Mr. B. Srinidhi, Mr. E. Suneel / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 4, Jul-Aug 2013, pp.2510-2516

An Object Shape Completion Using the Shape Boltzmann Machine

Mr. B. Srinidhi*and Mr. E. Suneel**
* M.Tech Student ** Associate Professor, Department of ECE
DVR & Dr HS MIC college of Technology, Kanchikacherla, Andhra Pradesh-521180, India

ABSTRACT

An object shape plays a crucial role in many computer vision applications such as segmentation, object detection, inpainting and graphics. An object shape mainly depends upon local and global variables. Local variables on the shape such as smoothness and continuity can help provide correct segmentations where the object boundary is noisy, unclear or lost in shadow. Global variables on the shape such as ensuring the correct number of parts (legs, wheels, wings etc) can resolve ambiguities where background clutter looks similar to part of the object. In this paper, we present a new learning algorithm for Boltzmann machines that contain two layers of hidden units that we call a Shape Boltzmann Machine (ShapeBM) for the task of modeling foreground/background (binary) and parts-based (categorical) shape images. We show that the ShapeBM can generate more realistic and generalized samples and ability to do shape completion suggests applications in a computer graphic setting.

Keywords – Boltzmann Machine, Generalized, Generative, Realistic, Sampling.

I. INTRODUCTION

Foreground/background classification of pixels is a crucial preprocessing step in many computer vision applications, such as those for object detection and segmentation, inpainting and graphics. The original learning algorithm for Boltzmann machines required randomly initialized Markov chains to approach their equilibrium distributions in order to estimate the data-dependent, data-independent expectations that a connected pair of binary variables would both be on. The difference of these two expectations is the gradient required for maximum likelihood learning. Even with the help of simulated annealing, this learning procedure was too slow to be practical.

There have been a wide variety of approaches to modeling 2D shape. The most commonly used models are grid-structured Markov Random Fields (MRFs) or Conditional Random Fields[8]. In such models, the pairwise potentials connecting neighboring pixels impose very local constraints like smoothness but are unable to capture more complex properties such as convexity or curvature, nor can they account for longer-range properties. Carefully designed high-order potentials [5] allow particular local or longer-range shape properties to be modeled within an MRF, but these potentials fall short of capturing all such properties so as to make realistic-looking samples. For example, a strong shape model of horses would know that horses have legs, heads and tails, that these parts appear in certain positions consistent with a global pose, that there are never more than four legs visible in any given image, that the legs have to support the horse's body, along with many more properties that are difficult to express in words but necessary to make the shape look plausible. A common approach when using a contour (or an image) is to use a mean shape in combination with some principal directions of variation, as captured by a Principal Components Analysis[9] or Factor Analysis[2]. Such models capture the typical global shape of an object and global variations on it (such as changes in the aspect ratio of a face). Non-parametric approaches employ what is effectively a large database of template shapes[6] or shape fragments[3]. In the former case, because no attempt is made to understand the composition of the shape, it is impossible to generalize to novel shapes not present in the database.

In this paper shows how a strong model of binary shape can be constructed using a form of DBM[10] with a set of carefully chosen capacity variables, which we call the Shape Boltzmann Machine (SBM). The model is a generative model of object shape and can be learned directly from training data. Due to its generative formulation the SBM can be used very flexibly, not just as a shape prior in segmentation tasks but also, for instance, to synthesize novel shapes in graphics applications, or to complete partially occluded shapes. We learn SBM models from several challenging shape datasets and evaluate them on a range of shape synthesis and completion tasks. We demonstrate that, despite the relatively small sizes of the training datasets, the learned models are both able to generate realistic samples and to generalize to generate samples that differ from images in the training dataset. We finally present an extension of the SBM that also allows it to simultaneously model the shape of multiple dependent regions such as the parts of an object,

which can in turn be used, for instance, as a prior in parts-based segmentation tasks.

