Learning Co-Visibility: Networks for Nuisance-Invariant Image Comparison

Abstract

We test the hypothesis that a generic learning architecture can be used to *train away* nuisance variability preset in images, owing to changes of viewpoint, illumination and partial occlusion. We distill the simplest possible classification task, a binary classification with no intrinsic variability, which amounts to the determination of *co-visibility* from different images of the same underlying scene. While we cannot outperform algorithms specifically engineered for occlusion detection, that make direct use of the phenomenology of the image formation process, we show that a Gated RBM successfully learns away the nuisance variability due to viewpoint changes and partial occlusion. More work is necessary to test this hypothesis in the presence of significant intra-class variability.

1 Introduction

Deep dearning architectures have shown the ability to learn class-specific variability despite significant nuisance variability [17, 25, 44]. The problem of nuisance variability is particularly acute in Computer Vision, where even the same object or scene can yield a large variety of images depending on vantage point, scale, illumination, partial occlusion etc. This point has been recently emphasized by Poggio [29], who set forth the hypothesis that much of the ventral stream is tasked with managing the infinite amount of nuisance variability, and by Sundaramoorthi et al. [39], who showed that the quotient of images modulo changes of viewpoint and illuimnation is supported on a set of measure zero, and therefore the intrinsic variability of objects in images is infinitesimal compared to nuisance variability.

These theses would seem to challenge the possibility that nuisance variability in images can be *learned away* by even powerful learning architectures.[1] In this manuscript, we put this challenge to the test, by distilling the simplest possible decidable visual classification task, and deploying a deep learning architecture to tackle it.

We have identified the simplest task to be the determination of *co-visibility*. This is a binary decision where, given putative correspondence between pixels in *two or more* images, we wish to determine whether or not these *back-project* onto the same point in physical space. This completely eliminates *intrinsic variability* (the underlying scene is known to be the same), and the diversity between images of the same scene is entirely attributable to nuisance factors such as different vantage points, possibly different illumination, and partial occlusion. We then deploy a Gated RBM to learn away such nuisance variability, given training samples consisting of multiple pairs of images of the same underlying scene.

[1] At the *ontogenic* level; of course nuisance variability in images can be learned-away at the *phylogenic* level, encoded in the architecture of an algorithm, or in the evolution of a biological sensing and processing organism.

The result is effectively an algorithm for *occlusion detection*. Violation of co-visibility occurs in regions of an image that back-project onto portions of the scene that are not visible from another. Therefore, we compare our results against algorithms that are specifically *engineered* to detect occlusions. While we cannot expect to outperform them in the specific task, we wish to test the hypothesis that a Gated RBM successfully learns away the nuisance variability in the image formation process. Our empirical analysis is described in Sect. 3, and our conclusions are drawn in Sect. 4. The upshot is that, even though in theory nuisances account for almost all the variability in the data [39], in practice the finite cardinality of data space acts as a regularizer, and since the classification occurs in data space (at least for this simplest task), nuisance variability can be learned away. More empirical analysis needs to be performed to assess this issue when significant intrinsic variability competes with nuisance variability, such as in the detection, localization, and recognition of object classes in multiple images.

1.1 Related work

The determination of co-visibility is related to the general problem of *correspondence*, that underlies a significant portion of Computer Vision work, that cannot realistically be reviewed here. When correspondence is trivial, for instance when the multiple images of the same scene are taken from a stationary camera at different time instants, this problem is known as *background subtraction* [28]. In this case, violations of co-visibility are due to the presence of a "foreground" object, which is the object of interest. More in general, the determination of co-visibility is entangled with correspondence, so this problem relates to *optical flow*, another broad concern in the Computer Vision literature. In [3] the problem of occlusion detection, and the simultaneous estimation of optical flow, is cast as a (relaxed) convex optimization; we take this as a paradigm for comparison when the "baseline" (inter-frame translational motion) is small. For large baseline, the correspondence problem is significantly more complex, thus our hope is that a learning architecture could tackle it too. Other recent work on occlusion detection includes [5, 22], that focuses on occlusion boundaries, as opposed to co-visibility, similar to [1, 18, 21, 11, 14, 35]). In [16] an online learning framework is proposed that combines dense optical flow estimation and pixel-wise information gathering. Other approaches infer the occluded regions by calculating the residual from optical flow estimation ([37, 43]). Other works formulate the occlusion detection as classification problem ([19, 21, 38]) and perform motion estimation in a discrete setting that is a well-known difficult problem.

