# Stereopsis via deep learning

**Roland Memisevic**[*], **Christian Conrad**
Department of Computer Science
University of Frankfurt
Germany

## Abstract

Estimation of binocular disparity in vision systems is typically based on a matching pipeline and rectification. Estimation of disparity in the brain, in contrast, is widely assumed to be based on the comparison of local phase information from binocular receptive fields. The classic binocular energy model shows that this requires the presence of local quadrature pairs within the eye which show phase- or position-shifts across the eyes. While numerous theoretical accounts of stereopsis have been based on these observations, there has been little work on how energy models and depth inference may emerge through learning from the statistics of image pairs. Here, we describe a probabilistic, deep learning approach to modeling disparity and a methodology for generating binocular training data to estimate model parameters. We show that within-eye quadrature filters occur as a result of fitting the model to data, and we demonstrate how a three-layer network can learn to infer depth entirely from training data. We also show how training energy models can provide depth cues that are useful for recognition. We also show that pooling over more than two filters leads to richer dependencies between the learned filters.

## 1 Introduction

To infer 3-dimensional structure from two or more images, it is necessary to establish which points in the different images depict the same 3-D location. There are at least two ways to approach this task: 1.) For each position in one image find a nearby *matching* point in the other image using some measure of similarity between local image patches (e.g. [23]). 2.) For each position in both images, extract features that describe phase and frequency content of the region around that point, and read off the *phase difference* across the two images from the set of filter responses.

While the first approach is more common in practice, the second is more plausible biologically, as it does not require loops over local patches. The most well-known account of phase-based disparity estimation is the *binocular energy* or *cross-correlation* model (e.g. [17], [5], [19]). In its most basic form, this model states that local disparities are encoded in the sum of the squared responses of two neurons, each of which has a binocular receptive field. Each binocular receptive field, in turn, shows a position-shift *across* the two eyes. Between them the two receptive fields show a quadrature relationship (within each eye). It can be shown that the position-shift across the eyes allows the energy model to encode local disparity, while the quadrature relationship within each eye allows it to be independent of the Fourier phase of the local stimulus [19, 4]. The requirement to combine exactly two quadrature pairs to estimate disparity has been challenged in favor of more complex models that pool over a larger set of phases and frequencies in roughly the same orientation [5]. In the monocular case, the presence of quadrature pairs itself has also been challenged in favor of the presence of additional inhibitory connections with roughly orthogonal orientation [21].

[*] ro@cs.uni-frankfurt.de

The energy model is widely accepted as a plausible standard model for stereopsis. However, in stark contrast to standard (monocular) receptive field learning and sparse coding (e.g. [10]), there has been little work on how the model may emerge from looking at training data, and even less work on how it may be applied in any real-world scenario. Here we describe an approach to *learning* local phase relationships between images from data. Our idea is based on the observation that many biological vision systems, and also many engineered systems, perform depth perception either with a *fixed* stereo rig that does not change over time, or by *fixations* on locally approximately smooth surfaces [13]. In both cases, the statistics of the relationship between left and right view are governed almost exclusively by variations in depth across a scene, raising the question whether it is possible to *learn* this relationship from data — and depth or disparity information with it.

### 1.1 Related Work

Some work on sparse coding in the context of depth estimation has been reported in the past. [27] apply sparse coding techniques to disparity and depth maps and show how this makes it possible to denoise the disparity estimates. Here, we consider the different task of learning to relate the sparse codes of *two* image patches. As we shall discuss, this requires multiplicative interactions between the codes and thus conventional sparse coding techniques do not apply.

Learning to encode binocular, as well as temporally related, image pairs has been considered using various conventional sparse coding techniques applied to the *concatenation* of the two images [10]. An overview and some examples are given in [10]. However, as we review and confirm experimentally below, hidden variables are dependent on the Fourier phase of the stimulus in this type of model and thus cannot encode disparity well.

Recently, several bi-linear (and non-linear) methods to learning image relations have been proposed (for example, [18], [16], [24], [28], [6]). Interestingly, it has been observed that many of these models tend to encode global transformations, including global shifts over a whole patch well, but do not tend to develop localized receptive fields (see, for example, [28], [16]). As a result, these methods have been applied predominantly to modeling global image- or patch-transformations (such as affine). We find that training on whitened data is crucial to be able to learn local transformations. A similar observation has been made by [9] who describe how phase-invariant filters can emerge when training an energy model on shifted images.