Table 1 : Comparison of a number of different shape models

Shape Models	Realism		Generalization
	Globally	Locally	
Mean [1]	✓	-	-
Factor Analysis[2]	✓	-	✓
Fragments[3]	-	✓	✓
Grid MRFs/CRFs[4]	-	✓	✓
High-order potentials[5]	Limited	✓	✓
Database [6]	✓	✓	-
ShapeBM [7]	✓	✓	✓

We show that the SBM characterizes a strong model of shape[7], in that samples from the model look realistic and it can generalize to generate samples that differ from training examples. The Realism ensures that the model captures shape characteristics at all spatial scales well enough to place probability mass only on images that belong to the "true" shape distribution. The Generalization ensures that there are no gaps in the learned distribution, i.e. that it also covers novel unseen but valid shapes.

II. BOLTZMANN MACHINES

A Boltzmann machine is a network of symmetrically coupled stochastic binary units. It contains a set of visible units $v \in \{0,1\}^D$, and a set of hidden units h$\in \{0,1\}^p$. The energy of the state {v,h} is defined as

$$E(v,h;\theta) = -\frac{1}{2}v^T Lv - \frac{1}{2}h^T Jh - v^T Wh \qquad (1)$$

Where $\theta = \{W, L, J\}$are the model parameters: W, L, J represent visible-to-hidden, visible-to-visible and hidden- to-hidden symmetric interaction terms.

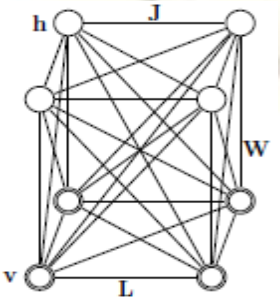


Fig. 1: General Boltzmann machine.

In Fig 1 the top layer represents a vector of stochastic binary "hidden" features and the bottom layer represents a vector of stochastic binary "visible" variables. The diagonal elements of L and J are set to 0. The probability that the model assigns to a visible vector v is

$$p(v;\theta) = \frac{p*(v;\theta)}{Z(\theta)} = \frac{1}{Z(\theta)}\sum_h e^{-E(v,h;\theta)} \qquad (2)$$

$$Z(\theta) = \sum_v \sum_h e^{-E(v,h;\theta)} \qquad (3)$$

Where p* denotes unnormalized probability and Z(θ) is the partition function. The conditional distributions over hidden and visible units are given by

$$p(h_j = 1/v, h_{-j}) = \sigma\left(\sum_{i=1}^{D} W_{ij} v_i + \sum_{m=1\setminus j}^{P} J_{im} h_j\right) \qquad (4)$$

$$p(v_j = 1/h, v_{-i}) = \sigma\left(\sum_{j=1}^{P} W_{ij} h_j + \sum_{k=1\setminus i}^{D} L_{ik} v_j\right) \qquad (5)$$

where $\sigma(x) = 1/(1+e^{-x})$ is the logistic function.

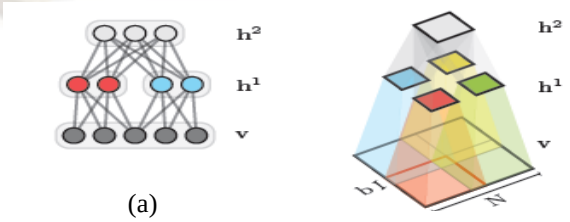
III. PROPOSED MODEL

RBMs and DBMs are powerful generative models, but also have many parameters. Since they are typically trained on large amounts of unlabeled data (thousands or tens of thousands of examples), this is usually less of a problem than in supervised settings. Segmented images, however, are expensive to obtain and datasets are typically small (hundreds of examples). In such a regime, RBMs and DBMs can be prone to over fitting.

In this section we will describe how we can impose a set of carefully chosen connectivity and capacity constraints on a DBM to overcome this problem: the resulting SBM formulation not only learns a model that accurately captures the properties of binary shapes, but that also generalizes well, even when trained on small datasets.

III. I. SHAPE BOLTZMANN MACHINE

The SBM used below has two layers of latent variables: h^1 and h^2. The visible units v are the pixels of a binary image of size N X M. In the first layer we enforce local receptive fields by connecting each hidden unit in h^1 only to a subset of the visible units, corresponding to one of four rectangular patches, as shown in Fig. 2.



(a) (b)

Fig. 2: The Shape Boltzmann Machine.

(a) 1D slice of a Shape Boltzmann Machine.
(b) The Shape Boltzmann Machine in 2D.