Our work is most closely related to [40] who design an architecture for stereo; in stereo, two images are captured concurrently, and the baseline is short and known. We therefore consider this as a special case of co-visibility, where the "optical flow" component is restricted to a scalar disparity field (assuming the stero rig has been pre-calibrated). We also employ a Gated Restricted Boltzmann Machine (RBM), but train on a far wider range of nuisance variability.

The flexibility of this method lies in the fact that by, selecting an appropriate training set, the model can be made insensitive to nuisance variability due to viewpoint and illumination, so the residual is easily partitioned into two classes: Visible or occluded. We use a variety of image pairs, captured indoor and outdoor. The features that are automatically extracted by the model during the learning phase are either appearance or motion features, but can be adapted in different settings and deal effectively with different types of nuisances.

2 The model

We model the joint probability of pairs of real- or binary valued image patches $\mathbf{x}$ and $\mathbf{y}$ through a set of binary hidden variables $\mathbf{h}$, which capture *elementary* operations, such as a translational shift, planar rotation and other small-dimensional (local) group transformations. More complex transformations can be obtained by marginalizing over subsets of the hidden variables. Thus, following [40], we define a *similarity measure* between two images x and y, as follows:

$$S(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \sum_{f=1}^{F}((\sum_{i=1}^{I} u_{if}x_i)(\sum_{j=1}^{J} v_{jf}y_j)(\sum_{k=1}^{K} w_{kf}h_k)) \quad (1)$$

Both the images and the hidden layer are projected onto the $\mathbf{F}$ corresponding filters and then the projections are multiplied. This is the factorized version of the similarity term, which has been championed by [23]. Intuitively this product represents a similarity score as it can be intepreted as a *co-occurence* of high projection scores of both $\mathbf{x}$ and $\mathbf{y}$ for a certain subset of the hidden variables $\mathbf{h}$. In case a *symmetric* model is used, where the filters of both visible layers are equal ($u_{if} = v_{if}, i = 1 \dots I, I = J$), high values in the projections for $\mathbf{x}$ and $\mathbf{y}$ expresses similarity between them. But even when the network is not symmetric, the high simlarity score S means that the two images respond to filters that are related via a subset of the hidden layer filters/transformations $\{w_{kf}, k = 1 \dots K\}$.

By assuming that the training set is a set of similar in some sense image pairs, the parameters of visible layers' filters $\mathbf{u}_f$ and $\mathbf{v}_f$ are trained to represent structures that detect similarity between image pairs, as the latter ones respond highly for them. On the other side, the natural interpretation of the hidden layer filters $\mathbf{w}_f$ is that they represent every single elementary transformation between similar in some sense images. For example, if we have two identical images, but the one is a rotated version of the other one, then the network will respond highly for some subset of the hidden variables that pertain to elementary rotation transformations.

Then the two quadratic terms $1/2 * \sum_{i=1}^{I}(x_i - a_i)^2$ and $1/2 * \sum_{j=1}^{J}(y_j - b_j)^2$ are introduced in the formulation, which represent the L_2 distance of the visible layers from some offsets $\{a_i, i = 1 \dots I\}$ and $\{b_j, j = 1 \dots J\}$ for every image pixel. In that way, the average offset over the training set is eliminated. Last, a *generalization* term $-\sum_{k=1}^{K} c_k h_k$ is added, so that the hidden layer filters are weighted sums over as many elementary similarity hidden variables as possible. This ensures better generalization over the image pairs that do not belong to the training set. Thus, the energy of the *joint* configuration is defined as:

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}; \theta) = -S(\mathbf{x}, \mathbf{y}, \mathbf{h}) + 1/2 * \sum_{i=1}^{I}(x_i - a_i)^2 + 1/2 * \sum_{j=1}^{J}(y_j - b_j)^2 - \sum_{k=1}^{K} c_k h_k \quad (2)$$

where $\theta = \{\mathbf{U}, \mathbf{V}, \mathbf{W}, \mathbf{a}, \mathbf{b}, \mathbf{c}\}$ are the model parameters. As a notation, the matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ are used for the F elementary filters of each layer and the vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ for the offsets.

