978-1-4673-1228-8/12/$31.00 ©2012 IEEE

# Unsupervised Learning of Translation Invariant Occlusive Components

Zhenwen Dai
Frankfurt Institute for Advanced Studies,
Dept. of Physics, Goethe-University Frankfurt
dai@fias.uni-frankfurt.de

Jörg Lücke
Frankfurt Institute for Advanced Studies,
Dept. of Physics, Goethe-University Frankfurt
luecke@fias.uni-frankfurt.de

## Abstract

*We study unsupervised learning of occluding objects in images of visual scenes. The derived learning algorithm is based on a probabilistic generative model which parameterizes object shapes, object features and the background. No assumptions are made for the object orders in depth or the objects' planar positions. Parameter optimization is thus subject to the large combinatorics of depth orders and positions. Previous approaches constrained this combinatorics but were still only able to learn a very small number of objects. By applying a novel variational EM approach, we show that even without constraints on the object combinatorics, a relatively large number of objects can be learned. In different numerical experiments, our unsupervised approach extracts explicit object representations with object masks and object features closely aligned with the true objects in the scenes. We investigate the robustness of the approach and the use of the learned representations for inference. Furthermore, we demonstrate generality of the approach by applying it to grayscale images, color-vector images, and Gabor-vector images as well as to motion trajectory data for which the extracted components correspond to motion primitives.*

## 1. Introduction

Visual scenes consist of mutually occluding objects at different positions. A long-standing goal of unsupervised learning on images is to be able to learn object representations from scenes without any label information pertaining to these objects. The problem faced by all the approaches pursuing this goal is the large combinatorics of objects in the images. Potentially, a scene can consist of virtually any number of objects at any planar position and in any orders of depth.

Approaches to learn the low-level statistical properties of images (*e.g.*, sparse coding [1] or ICA and their variants) address the combinatorics of low-level object parts such as edges but usually avoid explicit treatments of edge positions or edge occlusions. Nevertheless, they require approximate learning methods to avoid the still large combinatorics of different edges in an image patch. By using approximations they can, however, learn large numbers of image components in terms of hundreds or even thousands of basis functions (*e.g.*, [2]). Similarly, restricted Boltzmann machines (RBMs, *e.g.*, [3]) have been used to learn image features. Image representations learned without supervision have been shown to be very successful in computer vision, especially if they are combined with other methods [2, 4]. This success is driving further research on unsupervised learning approaches for computer vision.

To learn statistical models of whole visual scenes rather than small patches, similar probabilistic approaches are thus developed and applied increasingly frequently. The required modeling of object occlusions and object positions is considerably more challenging than the task to extract low-level features as it is addressed by sparse coding and ICA. The problem of object occlusions has recently been addressed by modifying and generalizing established approaches: Sparse coding and ICA have thus been generalized to Occlusive Components Analysis (OCA, [5]), while RBMs have been generalized to masked RBMs [6]. By using deterministic approximations to Expectation Maximization (EM) in [5], explicit object representations can be learned from cluttered scenes. By applying sampling in RBMs [6], hierarchical representations that take occlusions into account can be learned. Training both these occlusion models is difficult because occlusions result in a greater combinatorics and typically involve more pronounced local optima than preceding models. Furthermore, OCA and masked RBMs both do not model object positions explicitly – a property they inherit from standard sparse coding and standard RBMs, respectively. In contrast, earlier approaches based on sprites [7, 8] explicitly model object occlusions *and* object translations. Sprite models are generative models of visual scenes allowing for arbitrary planar positions but fixed the depth order (an object always has the same distance from the camera). This restriction of depth order combinatorics allows for a more efficient learning. Still only few objects can be learned because of the additional combinatorics of different translations (usually two or three objects are learned from videos but [8] also report experiments on data with up to five objects). The sprites model has later been extended with linear subspace object representations and a restricted form of depth order inference [9], but still only experiments with up to four objects are reported. In approaches that maintain explicit models of translation invariance but only consider one object per image more objects can be learned: work on transformation-invariant clustering [10] reports applications to up to eight objects (also compare [11]). However, note that the constraint to just one object per image avoids both the combinatorics of object occlusions as well as of different object positions.

In this work, we consider the full combinatorics of planar object positions and depth orders in visual scenes. For modeling and learning, the new approach combines methods of sprite models with translation invariance [7, 8] and of models with unconstrained depth orders [5]. To overcome the combinatorial challenges of the model, we apply a novel variational EM approach for parameter optimization [12] as well as sophisticated annealing schemes to avoid local optima. The derived learning algorithm is hereby designed to be applicable to general feature representations of images, *i.e.*, it is applicable to grayscale and color images as well as to arrays of more complex feature vectors such as Gabor vectors. Furthermore, the generality of the underlying generative model allows for applications to data of domains other than images. We demonstrate the performance of the learning algorithm by applying it to visual scenes of different complexity, and show the generality of the approach by extracting motion primitives from the trajectories of handwritten characters.

## 2. The Generative Model

Our generative model of visual scenes is defined for images in the form of two-dimensional arrays of feature vectors $\vec{y}$. Depending on the application, a feature vector can be a scalar gray value, an RGB color vector, or a more complex feature vector. Given the resolution $(D_1, D_2)$ of an image, the data points are thus of the form $Y = (\vec{y}_{(1,1)}, \ldots, \vec{y}_{(D_1,D_2)})$.