In order to encourage boundary consistency each patch overlaps its neighbor by b pixels and so has side lengths of N/2 + b/2 and M/2 + b/2. We furthermore share weights between the four sets of hidden units and patches. In the SBM the receptive field overlap of adjacent groups of hidden units is particularly small compared to their sizes.

Overall, these modifications reduce the number of first layer parameters by a factor of about 16 which reduces the amount of data needed for training by a similar factor. At the same time these modifications take into account two important properties of shapes: first, the restricted receptive field size reflects the fact that the strongest dependencies between pixels are typically local, while distant parts of an object often vary more independently (the small overlap allows boundary continuity to be learned primarily at the lowest layer); second, weight sharing takes account of the fact that many generic properties of shapes (e.g. smoothness) are independent of the image position. For the second layer we choose full connectivity between h^1 and h^2, but restrict the relative capacity of h^2: we use around 4 X 500 hidden units for h^1 vs. around 50 for h^2 in our single class experiments. While the first layer is primarily concerned with generic, local properties, the role of the second layer is to impose global variables, e.g. with respect to the class of an object shape or its overall pose. The second layer mediates dependencies between pixels that are far apart (not in the same local receptive field), but these dependencies will be weaker than between nearby pixels that share first level hidden units. Limiting the capacity of the second layer encourages this separation of concerns and helps to prevent the model from over fitting to small training sets. Note that this is in contrast to who use a top-most layer that is at least as large as all of the preceding layers.

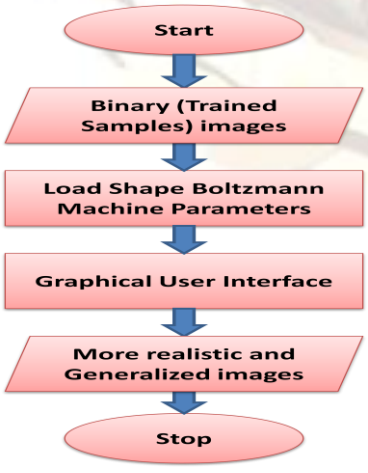


Fig.3: Flow Chart for the Shape Boltzmann Machine

Our MATLAB implementation completed training around 4 hours, running on a dual–core 3GHz PC with 4GB memory. We use advanced versions of MATLAB 2009b. In the below flow chart training samples collected after processing the learning procedure. SBM parameters are discussed in section III.II. Graphical user interface(GUI) is the one of the major tool in the MATLAB platform. Based on SBM parameters more realistic and generalized samples generated through the GUI form trained binary samples.

1) **Realism** - samples from the model look realistic;
2) **Generalization** - the model can generate samples that differ from training examples.

III. II. A MULTI REGION SBM

The SBM model described in the previous section represents Shapes as binary images and can be used, for example, as a prior when segmenting a foreground object from its background. While it is often sufficient to consider the foreground object as a single region without internal structure, there are situations where it is desirable to explicitly model multiple, dependent regions, e.g. in order to decompose the foreground object into parts. In the SBM this can be achieved by using categorical visible units instead of binary ones: Visible units with L + 1 different states (i.e. $v_i \epsilon \{0, \ldots . L\}$) allow the modeling of shapes with L parts. The visible unit representing the i^{th} pixel then indicates which of the L parts or the background the pixel belongs to (here we treat the background as part 0).

We use a "one-of-L + 1" encoding for v_i, i.e. we choose vi to be L + 1 dimensional binary vectors, for $v_i = l$ we set $v_{il} = 1$, $v_{il'} = 0, \forall l' \neq l$. The energy function of this model given by

$$E\left(v, h^1, \frac{h^2}{\theta^s}\right) = \sum_{i,l} b_{li}\, v_{li} + \sum_{i,j,l} w^1_{lij}\, v_{li}\, h^1_j + \sum_j c^1_j h^1_j + \sum_{j,k} w^2_{jk}\, h^1_j h^2_k + \sum_k c^2_k h^2_k \quad (6)$$

Where we use V to denote the the matrix with the L+1 dimensional vectors v_i in its rows.