The joint probability is given by the *Boltzmann* distribution over the triplets $(\mathbf{x}, \mathbf{y}, \mathbf{h})$:

$$P_\theta(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \frac{e^{-E(\mathbf{x}, \mathbf{y}, \mathbf{h}; \theta)}}{Z(\theta)} \quad (3)$$

where $Z(\theta) = \sum_{\mathbf{x}, \mathbf{y}, \mathbf{h}} e^{-E(\mathbf{x}, \mathbf{y}, \mathbf{h}; \theta)}$ is the *partition* function. The not symmetric model can be applied over image pairs of different size ($I \neq J$), while the number of hidden variables K and the number of the filter triplets $\{(\mathbf{u}_f, \mathbf{v}_f, \mathbf{w}_f), f = 1 \dots F\}$ are defined by the user. Some representative values are: $I = J = 13 * 13 = 169$, $K = 100$, $F = 200$. The model is drawn in Fig. 1.

2.1 Conditionals and Marginals

The large complexity advantage of the Restricted Boltzmann Machines over the Boltzmann Machines is that all the variables of a layer (hidden or visible) are *conditional independent* to each other, so the joint conditional distribution over a layer is just the product of all the conditional distribution and the calculations can be done in parallel. So by using the Eqs. 1-3 and the statistical independence, we receive the following conditionals (from now on, the notation is simplified by omitting θ, athough it is implied):

$$P(\mathbf{x}|\mathbf{y}, \mathbf{h}) = \prod_{i=1}^{I} \mathcal{N}(a_i + \sum_{f=1}^{F} u_{if}(\sum_{j=1}^{J} v_{jf} y_j)(\sum_{k=1}^{K} w_{kf} h_k); 1.0) \quad (4)$$

$$P(\mathbf{y}|\mathbf{x}, \mathbf{h}) = \prod_{j=1}^{J} \mathcal{N}(b_j + \sum_{f=1}^{F} v_{jf}(\sum_{i=1}^{I} u_{if} x_i)(\sum_{k=1}^{K} w_{kf} h_k); 1.0) \quad (5)$$

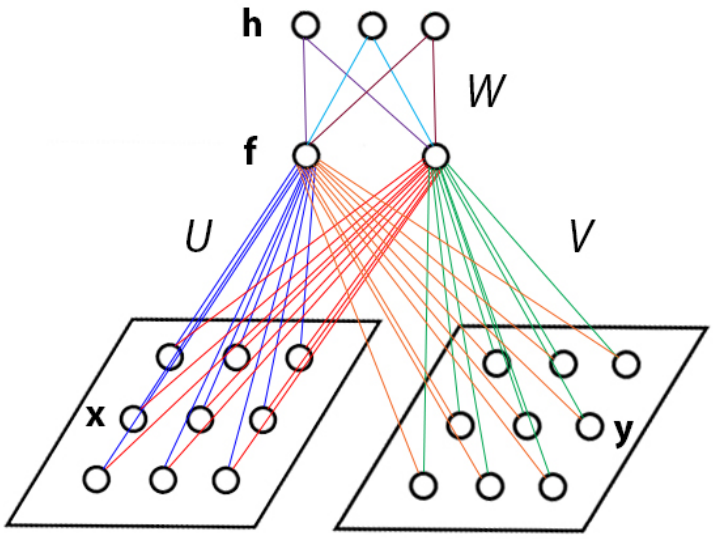

Figure 1: The Gated RBM.

$$P(\mathbf{h}|\mathbf{x},\mathbf{y}) = \prod_{k=1}^{K} ber(c_k + \sum_{f=1}^{F} w_{kf}(\sum_{i=1}^{I} u_{if}x_i)(\sum_{j=1}^{J} v_{jf}y_j)) \tag{6}$$

In the case that the visible layers are not real-valued, but binary-values like the hidden layers, the Eqs. 4 and 5 become:

$$P(\mathbf{x}|\mathbf{y},\mathbf{h}) = \prod_{i=1}^{I} ber(a_i + \sum_{f=1}^{F} u_{if}(\sum_{j=1}^{J} v_{jf}y_j)(\sum_{k=1}^{K} w_{kf}h_k)) \tag{7}$$

$$P(\mathbf{y}|\mathbf{x},\mathbf{h}) = \prod_{j=1}^{J} ber(b_j + \sum_{f=1}^{F} v_{jf}(\sum_{i=1}^{I} u_{if}x_i)(\sum_{k=1}^{K} w_{kf}h_k)) \tag{8}$$

where the Gaussian (Normal) and Bernoulli distributions are the following:

$$\mathcal{N}(\mathbf{x};\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}} \tag{9}$$

$$ber(\mathbf{x};p) = \begin{cases} p, & \text{if } x = 1 \\ 1-p, & \text{if } x = 0 \end{cases} \tag{10}$$

