Unsupervised Learning of Object Deformation Models

Iasonas Kokkinos *
Department of Statistics, UCLA
jkokkin@stat.ucla.edu

Alan Yuille
Department of Statistics, UCLA
yuille@stat.ucla.edu

Abstract

The aim of this work is to learn generative models of object deformations in an unsupervised manner. Initially, we introduce an Expectation Maximization approach to estimate a linear basis for deformations by maximizing the likelihood of the training set under an Active Appearance Model (AAM). This approach is shown to successfully capture the global shape variations of objects like faces, cars and hands. However the AAM representation cannot deal with articulated objects, like cows and horses. We therefore extend our approach to a representation that allows for multiple parts with the relationships between them modeled by a Markov Random Field (MRF). Finally, we propose an algorithm for efficiently performing inference on part-based MRF object models by speeding up the estimation of observation potentials. We use manually collected landmarks to compare the alternative models and quantify learning performance.

1. Introduction

In this paper we pursue the learning of probabilistic models for object categories using minimal supervision. There is important need for such methods to facilitate the application of computer vision to large scale problems such as detecting large numbers of objects in images. Recent work has shown the practicality of learning models of object categories using sparse features [24], demonstrating good performance for tasks like object recognition and localization. However, the use of sparse features means that: (i) the models only exploit a limited amount of the image information, and (ii) the models are not suited for tasks such as top-down segmentation, tracking, identification, etc.

There have been a variety of attempts for learning dense models of objects. Some notable examples include [27, 14, 9, 25], as well as the Active Appearance Model (AAM) learning literature, e.g. [23, 4, 1]. But all of these have some limitations. For example, clean segmentation maps are required in [27], a movie of the deforming object is needed in [14], while other approaches [9, 25] model the object deformations using generic, smoothness constraints instead of learning the statistics of the object deformations.

We learn a dense representation of the object by formulating the task as parameter estimation for a generative model. There are two different types of unknowns: (a) the model parameters which characterize the possible deformations of the object category, and (b) the deformation variables which specify the shape of each object example. We apply the EM algorithm [6], where the M-step estimates the model parameters and the E-step takes the expectation with respect to the deformation variables, which are treated as hidden variables. Our minimal supervision consists in requiring that the object is present in the image with roughly constant position and scale but with variable background.

The image representation we use is based on the edge and ridge primal sketch of Lindeberg [15]. This contains useful shape-related information about the object as edges are typically active along object boundaries and ridges indicate the symmetry axes of objects. This representation can enable both object segmentation and detection, e.g. [13] but has not yet been used for unsupervised learning. Further, it is largely insensitive to appearance variation, thereby allowing us to focus on learning deformations. Finally, the edge and ridge curves can be treated as sets of points; this allows us to use the clustering procedure of Mean-Shift [2] for two crucial problems, namely AAM initialization and finding the object parts.

In Section 2 we first apply our strategy to deformable objects without articulated parts, which we represent by AAM's. We apply the EM algorithm, where the M-step estimates the deformation basis elements of the AAM while the E-step takes the expectation with respect to the AAM expansion coefficients. AAM's are good models for certain types of objects and, as we will show, learning AAM's is useful to initialize the learning of more complex models.

In Section 3 we turn to the more challenging task of learning articulated objects, and represent the object by deformable parts with relationships modeled by Markov Random Fields (MRF's). Our strategy is to first model the object by a single AAM, which we learn as described above, and use it as initialization for learning a more complex model containing multiple parts, which are obtained auto-

*This work was supported by a Keck Foundation Grant and NSF Grant 0413214.

matically. Our algorithm proceeds to learn their appearances and spatial relationships, by extending the EM algorithm to estimate the deformations of the parts (E-step) and the MRF model which describes the spatial relations between them (M-step). Finally, we propose an algorithm for efficiently performing inference on part-based MRF object models by speeding up the estimation of observation potentials.

We validate the merit of our approach by learning models for a range of different objects, including e.g. cars, hands and cows. We provide systematic results using as ground truth manually collected landmarks that we make publicly available.

2. Learning AAMs

2.1. Previous Work

The AAM representation [3, 23] captures the variability of a deformable object in terms of *shape* variations and *appearance* variations. Both are expressed by expansions on a linear basis, but their combination yields a nonlinear model. The appearance $\mathcal{T}(\mathbf{x})$ is synthesized on a deformation-free ('template') grid and is then warped onto the image coordinate system, using a deformation field $\mathbf{S}$:

$$\mathbf{S}(\mathbf{x};\mathbf{s}) \equiv (S_x(\mathbf{x};\mathbf{s}), S_y(\mathbf{x};\mathbf{s})) = \sum_{i=1}^{N_S} \mathbf{s}_i \mathcal{S}_i(\mathbf{x}), \qquad (1)$$

where $\mathbf{s}$ are the shape coefficients, and $\mathcal{S}_1, \ldots, \mathcal{S}_{N_S}$ are the shape basis elements. The deformation $\mathbf{S}$ brings the image pixel $(x + S_x(\mathbf{x};\mathbf{s}), y + S_y(\mathbf{x};\mathbf{s}))$ in registration with the template pixel $\mathbf{x} = (x, y)$, where its appearance $\mathcal{T}(\mathbf{x})$ is predicted.