The generative model assumes such images to be composed of possibly $H$ occluding components (objects) at arbitrary planar positions. The object positions are modeled by $h = 1, \ldots, H$ hidden variables $\vec{x}_h \in \{1, \ldots, D_1\} \times \{1, \ldots, D_2\}$. Each object is modeled to appear in an image with probability $\pi \in [0, 1]$. Instead of modeling object presence and absence explicitly, we, for mathematical convenience, assign the special "position" $(-1, -1)$ to all objects which are not chosen to generate the image. Assuming a uniform distribution for the positions, the prior

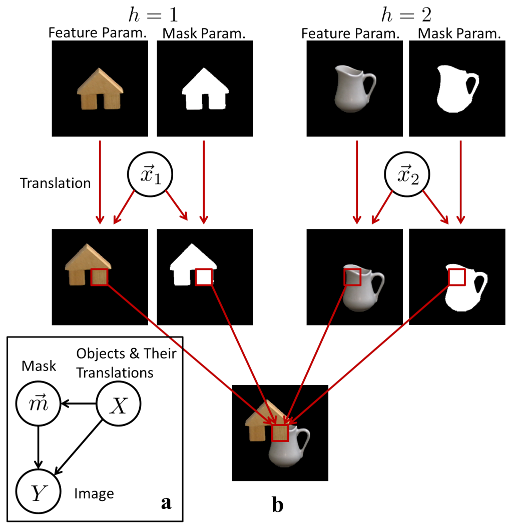

Figure 1. **a** The Graphical Model. **b** An illustration of the image generation procedure.

distribution for objects and their positions is thus given by:

$$p(X|\pi) = \prod_{h=1}^{H} p(\vec{x}_h|\pi), \; p(\vec{x}_h|\pi) = \begin{cases} 1-\pi, & \vec{x}_h = (-1,-1) \\ \frac{\pi}{D_1 D_2}, & \text{otherwise} \end{cases}, \tag{1}$$

where the hidden variable $X = (\vec{x}_1, \ldots, \vec{x}_H)$ contains the information on presence/absence and position of all the image components.

The objects in the model are represented as sets of parameters for object masks and object features. High values of mask parameters $\alpha_{\vec{i}}^{h}$ encode the shape of an object $h$ while the values of feature parameters $\vec{w}_{\vec{i}}^{h}$ encode the values of the object's planar feature arrangement. Mask and feature parameters are visualized for two objects in Fig. 1b (top row). Given translations of the objects, masks and feature parameters are shifted by changing parameters indices w.r.t. cyclic boundary conditions (see Fig. 1, second row), which is given by

$$\vec{d} = (\vec{i} + \vec{x}) := \big((i_1 + x_1) \bmod D_1, (i_2 + x_2) \bmod D_2\big)^T. \tag{2}$$

Such a modulus operation is used because it reduces the translation space of every object by a factor of four.

After translation, two or more objects may occupy the same pixel $\vec{d}$, but only one of them can determine the pixel value due to occlusion (see Fig. 1). We model this property by defining a hidden variable $\vec{m} = (m_{(1,1)}, \ldots, m_{(D_1,D_2)})$. For each pixel $\vec{d}$ of the image, the value of $m_{\vec{d}}$ determines which of the components is used to set the pixel's vectorial value. $\vec{m}$ takes all objects, their positions and their mask parameters into account. We collect all mask parameters into a matrix $A = (\vec{\alpha}^1, \ldots, \vec{\alpha}^H)$ with $\vec{\alpha}^h = (\alpha_{(1,1)}^h, \ldots, \alpha_{(D_1,D_2)}^h)$ and define the set $\Gamma = \{h | \vec{x}_h \neq (-1,-1)\}$ to contain all objects present in the scene. The distribution of $\vec{m}$ is then defined by:

$$p(\vec{m}|X,A) = \prod_{\vec{d}=(1,1)}^{(D_1,D_2)} p(m_{\vec{d}}|X,A),$$
$$p(m_{\vec{d}}|X,A) = \begin{cases} \Lambda, & m_{\vec{d}}=0 \\ (1-\Lambda)\dfrac{\alpha^{m_{\vec{d}}}_{(\vec{d}-\vec{x}_{m_{\vec{d}}})}}{\sum_{h'\in\Gamma}\alpha^{h'}_{(\vec{d}-\vec{x}_{h'})}}, & m_{\vec{d}}\in\Gamma. \end{cases} \quad (3)$$

We will refer to the entries of the variable $\vec{m}$ as *mask variables* because of their close association with the mask parameters. $\Lambda = \prod_{h\in\Gamma}(1-\alpha^h_{(\vec{d}-\vec{x}_h)})$ denotes the probability that the background determines the pixel value (in this case $m_{\vec{d}}=0$). The mask variable $m_{\vec{d}}$ is assigned to component $h$ ($m_{\vec{d}}=h$) with a high likelihood if the translated mask parameter of this component is high at $\vec{d}$. Note that the considered mask parameters are normalized according to all other objects in $\Gamma$, *i.e.*, all other objects in the image. The value of each $m_{\vec{d}}$ is thus chosen according to the mixture distribution defined by the mask parameters and the background in (3). This is similar to [7, 8] with the difference that our used mask variable is not assuming the same depth order of objects for all images. Modeling occlusion using the mixture approach in (3) represents an approximation but inference and learning become more efficient than iterating through all depth orders (compare [5]).