Finally, the distribution over the image pairs $\mathbf{x}$ and $\mathbf{y}$ is computed by marginalizing the joint distribution of Eq. 3 over $\mathbf{h}$:

$$P(\mathbf{x},\mathbf{y}) = \sum_{\mathbf{h}\in\{0,1\}^K} P(\mathbf{x},\mathbf{y},\mathbf{h}) \tag{11}$$

The number of possible $\mathbf{h}$ increases exponentially with the number of hidden variables K and makes the computation really difficult for bigger K. However, working with the conditionals, for which sampling is "easy" and the statistical independence simplifies the computations, is enough for both the model training and the distance (similarity) calculations for the testing images pairs.

2.2 Maximum Likelihood Learning

Given a set of *i.i.d.* training examples $D = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \dots (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})\}$, we want to learn the model parameters θ. An unsupervised learning framework is used. Our strategy is to maximize the penalized log-likelihood:

$$L(\theta) = \frac{1}{N}\sum_{n=1}^{N} logP(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) - \frac{\lambda}{N}(\|\mathbf{U}\|^2_{Frob} + \|\mathbf{V}\|^2_{Frob} + \|\mathbf{W}\|^2_{Frob}) \tag{12}$$

for the set of training pairs D. A regularization term $-\frac{\lambda}{N}(\|\mathbf{U}\|^2_{Frob} + \|\mathbf{V}\|^2_{Frob} + \|\mathbf{W}\|^2_{Frob})$ is added to the objective function, so that the learning is speeded up. $Frob$ stands for the *Frobenius* norm, which for matrix $\mathbf{U}$ is defined as:

$$\|\mathbf{U}\|_{Frob} = \sqrt{trace(\mathbf{U}^*\mathbf{U})} = \sqrt{\sum_{f=1}^{F}\sum_{i=1}^{I}|u_{if}|^2} \tag{13}$$

Similar definitions apply for $\mathbf{V}$ and $\mathbf{W}$. When combining Eqs. 3, 11 and 12, the derivative of $L(\theta)$ wrt. $\mathbf{U}$ is given by:

$$\frac{\partial L(\theta)}{\partial U_{if}} = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial}{\partial U_{if}} log(\sum_{\mathbf{h}} e^{-E(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{h}; \theta)}) - \frac{\partial logZ(\theta)}{\partial U_{if}} - \frac{2\lambda U_{if}}{N} \tag{14}$$

$$= -\frac{1}{N}\sum_{n=1}^{N}\left\langle \frac{\partial E(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{h}; \theta)}{\partial U_{if}} \right\rangle_{\mathbf{h}} + \left\langle \frac{\partial E(\mathbf{x}, \mathbf{y}, \mathbf{h}; \theta)}{\partial U_{if}} \right\rangle_{\mathbf{x},\mathbf{y},\mathbf{h}} \tag{15}$$

The second term in Eq. 15 is intractable, as it is computed over all the exponentially many configurations $(\mathbf{x}, \mathbf{y}, \mathbf{h})$. However, following the approach proposed in [40], the complexity of the problem becomes reasonable. The expected value over all the configurations is approximated by using *Gibbs sampling*. Samples are drawn alternatingly from the conditional distributions $P(\mathbf{h}|\mathbf{x}, \mathbf{y})$, $P(\mathbf{x}|\mathbf{h}, \mathbf{y})$ and $P(\mathbf{y}|\mathbf{h}, \mathbf{x})$ in random order and the process is terminated before the equilibrium distribution [12]. Given the tri-partite structure of the model, the learning process can be characterized as *3-way Contrastive Divergence*. The algorithm 1 from [40] is followed.

