbf{x}_i|\mathbf{h}_i, \boldsymbol{\theta}_\bullet, \sigma^2)\right] \quad , \end{aligned} \tag{11}$$

where we have dropped the prior term which does not depend on the unknowns. From equations 1 and 3, we can write the likelihood from equation 11 as

$$\begin{aligned} Pr(\mathbf{x}_i|\mathbf{h}_i, \boldsymbol{\theta}_\bullet, \sigma^2) \quad &= \\ \prod_{j=1}^{J} \text{Norm}_{x_{ij}}&\left[\mu\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_\mu\right] + \sum_{f=1}^{F} \phi\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_f\right] h_{if}, \sigma^2\right] . \end{aligned} \tag{12}$$

We can simplify this expression if we assume that the function $\mu\left[\cdot,\cdot\right]$ takes the same form as $\phi\left[\cdot,\cdot\right]$ and then define $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_\mu$ and $h_{i0} = 1$ so that

$$Pr(\mathbf{x}_i|\mathbf{h}_i, \boldsymbol{\theta}_\bullet, \sigma^2) = \prod_{j=1}^{J} \text{Norm}_{x_{ij}}\left[\sum_{f=0}^{F} \phi\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_f\right] h_{if}, \sigma^2\right] . \tag{13}$$

When we substitute equation 13 into equation 11 we see that we are optimising the logarithm of normal distributions and it follows that the parameter estimation takes the form of a non-linear least squares problem

$$\hat{\boldsymbol{\theta}}_\bullet = \underset{\boldsymbol{\theta}_\bullet}{\operatorname{argmax}} \sum_{ij} \left(x_{ij} - \sum_{f=0}^{F} \phi\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_f\right] h_{if}\right)^2 . \tag{14}$$

Unfortunately, solving for a general form of the function $\phi\left[\cdot,\cdot\right]$ is not easy and we would have to apply an iterative method such as Gauss-Newton to find a local minimum.

4.5. Choosing the form of the functions $\phi[\cdot,\cdot]$

Fortunately, there is an important class of functions $\phi[\cdot,\cdot]$ for which we can compute a closed-form solution for equation 14. In particular, we assume that the functions take the form of a pre-determined non-linear vector function $\mathbf{a}\left[\mathbf{c}_{ij}\right]$ of the context vectors; this is then linearly projected onto the parameter vector $\boldsymbol{\theta}_f$, such that

$$\phi\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_f\right] = \mathbf{a}\left[\mathbf{c}_{ij}\right]^T \boldsymbol{\theta}_f \quad . \tag{15}$$

This assumption still allows us to approximate complex functions since many regression approaches (including Gaussian processes [23], K-Nearest Neighbors [12] and relevance vector machines [27]) can be written in this form.

When we substitute equation 15 into the least squares criterion (equation 14) we see that the solution for the parameters $\boldsymbol{\theta}_\bullet$ now becomes the *linear* least squares problem

$$\hat{\boldsymbol{\theta}}_\bullet = \underset{\boldsymbol{\theta}_\bullet}{\operatorname{argmax}} \sum_{ij} \left(x_{ij} - \sum_{f=0}^{F} \mathbf{a}\left[\mathbf{c}_{ij}\right]^T \boldsymbol{\theta}_f h_{if}\right)^2 , \tag{16}$$

that can be solved in closed form.

Unfortunately, even this might be difficult to compute in practice. Let M be the dimensionality of the non-linear projection vector $\mathbf{a}[\mathbf{c}_{ij}]$. Then, we are solving a system of (IJ) equations (where I is the image number and J is the number of pixels per image) and (FM) unknowns, and this may be very slow and memory intensive.

