Supplementary Materials: 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

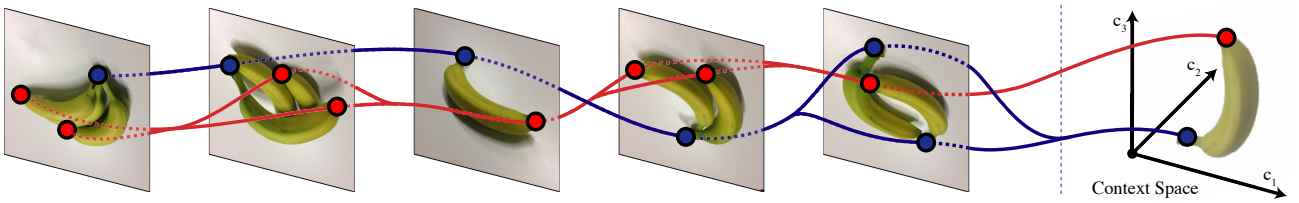


Figure 1: An example of an unstructured dataset: due to multiple instances and occlusions there is no clear way of aligning the images via a global transformation, or indeed estimating a dense warping (one-to-one mapping) between them. However, additional information, such as part labelling or segmentation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

1. Relation to AAM

In this section we show that Active Appearance Model [2] is another special case of Context-Conditioned Component Analysis when a specific form of $\phi[\cdot,\cdot]$ is adopted. As a reference, we include a motivational example of an unstructured dataset in Figure 1 explicitly demonstrating the mapping between locations in image and context space.

1.1. AAM

We start by describing AAM. The AAM assumes that the input consists of a set of images $\{\mathbf{x}_{ij}\}_{i=1,j=1}^{I,J}$ and a corresponding set of x-axis and y-axis coordinates of the fiducial points $\{\mathbf{u}_{iq}\}_{i=1,q=1}^{I,Q}$. (For example, for faces, the fiducial points are points such as corners of the eyes, the tip of the nose, chin, etc.)

Next, the mean coordinate of each of the fiducial points is computed: $\{\bar{\mathbf{u}}_q\}_{q=1}^{Q}$. It is assumed that these coordinates define the mean shape of the object, which serve as a "common template" for the images of the dataset.

The mean coordinates $\{\bar{\mathbf{u}}_q\}_{q=1}^{Q}$ are used as vertices of a mesh with triangular faces computed by Delaunay triangulation [3] or some other variation of it. Thus, each pixel of the common template has corresponding spatial coordinates, the triangle of the mesh and barycentric coordinates to the vertices of the corresponding triangle.

All images are warped to this common template. This is done with a piece-wise affine warp that maps each triangle of the mesh from image i to the common template. So, for each image i and its triangle τ, AAM computes the affine transformation that maps coordinates of the vertices of the triangle τ to the coordinates of the corresponding vertices in the common template. Hence, the coordinates of the pixel and the fiducial points $\{\mathbf{u}_{iq}\}_{i=1,q=1}^{I,Q}$ are used to *deterministically* compute the pixel coordinates in the common template.

Once all images are warped to the same template, the vectorized representation of the warped images can be used as input to a subspace model such as PCA. The subspace models the appearance of the object. To fit the learned model to a new image and its set of fiducial points, one must compute the transformation from the common template to the new image.

1.2. C-CCA

Next, we show that AAM is equivalent ot C-CCA when a specific $\phi[\cdot,\cdot]$ is adopted. As in the paper, we define the form of $\phi[\cdot,\cdot]$ as

$$\phi\left[\mathbf{c}_{ij}, \boldsymbol{\theta}_f\right] = \mathbf{a}\left[\mathbf{c}_{ij}\right]^T \boldsymbol{\theta}_f \quad . \tag{1}$$

So, image i has fiducial coordinates $\{\mathbf{u}_{iq}\}_{i=1,q=1}^{I,Q}$. Furthermore, the j-th pixel of image i has a pixel intensity of x_{ij}

and it's x-axis and y-axis coordinates $\mathbf{v}_{ij}$. We define its context vector as:

$$\mathbf{c}_{ij} = [\mathbf{v}_{ij}^T \quad (2)$$
$$\mathbf{u}_{i1}^T \dots \mathbf{u}_{iQ}^T \quad (3)$$
$$\sqrt{|\mathbf{v}_{ij} - \mathbf{u}_{i1}|^2} \dots \sqrt{|\mathbf{v}_{ij} - \mathbf{u}_{iQ}|^2}]^T \quad (4)$$

which is a concatenation of pixel's coordinates $\mathbf{v}_{ij}$ (2 values), the coordinates of the fiducial points ($2 \times Q$ values) and Euclidean distances to all of the fiducial points (Q values).