After drawing object presence/absence, object positions and the mask variables, the image is finally generated using background and object features. Based on a Gaussian noise model, the distribution of visual scenes is given by:

$$p(Y|\vec{m},X,\Theta) = \prod_{\vec{d}=(1,1)}^{(D_1,D_2)} p(\vec{y}_{\vec{d}}\,|\,m_{\vec{d}},X,\Theta),$$
$$p(\vec{y}_{\vec{d}}|m_{\vec{d}},X,\Theta) = \begin{cases} \mathcal{N}(\vec{y}_{\vec{d}};\,B,\sigma_B^2), & m_{\vec{d}}=0 \\ \mathcal{N}(\vec{y}_{\vec{d}};\,\vec{w}^{m_{\vec{d}}}_{(\vec{d}-\vec{x}_{m_{\vec{d}}})},\sigma^2), & m_{\vec{d}}\in\Gamma, \end{cases} \quad (4)$$

where $\sigma^2$ is the variance of all the component features. The background distribution is a Gaussian distribution with mean $B$ and variance $\sigma_B^2$. Equations 1 to 4 define our generative model. Mask parameters of the objects will be denoted by $A$ as defined above and, analogously, their features will be denoted by $W = (W^1,\ldots,W^H)$ with $W^h = (\vec{w}^h_{(1,1)},\ldots,\vec{w}^h_{(D_1,D_2)})$. The set of all model parameters is denoted by $\Theta$.

## 3. Likelihood Optimization

Given the dataset $\mathcal{Y} = \{Y^{(1)},\ldots,Y^{(N)}\}$, unsupervised learning can be formulated as finding the best model parameters w.r.t. the data likelihood $L = p(Y^{(1)},\ldots,Y^{(N)}|\Theta)$. Following the Expectation Maximization (EM) approach, we derived parameter update rules by setting the derivatives of the model's free energy w.r.t. the parameters to zero. For the feature parameters $W$, we obtain:

$$\vec{w}^h_{\vec{i}} = \frac{\sum_n\sum_X q_2^{(n)}(X)q_1^{(n)}(m_{(\vec{i}+\vec{x}_h)}\!=\!h;X)\vec{y}^{(n)}_{(\vec{i}+\vec{x}_h)}}{\sum_n\sum_X q_2^{(n)}(X)q_1^{(n)}(m_{(\vec{i}+\vec{x}_h)}\!=\!h;X)}, \quad (5)$$

with the abbreviations: $q_2^{(n)}(X) = p(X|Y^{(n)},\Theta)$ and $q_1^{(n)}(m_{\vec{d}};X) = p(m_{\vec{d}}|Y^{(n)},X,\Theta)$. For the mask parameters $A$, a straightforward derivation is unfortunately not possible and with usual approximations it is difficult to keep $A$ in the interval $(0,1)$. Instead, we define the mask parameter as the average of the mask posteriors,

$$\alpha^h_{\vec{i}} \approx \frac{1}{N}\sum_n\sum_X q_2^{(n)}(X)q_1^{(n)}(m_{(\vec{i}+\vec{x}_h)}\!=\!h;X). \quad (6)$$

This definition is inspired by the update rules of the mask variable in fore/background models (see [13]), but has here been extended to handle multiple occluding objects without a predefined order.

For computational efficiency, we decompose the very large joint space of $\vec{m}$ and $X$ by exploiting the standard assumption of independent observed variables (compare, *e.g.*, [7, 8]):

$$p(\vec{m},X|Y,\Theta) = p(X|Y,\Theta)\prod_{\vec{d}=(1,1)}^{(D_1,D_2)} p(m_{\vec{d}}|Y,X,\Theta). \quad (7)$$

It follows that the update equations can be computed with the posterior over individual pixels,

$$p(m_{\vec{d}}|Y,X,\Theta) = \frac{p(Y,m_{\vec{d}}|X,\Theta)}{\sum_{m'_{\vec{d}}\in\Gamma\cup\{0\}}p(Y,m'_{\vec{d}}|X,\Theta)}, \quad (8)$$

and the posterior over the translation variable $X$,

$$p(X|Y,\Theta) \propto \Big(\prod_{\vec{d}=(1,1)}^{(D_1,D_2)}\sum_{m_{\vec{d}}\in\Gamma\cup\{0\}} p(Y,m_{\vec{d}}|X,\Theta)\Big)p(X|\Theta). \quad (9)$$

**Variational EM Optimization.** With the above posterior equations, parameter optimization can be performed using EM without approximations. However, the computation is intractable for anything else than very small-scale problems because the computational complexity of $q_2^{(n)}(X)$, $\mathcal{O}\big((D_1D_2+1)^H\big)$, increases exponentially with the number of components. Therefore, we apply a novel variational approximation to EM in the form of Expectation Truncation [12]. Expectation Truncation has the benefit of being very efficient while maintaining rich posterior representations. With the variational approximation the posterior is not factored but truncated. For our model it is given by:

$$p(X|Y^{(n)},\Theta) \approx \frac{p(X,Y^{(n)}|\Theta)}{\sum_{X\in\mathcal{K}_n}p(X,Y^{(n)}|\Theta)}, \text{if } X\in\mathcal{K}_n, \quad (10)$$

and zero otherwise. High precision is achieved if the set $\mathcal{K}_n$ contains those configurations of $X$ with most posterior mass. As the number of objects in an image is usually small compared to the total number of objects, we can expect relatively small sets of $\mathcal{K}_n$ to finally provide good approximations. To find sets $\mathcal{K}_n$ we, in practice, define a selection function assigning a score to every translation of every component,

$$\mathcal{S}_{\Theta}^{(n)}(\vec{x}^h) = \prod_{\vec{i}\in\mathcal{P}_h} \mathcal{N}(\vec{y}^{(n)}_{(\vec{i}+\vec{x}_h)}; \vec{w}^h_{\vec{i}}, \sigma^2), \forall h \in \{1,\ldots,H\}. \tag{11}$$