This change in the nature of the visible units preserves all of the appealing properties of the SBM. In particular the conditional distributions over the three sets of variables V, h1, and h2 remain factorial. The only change is in the specific forms of the two conditional distributions p(v/h^1) and p(h^1/v; h^2) :

$$p(v_i = l|h^1) = \frac{exp\left(\sum_j w^1_{lij}\, h^1_j + b_{li}\right)}{\sum_{l'=0}^{L} \exp\left(\sum_j w^1_{l'ij}\, h^1_j + b_{l'i}\right)} \quad (7)$$

$$p\left(h^1_j = 1/V, h^2\right) = \sigma\left(\sum_{i,l} w^1_{lij}\, v_{li} + \sum_k w^2_{jk}\, h^2_k + c^1_j\right) \quad (8)$$

where in the left-hand-side of eq. (7) we use $v_i = l$ to denote the fact that $v_{il} = 1, v_{il'} = 0, \forall l' \neq l$ as explained above.

The conditional distribution given in eq. (7) implements the constraint that for each pixel only one of these L + 1 binary units can be active, i.e. only one of the parts can be present. Due to the particular form of the conditional distribution (7) categorical visible units are often referred to as "softmax" units. It should be noted that the above formulation of the multi-part SBM is especially suited to model the shapes of several dependent regions such as non-occluding (or lightly occluding) object parts. For modeling the shapes of multiple independent regions, as arise in the case of multiple occluding objects, it might be more suitable to model occlusion explicitly.

IV. LEARNING

Learning of the model involves maximizing log p(v;Θ) of the observed data v with respect to its parameters Θ = {b;W^1;W^2; c^1; c^2}. The gradient of the log-likelihood of a single training image with respect to the parameters is given by

$$\nabla\Theta \log p(v;\Theta) = \langle \nabla\Theta E(v, h^1, h^2; \Theta)\rangle_{p\Theta(h^1,h^2/v)} - \langle \nabla\Theta E(v', h^1, h^2; \Theta)\rangle_{p\Theta(v',h^1,h^2)} \quad (9)$$

and the total gradient is obtained by summing the gradients of the individual training images.

The first term on the right hand side is the expectation of the gradient of the energy where the expectation is taken with respect to the posterior distribution over h^1, h^2 given the observed image v. The second term is also an expectation of the gradient of the energy, but this time taken with respect to the joint distribution over v, h^1, h^2 defined by the model. Although the gradient is readily written out, maximization of the log-likelihood is difficult in practice. Firstly, except for very simple cases it is intractable to compute as both expectations involve a sum over a number of terms that is exponential in the number of variables (visible and hidden units). Secondly, gradient ascent in the likelihood is prone to getting stuck in local optima. In this work we minimizes these difficulties in three ways: (a) it approximates the first expectation in eq. (9) using a mean-field approximation to the posterior; (b) it approximates the second expectation with samples drawn from the model distribution via MCMC; and (c) it employs a pre-training strategy that provides a good initialization to the weights W^1, W^2 before attempting learning in the full model. Learning proceeds in two phases. In the pre-traning phase we greedily train the model bottom-up, one layer at a time. The purpose of this phase is to find good initial values for all parameters of the model. In the second phase we then perform approximate stochastic gradient ascent in the likelihood of the full model to fine-tune the parameters in an expectation-maximization-like scheme. This involves the same sample-based approximation to the gradient of the normalization constant.

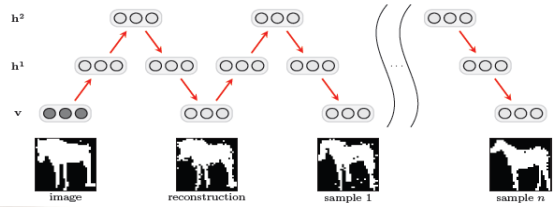


Fig.4: Block-Gibbs MCMC sampling Scheme

In Fig. 9 which v, h^1 and h^2 variables are sampled in turn. Note that each sample of h^1 is obtained conditioned on the current state of v and h^2. For sufficiently large values of n, sample n will be uncorrelated with the original image.

V. RESULTS

In this section we demonstrate that the SBM can be trained to be a strong model of object shape. For this purpose we consider a challenging dataset: Weizmann horses. Weizmann horse dataset The Weizmann horse dataset contains 327 images, all of horses facing to the left, but in a variety of poses. The dataset is challenging because in addition to their overall pose variation, the positions of the horses' heads, tails and legs change considerably from image to image.