To make this system practical, we use non-linear projections which are sparse (i.e., most of the elements of $\mathbf{a}[\mathbf{c}_{ij}]$ are zero). A simple example of this is *K-Nearest Neighbors* regression [12]. In this case, we compare the context $\mathbf{c}_{ij}$ to M prototype context vectors $\{\mathbf{z}_m\}_{m=1}^{M}$ and $\mathbf{a}$ becomes an $M \times 1$ vector which is zeros except for the K indices corresponding to the K nearest neighbors. These K indices are set to the inverse of the euclidean distance in the context vector space and $\mathbf{a}$ is normalized so that elements sum to 1. We normalize the context vectors to zero mean and unit variance.

4.6. Modeling Color

To model appearance, we use our C-CCA on 3 color channels $\{\mathbf{x}_i^{(r)}, \mathbf{x}_i^{(g)}, \mathbf{x}_i^{(b)}\}_{i=1}^{I}$, by using a separate function parameter $\boldsymbol{\theta}_\bullet^{(r)}, \boldsymbol{\theta}_\bullet^{(g)}, \boldsymbol{\theta}_\bullet^{(b)}$ for each of the color channels. We process the colorspace of each image in the training set $\{\tilde{\mathbf{x}}_i^{(r)}, \tilde{\mathbf{x}}_i^{(g)}, \tilde{\mathbf{x}}_i^{(b)}\}_{i=1}^{I}$ with the photometric transformation technique introduced in [20].

We estimate a rotation matrix $\mathbf{R}_i$ and translation vector $\mathbf{t}_i$, for each image, that transforms the "common" color space modeled by the C-CCA to each of the individual image color spaces, such that

$$\tilde{\mathbf{x}}_{ij}^\star = \mathbf{R}_i \mathbf{x}_{ij}^\star + \mathbf{t}_i \quad , \tag{17}$$

where $\tilde{\mathbf{x}}_{ij}^\star = \left(\tilde{x}_{ij}^{(r)}, \tilde{x}_{ij}^{(g)}, \tilde{x}_{ij}^{(b)}\right)^T$ and the notation $(\cdot^\star)$ is used to denote variables that represent values over 3 color channels.

Since $\mathbf{R}_i$ is a rotation matrix, its inverse can be applied to the images before each iteration of the generalized EM as

$$\mathbf{x}_{ij}^\star = \mathbf{R}_i^{-1}\left(\tilde{\mathbf{x}}_{ij}^\star - \mathbf{t}_i\right) \quad . \tag{18}$$

The rotation and translation matrices are estimated at the end of each of the iterations by solving

$$\{\mathbf{R}_i, \mathbf{t}_i\}_{i=1}^{I} = \underset{\mathbf{R}_i, \mathbf{t}_i}{\operatorname{argmin}} \sum_{j} \left(\tilde{\mathbf{x}}_{ij}^\star - \mathbf{R}_i \mathbf{y}_{ij}^\star - \mathbf{t}_i\right)^2 , \tag{19}$$

Algorithm 1 C-CCA Learning Procedure

Require: *RGB values* $\{\tilde{\mathbf{x}}_i^\star\}$, *Context vectors* $\{\mathbf{c}_{ij}\}$

$\forall i : \mathbf{h}_i \leftarrow$ Random sample of $\text{Norm}_{\mathbf{h}_i}[0, \mathbf{I}]$

$\forall i : \mathbf{R}_i \leftarrow \mathbf{I}; \mathbf{t}_i \leftarrow \vec{\mathbf{0}}$

$\sigma^2 \leftarrow 1$

$\forall i, j : \mathbf{c}_{ij} \leftarrow (\mathbf{c}_{ij} - \texttt{mean}[\mathbf{c}]) \,/\, \texttt{std}[\mathbf{c}]$

$\{\mathbf{z}_m\}_{m=1}^M \leftarrow$ M random samples of $\{\mathbf{c}_{ij}\}$

$\forall i, j : \mathbf{a}[\mathbf{c}_{ij}] \leftarrow \texttt{kNN}\,[\mathbf{c}_{ij}, \{\mathbf{z}_m\}]$

for number of iterations **do**

$\forall i, j : \mathbf{x}_{ij}^\star \leftarrow \mathbf{R}_i^{-1}(\tilde{\mathbf{x}}_{ij}^\star - \mathbf{t}_i)$