2.3 Occlusion testing

The model is trained with pairs of images of the same underlying scene, taken under different illumination, different vantage point – yielding (self-)occlusions. We then design a score to quantify co-visibility in a testing pair. This is performed at the patch level. The log-likelihood that is assigned to an image pair $(\mathbf{x}, \mathbf{y})$ is given by combining Eqs. 2, 3 and 11:

$$logP(\mathbf{x}, \mathbf{y}) = -logZ + \sum_{k=1}^{K} log(1 + exp(c_k + \sum_{f=1}^{F} w_{kf}(\sum_{i=1}^{I} u_{if}x_i)(\sum_{j=1}^{J} v_{jf}y_j)) - 1/2 * \sum_{i=1}^{I}(x_i - a_i)^2 - 1/2 * \sum_{j=1}^{J}(y_j - b_j)^2 \tag{16}$$

The normalizing term $logZ$ is the most demanding to compute, as it includes marginalization over $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$, but fortunately it is not necessary to be computed. When two image patch pairs are compared, this term is eliminated. By adopting a naive strategy of comparing the log-likelihood of couples of image pairs, there is a danger of making the likelihood of a single pair $(\mathbf{x}, \mathbf{y})$ arbitrarily bad by rescaling both images with some constant. To deal with that, the following metric is used:

$$d(\mathbf{x}, \mathbf{y}) = -logP(\mathbf{x}, \mathbf{y}) - logP(\mathbf{y}, \mathbf{x}) + logP(\mathbf{x}, \mathbf{x}) + logP(\mathbf{y}, \mathbf{y}) \tag{17}$$

similar to [40]. The normalizing terms are eliminated, plus the likelihood of any single image as input in both observed layers is normalized.

3 Experimental Evaluation

Overlapping patches were extracted from test images and the corresponding patches in the two images were compared using the distance of Eq. 17. Classification was performed by thresholding the log-likelihood. Both the *scale* and the *threshold* t are parameters of our method and need to be selected empirically or via cross-validation over a set of training video sequences with the occlusion maps provided as ground-truth. The scale is dependent on the magnitude of the image transformations that are not occlusions. Intuitively, the (average) dominant transformation of the background scene is distinguishable inside the patches (observed layers) by the model, so the latter one is able to eliminate it.

We tested our method on sequential frames in the Berkeley Motion Segmentation and Middlebury datasets, using 13×13 patches between the corresponding areas of every two frames (that is $I = J = 169$). We used $F = 200$ filters for each layer (the two observed and the hidden one) and $K = 100$ hidden variables. In Fig. 2 the occluded areas between the frames $20 - 21$ in the *cars8* case of the Motion Segmentation dataset are demonstrated, where thresholding with $t = 0.7$ is applied on the log-likelihood metric.

As a baseline, we compare our method with simple difference of averaged intensities in the patches. This yields a large number of false alarms (grass or road patches movements) that are instead easily handled by the network. We trained the model for $10,000$ iterations. The training set includes shifted and rotated random images, and general affine transformations in images cropped from the CIFAR dataset. All the samples were of size 13×13 exactly as the size of the observed layers of the model. In this case $25,000$ training image pairs were used.

3.1 Statistics

Next, we present a comparison of our occlusion detection results in some examples of the Middlebury dataset with [4] and [19] in terms of precision and recall statistics. In [19] the binary occlusion maps are provided, so for certain values of their recall we all compare the precision values.

	Venus	RubberWhale	Hydrangea	Grove2	Grove3	Urban2	Urban3
Recall [19]	0.66	0.20	0.20	0.55	0.45	0.50	0.51
Precision [19]	0.61	0.46	0.68	0.72	0.79	0.26	0.56
Precision [4]	0.69	0.91	0.96	0.96	0.86	0.95	0.94
Precision (ours)	0.64	0.93	0.95	0.78	0.83	0.74	0.62

Table 1: Comparison with [4] and [19] on Middlebury. Both [4] and we compare with [19] in terms of the precision for certain recall values.

As we anticipated, we do not expect to outperform algorithms that are specifically engineered for occlusion detection. Nevertheless, in Table 2 we show the $F1-score$[2], that measures both precision and recall and we present its maximum value for each instance.

	Venus	RubberWhale	Hydrangea	Grove2	Grove3	Urban2	Urban3
F1-Score (ours)	0.83	0.80	0.81	0.70	0.86	0.83	0.63

Table 2: Our F1-score values for some instances on Middlebury dataset.

4 Discussion

We have empirically tested the hypothesis that a fairly simple learning architecture can manage the nuisance variability in the imaging process. To this end, we have distilled the simplest classification

[2]The $F1 - score$ (or balanced $F - score$) defined as the harmonic mean of precision and recall.