If the parameters of the AAM are known (i.e. $\mathcal{T}(\mathbf{x})$ and $\{\mathcal{S}_i(\mathbf{x}) : i = 1, ..., N_S\}$), then we can fit an input image to the model by minimizing a least squares criterion $E(\mathbf{s})$ with respect to $\{\mathbf{s}_i : i = 1, ..., N_S\}$, where:

$$E(\mathbf{s}) = \sum_{\mathbf{x}} \left(I(\mathbf{S}(\mathbf{x};\mathbf{s})) - \mathcal{T}(\mathbf{x})\right)^2 \qquad (2)$$

It has been observed [1] that this criterion can be linked to the Transformed Components Analysis (TCA) method [8] by interpreting the criterion as the negative log-likelihood of a probability model.

Even though the literature on AAM fitting is well-developed, the task of learning an AAM is less explored. In the seminal work of [23] a bootstrapping method was devised for learning AAMs by iteratively fitting the images with the model and then updating the AAM model using optical flow. However, the optical flow calculation is nontrivial and is not guaranteed to decrease some global goodness of fit criterion. An elegant formulation for AAM learning is proposed in [1] which is also intuitively similar to EM, but the procedure used to estimate the deformation eigenspace is different. First deformation fields are estimated separately for each image and then they are projected on a PCA basis; this is not guaranteed to decrease their goodness of fit criterion monotonically and as we show, one can directly minimize this criterion w.r.t. the shape basis elements.

In [4], the authors minimize an information-theoretic criterion using diffeomorphisms in conjunction with an MDL term to enforce simplicity of the learned model. However, the resulting model is not of the typical AAM form, since the deformations generated even by a simple model do not necessarily lie on a low-dimensional linear space.

2.2. Learning Linear Deformation Models with EM

We now address the learning of AAM's by our EM strategy. We formulate this problem as probabilistic estimation using a generative model for the data images $\{I_\mu\}$:

$$P(I_\mu|\mathcal{S}, \mathbf{s}^\mu, \mathcal{T}) \propto \exp -\frac{1}{\sigma^2} \sum_{\mathbf{x}} \left(I_\mu(\mathbf{S}(\mathbf{x};\mathbf{s})) - \mathcal{T}(\mathbf{x})\right)^2 \qquad (3)$$

$$P(\mathbf{s}^\mu|\sigma) \propto \exp -\lambda \sum_i \frac{(\mathbf{s}_i^\mu)^2}{\sigma_i^2}, \qquad (4)$$

where $\mathcal{T}$ is the appearance model, $\mathbf{S}(\mathbf{x};\mathbf{s}) = \sum_{i=1}^{N_S} \mathbf{s}_i^\mu \mathcal{S}_i(\mathbf{x})$ as in (1), $\{\mathcal{S}_i\}$ are the deformation basis elements, $\{\mathbf{s}_i^\mu\}$ are the shape coefficients, and σ is the assumed noise variance. The deformation variables are assumed to be drawn from the Gaussian prior of (4), where the parameters $\{\sigma_i\}$ are the variances of the expansion coefficients and λ is a design parameter, determining the tradeoff between data fidelity and the prior.

Given a dataset of images $\{I_\mu : \mu = 1, ..., N\}$, and a set of shape coefficients $\mathbf{s}^\mu$, the observation likelihood is given by $\prod_{\mu=1}^{N} \sum_{\mathbf{s}^\mu} P(I_\mu|\mathcal{S}, \mathbf{s}^\mu, \mathcal{T}) P(\mathbf{s}^\mu|\sigma)$. Inspired from [18] we treat the shape coefficients as hidden variables, so that the EM algorithm [6] can be applied to find the parameter estimates $\mathcal{T}, \{\mathcal{S}_i\}$ that lie on a local maximum of this expression.

Specifically one can formulate the EM algorithm in terms of minimizing a free energy function $F[\{\mathcal{S}_i\}, \mathcal{T}; Q]$ with respect to both $\{\mathcal{S}_i\}, \mathcal{T}$ and Q [16], with $Q(.) = \prod_\mu Q_\mu(.)$ where $Q_\mu(.)$ is an unknown probability distribution on the shape coefficients $\mathbf{s}^\mu$. The negative of the free energy is:

$$\sum_{\mu=1}^{N} \int_{\mathbf{s}^\mu} Q_\mu(\mathbf{s}^\mu) \log P(I_\mu|\mathcal{S}\mathbf{s}^\mu, \mathcal{T}) P(\mathbf{s}^\mu|\sigma) + \sum_{\mu=1}^{N} H(Q_\mu),$$

where $H(Q_\mu) = \int_{\mathbf{s}} Q_\mu(\mathbf{s}) \log Q_\mu(\mathbf{s})$ is the negative entropy of the distribution Q_μ.

We restrict the family of distributions to which $Q_\mu(.)$ belongs to be of the form: $Q_\mu(\mathbf{s}) = \delta(\mathbf{s} - \mathbf{s}_\mu)$, where $\mathbf{s}_\mu$ is a parameter vector, and $\delta(.)$ is the delta function.