A selection can be computed efficiently as it only considers $D_1 D_2 H$ cases. For each component $h$, $\mathcal{P}_h$ denote the indices of the reliable pixels which are defined as those with the $\lambda$ highest mask parameters $\alpha^h_{\vec{i}}$. To construct $\mathcal{K}_n$, we first select the components and their translations with the $S$ highest scores of the selection function. Then we define $\mathcal{K}_n$ to contain all combinations of the selected translations of all the selected components (plus the conditions of the components not being in the image). We define $S$ as the largest $S$ with which $\mathcal{K}_n$ has less than or equal to $K$ entries. Note that the accuracy of this approximation is determined by approximation parameters $\lambda$ and $K$. The larger $\lambda$ and $K$ the higher is the accuracy, however, the computational cost is also higher.

**Directional Annealing.** The convergence to local optima is a commonly encountered problem in many optimization tasks. Deterministic Annealing (DA) is a widely used technique, which has demonstrated substantial performance improvement by avoiding local optima for a variety of supervised and unsupervised learning methods [14]. It can be smoothly integrated into EM algorithms with some small modifications on posterior calculations [15]. However, in numerical experiments on our model as well as other translation-invariant models, we found that the conventional DA requires substantial modifications to remain applicable. The reason is that the updated parameters are the sum of data points weighted by the posteriors (see Eqn. 5). Considering the posterior of the translation variable, the update parameters (*e.g.* $W$) can be viewed as a convolution of a filter with the data points (images), where the posterior is the filter. Therefore, when conventional DA smooths the posteriors, the posterior of the translations performs like a smoothing filter. As a consequence, the feature parameters are smoothed into a uniform value after few EM iterations. Overly strong smoothing can be avoided by careful tuning of the annealing temperature to a low value. However, at such values annealing usually becomes ineffective.

Taking these observations into account, we consider a novel type of annealing which avoids the over-smoothing effect also at high temperatures. The approach separates the hidden variables into two groups by whether they contain translations or not, and only performs annealing on the marginal posterior distribution of the group not containing translations. More specifically, the hidden variable $X$ in our model encodes the existence of components and the translations of the existing components. Following this approach,

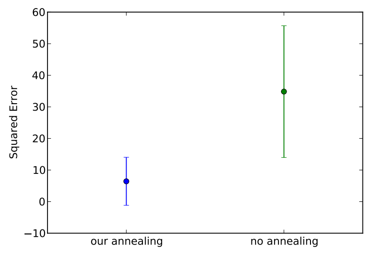

Figure 2. The comparison of the learning performance with and without our annealing algorithm. The y-axis shows the squared error between the learned mean feature $W$ and the ground truth.

we rewrite $X$ as a pair of hidden variables $(\vec{s}, Z)$, where the binary vector $\vec{s}$ defines whether each component exists and $Z$ gives the translations of the existing components. As stated before, annealing is only applied to the marginal posterior distribution of $\vec{s}$:

$$\tilde{p}(\vec{s}\,|Y,\Theta) = \frac{p(\vec{s}|Y,\Theta)^{\beta}}{\sum_{\vec{s}'} p(\vec{s}'|Y,\Theta)^{\beta}}, \;\; \tilde{p}(Z\,|\vec{s},Y,\Theta) = p(Z\,|\vec{s},Y,\Theta), \tag{12}$$

where $\beta = 1/T$ and $T$ is the annealing temperature. Substituting (12) into the joint posterior of $(\vec{s}, Z)$, we obtain:

$$\begin{aligned}\tilde{p}(\vec{s}, Z|Y,\Theta) &= \tilde{p}(\vec{s}\,|Y,\Theta)\cdot\tilde{p}(Z\,|\vec{s},Y,\Theta)\\ &= \frac{p(\vec{s}|Y,\Theta)^{\beta}}{\sum_{\vec{s}'} p(\vec{s}'|Y,\Theta)^{\beta}}\cdot\frac{p(\vec{s},Z|Y,\Theta)}{p(\vec{s}|Y,\Theta)}.\end{aligned} \tag{13}$$

The performance of our annealing algorithm is evaluated quantitatively on synthetic data (see Sec. 4). We ran our learning algorithm 10 times with and without annealing separately, while keeping other configurations the same. The performance is evaluated by computing the mean squared error between the learned features and their ground truth. The squared error is only computed for the pixels of learned components with their mask parameters greater than or equal to $0.5$, because only the pixels with high mask parameters reliably represent the components (see Fig. 3b). Due to the absence of supervision information, it is unknown which learned component corresponds to which ground truth component with what translation. Therefore, the squared error is computed for the best matched components with the corresponding translations. Fig. 2 shows the comparison. The results show that directional annealing more efficiently avoids local optima and thus addresses the important problem of more pronounced local optima in models of occlusion.

## 4. Experimental Results

We applied the derived algorithm to different types of data in numerical experiments. We will start with visual scene where ground truth is available then go to increasingly challenging settings and finally apply the approach to data from a domain other than images. All runs of the algorithm were executed on a GPU cluster of 12 GTX480 cards.