Figure 2: The detected occluded areas in the case *cars8* between frames $20 - 21$ from the Berkeley Motion Segmentation dataset. In the upper image we present the result that we obtain by calculating the average intensity difference patch-wise over the image between the two frames, while in the lower image we demonstrate the result of our method.

problem (binary), for the case where there is no intrinsic variability (the underlying scene is known to be the same), so learning away nuisance variability is tantamount to performing occlusion detection.

We have shown empirically that, although a network does not outperform algorithms specifically engineered to detect occlusions, it does manage to reduce nuisance variability significantly, thus challenging recent work that suggests that nuisance variability accounts for most of the complexity in imaging data, at least for the specific case when intrinsic variability is absent.

Of course, the case where significant intrinsic variability is present and competes with nuisance variability is far more complex to test, and a significant portion of the machine learning and computer vision communities are engaged in this process. Here we have tested a simple example to have some indication on the way forward.

The reader may question the fact that we are trying to *learn* occlusion detection, when there are perfectly viable algorithms to engineer the process. We stress that occlusion detection is just a vehicle to test our hypothesis, and we would be the first to dissuade anyone from using a generic architecture for occlusion detection.

References

[1] L. Álvarez, R. Deriche, T. Papadopoulo and J. Sánchez. "Symmetrical Dense Optical Flow Estimation with Occlusions Detection", *International Journal of Computer Vision*, pp. 371-385, 2007.

[2] A. Ayvaci and S. Soatto. "Detachable Object Detection: Segmentation and Depth Ordering from Short-Baseline Video", *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1942-1951, 2012.

[3] A. Ayvaci, M. Raptis and S. Soatto. "Sparse Occlusion Detection with Optical Flow", *International Journal of Computer Vision*, pp. 322-338, 2012.

[4] A. Ayvaci, M. Raptis and S. Soatto. "Occlusion Detection and Motion Estimation with Convex Optimization", *NIPS*, pp. 100-108, 2010.

[5] N. Apostoloff and A. W. Fitzgibbon. "Learning Spatiotemporal T-Junctions for Occlusion Detection", *CVPR*, pp. 553-559, 2005.

[6] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black and R. Szeliski. "A Database and Evaluation Methodology for Optical Flow", *International Journal of Computer Vision*, pp. 1-31, 2011.

[7] R. Ben-Ari and N. A. Sochen. "Stereo Matching with Mumford-Shah Regularization and Occlusion Handling", *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 2071-2084, 2010.

[8] A. F. Bobick and S. S. Intille. "Large Occlusion Stereo", *International Journal of Computer Vision*, pp. 181-200, 1999.

[9] R. Fransens, C. Strecha and L. J. V. Gool. "A Mean Field EM-algorithm for Coherent Occlusion Handling in MAP-Estimation Prob", *CVPR*, pp. 300-307, 2006.

[10] D. B. Grimes and R. P. N. Rao. "Bilinear Sparse Coding for Invariant Vision", *Journal of Neural Computation*, pp. 47-73, 2005.

[11] X. He and A. L. Yuille. "Occlusion Boundary Detection Using Pseudo-depth", *ECCV*, pp. 539-552, 2010.

[12] G. E. Hinton. "Training Products of Experts by Minimizing Contrastive Divergence", *Journal of Neural Computation*, pp. 1771-1800, 2002.

[13] G. E. Hinton, S. Osindero and Y. W. Teh. "A Fast Learning Algorithm for Deep Belief Nets", *Journal of Neural Computation*, pp. 1527-1554, 2006.

[14] A. Humayun, O. M. Aodha and G. J. Brostow. "Learning to find occlusion regions", *CVPR*, pp. 2161-2168, 2011.

[15] S. Ince and J. Konrad. "Occlusion-Aware View Interpolation", *EURASIP J. Image and Video Processing*, 2008.

[16] N. Jacobson, Y. Freund and T. Q. Nguyen. "An Online Learning Approach to Occlusion Boundary Detection", *IEEE Transactions on Image Processing*, pp. 252-261, 2012.

[17] K. Kavukcuoglu, M. A. Ranzato, R. Fergus and Y. LeCun. "Learning invariant features through topographic filter maps", *CVPR*, pp. 1605-1612, 2009.

[18] Y. H. Kim, A. M. Martnez and A. C. Kak. "Robust motion estimation under varying illumination", *Image Vision Comput.*, pp. 365-375, 2005.

[19] V. Kolmogorov and R. Zabih. "Computing Visual Correspondence with Occlusions via Graph Cuts", *ICCV*, pp. 508-515, 2001.