The M-step, minimizing $F[\mathcal{S}, \mathcal{T}, \sigma; Q]$ with respect to $\mathcal{S}, \mathcal{T}, \sigma$ with Q fixed, then leads to minimizing:

$$\sum_{\mu=1}^{N} \left[\sum_{\mathbf{x}} \left[I_\mu(S(\mathbf{x}; \mathbf{s}^\mu)) - \mathcal{T}(\mathbf{x}) \right]^2 + \lambda \sum_j \frac{(\mathbf{s}_j^\mu)^2}{\sigma_j^2} \right]. \quad (5)$$

We minimize with respect to $\mathcal{S}$ by steepest descent, taking the derivative with respect to the i-th basis element at location $\mathbf{x}$, $(\mathcal{S}_{x,i}(\mathbf{x}), \mathcal{S}_{y,i}(\mathbf{x}))$, to obtain the update rule

$$dS_i = -\left(\frac{\partial E}{\partial \mathcal{S}_{x,i}(\mathbf{x})}, \frac{\partial E}{\partial \mathcal{S}_{y,i}(\mathbf{x})} \right), \quad \text{where} \quad (6)$$

$$\frac{\partial E}{\partial \mathcal{S}_{\cdot,i}(\mathbf{x})} = \sum_{\mu=1}^{N} \mathbf{s}_i^\mu \left. \frac{\partial I}{\partial \cdot} \right|_{S^\mu} \left[I_\mu(S(\mathbf{x}; \mathbf{s}_\mu)) - \mathcal{T}_\mu(\mathbf{x}) \right] \quad (7)$$

Above $\left. \frac{\partial I}{\partial \cdot} \right|_{S_\mu}$ denotes the derivative of I along dimension $\cdot$ after warping I to the template grid using $S(\mathbf{x}; \mathbf{s}_\mu)$; the update step is estimated using line search.

The minimization with respect to $\mathcal{T}$ yields:

$$\mathcal{T}(\mathbf{x}) = \frac{1}{N} \sum_{\mu=1}^{N} I_\mu(\sum_{i=1}^{N_\mathcal{S}} \mathbf{s}_i^\mu \mathcal{S}_i(\mathbf{x})). \quad (8)$$

As mentioned, we discard a significant part of appearance variation by using the edge and ridge maps of [15] instead of the image intensity. The appearance model thus uses only the mean template instead of a linear expansion, as in typical AAMs. However, the EM approach can also be applied to learning an AAM which models the image intensity.

The E-step corresponds to finding the best fit for the shape coefficients in terms of the current estimates for $\mathcal{S}$ and $\mathcal{T}$. This is done using typical AAM fitting [3]. The initialization is performed by setting $\mathcal{T}$ to be the average of the set of data images; initial estimates for $\mathcal{S}$ are obtained as described below.

2.3. Extensions and Practical Issues

We now present refinements that make the EM algorithm effective for this task, namely (i) guaranteeing that the deformations do not cause the disappearance of template features (ii) using mean-shift to initialize the basis elements.

2.3.1 Feature Transport PDE

Non-trivial deformations result in local contractions and expansions. These naturally capture object scalings but can have a negative side effect, namely making object features disappear or inflate. For this reason we want the deformation fields to have zero acceleration in the direction perpendicular to image features; this guarantees that the features are only 'transported', without being distorted.

This requirement can be phrased as follows: consider a deformation field $\mathbf{h} = (h_x, h_y) = (x + g_x, y + g_x)$; g_x and g_y are the deformation increments calculated from the PCA synthesis. This deformation field moves features along orientation n_x, n_y by $g_x n_x + g_y n_y$; if this term is constant it means that the motion of features in this orientation is purely transporting them. Our constraint thus requires that the directional derivative of this function equals zero, i.e.

$$\partial_x \left(g_x n_x + g_y n_y \right) n_x + \partial_y \left(g_x n_x + g_y n_y \right) n_y = 0. \quad (9)$$

and can be imposed by projecting the available (g_x, g_y) fields onto the closest deformation field (f_x, f_y) satisfying (9).

In the supplemental material we use calculus of variations to prove that this projection f of g can be obtained by numerically solving the following PDE with respect to λ:

$$\sum_{i,j=(1,2)} n_i n_j \left[\partial_{1,j}(\lambda n_1 n_i) + \partial_{2,j}(\lambda n_2 n_i) \right] = \sum_{i,j=(1,2)} \partial_j f_i n_i n_j$$

and then setting $f_i = g_i - \frac{\partial \lambda}{\partial x_i}, i = 1, 2$. Above we replace the x, y indexes with $1, 2$ for notational simplicity.

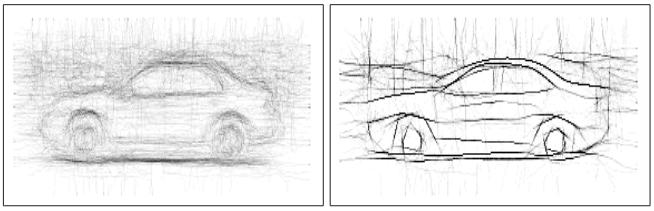
It is convenient that linear expansions are used to synthesize any object deformation: constraining all shape basis elements to have zero acceleration automatically guarantees that this holds also for any synthesized deformation. We thus solve this problem by updating the basis elements according to (7) and then projecting them onto the space of functions as above.

We note that this solution works because the edge and ridge maps are thin structures, obtained by smoothing the maxima contours of [15] with a small Gaussian. This allows the locally accurate estimation of orientation and makes the requirement of feature transport meaningful.