**Visual Scenes of COIL Objects.** We first demonstrated the performance of our algorithm on data generated using objects from the COIL-100 image dataset [16]. 16 different objects were selected, downscaled to $10 \times 10$ pixels and

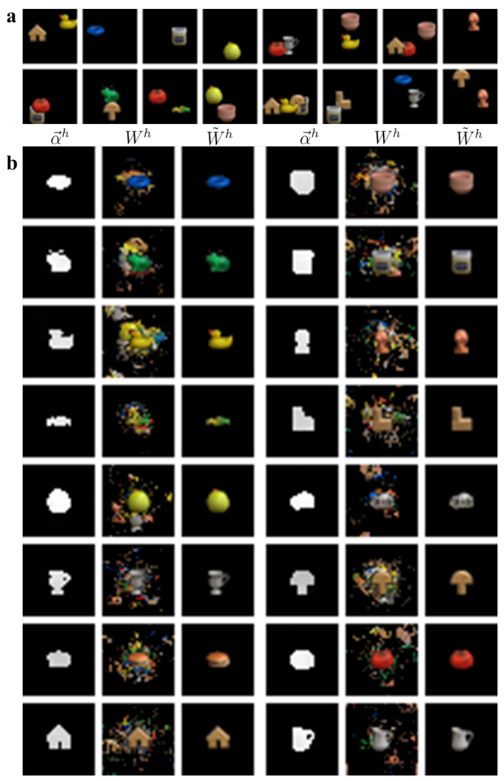

Figure 3. **a** 16 out of 2000 of the used data set. **b** Learned parameters. 1st column: mask parameters $\vec{\alpha}^h$. 2nd column: mean feature parameters $W^h$. 3rd column: mean features for mask parameters $\geq 0.5$.

segmented out from the black background. An image was generated by randomly selecting a subset of the 16 objects, where each object has a probability of 0.2 to appear. The appearing objects were placed at random positions on a $30 \times 30$ black image. When overlapping, the objects occluded each other with a different random depth order for each image. We generated $N = 2000$ such images in total and Fig. 3a shows some typical examples.

Our algorithm was applied with $H = 16$ components and a Gaussian background distribution with zero mean and variance $\sigma_B^2 = 0.001^2$. The initial features $W$ were set to randomly selected data points. The initial mask parameters $A$ were independently and uniformly drawn from the interval $(0, 1)$ for the area where the difference between the model features and the background features were larger than 0.1. The initial temperature for annealing was set to $T = 50$. After keeping it constant for 10 iterations, the temperature linearly decreased to 1 within 100 iterations. For the robustness of learning, the feature variance $\sigma^2$ decreased together with the temperature from $0.3^2$ to $0.02^2$. The algorithm terminated when the temperature was equal to 1 and the difference of the data log-likelihood of two consecutive iterations was sufficiently small (less than 0.01%). Learning was performed with the approximation parameters $\lambda = 30$, $K = 10,000$. Parameter optimization took 40 minutes, approximately. Fig. 3b shows the resulting parameters for object masks and object features. As can be observed, all 16 generating data components were recovered. Note that the feature parameters show non-zero values outside the region of high mask values. Their relevance is negligible, however, as only the features for high mask values significantly effect the likelihood (see third column in Fig. 3b). In different runs of the algorithm with different initial parameter values, the algorithm recovered all objects in about half of the runs (see Fig. 2).

These results demonstrate that efficient learning is possible for objects at arbitrary positions and with arbitrary depth ordering. Previous results did not report more than six or ten [5] objects even though depth order [8] or position combinatorics [5] was constrained. No results for more than very few objects (two to four) were reported for the unstrained case because the combinatorics scales super-exponentially with the number of components. Besides the speed-up provided by GPU parallelization, the algorithmic advancements of our approach are the key of conquering the super-exponential complexity of the combinatorics.

**Video Data.** After having verified parameter recovery of the algorithm on ground truth data, we then applied the approach to data of real visual scenes. We used two video sequences of objects placed in front of a camera. Both videos contained objects changing their positions and occluding each other if overlapping each other (see Fig. 4a for some sample frames and Supplement for the full sequence). The resolution of both videos was $320 \times 240$ and images for training were extracted at a rate of three frames per second. For more robustness, each extracted image was represented by Gabor feature vectors (one scales, eight orientations, see Supplement for details) instead of RGB colors.

From the first video 523 images were extracted and represented by a grid of $40 \times 30$ Gabor vectors. Our algorithm was applied with $H = 5$ components and the object appearance frequency of $\pi = 0.2$. The mean of the Gaussian background distribution was set to the first image and its variance to $\sigma_B^2 = 0.04^2$. The initial parameters and the annealing scheme were the same as in the synthetic experiment, and the feature variance decreased from $0.1^2$ to $0.06^2$. By using approximation parameters of $K = 200$ and $\lambda = 100$, parameter optimization required 6.2 minutes. Fig. 4b (left two columns) shows the learned parameters for masks and features by heat maps. As can be observed, the *box* has been represented by the first component and the *mug* by the fourth. The *stapler* has been represented twice, by the second and the third component. The last component has captured some background features. Note that the stapler representations are associated with two different scales as in the video it is indeed placed at different depth. As an example application of the learned representation, we show in Fig. 4c