$\boldsymbol{\theta} \leftarrow$ Solution of Equation 16

$\sigma^2 \leftarrow$ Solution of Equation 10

$\forall i : \mathbf{h}_i \leftarrow$ Solution of Equation 9

$\forall i : \mathbf{R}_i, \mathbf{t}_i \leftarrow$ Solution of Equation 19

end for

return $\{\mathbf{z}_m\}, \boldsymbol{\theta}, \{\mathbf{h}_i\}, \sigma^2, \{\mathbf{R}_i\}, \{\mathbf{t}_i\}$

where $\mathbf{R}_i$ is constrained to be a rotation matrix and $\mathbf{y}_{ij}^\star = \left(y_{ij}^{(r)}, y_{ij}^{(g)}, y_{ij}^{(b)}\right)^T$ is the reconstruction before the color transform with

$$y_{ij}^{(\kappa)} = \sum_{f=0}^{F} \phi\left[\mathbf{c}_{ij}\,,\, \boldsymbol{\theta}_f^{(\kappa)}\right] h_{if} \quad , \tag{20}$$

for $\kappa \in \{r, g, b\}$. Equation 19 has a known closed form solution [15, 20].

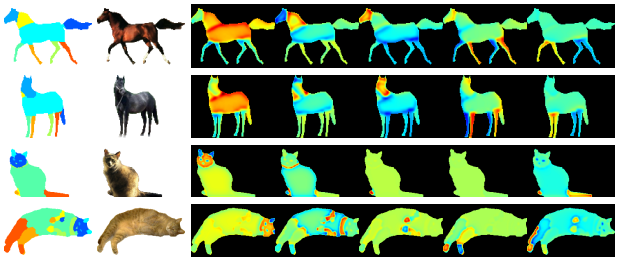
5. Experiments

5.1. Datasets and Context Vectors

Horses We use the Weizmann Horse Dataset [3] that consists of images of horses and corresponding segmentation masks. Additionally, we labeled images with 7 semantic parts (head, neck, torso, each of the 4 legs). Each pixel's context vector consists of spatial coordinates of the pixel (2 values), and filter responses of the segmentation mask and the 7 part masks. The filterbank consists of 15 filters (a subset of Leung-Malik Filter Bank [16]), namely the first and second derivatives of Gaussians in 6 orientations, Laplacian of Gaussian at 2 scales and 1 Gaussian filter.

Cats We use images with cats from Pascal VOC2010 dataset [11] with segmentation masks and semantic labels (17 body parts of the cat) provided by [5]. We discarded images that had low resolution and rescaled images such that the distance between eyes is consistent. Each pixel's context vector consists of the distance to the middle of the eyes and filter responses of the segmentation mask and the 17 part masks. The filterbank is same as for the horses example.

Elephants We compiled a dataset of web images of elephants with corresponding segmentation masks and body part labels (head, torso, trunk and 4 legs). Each pixel's context vector consists of the spatial coordinates of the pixel (2 values), and filter responses of the segmentation mask and the 7 part masks. The filterbank is same as for the horses example.



(a) (b) (c)

Figure 2. Visualization of the context vectors. (a) Part labels. (b) RGB Image. (c) Projections onto the first five principal components of the context vectors (the true dimensionality of the context vectors is 122 and 271 for horses and cats respectively).

Dataset	# Images	# Parts	Length of $\mathbf{c}_{ij}$	Avg. # pixels	Resolution
Horses	295	7	122	3068	150x150
Cats	567	17	271	5069	246x249
Elephants	275	7	122	3672	136x136
Facades	104	5	42	10004	187x174

Table 1. Dataset Statistics.