V. I. REALISM

The realism ensures that the model captures shape characteristics at all spatial scales well enough to place probability mass only on images that belong to the "true" shape distribution. The SBM aims to overcome these problems through a combination of connectivity constraints, weight sharing and model hierarchy. The combination of these ingredients is necessary to obtain a strong model of shape. Samples from the SBM for horses and motorbikes are shown in Fig. 5.

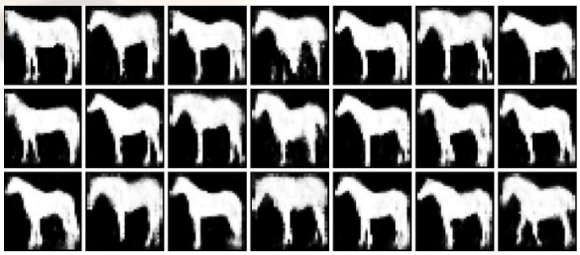


Fig. 5 Realism Criterion

First, we note that the model generates natural shapes from a variety of poses. Second, we observe that details such as legs (in the case of horses) or handle bars, side mirrors, and forks (in the

case of motorbikes) are preserved and remain sharply defined in the samples. Third, we note that the horses have the correct number of legs while motorbikes have, for instance, the correct number of handle bars and wheels. Finally, we note that the patch overlap ensures seamless connections between the four quadrants of the image. Indeed, horse and motorbike samples generated by the model look suficiently realistic that we consider the model to have fulfilled the Realism requirement.

V. II. GENERALIZATION

The generalization ensures that there are no gaps in the learned distribution, i.e. that it also covers novel unseen but valid shapes. We next investigated to what extent the SBM meets the Generalization requirement, to ensure that the model has not simply memorized the training data. In Fig. 6 we show for horses the difference between the sampled shapes from Fig. 5 and their closest images in the training set. We use the Hamming distance between training images and a thresholded version of the conditional probability (> 0.3), as the similarity measure. This measure was found to retrieve the visually most similar images. Red indicates pixels that are in the sample but not in the closest training image, and yellow indicates pixels in the training image but not in the sample. Both models generalize from the training data-points in non-trivial ways whilst maintaining validity of the overall object shape. These results suggest that the SBM generalizes to realistic shapes that it has not encountered in the training set.

Fig. 6 Generalization Criterion

V. III. MULTIPLE OBJECT CATEGORIES

Class-specific shape models are appropriate if the class is known, but for segmentation / detection applications this may not be the case. A similar situation arises if the view point is not fixed (e.g. objects can appear right or left facing). In both cases there is large overall variability in the data but the data also form relatively distinct clusters of similar shapes (e.g. all objects from a particular category, or all right-facing objects).

To investigate whether the SBM is able to successfully deal with such additional variability and structure in the data we applied it to a dataset consisting of shapes from multiple object classes and tested whether it would be able to learn a strong model of the shapes of all classes simultaneously.We trained an SBM on a combination of the Weizmann data and 3 other animal categories from Caltech-101. In addition to 327 horse images, the dataset contains images of 798 motorbikes, 68 dragonflies, 78 llamas and 59 rhinos (for a total of 1329 images).

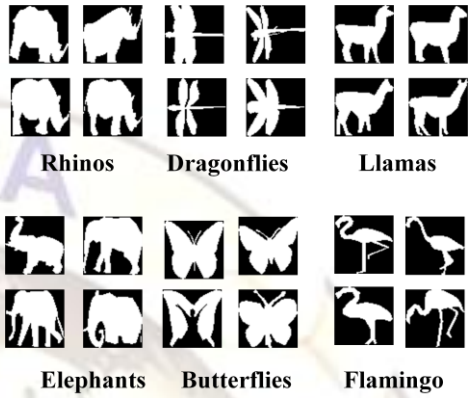


Fig. 7 Multiple objects

VI. SHAPE COMPLETION

We further assessed both the realism and generalization capabilities of the SBM by using it to perform shape completion, where the goal is to generate likely configurations of pixels for a missing region of the shape, given the rest of the shape. To perform completion we obtain samples of the missing - or unobserved – pixels v_U conditioned on the remaining (observed) pixels v_O (U and O denote the set indices of unobserved and observed pixels respectively). This is achieved using a Gibbs sampling procedure that samples from the conditional distribution.