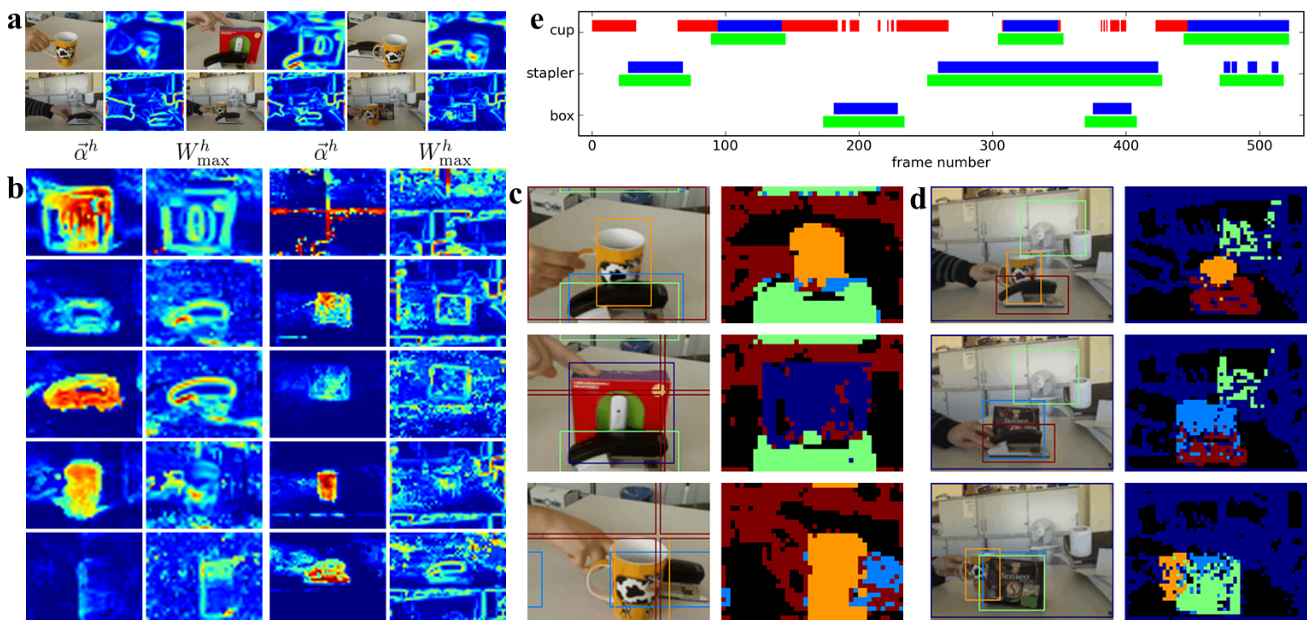

Figure 4. **a** 3 frames from each video: 1st row for Video 1, 2nd row for Video 2. The heat maps show the max of Gabor responses ($y_{\vec{d},\max} = \max_f y_{\vec{d},f}$). **b** Learned parameters: left two columns for Video 1 and right two columns for Video 2. **c&d** The inference result for some frames. The left column indicates the detection of components (every component a color). The right column denotes the inferred mask variable (black for background). **e** A measure of accuracy for Video 1 (blue for detected at correct positions, red for detected at wrong positions, green for the ground truth). The (precision, recall) percentages: cup (45%,90%), stapler (100%,82%), box (100%,77%).

the inferred object classes and positions (left column), and the inferred mask values for each Gabor grid point (right column). As can be observed, object classes and positions are correctly inferred, and the inferred mask variable gives an appropriate estimate on the regions occupied by the objects. Note that our model does not use pixel proximity but its explicit object representation that is learned statistically across the sequence. The model is thus providing image information complementary, *e.g.*, to segmentation algorithm. To give a quantitative measure of performance, the accuracy of detection is evaluated by manually checking every frame of the video (see Fig. 4e and Suppl. for details).

To provide an increased challenge for the algorithm, the objects used in the second video were smaller, the light was dimmer, and the background was more cluttered, on 444 frames. This time, we compute Gabor features on a finer grid of resolution $54 \times 40$. On these data the learning algorithm with $H = 6$ components and approximation parameters $K = 200$ and $\lambda = 50$ required 8.3 minutes for parameter optimization. Fig. 4b (right two columns) again shows the learned object parameters. This time the used box object is represented by two components while the first component represents part of the background. Fig. 4d shows inferred object classes and positions as well as the inferred values of the mask variable. A decreased continuity of the mask values is evidence of the increase difficulty of the task. However, mask values still show appropriate regions for the different objects. The top two images hereby show an example of false positive detections where part of the background gets associated with the box object. Such false positives could, however, be detected using post-processing based on other cues (*e.g.* continuity in time).

**Motion Trajectories.** Modeling motion primitives has been addressed in neuroscience and robotics in the context of motor synergies [17]. Most approaches of motion primitive learning either only extract static primitives from an individual time step [18], or do a linear subspace analysis on pre-segmented motion clips [19], of which the results are strongly influenced by the segmentation points. Some recent works aim to learn linearly superimposed motion primitives with various time spans and flexible activation points by nonnegative matrix factorization [20], factorial hidden markov model [21] and multivariate orthogonal matching pursuit [22]. Besides linear superimposed motion primitives, which are suitable in the context of limb control where the accuracy of path is more concerned, there is another type of motion primitives where the global motion switches among motion primitives sequentially. Namely, at every time step only one motion primitive exclusively controls the global motion. Such primitive types are useful for global motion planning where computation is a major concern, *e.g.* robotic planning and controlling [23, 24]. However, there has been few works about learning such exclusive motion primitives without pre-segmentation. By applying our approach to motion trajectory data, we found that it is able to extract motion primitives as the components of