Facades We use facades of buildings of Paris from the Ecole Centrale Paris Facades Database [26]. We use pixels that were labeled as wall, window, door, roof or balcony. Each pixel's context vector consists of the spatial coordinates and filter responses semantic labeling masks. The filterbank consists of 8 Haar-like features at 2 different scales.

Table 1 summarizes the relevant statistics of each of the datasets. Figure 2 visualizes the context vectors for the horses and the cats datasets.

5.2. Quantitative Evaluation

Table 2 shows the parameters that were empirically chosen for fitting each of the datasets. Only foreground pixels are used for training and testing. Table 2 also shows timing of EM iterations computed on 6 core @ 3GHz XEON CPU. We used the fitted models to compute the reconstruction images of the training and test sets. Table 2 shows the per-pixel error of both training and test sets.

We can compare our model to PPCA, as it is a special case of our model (see section 3.3). We change the context vectors to only include pixel coordinates, set K of the k-NN regression learner to 1; and use all of the unique pixel coordinates in our training data as the prototype context vectors $\{\mathbf{z}_m\}_{m=1}^M$. We use the same number of components (F) as used for C-CCA for each dataset. The reconstruction error is reported in Table 2. C-CCA outperforms PPCA across all datasets while using a significantly smaller number of parameters (M).

				C-CCA					PPCA		
Dataset	I	Test I	F	K	M	Iter time (min)	Training Set Mean Error	Test Set Mean Error	M	Training Set Mean Error	Test Set Mean Error
Horses	200	95	24	16	2000	9	0.0258 ± 0.0079	0.0547 ± 0.0432	11459	0.0268 ± 0.0092	0.0886 ± 0.0661
Cats	450	117	16	16	4500	27	0.0280 ± 0.0119	0.0511 ± 0.0351	44015	0.0315 ± 0.0131	0.0548 ± 0.0371
Elephants	200	75	20	16	3000	10	0.0120 ± 0.0031	0.0346 ± 0.0163	11803	0.0169 ± 0.0048	0.0470 ± 0.0220
Facades	80	24	20	16	2000	9	0.0333 ± 0.0094	0.0593 ± 0.0264	23629	0.0350 ± 0.0198	0.0889 ± 0.0330

Table 2. Parameters and Mean Per-pixel Reconstruction Error (mean squared error in RGB colorspace) for C-CCA and PPCA.

Figure 3. Appearance Transfer Results (Horses). (a) A subset of images of the training set. (b) Reconstruction of the images using the fitted subspace model with the estimated weights h_g. (c) Reconstruction of the images using the fitted subspace model with the fixed function weights h_g, rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ of the first (leftmost) image.

5.3. Appearance Transfer

The estimated function weights $\mathbf{h}_i$ and the colorspace rotation $\mathbf{R}_i$ and translation vector $\mathbf{t}_i$ of the image i of the dataset define the appearance of the visual object. So, for appearance transfer, we select another image j from the dataset, and reconstruct it by weighing the basis matrix $\mathbf{\Phi}_j$ with $\mathbf{h}_i$, and apply the colorspace transformation using $\mathbf{R}_i$ and $\mathbf{t}_i$. Figures 3 and 4 show examples.

Notice that computing a dense correspondence between some of these images is challenging. One of techniques for solving image alignment is SIFT flow [19]; Figure 5 illustrates how SIFT flow fails to compute a correspondence, as the pairs of images have animals in very different poses with multiple occlusions.

5.4. Structured Inpainting

We use the *test set* to do image inpainting. Half of the image's intensities are used to estimate regression function weights h_g and colorspace rotation matrix $\mathbf{R}_i$ and translation vector $\mathbf{t}_i$. We use the context vectors of *all* the pixels to reconstruct the whole image. Figures 6, 7 and 8 show the results. Notice the difference between the reconstructed and the inpainted images.