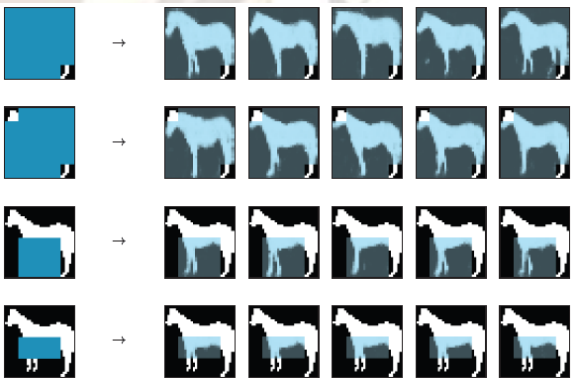


Fig. 8 Shape completion variability

In this procedure, samples are obtained by running a Markov chain as before, sampling v, h^1, and h^2 from their respective conditional distributions, but every time v is sampled we "clamp" the observed pixels v_O of the image to their given values, updating only the state of the unobserved pixels v_U. Since the

model specifies a distribution over the missing region $p(v_U/v_O)$, multiple such samples capture the variability of possible solutions that exist for any given completion task. In Fig. 8 we show how the samples become more constrained as the missing region shrinks. Blue in the first column indicates the missing regions. The samples highlight the variability in possible completions captured by the model. As the missing region shrinks, the samples become more constrained.

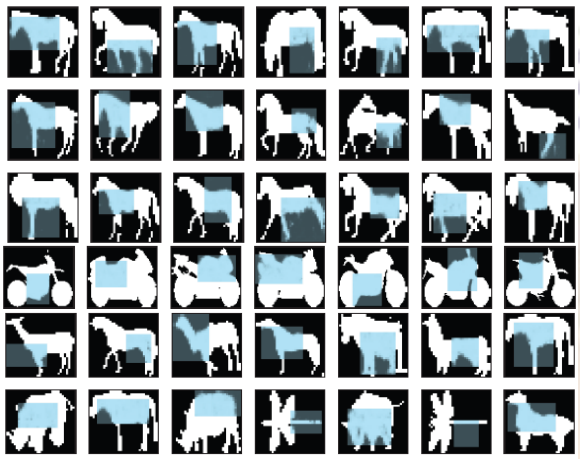
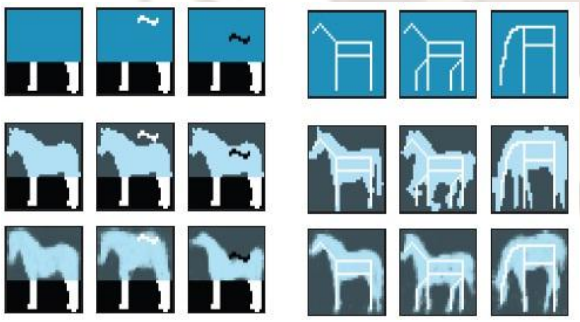


Fig. 9: Sampled image completion.

Fig. 9 shows sampled completions of regions of horse, motorbike, llama, dragonfly and rhino images that the model had not seen during training. Despite the large sizes of the missing portions, and the varying poses of the horses, motorbikes, llamas, dragonflies and rhinos completions look realistic. The SBM's ability to do shape completion suggests applications in a computer graphics setting. Sampled completions can be constrained in real-time by simply clamping certain pixels of the image.



(a) (b)

Fig. 11: Constrained shape completion.

In Fig. 11a we show snapshots of a graphical user interface in which the user modifies a horse silhouette with a digital brush. The model's ability to generalize enables it to generate samples that satisfy the user's constraints. The model's accurate knowledge about horse shapes ensures that the samples remain realistic. As a direct comparison we also consider a simple data-base driven ("non-parametric") approach where we try to find suitable completions via a nearest-neighbor search in our database of training shapes. Missing regions (blue pixels, top row) are completed using the SBM and by finding the closest match (middle row) to the prescribed pixels in the training data. Fig. 11a The horse's back is pulled up by the SBM (bottom row) using an appropriate \on" brush. Notice how the stomach moves up and the head angle changes to maintain a valid shape. The horse's back is then pushed down with an "on" brush. Fig. 11b given only minimal user input, the model completes the images to generate realistic horse shapes. As shown in Fig. 11 such a database-driven approach can fail to find shapes that match the constraints.