2.3.2 Basis Initialization using Mean Shift

As in [23] for algorithm stability we use a basis pursuit-type algorithm which introduces basis elements iteratively. At each iteration a new basis element is introduced into the model and then the EM learning loop is iterated until convergence. The new basis element $\mathcal{S}_i$ should be in a direction that improves the registration of the training images. To achieve this, for example in [23] optical flow is used to align each image separately with the template image, followed by PCA to yield a new basis element.

Instead, we exploit the versatile nature of the primal sketch representation. Apart from 2-D images, the primal sketch provides us with 1-D curves, which can in turn be represented as sets of points. We can then see the task of registering the image sketches as clustering these points onto sharp, template contours. For this we use the Mean Shift algorithm [2] which is a non-parametric clustering algorithm that is ideally suited for our task.



Initial After Mean Shift

Figure 1: Using Mean-Shift Clustering for aligning sketch contours. Left: superimposed edge contours from the whole car training set, as aligned by the previous AAM learning iteration. Right: after 6 iterations of Mean-Shift clustering.

Specifically, the previous iteration's AAM deformation estimates align to some extent the edge and ridge contours of the initial images. After this alignment, we take the points lying on these contours and represent them by a feature vector (x, y, θ, s), which encodes their position (x, y) their orientation estimate θ, obtained from the local curve structure, and their width s, estimated by the method of [15]. Ridges and edges are then separately clustered using Mean Shift. An essential modification is that we restrict the motion of points so that they can only move perpendicular to the curves they are on. This is required to prevent the curves from contracting to points.

The output of this stage is a set of thin contours as shown in Fig. 1. The motion of the sketch points of each training image thus provides an estimate of the deformation increment. We extend these increments to the rest of the image domain to obtain a dense deformation field and use PCA over the whole training set to derive the new deformation basis element.

Summarizing, the location of the previous material in the procedure that produces the basis elements is given by the following pseudocode:

$\mathcal{S} = \mathrm{BASIS}(I)$	$\mathcal{S} = \mathrm{EM}(\mathcal{S}, I)$
for $i = 1$ to $N_{\mathcal{S}}$ **do**	**repeat**
Basis pursuit:	E-step:
Introduce $\mathcal{S}_i$ (2.3.2)	$\mathbf{s} = \mathrm{AAM\_FIT}(\mathcal{S}, I)$ [3]
EM update:	M-step:
$\mathcal{S}_{1\ldots,i} \Leftarrow \mathrm{EM}(\mathcal{S}_{1\ldots i}, I)$	$\mathcal{S}$=UPDATE$(\mathcal{S}, \mathbf{s}, I)$ (2.2)
end for	$\mathcal{S}$=PROJECT$(\mathcal{S})$ (2.3.1)
	until convergence

2.4. Experimental Results on Learning the AAM

Our main interest has been to apply our framework to real, noisy, unsegmented images and see whether shape information can be extracted without manual annotation. We have therefore applied the AAM learning algorithm described above to five object categories, namely Caltech faces, UIUC cars, IMM hands, ETH cows and Weizmann horses.

In order to systematize our evaluation we have manually placed landmarks on 50 images of each object category; we make the annotations available from our website.

To evaluate the performance of AAM fitting, we examine how close they are brought to each other by the AAM learning. Several measures for evaluating learning performance have been proposed, e.g. in [5, 19] but for simplicity we use the following procedure. First, we backward wrap the images to the template coordinate system, using the AAM-based deformations. Then we estimate the covariance matrix C_i for each object landmark i, and use $\sqrt{|C_i^{\frac{1}{2}}|}$ as a measure of registration quality, as it is coarsely proportional to the radius of a circle enclosing the points. For brevity we call this measure Radius from Covariance Determinant (RCD).

In Fig. 2 we plot the values of RCD for three different scenarios: (i) AAM with 7 basis elements trained with ground truth deformations (ii) AAM utilizing only translation (iii) AAM with 5 learned basis elements. The performance of the first is used as a reference to assess the difficulty of the considered dataset. As we see, the results of (iii) are closing the gap between (i) and (ii), while for the easier classes, namely cars, faces and hands, (i) and (iii) are very close. We also observe, as expected, that the results for cows and horses are poor at specific landmarks, which correspond to the feet locations.

We visually verify the registration of the data in Fig. 2, by comparing the mean feature maps $\mathcal{T}(\mathbf{x})$ obtained by using only translation (mid-row) and the deformation basis learnt by our approach (bottom-row). The templates result from averaging the backward-wrapped edge and ridge maps, where the deformations are estimated by the AAM. We observe that the average back-projections using our model are significantly sharper, indicating that the model has captured the shape variations better. But for complex categories like cows or horses the averaged back-projections are fuzzy at complex areas, like feet.

Summing up, the EM algorithm is successful for learning the AAM for certain types of objects. The results are good for cars, faces, and hands, but are weaker for articulated objects like cows and horses, for which the AAM representation is inherently inadequate.

3. Learning Part-Based Models

We now proceed to the more challenging task of learning part-based models of articulated motion. AAM's are poorly suited to modeling articulated objects because global eigenvectors cannot easily capture the variety of local deformations than can occur, for example, at the legs of a cow. Moreover, parts of articulated objects can become self-occluded and AAM's are not designed to deal with this either.