Let $\boldsymbol{\theta}_f$ be a vector of size $J \times 1$, where J is the number of pixels in the "common template". First, assume that $\mathbf{a}[\mathbf{c}_{ij}]$ returns a vector of size $J \times 1$ that consists of zeros and a single 1.

It follows that using the mean coordinates $\{\bar{\mathbf{u}}_q\}_{q=1}^Q$ and the Delaunay triangulation, one can define such $\mathbf{a}[\mathbf{c}_{ij}]$ that *deterministically* defines which index of the returned vector has value of 1. This effectively maps every pixel of the input image to the common template similarly to AAM.

Indeed, the first $2 + 2 \times Q$ values of $\mathbf{c}_{ij}$ (the pixel's coordinates and the coordinates of the fiducial points) can be used to determine the triangle of the mesh's triangulation that the pixel belongs to. Furthermore, the last Q values of $\mathbf{c}_{ij}$ (Euclidean distances to all of the fiducial points) can be used to compute the barycentric coordinates of the pixel. The barycentric coordinates can be used to deterministically define which of the pixels in the common template x_{ij} corresponds to, which defines which index of the returned vector have value of 1.

Hence, such $\mathbf{a}[\mathbf{c}_{ij}]$ maps images to the common template similarly to AAM. Thus, $\boldsymbol{\theta}_f$ are equivalent to the components of PPCA.

By a similar line of reasoning one can show that Layered AAM [5] is also a special case of C-CCA.

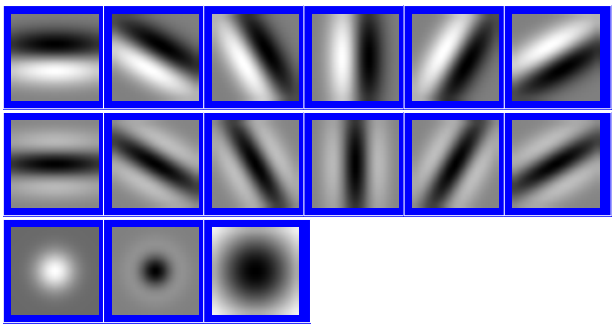
2. Context vectors

As it was mentioned in the paper, the context vectors that we use contain filter responses. For horses, cats and elephants datasets, the filterbank consists of 15 filters (a subset of Leung-Malik Filter Bank [6]), namely the first and second derivatives of Gaussians in 6 orientations, Laplacian of Gaussian at 2 scales and 1 Gaussian filter. For facades, the filterbank consists of 8 Haar-like features at 2 different scales. Figure 2 shows visualizations of the filterbanks.

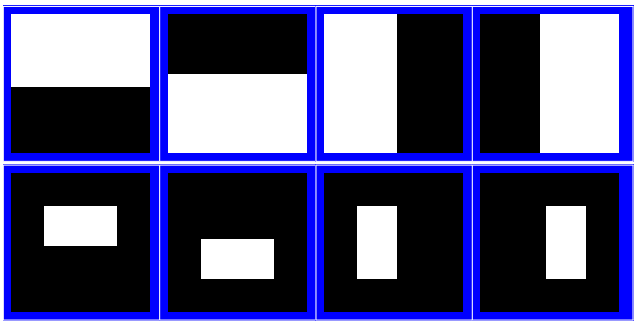
3. Exploring Components

In this section we would like to demonstrate the components that the model learns from the data.

Our model assumes the prior over the hidden variables to be spherical, which is different from PCA, where the components are ranked by the singular value. Thus, for better visualization of the components, we relearned the model with a smaller F. The parameters are in Table 1.



(a)



(b)

Figure 2: Visualization of the filterbanks. (a) Filters used on horses, cats and elephants datasets. (b) Filters used on facades dataset.

Dataset	I	Test I	F	M	K
Horses	200	95	8	3500	16

Table 1: Model Parameters.

We demonstrate the effect of moving along each of the components in the positive and negative directions in Figure 3 and Figure 4. Notice the effect of the same components on different poses: for example, the effect of $\phi[\cdot, \boldsymbol{\theta}_3]$ is the brightness of the head and tail of the horse as shown in Figure 3(f) and Figure 4(f). Similarly, $\phi[\cdot, \boldsymbol{\theta}_4]$ defines the left-right change of the brightness of the horse as show in Figure 3(g) and Figure 4(g).

4. Results

We show figures from the paper in a larger scale, with more examples, and more results.

Figures 6 to 8 show appearance transfer results for horses and cats.

Figures 10 to 13 show structured inpainting results for each of the datasets.