[20] H. Larochelle, D. Erhan, A. C. Courville, J. Bergstra and Y. Bengio. "An empirical evaluation of deep architectures on problems with many factors of variation", *ICML*, pp. 473-480, 2007.

[21] K. P. Lim, A. Das and M. N. Chong. "Estimation of Occlusion and Dense Motion Fields in a Bidirectional Bayesian Framework", *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 712-718, 2002.

[22] D. R. Martin, C. Fowlkes and J. Malik. "Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues", *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 530-549, 2004.

[23] R. Memisevic and G. E. Hinton. "Learning to Represent Spatial Transformations with Factored Higher-Order Boltzmann Machines", *Journal of Neural Computation*, pp. 1473-1492, 2010.

[24] R. Memisevic, C. Zach, G. E. Hinton and M. Pollefeys. "Gated Softmax Classification", *NIPS*, pp. 1603-1611, 2010.

[25] R. Memisevic and G. E. Hinton. “Unsupervised Learning of Image Transformations”, *CVPR*, 2007.

[26] Y. Nakamura, T. Matsuura, K. Satoh and Y. Ohta. “Occlusion Detectable Stereo - Occlusion Patterns in Camera Matrix”, *CVPR*, 1996.

[27] S. A. Niyogi. “Detecting Kinetic Occlusion”, *ICCV*, pp. 1044-1049, 1995.

[28] M. Piccardi. “Background subtraction techniques: a review”, *SMC*, pp. 3099-3104, 2004.

[29] T. Poggio. “How the ventral stream should work”, *Nature Proceedings*, 2011.

[30] M. Proesmans, L. J. Van Gool, E. J. Pauwels and A. Oosterlinck. “Determination of Optical Flow and its Discontinuities using Non-Linear Diffusion”, *ECCV*, pp. 295-304, 1994.

[31] M. A. Ranzato, A. Krizhevsky and G. E. Hinton. “Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images”, *Journal of Machine Learning Research*, pp. 621-628, 2010.

[32] M. A. Ranzato and G. E. Hinton. “Modeling pixel means and covariances using factorized third-order boltzmann machines”, *CVPR*, pp. 2551-2558, 2010.

[33] M. E. Sargin, L. Bertelli, B. S. Manjunath and K. Rose. “Probabilistic occlusion boundary detection on spatio-temporal lattices”, *ICCV*, pp. 560-567, 2009.

[34] S. Soatto. “Actionable information in vision”, *ICCV*, pp. 2138-2145, 2009.

[35] A. N. Stein and M. Hebert. “Combining Local Appearance and Motion Cues for Occlusion Boundary Detection”, *BMVC*, 2007.

[36] A. N. Stein and M. Hebert. “Occlusion Boundaries from Motion: Low-Level Detection and Mid-Level Reasoning”, *International Journal of Computer Vision*, pp. 325-357, 2009.

[37] C. Strecha, R. Fransens and L. J. Van Gool. “A Probabilistic Approach to Large Displacement Optical Flow and Occlusion Detection”, *ECCV Workshop SMVP*, pp. 71-82, 2004.

[38] J. Sun, Y. Li, S. Bing and K. H. Y Shum. “Symmetric stereo matching for occlusion handling”, *CVPR*, pp. 399-406, 2005.

[39] G. Sundaramoorthi, P. Petersen, V. S. Varadarajan and S. Soatto. “On the set of images modulo viewpoint and contrast changes”, *CVPR*, 2009.

[40] J. Susskind, G. E. Hinton, R. Memisevic and M. Pollefeys. “Modeling the joint density of two images under a variety of transformations”, *CVPR*, pp. 2793-2800, 2011.

[41] G. W. Taylor, R. Fergus, Y. LeCun and C. Bregler. “Convolutional Learning of Spatio-temporal Features”, *ECCV*, pp. 140-153, 2010.

[42] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P. A. Manzagol. “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion”, *Journal of Machine Learning Research*, pp. 3371-3408, 2010.

[43] J. Xiao, H. Cheng, H. S. Sawhney, C. Rao and M. A. Isnardi. “Bilateral Filtering-Based Optical Flow Estimation with Occlusion Detection”, *ECCV*, pp. 211-224, 2006.

[44] W. Y. Zou, A. Y. Ng, S. Zhu and K. Yu. “Deep Learning of Invariant Features via Simulated Fixations in Video”, *NIPS*, pp. 3212-3220, 2012.