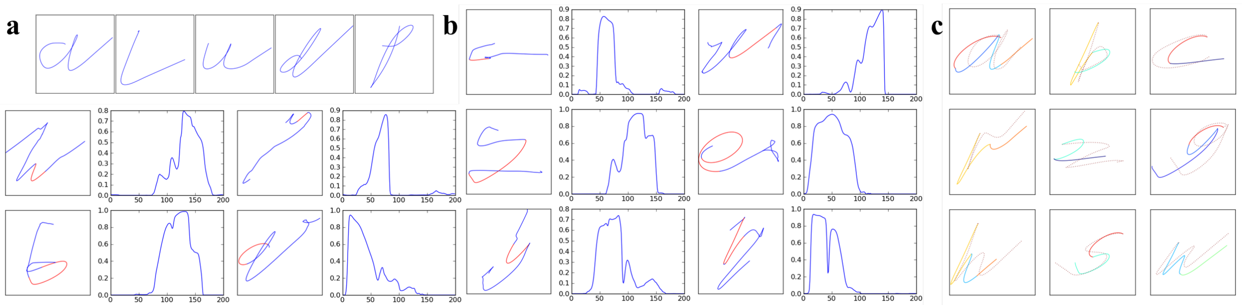

Figure 5. **a** 5 samples of character trajectories, generated by integrating the velocities over time. **b** Learned parameters. The 1st&3rd columns show the integrated trajectories (red segments denote where its mask parameter is $\geq 0.5$). The 2nd&4th columns show the mask parameters. **c** Some reconstruction results. The solid curves indicate the reconstruction results (every component a color) and the dotted curves show the original trajectories.

data. As demonstrated in the following experiment, without changing the model or imposing sparsity constraints on temporal domain, motion primitives naturally arise as a result of optimization.

We applied our model to handwriting trajectory data [25]. The data was captured using a WACOM tablet at 200Hz. It has 2858 characters in total and each character is a 3-dimension time series consisting of the 2-dimension pen velocity and the pen tip force. For simplicity, we ignored the tip force dimension and aligned the time series to 200 time steps by truncating the longer ones and padding shorter ones with zeros. Fig. 5a shows 5 samples of the character trajectories by integrating the velocities over time. Our algorithm was applied with $H = 10$ components. The Gaussian background distribution was set to zero mean and variance $\sigma_B^2 = 0.1^2$. The initial parameters and the annealing scheme had the same setting as previous two experiments except the initial annealing temperature was $T = 20$. The feature variance decreased from $2.0^2$ to $1.0^2$. Fig. 5b shows the learned component parameters. As the data are time series, the mask parameters are shown as curves along the time line. Note that the learned mask parameters are only high over a small part of the time series and low elsewhere. The trajectories over those time segments (see the red segments in Fig. 5b) are examples of motion primitives which are used to construct most of the characters. To show this, we reconstructed the data points using the MAP estimates of hidden variables. The MAP estimates determine the most likely component and translation for each time step of a given datapoint. Using the MAP estimates we reconstructed data points by replacing, at each time step, the velocity by the mean velocity from the corresponding component. The reconstruction results of some characters are shown in Fig. 5c (see supplement for more examples). Note that the velocity integration is very sensitive to small differences. Such differences can, over time, cause the integrated curve to divert significantly from the original. Importantly, however, the structures of the reconstructed characters remained very similar to the original and reconstructions were comprised of several motion primitives.

## 5. Discussion

We have studied unsupervised learning from visual scenes with arbitrary object positions and orders in depth. Our approach is based on a probabilistic generative model with hidden variables for the planar positions of the components and a selection of one component per pixel through a mixture distribution. A computationally tractable learning algorithm was derived by applying a truncated variational EM approach [12]. In numerical experiments we have shown that representations of objects can be learned in the form of object masks and object features. By applying the approach to motion trajectory data, we have, furthermore, demonstrated the applicability to other data domains.

As for earlier approaches such as sprite models [7, 8] or occlusive components analysis (OCA, [5]), the hidden variables and parameters in our approach directly reflect the properties of observed data: the mask and feature parameters reflect object shapes and appearances, and the hidden variables explicitly encode the component positions and their responsibilities for the pixels in a given image. Along with OCA and sprites models, our approach thus follows one of the original goals of generative modeling: deriving learning algorithms based on the models replicating the true generating process as explicitly as possible. Other approaches follow alternative paths: Boltzmann machines map learning of visual properties to learning across hierarchical stages [3, 6] while other models, *e.g.*, use a condensed image representation, an epitome, for learning and inference on images. In RBMs and epitome models, image generation is more indirectly associated with the physical generation of images, which can have positive and negative consequences often depending on the addressed task. Epitome models or RBMs have thus been applied very successfully, *e.g.*, to fill-in missing image parts while the more direct sprites models or OCA provide explicit representations of individual objects, which can directly be used, *e.g.*, for object detection, removal, position inference *etc*. Still other approaches learn object shapes and appearances using bilinear models (*e.g.*, [26]), which can indirectly address the invariance problem by learning different component styles. However, for mathematical convenience, components in these models are assumed to combine linearly – an assumption that can not capture occlusion of objects. Layered models have also been explored in the literature of video layer decomposition. Recent works [27, 28, 29] usually assume manual initialization, while some earlier works of flow-based layered models [30, 31] do not require manual initialization but use fixed depth orders.