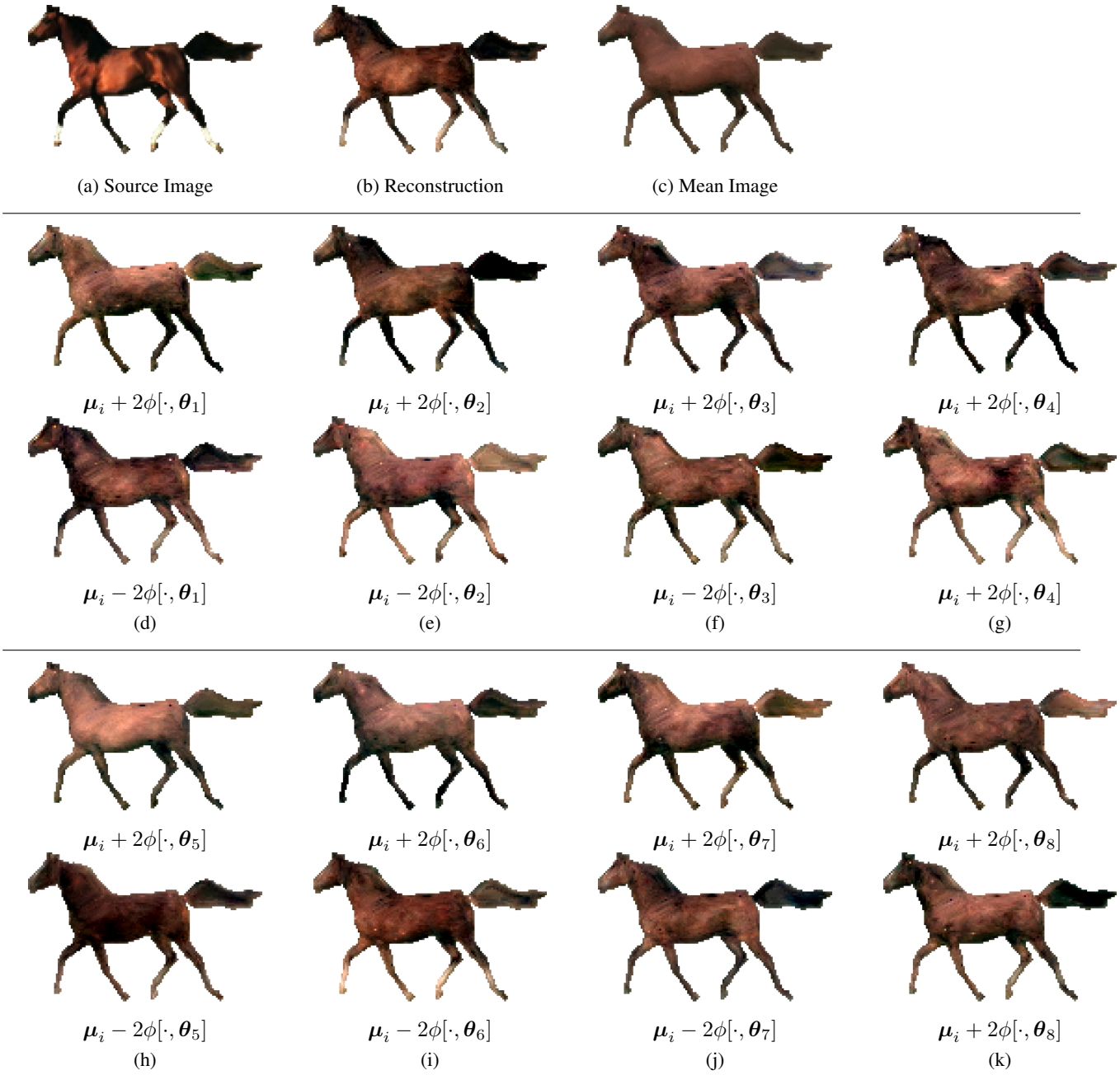


Figure 3: Moving along the components.