Instead we use a part-based representation of deformable objects, that models motion separately within each part, us-

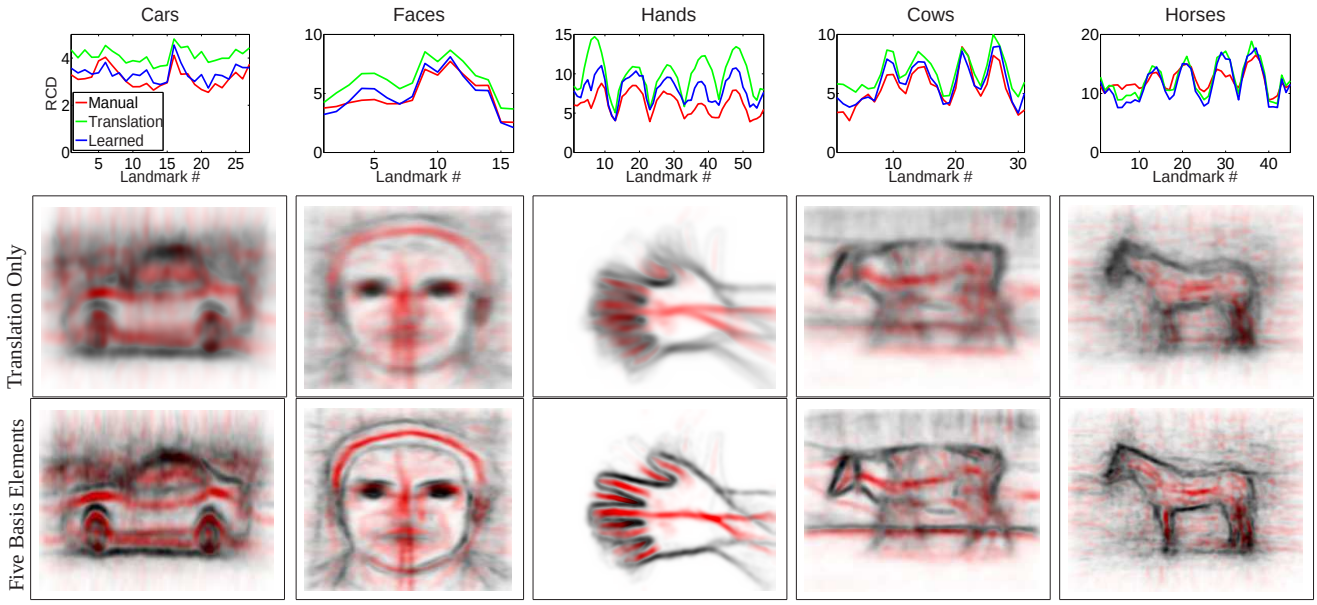


Figure 2: AAM Learning performance: On the top row we compare the Radius of Covariance Determinant (RCD) measure for three different registration scenarios: using an AAM trained with ground truth landmarks (red), an AAM accounting only for translation (green) and an AAM using the learned deformation basis (blue) - less is better. The learned AAM exhibits almost equally good performance as the manually trained one. Bottom rows: the improvement of the estimated deformations using the shape model leads to sharper template features: starting from a messy average, a clean 'mean sketch' is obtained where symmetry axes and structures like reveals object parts like eyes, lips, car wheels, finger tips, etc. Edges are shown in black and ridges in red. (***Most improvements are only visible in color***)

ing a Markov Random Field (MRF) model to enforce the consistency of the estimated deformations. At the high level we proceed along the same lines as in the previous part of the paper. We learn the parameters of the deformable part model with an EM strategy by iteratively matching the model to the dataset images (E-step) and then estimating its parameters (M-step).

To the best of our knowledge this is a problem that has not been dealt with previously. Other work, e.g. [7, 20, 22], requires manual annotation of training data sets, which has constrained the application of such models to a few important applications such as hand or person tracking. In [14] a movie of a deforming object is used to determine its rigidly moving parts, and an additional system is used to estimate an MRF model for its parts. Using motion information simplifies the problem, since it gives reliable information about the position of the boundary and it also means that only small deformations must be estimated for each time step. The object variation is also only learned from a specific member of the object category. By contrast, our approach uses images from different members of the same category that do not have to be in related, consecutive poses.

3.1. Part-based Models of Deformations

Part-based deformable models have become popular in the articulated object detection/tracking literature e.g. [7, 20, 26], as composite motions can be explained in terms of simpler motions of the object parts. For example in [7, 26], the deformation of each part is described in terms of translation, rotation, and scaling along either of its two axes; in the coordinate system determined by the i^{th} box' center and axes, the object point $\mathbf{x} = (x, y)$ is mapped to the image point $\mathbf{x}' = (x', y')$:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_i^x \cos(\theta_i) & -s_i^y \sin(\theta_i) \\ s_i^x \sin(\theta_i) & s_i^y \cos(\theta_i) \end{bmatrix} \begin{bmatrix} x - x_i \\ y - y_i \end{bmatrix} \quad (10)$$

$$= \begin{bmatrix} 1 & 0 & x & y & 0 & 0 \\ 0 & 1 & 0 & 0 & x & y \end{bmatrix} c_i, \quad (11)$$

where (s_i^x, s_i^y), θ_i, and (x_i, y_i) are the scaling, rotation and translation parameters respectively, while c_i, is used for an equivalent parametrization of the deformation; see e.g. [26] for details. The composite deformation of the object is synthesized by subjecting each point $\mathbf{x}$ to transformation (11), where i is the part containing the point.