[26] also describe an approach to stereo sparse coding. Our work differs from theirs in that it is not based on explicitly *modeling* the depth-induced geometric relationships between images, but rather on learning such relationships from data.

## 2 Learning a Binocular Cross-Correlation Model

Given a pair of binocular ("left" and "right") image patches, $I_L(x)$ and $I_R(x)$, at position $x$, the task of depth inference amounts to estimating the position shift that each pixel in $I_L$ has to undergo in order to move to its representation in $I_R$. It is often assumed, for simplicity, that $I_L(x) = I_R(x-d)$, that is, the right patch is a shifted version of the left patch [5] with some offset $d$. A more realistic assumption is that $I_L(x)$ is an affine transformation of $I_R(x)$, as this allows for modeling locally slanted, rather than constant-distant, surfaces [7]. We make this assumption in this work.

One way to obtain an estimate of the disparity at $x$ is by computing the responses of Gabor functions $f_L^0(I_L(x))$ and $f_R^0(I_R(x))$ in the left and right patch, respectively, which are tuned to the same frequency and orientation, and which differ only with respect to their phases and/or exact positions. We shall abbreviate the responses $f_L^0(x)$ and $f_R^0(x)$.

The *product* $p_0 = f_L^0(x) \cdot f_R^0(x)$ will tend to be large for patch pairs that are shifted by exactly the amount that aligns the filters. However, since the product (much like the sum) will also depend on the *content* of the patches, $p_0$ is not a good estimate of disparity [19].

It is possible to reduce that dependency, however, by adding two further Gabor filters $f_L^1(x)$ and $f_R^1(x)$, which are in quadrature relationship with $f_L^0(x)$ and $f_R^0(x)$, respectively ([19, 5]). Now, the *sum of the two products*

$$c(x) = f_L^0(x) \cdot f_R^0(x) + f_L^1(x) \cdot f_R^1(x) \tag{1}$$

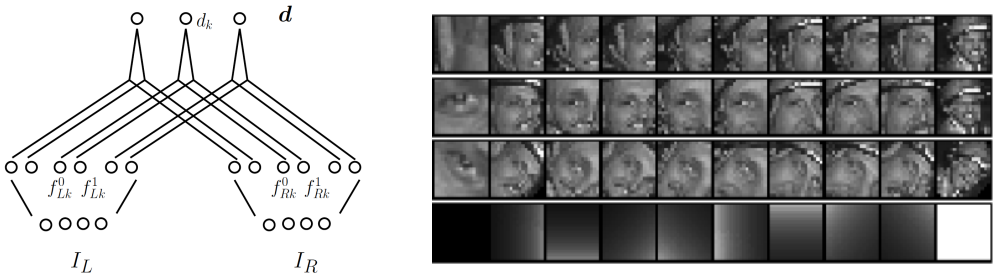


Figure 1: **Left:** Schematic representation of the binocular Boltzmann machine, resembling the basic disparity cross-correlation model. Each latent variable $d_k$ receives activation from the product of two pairs of filters $(f^0_{Lk}, f^0_{Rk})$ and $(f^1_{Lk}, f^1_{Rk})$. **Right:** Example training data: From top to bottom: Row 1, row 2 and row 3 show rendered image planes for the left and right camera, where in row 3 the right camera has been rotated by $45°$ around the $z$ axis. These rendered images are based on the depth maps shown in row 4 and a randomly selected texture map from the Berkeley Segmentation Database [14]. Depth maps are coded as gray value images, where white denotes depth $d_{min} = 1$ (closer to the camera) and black $d_{max} = 6$ (farther away from the camera).

is proportional to $\cos(\phi_{LR})$, where $\phi_{LR}$ is the angle (in 2d) between the 2d-projection of the left and right image patch, respectively, onto each quadrature pair. Since the $\cos$ function is maximal when its argument is zero, the response, $c(x)$, is maximal, when the quadrature filters are able to exactly align the patches.