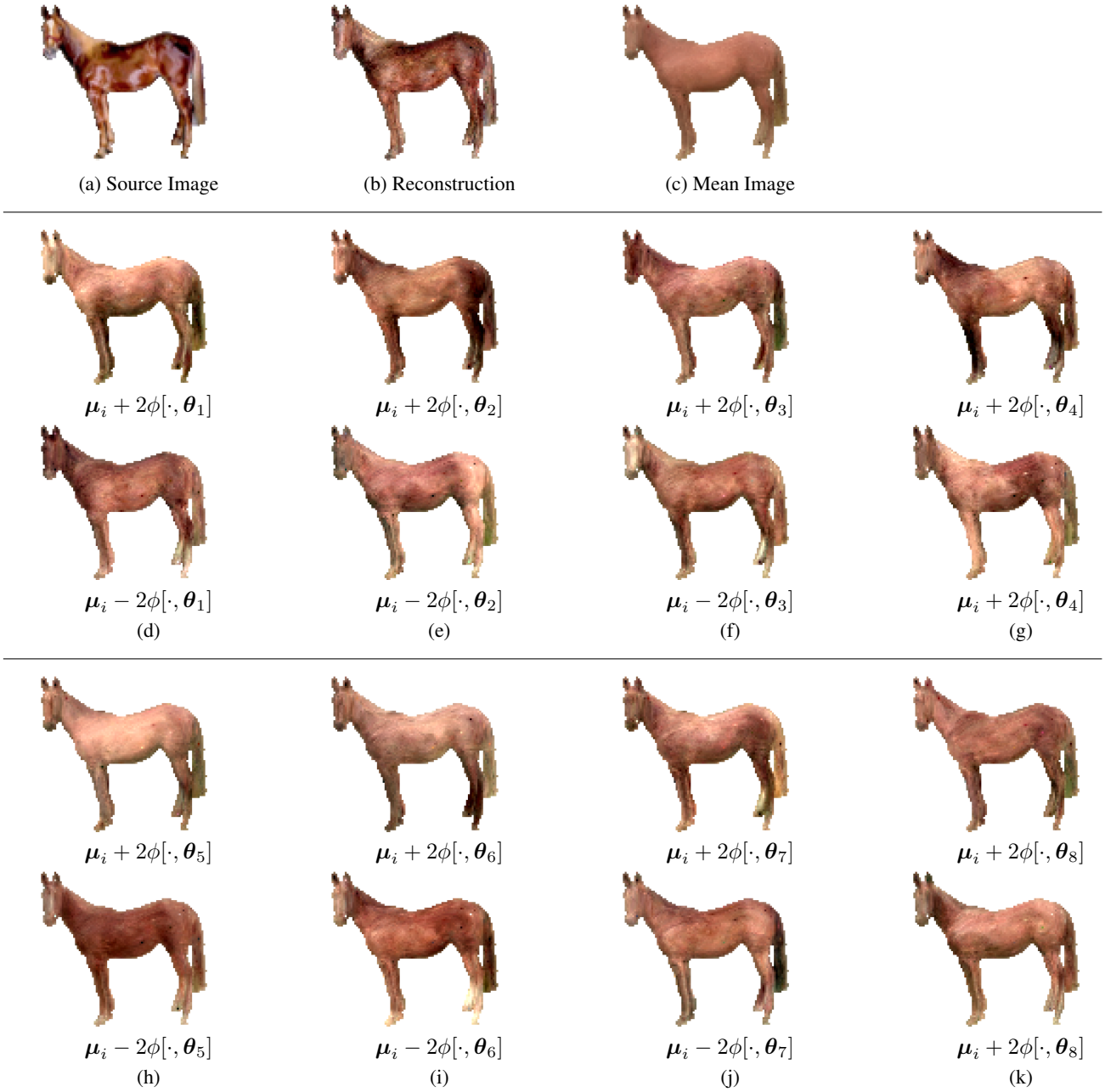
Figure 9 demonstrates dense correspondence computed with SIFT flow [7], one of the techniques for solving image alignment problem. The SIFT flow algorithm performs poorly in these examples for several reasons. First, the appearance in SIFT space may not match due to the difference of textures. More importantly, the assumption of the smoothness of the deformation field doesn't hold for self-occlusions or missing parts, which is especially relevant for the cats example. The problems associated with this assumption also hold for Compositional Model [8].

References

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

[2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Computer Vision – ECCV 1998*, pages 484–498. Springer, 1998. 1

[3] B. Delaunay. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934. 1



(a) Source Image (b) Reconstruction (c) Mean Image

(d) (e) (f) (g)

(h) (i) (j) (k)

Figure 4: Moving along the components.