This clarifies a relationship with the AAM's used earlier: Both models pertain to estimating a deformation of a prototypical object to an observed image instance. AAM's do so by generating a deformation field with a linear model using *global, object-specific* basis elements, while deformable parts models with MRF's do so by using *localized, generic*, basis elements. A part-based model using MRF's is therefore advantageous in that the locality of the models allows relatively simple models to account for complex motions, in a divide-and-conquer strategy. Further, one can treat each

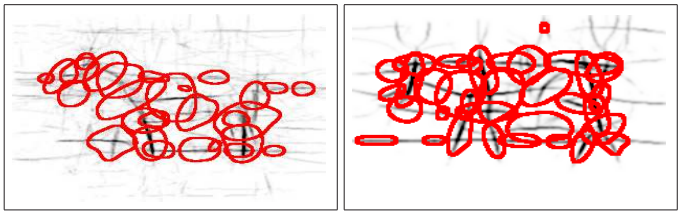


Figure 3: Object parts found by Mean Shift clustering of 50 ridge maps, subsequent to AAM registration.

part separately, allowing it e.g. to be missed, or lying at a different layer, which is not straightforward when using an AAM.

3.2. Initialization Strategy

We use the AAM learning results to initialize the MRF model learning, by pre-registering the objects and removing global pose variation. We next proceed to do mean-shift clustering on the ridge contours which respond to symmetry axes, and therefore can indicate the object parts. After converting the registered ridge curves into a set of points and clustering them as in Sec. (2.3.2), we repeat the clustering, but this time constraining the motion to be only *along* the orientation of the curves, thereby collapsing straight-line segments to points. The output is a set of points, representing different clusters of ridge curves from the data images.

This gives us both the number and extent of the parts as shown in Fig. 3, by using the average scale of the ridges to determine the part width. We note that these parts are not always in strict correspondence with the actual object parts; still, they indicate areas of the object that should move rigidly, and can therefore successfully initialize EM.

3.3. EM Learning of the Part-Based Model

3.3.1 M-step

For learning the MRF structure and clique potentials we rely on the work of [7], due to its simplicity and clarity: The kinematic constraints among parts i, j are expressed using a potential of the form

$$\phi^{i,j}(c_i, c_j) = -\log N(T_{i,j}(c_i) - T_{j,i}(c_j), 0, \Sigma_{i,j}) \quad (12)$$

where $T_{i,j}$ and $T_{j,i}$ are linear transformations that map the part parameters c_i, c_j to their locations, scale, and orientation in the image – and $N(x, \mu, \sigma)$ is the value of the Gaussian distribution with mean μ and variance σ, evaluated at x. Since this is a model for the relative motion of articulated parts, it will assign high likelihood to observations from pairs of parts forming actual joints, since the underlying model will be valid, while its predictions for unrelated parts will be almost random. As in [7], the Minimum Spanning Tree algorithm is used to construct a tree-structured graph connecting all MRF nodes.

Finally, due to computational efficiency considerations described below, a binary pattern is desirable, indicating areas where there should be edges/ridges and where there should not. These are obtained by the thresholding with the Niblack method [17] the templates obtained by registering the whole training set. For example, a binary template obtained for a cow's head is shown in Fig. 4.

3.3.2 E-step

The E-step amounts to estimating the posterior distribution on part positions conditioned on the observed image and the current MRF parameter estimates. This requires performing inference on the MRF; since we have a tree structured MRF we can apply Belief Propagation (BP) to derive the required marginal distributions. BP employs a distributed message passing scheme, where each node i sends to its neighbors $\mathcal{N}(i)$ messages computed as:

$$m_{i,j}(c_j) = \sum_{c_i} \Phi_i(c_i)\Psi_{i,j}(c_i, c_j) \prod_{k \in \{\mathcal{N}(i) \setminus j\}} m_{k,i}(c_i). \quad (13)$$

The belief for each state c_i of node i is estimated as the product of the incoming messages with the potential function Φ_i, $B_i(c_i) = \Phi_i(c_i) \prod_{j \in \mathcal{N}(i)} m_{i,j}(c_i)$ and for a tree-structured graph $B_i(c_i)$ will equal $P(c_i|I)$.

The computational burden in (13) lies in the summation over the state space c_i, which entails estimating the summand for every possible rotation, scaling and translation of the corresponding part. Particle filtering methods for general graphical models like NBP/PAMPAS [21, 11, 20, 10] deal with this problem, using sampling-based approximations to the messages and posteriors, and focusing on a smaller region of the state space; we use the efficient implementation of NBP made available by [10].

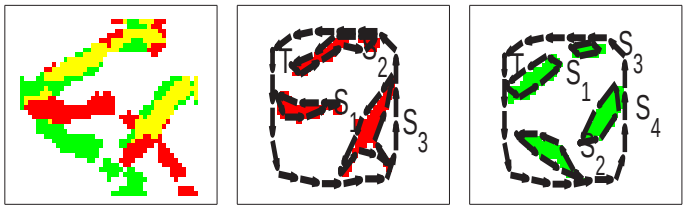
Still, the evaluation of observation potentials for all samples can render the approach impractical due to the large number of operations needed to compute the observation likelihood. We address this in the following

Efficient estimation of the observation potentials: Here we use the efficient technique introduced in [13] to estimate observation potentials; hand-crafted templates for generic features were used there, while here we use it for the automatically constructed part templates.