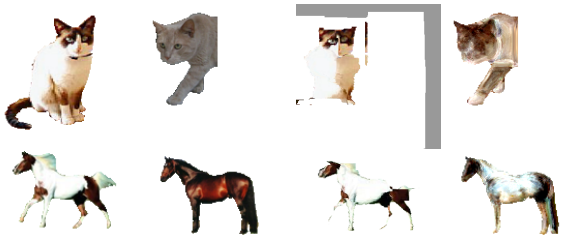


Figure 5. Left to Right: Source image. Target image. The warping of the source image onto the target image using the correspondence computed by SIFT flow [19]. The appearance transfer using C-CCA.

The inpainting results in Figure 9 of the Facades dataset demonstrate the importance of context vectors in our model. Due to the provided labeling, all windows in each of the facades have same context vectors. This results in the function $\phi[\cdot,\cdot]$ returning equivalent values for all pixels that were labeled as window. As the result, all windows in our reconstruction and inpainted images look the same.

6. Conclusion

We have proposed a novel generative subspace model that exploits additional information such as segmentation or semantic labeling to model the appearance of visual objects.

Figure 4. Appearance Transfer Results (Cats): For each column, rows top to bottom: (1) An image of the training set. (2-4) Reconstruction of the image of *another* cat in a different pose by fixed function weights h_g, rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ of the top row image.

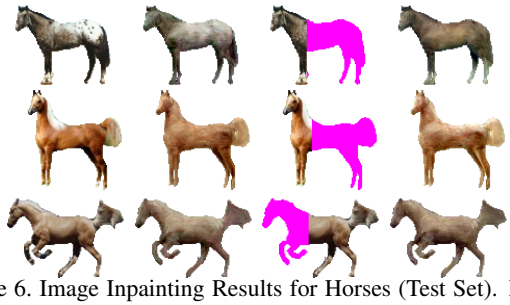


Figure 6. Image Inpainting Results for Horses (Test Set). Left to Right: Image from test set. Reconstruction of the image using the fitted model. Input for inpainting (context vectors of all pixels are known). Inpainted result.

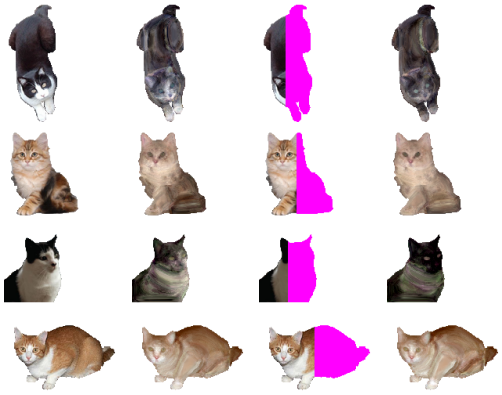


Figure 7. Image Inpainting Results for Cats (Test Set). Left to Right: Image from test set. Reconstruction of the image using the fitted model. Input for inpainting (context vectors of all pixels are known). Inpainted result.

The "components" learned by our model are conditioned on the per-pixel context vectors that were computed using additional information. This enables sharing of appearance information in a complex, non-linear way. We demonstrated that C-CCA can successfully model challenging datasets

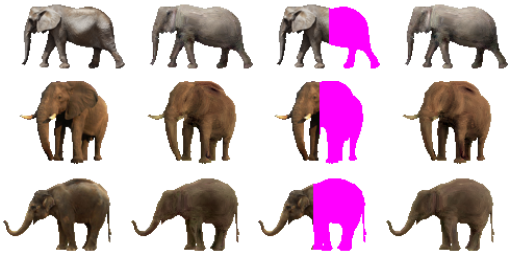


Figure 8. Image Inpainting Results for Elephants (Test Set). Left to Right: Image from test set. Reconstruction of the image using the fitted model. Input for inpainting (context vectors of all pixels are known). Inpainted result.