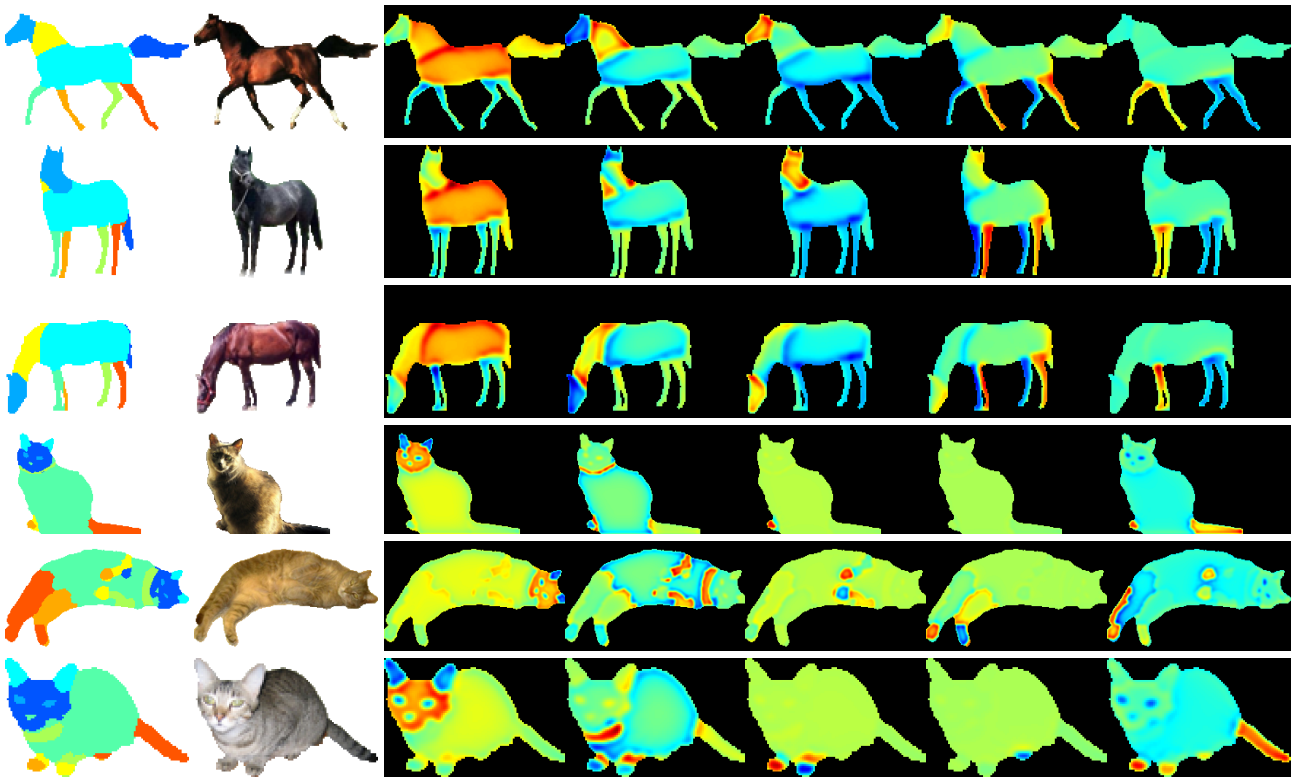
[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html. Accessed: 2014-09-25. 6

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

[6] 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. 2

[7] 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. 3, 7

[8] 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



(a) (b) (c)

Figure 5: 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).

Figure 6: Appearance Transfer Results. (a) A subset of images of the horses training set [1]. (b) Reconstruction of the images using the fitted model. (c) Reconstruction of the images using the fitted model with the fixed function weights h_g, rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ of the first (leftmost) image.

Figure 7: Appearance Transfer Results. Rows top to bottom: (a) A subset of images of the cats training set [4]. (b) Reconstruction of the images using the fitted model. (c) Reconstruction of the images using the fitted model with the fixed function weights h_g, rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ of the first (leftmost) image.

Figure 8: Appearance Transfer Results: For each column, rows top to bottom: (1) An image of the training set. (2-5) Reconstruction of the image of *another* cat in a different pose using the fixed function weights h_g, rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ of the top row image.