This technique is based on extending the integral image technique using Stoke's theorem, which allows to express are integrals in terms of curvilinear ones:

$$\iint_S f(x,y)dxdy = \int_{\partial S} Pdx + Qdy = \int_0^l (P,Q) \cdot \mathcal{T} ds.$$

Above $\mathcal{T}$ is the tangent vector, l is the curve's length, and Q and P must satisfy $\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} = f(x,y)$, e.g. $Q(x,y) = \frac{1}{2}\int_0^x f(x,y)dx$, $\quad P(x,y) = -\frac{1}{2}\int_0^y f(x,y)dy$. This only



Template Ridge Interior Edge Interior

Figure 4: Efficient estimation of the observation potential: The sum of e.g. ridge strength R in the interior of the binary template $\int_{S_1 \cup S_2 \cup S_3} RdS$ and in its complement $\int_{T \setminus \{S_1 \cup S_2 \cup S_3\}} RdS$, can be calculated using curvilinear integrals, based on Stoke's theorem.

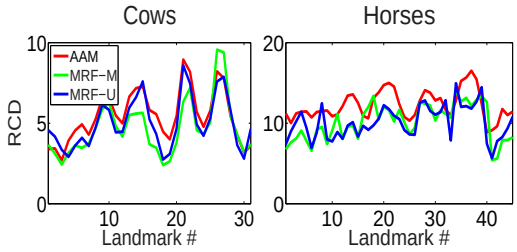


Figure 5: Evaluation results comparing the RCD for manually constructed AAM, part-based (MRF-M) and learned part-based (MRF-L) model; less is better. The learned MRF typically performs worse than the manual MRF and better than the manual AAM models.

requires retrieving the values of the integral images P, Q on the starting points of the arrows, instead of summing over the whole image domain. The use of this efficient method is facilitated by the construction of binary ridge and edge masks in the M-step.

For a proposed part location, (particle) the sum of ridge/edge strength lying inside and outside the predicted template borders is estimated and used as a feature summarizing the image strength as shown in Fig. 4. The extracted measurements are input to a classifier that estimates the probability of the part being present given the feature values. The classifier is trained using the feature values extracted at part locations estimated at the previous EM iteration as positives and features extracted by randomly perturbing these estimates as negatives. Details are given in the supplemental material.

Avoiding Part Collision: A problem we have encountered is the 'collision' of distinct object parts onto the same image locations; for example both front cow feet often lock onto the single foot where the feature strengths are larger, leaving the other unmatched. A more accurate shape model may deal with this problem by penalizing such configurations, but in our case this is not given beforehand.

In [22] a single part is allowed to explain each observation by accounting for self-occlusion in the likelihood criterion. In brief, a hidden variable is introduced indicating whether an observation can be modeled by one part, or is already modeled by another. This allows for a distributed inference scheme that repels collapsing parts, since once a part has 'occupied' an image location, there is no gain in likelihood for the other parts by modeling that area. Details can be found in [22].

To use curvilinear integrals for estimating the observation potentials we slightly modify this idea. We use the edge and ridge strengths predicted by each part to block other parts from falling on the same locations as follows: For each part we use the particle from its marginal distribution with the highest posterior likelihood, and back-project its template onto the image according to that particle's parameters. This gives us a mask indicating the features explained by that part. We discount feature strength on these locations, by multiplying the strength with a small constant smaller than one. This often resolves the collision problem.

3.4. Experimental Results

We first build a part-based model using manually determined parts and deformation parameters estimated from the ground truth data. We then perform NBP on its MRF and compare its performance with that of an AAM trained using the same deformations. As is shown in Fig. 5 the part-based model performs typically better than the AAM on the hard parts, namely the legs, where the RCD measure peaks for AAMs.

We also evaluate the performance of the part-based model using the automatically learned clique potentials and object parts. As we show, it performs worse than the manually trained model, but better than the manually trained AAM. The gain compared to the AAM model is mainly due to the ability of the parts to move independently, which is the case for both part-based models.

In Fig. 6 we show top-down matching results of our part-based model to the primal sketch maps computed from an image, demonstrating our model's ability to act as a generative model for the image sketch. Matching results are shown in Fig. 7, using -for visualization- a model with manually delineated parts, but with MRF parameters learned without supervision, using EM. Similar deformations are estimated using the automatically learned parts.

4. Conclusions

In this work we have pursued the automated construction of generative models for object deformations, demonstrating that this is feasible for a broad set of object categories. We have used dense feature maps extracted from real, noisy images, and demonstrated that unsupervised learning of AAM and part-based models can be accomplished with

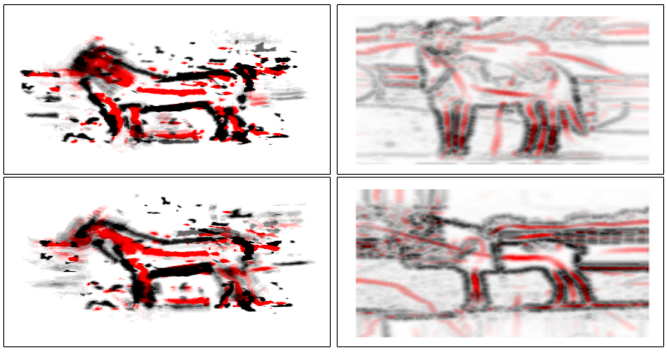


Figure 6: Top-Down synthesis (left) of the object sketch (right) using the marginal distribution of the part-based model. Object-related image information is synthesized adequately well.