The idea of using two quadrature pairs to detect disparity is known as the *binocular cross-correlation model* of disparity (e.g. [5, 2]). Instead of computing $c(x)$ using two products (Eq. 1), one can equivalently compute it implicitly as the sum $(f^0_L(x) + f^0_R(x))^2 + (f^1_L(x) + f^1_R(x))^2$, which will be the same up to four additional square terms that do not change the meaning of $c(x)$ if images are contrast normalized [5]. This model is known as the *binocular energy model* [17].

Energy models do not require explicit products, so they are sometimes preferred over Eq. 1. However, efficient learning methods exist for the cross-correlation model, and they permit inhibitory connections, as we shall discuss. In general, the question which form is more plausible biologically has not been settled nor has the question whether squaring of filter components can be achieved more plausibly by a suitable transfer function or by local products [15].

[5] among others suggested that performing disparity estimation based on exact quadrature components can be suboptimal. They argue that pooling over more than a pair is more appropriate due to the presence of discontinuities of the Gabor phase differences across different scales (i.e. frequencies). [21] describe a monocular model, which also provides further evidence in support of this argument. Their approach furthermore suggest that besides pooling, inhibitory connections may improve the estimates and thereby the performance of the model.

### 2.1 Disparity as a Latent Variable

A typical cross-correlation/energy model consists of a set of $K$ left quadrature filters $f^0_{Lk}, f^1_{Lk}$ and $K$ corresponding right filters $f^0_{Rk}, f^1_{Rk}$, with $k = 1, \ldots, K$. The filters are chosen, such that they show various phases, positions and frequencies [5, 19, 22].

In any such system, disparity will manifest itself as a *pattern of matching filter responses*, where, due to smoothness, similar Gabor-pairs will typically show similar responses, and furthermore, a change in one Gabor parameter, such as frequency or phase, can be offset to some degree by an opposite change in another parameter, such as position. [5] demonstrate how these ambiguities can lead to instabilities in disparity estimates and hypothesize that any phase-based disparity estimate should involve *pooling* over multiple energy/cross-correlation units.

With this prospect in mind, we define a probabilistic model of disparity as follows: We introduce *binary latent variables* $d_k$ $(k = 1, \ldots, K)$ each of which encodes the probability that an associated quadrature pair $(f^0_{Lk}, f^0_{Rk})$, $(f^1_{Lk}, f^1_{Rk})$ constitutes a match. As such, each instantiation of the *set of*

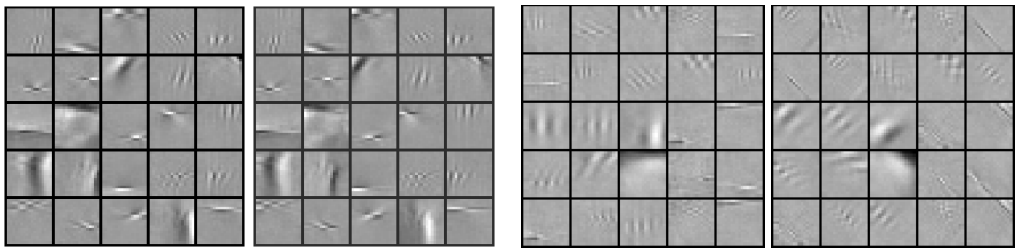

Figure 2: Filters learned from rectified cameras **(left)** and rotated cameras **(right)**.

*latent variables* can be thought of as a population code, that represents the observed disparity. In the following, we shall stack the set $d_k$ of latent variables in the vector $\boldsymbol{d}$.