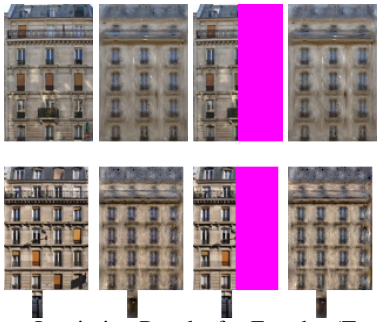


Figure 9. Image Inpainting Results for Facades (Test Set). Left to Right: Image from test set. Reconstruction of the image using the fitted model. Input for inpainting (context vectors of all pixels are known). Inpainted result.

that have complex occlusions non-rigid, deformable visual objects and varying numbers of parts. We reported both qualitative and quantitative results of our model with examples of structured inpainting and appearance transfer.

In future work, we would like to learn the weights for the dimensions of the context vector using the training data as opposed to naively normalizing the context vectors.

Acknowledgements This work was funded by EPSRC grant EP/I031170/1.

References

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997. 2

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 187–194, 1999. 2

[3] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Computer Vision – ECCV 2002*, pages 109–122. Springer, 2002. 6

[4] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu. A new lda-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713–1726, 2000. 2

[5] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 6

[6] T. Cootes, C. Twining, V. Petrovic, K. Babalola, and C. Taylor. Computing accurate correspondences across groups of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(11):1994–2005, Nov 2010. 2

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Computer Vision – ECCV 1998*, pages 484–498. Springer, 1998. 2

[8] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Interpreting face images using active appearance models. In *Automatic Face and Gesture Recognition, Third IEEE International Conference on*, pages 300–305. IEEE, 1998. 2

[9] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer graphics and Interactive Techniques*, pages 341–346. ACM, 2001. 3

[10] S. A. Eslami, N. Heess, and J. Winn. The shape boltzmann machine: a strong model of object shape. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 406–413. IEEE, 2012. 1

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html. Accessed: 2014-09-25. 1, 6

[12] E. Fix and J. L. Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, DTIC Document, 1951. 5

[13] B. J. Frey and N. Jojic. Transformed component analysis: joint estimation of spatial transformations and image components. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1190–1196, 1999. 3

[14] E. Jones and S. Soatto. Layered active appearance models. In *Computer Vision, International Conference on*, volume 2, pages 1097–1102. IEEE, 2005. 2

[15] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. 6

[16] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001. 6

[17] C. Liu, H.-Y. Shum, and W. T. Freeman. Face hallucination: Theory and practice. *International Journal of Computer Vision*, 75(1):115–134, 2007. 2

[18] C. Liu, H.-Y. Shum, and C.-S. Zhang. A two-step approach to hallucinating faces: global parametric model and local nonparametric model. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–192. IEEE, 2001. 2

[19] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT flow: Dense correspondence across different scenes. In *Computer Vision – ECCV 2008*, pages 28–42. Springer, 2008. 1, 7

[20] H. Mobahi, C. Liu, and W. T. Freeman. A compositional model for low-dimensional image set representation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2014. 3, 5, 6

[21] U. Mohammed, S. J. D. Prince, and J. Kautz. Visiolization: generating novel facial images. *ACM Trans. Graph.*, 28(3):57:1–57:8, July 2009. 2

[22] S. J. D. Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, 2012. 4

[23] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. 5

[24] E. Sánchez-Lozano, F. De la Torre, and D. González-Jiménez. Continuous regression for non-rigid image alignment. In *Computer Vision – ECCV 2012*, pages 250–263. Springer, 2012. 2

[25] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 2

[26] O. Teboul. Ecole Centrale Paris Facades Database. http://vision.mas.ecp.fr/Personnel/teboul/data.php. Accessed: 2014-07-01. 6

[27] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *The journal of machine learning research*, 1:211–244, 2001. 5

[28] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591. IEEE, 1991. 2

[29] X. Wang and X. Tang. Dual-space linear discriminant analysis for face recognition. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 564–569, 2004. 2

[30] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *Computer Vision, International Conference on*, volume 1, pages 756–763. IEEE, 2005. 3