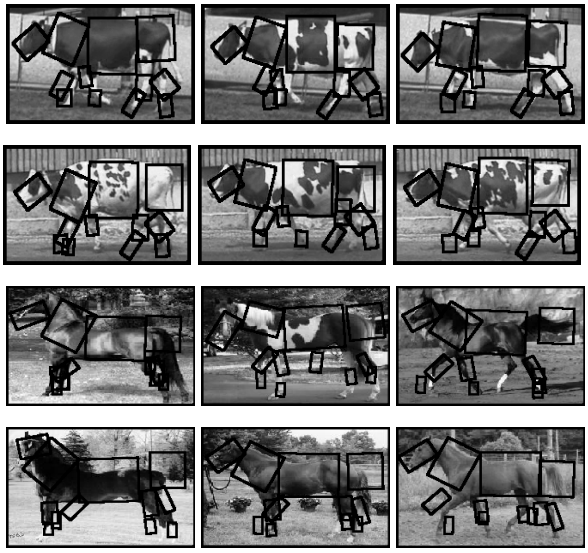


Figure 7: Additional Matching results on horses and cows.

minimal manual annotation. This allows for the construction of more accurate models of complex objects, which we intend to use for tasks like object detection and top-down segmentation.

Acknowledgements

I. Kokkinos thanks P. Maragos for support and discussions during the early stages of this work, while working on [12].

References

[1] S. Baker, I. Matthews, and J. Schneider. Automatic Construction of Active Appearance Models as an Image Coding Problem. *IEEE Trans. PAMI*, 26:1380–1384, 2004.

[2] D. Comaniciu and P. Meer. Mean shift: A Robust Approach toward Feature Space Analysis. *IEEE Trans. PAMI*, 24(5):603–619, 2002.

[3] T. Cootes, G. J. Edwards, and C. Taylor. Active Appearance Models. In *ECCV*, 1998.

[4] T. Cootes, S. Marsland, C. Twining, K. Smith, and C. Taylor. Groupwise Diffeomorphic Non-rigid Registration for Automatic Model Building. In *ECCV*, 2004.

[5] T. Cootes, C. Twining, V. ad R.Schestowitz, and C. Taylor. Groupwise Construction of Appearance Models using Piece-wise Affine Deformations. In *BMVC*, 2005.

[6] A. Dempster, N. Laird, and D. Rudin. Maximum Likelihood from Incomplete Data via the EM algorithm. *Journal of The Royal Statistical Society, Series B*, 1977.

[7] P. Felzenszwalb and D. Huttenlocher. Efficient Matching of Pictorial Structures. In *CVPR*, 2000.

[8] B. Frey and N. Jojic. Transformation-Invariant Clustering Using EM. *IEEE Trans. PAMI*, 25(1):1–17, 2003.

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

[10] A. Ihler, E. Sudderth, W. Freeman, and A. Willsky. Efficient Sampling of Gaussian Distributions. In *NIPS*, 2004.

[11] M. Isard. Pampas: Real Valued Graphical Models for Computer Vision. In *CVPR*, 2003.

[12] I. Kokkinos and P. Maragos. Synergy between Image Segmentation and Object Recognition using the Expectation Maximization Algorithm. 2007. submitted for publication.

[13] I. Kokkinos, P. Maragos, and A. Yuille. Bottom-Up and Top-Down Object Detection Using Primal Sketch Features and Graphical Models. In *CVPR*, 2006.

[14] M. Kumar, P. Torr, and A. Zisserman. Learning layered pictorial structures from video. In *Indian C. Comp. Vis.*, 2003.

[15] T. Lindeberg. Edge Detection and Ridge Detection with Automatic Scale Selection. *IJCV*, 30(2), 1998.

[16] R. Neal and G. Hinton. A View of the EM Algorithm that Justifies Incremental, Sparse and Other Variants. In M. Jordan, editor, *Learning in Graphical Models*. 1998.

[17] W. Niblack. *An Introduction to Digital Image Processing*. Prentice Hall, 1986.

[18] S. Roweis. EM Algorithms for PCA. In *NIPS*, 1998.

[19] R. Schestowitz, C. Twining, C. Cootes, V. Petrovic, C. Taylor, and W. Crum. Assessing the Accuracy of Non-Rigid Registration with and without Ground Truth. In *ISBE*, 2006.

[20] L. Sigal, M. Isard, R. Sigelman, and M. Black. Attractive people. In *NIPS*, 2003.

[21] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *CVPR*, 2003.

[22] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Distributed Occlusion Reasoning for Tracking with Nonparametric Belief Propagation. In *NIPS*, 2004.

[23] T. Vetter, M. Jones, and T. Poggio. A Bootstrapping Algorithm for Learning Linear Models of Object Classes. In *CVPR*, 1997.

[24] M. Welling, M. Weber, and P. Perona. Unsupervised Learning of Models for Recognition. In *ECCV*, 2000.

[25] J. Winn and N. Jojic. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *ICCV*, 2005.

[26] S. Yu, M. Black, and Y. Yacoob. Cardboard People: A Parametrized Model of Articulated Motion. In *CVPR*, 1996.

[27] S. Zhu and A. Yuille. FORMS: A Flexible Object Recognition and Modeling System. *IJCV*, 20(3), 1996.