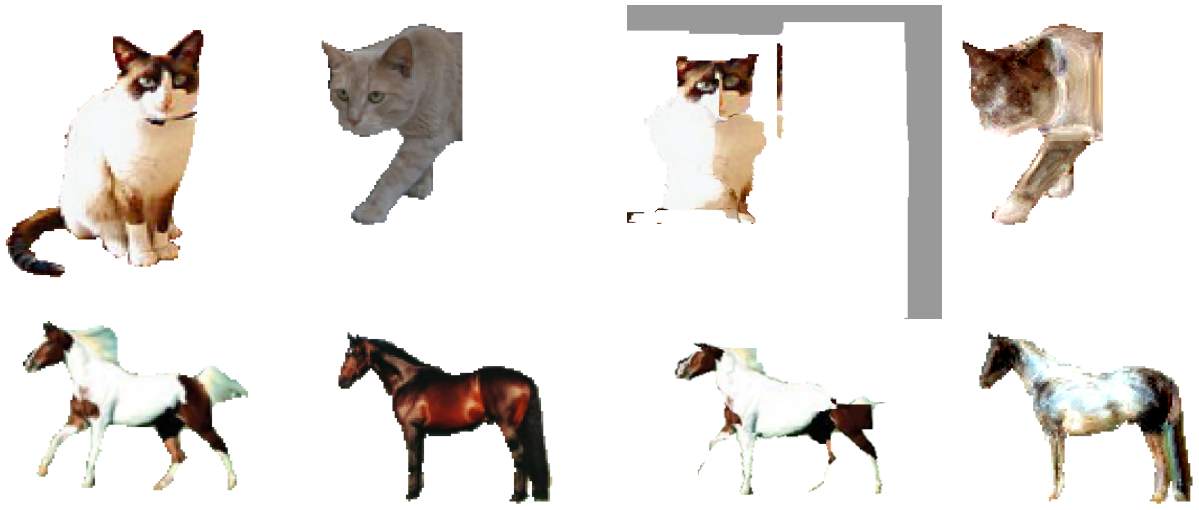


Figure 9: Left to Right: Source image. Target image. The warping of the source image onto the target image using the correspondence computed by SIFT flow [7]. The appearance transfer using C-CCA.

Figure 10: 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. Image from test set. Reconstruction of the image using the fitted model. Input for inpainting (context vectors of all pixels are known). Inpainted result.