Similar to [16, 20] we can then define the joint probability distribution over patch-pairs *and* disparity variables using the "Boltzmann" distribution[1]

$$p\left(I_L, I_R, \boldsymbol{d}\right) = \frac{1}{Z} \exp\left(\sum_k d_k \big(f^0_{Lk}(I_L) f^0_{Rk}(I_R) + f^1_{Lk}(I_L) f^1_{Rk}(I_R)\big)\right), \tag{2}$$

$$\text{with} \quad Z = \sum_{\boldsymbol{d}} \sum_{I_L, I_R} \exp\left(\sum_k d_k (f^0_{Lk}(I_L) f^0_{Rk}(I_R) + f^1_{Lk}(I_L) f^1_{Rk}(I_R))\right). \tag{3}$$

Under this model, the conditional probability over disparities $p(\boldsymbol{d}|I_L, I_R)$ is given by

$$p(\boldsymbol{d}|I_L, I_R) = \prod_k \frac{1}{1 + \exp\left(-\left(f^0_{Lk}(I_L) f^0_{Rk}(I_R) + f^1_{Lk}(I_L) f^1_{Rk}(I_R)\right)\right)}. \tag{4}$$

The model is a Boltzmann machine [8]. More specifically, it is a type of Boltzmann machine modeling *pairs* [24]. Training amounts to finding model parameters (filters) that maximize the average log-probability ($\log p(I_L, I_R) = \sum_{\boldsymbol{d}} p(I_L, I_R, \boldsymbol{d})$) for a set of training image pairs, $(I_L, I_R)$. Like in similar models one can use gradient-based optimization and contrastive divergence for training [8, 24], or train the model like an auto-encoder or using score-matching [25, 10]. In contrast to recent work on higher-order restricted Boltzmann machines [20], here it is not necessary to constrain the parameters to be positive. In fact, we can allow for inhibitory connections from the filter responses to disparity codes as we shall show. During inference, each latent variable receives activity from exactly two products of matched filter-responses (cf. Eq. 4). Figure 1 (left) shows the corresponding graphical model. One can incorporate *pooling* as suggested by [5], by replacing the "energy" inside the $\exp$-function (Eq. 2) with the more flexible form

$$\sum_k d_k \sum_{l \in P_k} w_{lk} f^l_{Lk}(I_L) f^l_{Rk}(I_R), \tag{5}$$

where $P_k$ is the set of filter pairs that latent disparity-variable $k$ connects to and $w_{lk}$ are parameters that weight the relative contribution of the $l^{\text{th}}$ filter-pair for latent variable $k$. These parameters can be learned from training data along with the filters. When using pooling, each latent variable receives activity from more than just two filter-products. Thus, unlike Eq. 1 not only quadrature pairs but larger sets of related filters can connect to one latent disparity variable. Note that the connections $w_{lk}$ need not be positive. Therefore, the contribution of a particular filter-pair may be *inhibitory* (cf. Eq. 4). We found that learning leads to inhibitory connections, which tend to differ with respect to phase, position and frequency from the excitatory connections.

## 3 Methods and Data-Generation

In contrast to standard (monocular) feature learning (see, e.g. [10] and references therein), acquiring training data for binocular feature learning is more challenging. Due to fixations and vergence,

[1] Similar to [16, 20] one can add "bias" terms in the definition of probability. Alternatively one can think of all variables in homogeneous notation, for example, using an additional constant "1"-dimension, or add "confinement" terms for real-valued observables [16].

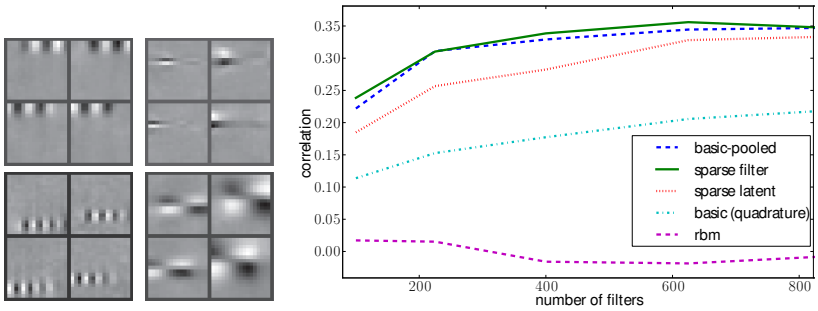


Figure 3: **Left:** Filters learned from rectified camera setup. Each group of four filters corresponds to one latent variable. Top row within each group of four: Left camera filters, Bottom row: Right camera filters. There is no pooling, thus each latent variable is connected to exactly four filters (two for each camera). Learning leads to within-camera quadrature pairs and across-camera position shifts. **Right:** Performance of various models on the task of predicting disparity from the activities of latent variables where `basic-pooled` and `basic (quadrature)` refers to the presented model incorporating pooling and without pooling, respectively. See text for details.

biological receptive fields are confronted mainly with locally smooth surfaces [7, 10]. In contrast to the monocular case, cropping patches from pairs of images is therefore not possible. Furthermore, in contrast to the task of denoising *pre-computed* depth-structure monocularly [27], one also cannot utilize databases of natural depth-estimates.