A natural way to directly evaluate a generative model quantitatively is by computing the likelihood of some held-out data under the model. As an alternative we therefore introduce what we will refer to as an "imputation score" for the shape completion task as a measure of the strength of a model. We collect additional horse and motorbike silhouettes from the web (25 horses and 25 motorbikes), and divide each into 9 segments. We then perform multiple imputation tests for each image. In each test, we remove one of the segments and estimate the conditional probability of that segment under the model, given the remaining 8 segments. The log probabilities are then averaged across the different segments and images to give the score. Except for the mean model (where they are trivial) the conditional distributions over the subsets of unobserved pixels given the rest of the image are infeasible to compute in practice due to the dependencies introduced by the latent variables. We therefore approximate the required conditional log-probabilities via MCMC: for a particular image and segment we draw configurations of the latent variables from the posterior given the observed part of the image and then evaluate the conditional probability of the true configuration of the unobserved segment given the latent variables, i.e. we compute

$$p(V_U|V_O) \approx \frac{1}{S}\sum_s p\left(V_U|\hat{h}^s\right) \quad (10)$$

Provided that our MCMC scheme allows us to sample from the true posterior the right hand side of eq. 10 provides us with an unbiased estimate of $p(v_U|v_O)$.A high score in this test indicates both the realism of samples and the generalization capability of a model, since models that do not allocate probability mass on good shapes (from the "true" generating distribution of horses) and models that waste probability mass on bad shapes are both penalized. The ShapeBM significantly outperforms our baseline models at this task.

VII. CONCLUSION

In this paper we have presented the Shape Boltzmann Machine, a strong generative model of object shape. The SBM is based on the general DBM architecture, a form of undirected graphical model that makes heavy use of latent variables to model high-order dependencies between the observed variables. We believe that the combination of (a) carefully chosen connectivity and capacity constraints, along with (b) a hierarchical architecture, and (c) a training procedure that allows for the joint optimization of the full model, is key to the success of the SBM.

These ingredients allow the SBM to learn high quality probability distributions over object shapes from small datasets, consisting of just a few hundred training images. The learned models are convincing in terms of both realism of samples from the distribution and generalization to new examples of the same shape class. Without making use of specialist knowledge about the particular shapes the model develops a natural representation with some separation of concerns across layers.

REFERENCES

[1] N. Jojic and Y. Caspi, Capturing Image Structure with Probabilistc Index Maps, *In CVPR*, 2004, 212-219.

[2] T. Cemgil, W. Zajdel and B. Krose, A Hybrid Graphical Model for Robust Feature Extraction from video, *In CVPR*, 2005, 1158-1165.

[3] E. Borenstein, E. Sharon, and S. Ullman. Combining Top-Down and Bottom-Up Segmentation. *In CVPR Workshop on Perceptual Organization in Computer Vision*, 2004.

[4] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 2004, 309–314.

[5] S. Nowozin and C. Lampert. Global connectivity potentials for random field models. *In CVPR*, 2009, 818–825.

[6] D. Gavrila. An Exemplar-Based Approach to Hierarchical Shape Matching. *PAMI*, 2007, 1408–1421.

[7] S. M. Ali Eslami, Nicolas Heess, and John Winn. The Shape Boltzmann Machine: a Strong Model of Object Shape. *In IEEE CVPR*, 2012, 406-413.

[8] Y. Boykov and M.P. Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D images. *In ICCV*, 2001, 105–112.

[9] T. Cootes, C. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application, *Computer Vision and Image Understandin*g, 1995, 61:38–59.

[10] R. Salakhutdinov and G. Hinton. Deep Boltzmann Machines, *In AISTATS*, 2009, vol. 5, 448–455.

[11] T. Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. *In ICML*, 2008, 1064-1071.