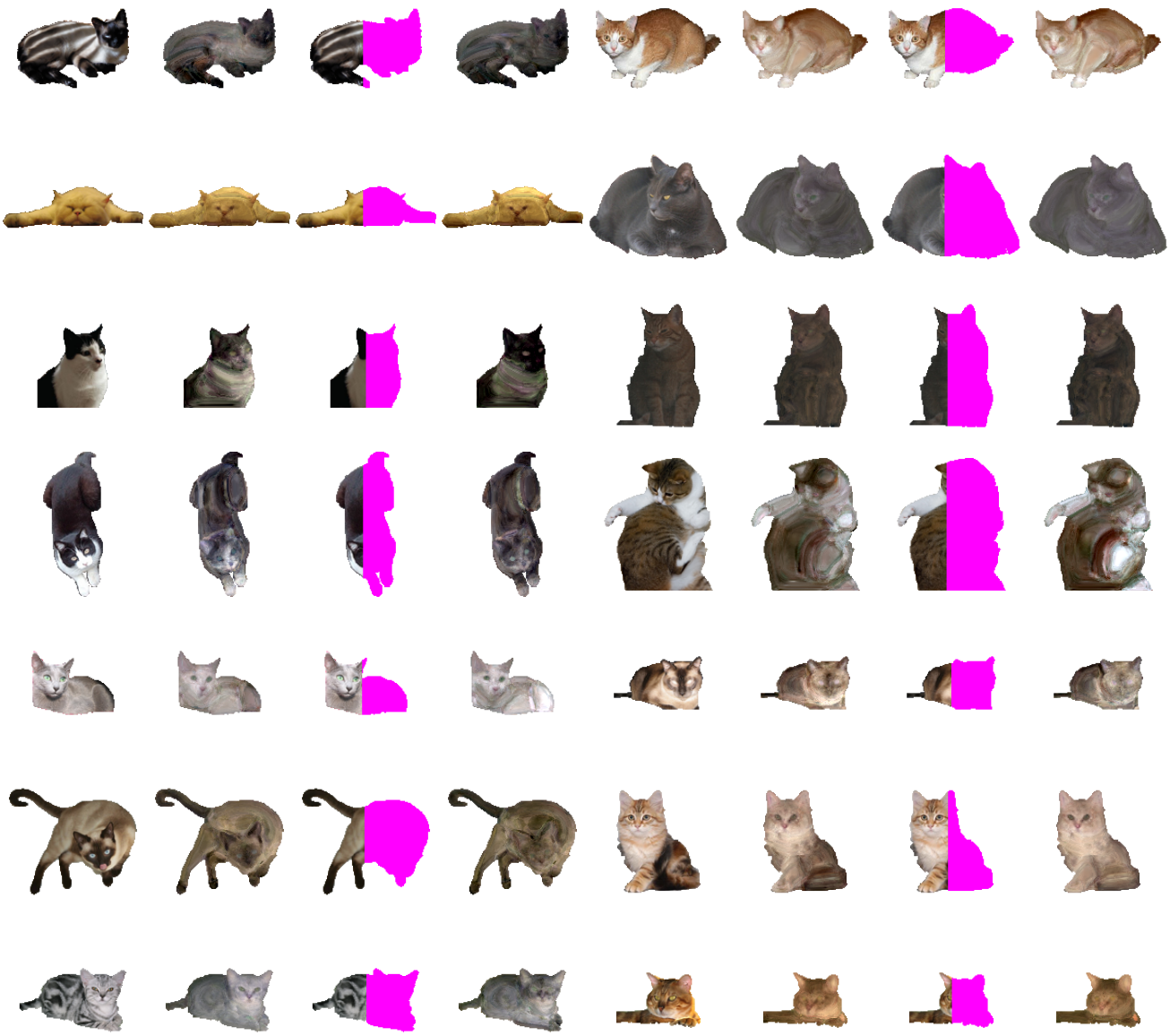


Figure 11: Image Inpainting Results for Cats (Test Set). Left to Right: Image from test set. Reconstruction of the image using the fitted model. Input for inpainting (context vectors of all pixels are known). Inpainted result. Image from test set. Reconstruction of the image using the fitted model. Input for inpainting (context vectors of all pixels are known). Inpainted result.

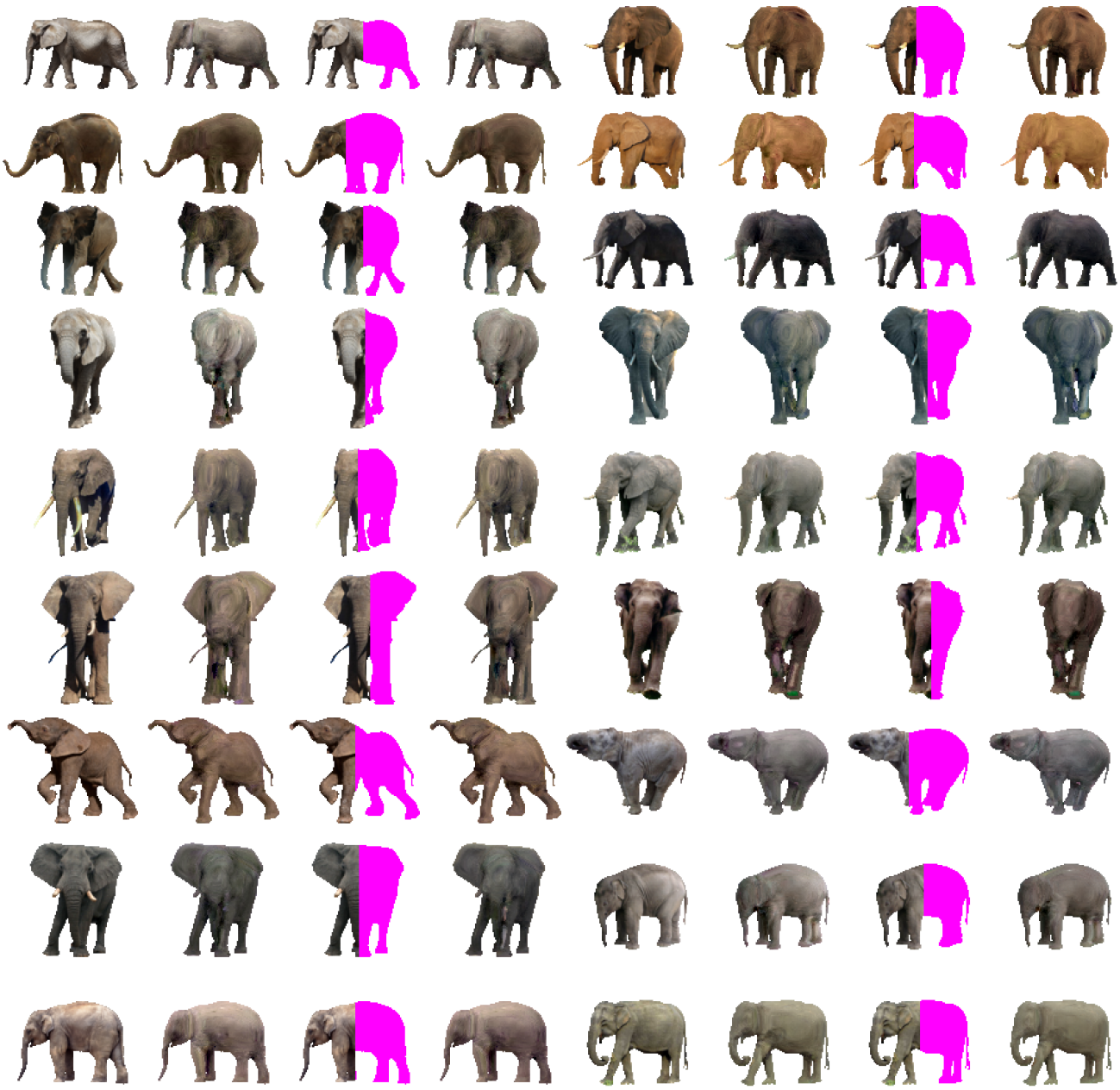


Figure 12: Image Inpainting Results for Elephants (Test Set). Left to Right: Image from test set. Reconstruction of the image using the fitted model. Input for inpainting (context vectors of all pixels are known). Inpainted result. Image from test set. Reconstruction of the image using the fitted model. Input for inpainting (context vectors of all pixels are known). Inpainted result.