All the approaches discussed above will exhibit their full potential if they are combined with other technology for computer vision, similar to, *e.g.*, sparse coding approaches which recently have been widely used in computer vision applications. Because of the large complexity of visual scenes compared to patches, much more experience has to be gathered in this line of research, however. In addition to object positions and depths, in principle, other object transformations such as object scaling or rotation, as well as illumination changes, could be formulated by a generative model. In general, the more properties of visual scenes are modeled the more powerful the model becomes. The major difficulty remains, however, practicality and computational tractability. In the numerical experiments for our approach we have shown that, compared to earlier approaches, relatively large numbers of objects/components can be learned efficiently without imposing constraints on the position or object depth combinatorics. Our approach thus goes significantly beyond the scene complexities that could previously be addressed by similar approaches.

**Acknowledgement.** This work was funded by the German Research Foundation (DFG) under grant LU 1196/4-1.

## References

[1] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996. 1

[2] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *NIPS*, pp. 801–808, 2007. 1

[3] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comp*, vol. 18, pp. 1527–1554, 2006. 1, 7

[4] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *ISCAS*, pp. 253 – 256, 2010. 1

[5] J. Lücke, R. Turner, M. Sahani, and M. Henniges, "Occlusive Components Analysis," in *NIPS 22*, pp. 1069–1077, 2009. 1, 2, 3, 5, 7

[6] N. L. Roux, N. Heess, J. Shotton, and J. Winn, "Learning a generative model of images by factoring appearance and shape," *Neural Comp*, vol. 23, pp. 593–650, 2011. 1, 7

[7] N. Jojic and B. J. Frey, "Learning flexible sprites in video layers," *CVPR*, 2001. 1, 2, 3, 7

[8] C. K. I. Williams and M. K. Titsias, "Greedy learning of multiple objects in images using robust statistics and factorial learning," *Neural Comp*, vol. 16, pp. 1039–1062, 2004. 1, 2, 3, 5, 7

[9] B. J. Frey, N. Jojic, and A. Kannan, "Learning appearance and transparency manifolds of occluded objects in layers," in *CVPR*, 2003. 2

[10] B. J. Frey and N. Jojic, "Transformation-invariant clustering using the em algorithm," *PAMI*, vol. 25, pp. 1–17, 2003. 2

[11] A. Kannan, N. Jojic, and B. J. Frey, "Fast Transformation-Invariant Component Analysis," *IJCV*, vol. 77, pp. 87–101, 2007. 2

[12] J. Lücke and J. Eggert, "Expectation truncation and the benefits of preselection in training generative models," *JMLR*, vol. 11, pp. 2855 – 2900, 2010. 2, 3, 7

[13] B. J. Frey and N. Jojic, "A comparison of algorithms for inference and learning in probabilistic graphical models.," *PAMI*, vol. 27, pp. 1392–1416, 2005. 3

[14] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE*, vol. 86, pp. 2210–2239, 1998. 4

[15] N. Ueda and R. Nakano, "Deterministic annealing EM algorithm," *Neural Networks*, vol. 11, pp. 271–82, Mar. 1998. 4

[16] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-100)," tech. rep., CUCS-006-96, 1996. 4

[17] Z. Ghahramani, "Building blocks of movement," *Nature*, 2000. 6

[18] E. Todorov and Z. Ghahramani, "Analysis of the synergies underlying complex hand manipulation," in *EMBC*, vol. 6, pp. 4637–40, 2004. 6

[19] A. Fod, M. Matarić, and O. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, pp. 39–54, 2002. 6

[20] A. D'Avella and E. Bizzi, "Shared and specific muscle synergies in natural motor behaviors.," *PNAS*, vol. 102, pp. 3076–81, 2005. 6

[21] B. H. Williams, *Extracting Motion Primitives from Natural Handwriting Data*. PhD thesis, 2008. 6

[22] Q. Barthélemy, A. Larue, A. Mayoue, D. Mercier, and J. I. Mars, "Shift & 2D Rotation Invariant Sparse Coding for Multivariate Signals," *TSP*, vol. 60, no. 4, pp. 1597–1611, 2012. 6

[23] D. D. Vecchio, R. M. Murray, and P. Perona, "Decomposition of human motion into dynamics-based primitives with application to drawing tasks," *Automatica*, 2003. 6

[24] D. C. Conner, H. Choset, and A. Rizzi, "Integrated planning and control for convex-bodied nonholonomic systems using local feedback control policies," in *Robotics: Science and Systems II*, 2006. 6

[25] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. 7

[26] J. B. Tenenbaum and W. T. Freeman, "Separating style and content with bilinear models.," *Neural Comp*, vol. 12, pp. 1247–83, 2000. 8

[27] J. D. Jackson, A. J. Yezzi, and S. Soatto, "Dynamic Shape and Appearance Modeling via Moving and Deforming Layers," *IJCV*, vol. 79, no. 1, pp. 71–84, 2008. 8

[28] C. Wang, M. D. L. Gorce, and N. Paragios, "Segmentation , ordering and multi-object tracking using graphical models," in *ICCV*, pp. 747–754, 2009. 8

[29] T. Schoenemann and D. Cremers, "High Resolution Motion Layer Decomposition using Dual-space Graph Cuts," in *CVPR*, 2008. 8

[30] A. D. Jepson, D. J. Fleet, and M. J. Black, "A Layered Motion Representation with Occlusion and Compact Spatial Support," in *ECCV*, vol. 1, pp. 692–706, 2002. 8

[31] J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *TIP*, vol. 3, no. 5, pp. 625–38, 1994. 8