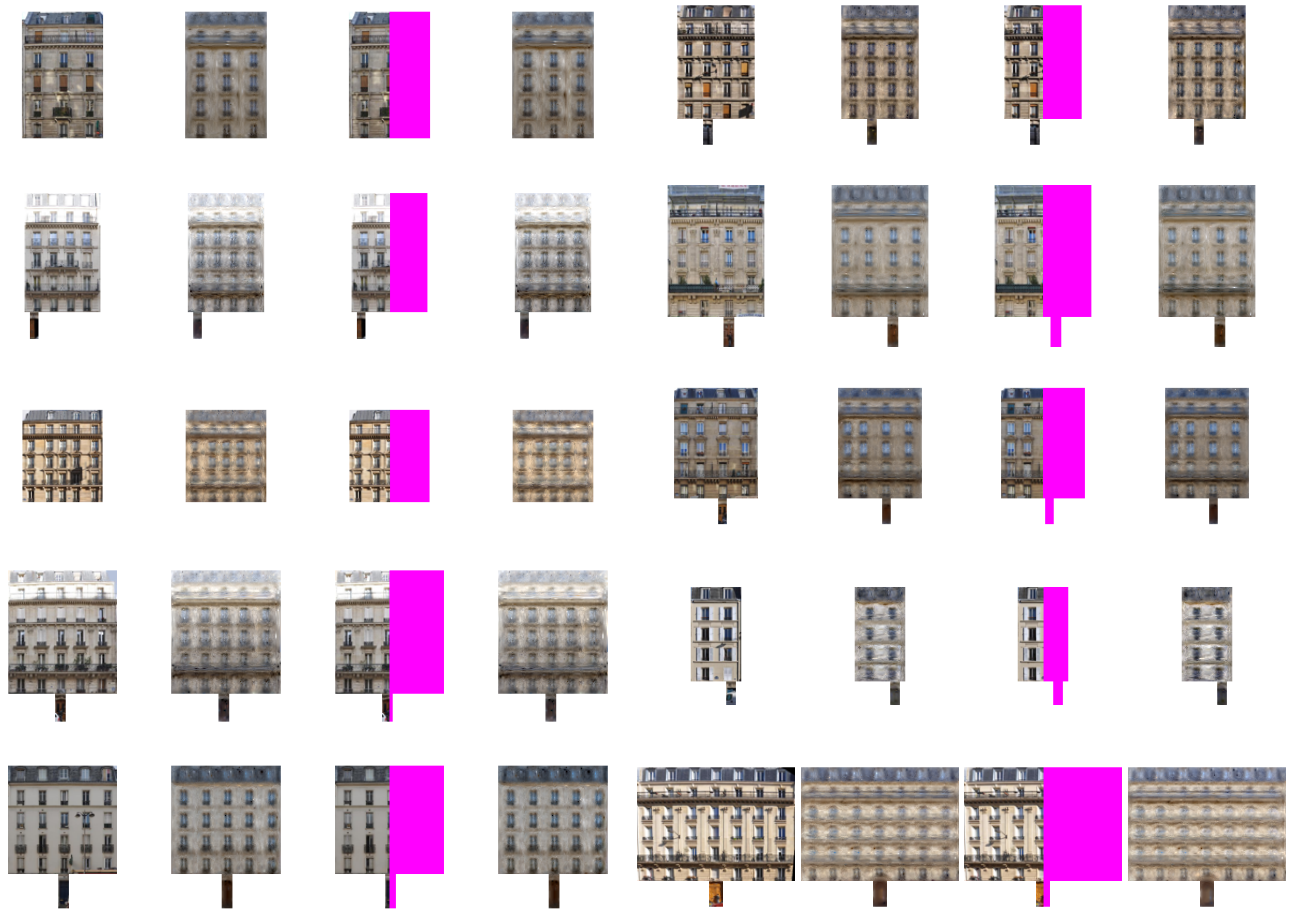


Figure 13: Image Inpainting Results for Facades (Test Set). Left to Right: Image from test set. Reconstruction of the image using the fitted model. Input for inpainting (context vectors of all pixels are known). Inpainted result. Image from test set. Reconstruction of the image using the fitted model. Input for inpainting (context vectors of all pixels are known). Inpainted result.