In this work, our goal is to demonstrate that we can get depth-estimates entirely through learning from raw stereo images. We therefore generated our own training data with known ground-truth disparities. Note that existing stereo datasets, like NORB and Middlebury, do not come with ground-truth, or are much too small for learning depth inference from data alone. However, in section 4 we will use the NORB dataset to show that one can learn depth cues *implicitly* and utilize them for recognition. We use the following approach to generating binocular image data: (i) generate a depth map as a slanted plane in 3D space, (ii) generate a texture map as a patch from a natural image of the same size as the depth map and (iii) project the 3D scene onto a set of cameras. We model the cameras as standard pinhole cameras, defined by their camera matrix $\mathbf{P_1}, \mathbf{P_2} \in \mathbb{R}^{3\times 4}$, respectively (see [7] for details). Depth planes are generated by uniformly sampling planes in 3D parameterized by the camera matrices to constrain the minimum and maximum depth value to lie within $[d_{min}, d_{max}]$. While 3D structure can be modeled sufficiently well using locally smooth surfaces, texture should be modeled according to the statistics of natural images, e.g., by cropping patches of natural images. The assumption underlying this approach is that appearance is independent of depth. While this is not true in general, it is a good approximation for the purpose of data-generation. Note that in the 3D scenes generated this way, no occlusion effects, and no artificial edges are induced in the rendered images, as the depth maps are always smooth. See Fig. 1 (right) for example data.

### 3.1 Learning Filter Pairs

We performed experiments on 2 data sets, containing $650.000$ image pairs, that have been generated as described. In the first data set, the cameras have the same orientation as the world reference frame, while the first camera coincides with the world frame the second camera is shifted by 15 pixels along the $x$ axis. The image planes are of size $17 \times 17$ pixels. We generated $650$ depth maps, where the scene depth varies between $d_{min} = 1$ and $d_{max} = 7$. For every depth map $1000$ texture maps of the same size were created by randomly cropped patches from natural images taken from the Berkeley Database [14] and PASCAL database [3] . The second data set was generated in the same way, but now the second camera is rotated by $45°$ around the $z$ axis.

We trained the basic model (cf. Fig. 1 (left) and Eq. (2)) containing no pooling and an extended model incorporating pooling, where the "energy" term in Eq. (2) was replaced with Eq. (5). Recall that in the basic model each latent variable is connected to exactly four filters — two in each image whereas in the model using pooling each latent variable can connect to more that four filters in total.

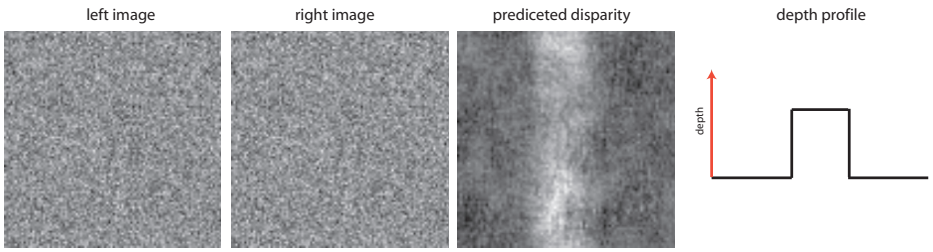


Figure 4: Learning to solve stereograms.

Within the results presented here, for both data sets, the model incorporating pooling was trained using $F = 1024$ factors and $K = 256$ hidden units. We also experimented with different numbers of these parameters in the range of 1000 to 1600 factors and 100 to 260 hidden units and found that the exact number did not have a strong influence on the learned filters. Based on contrastive divergence learning with a learning rate of typically $0.01$ and a batch size of $128$, the model was trained for $10$ to $100$ epochs. These parameters apply to the basic model as well with the additional constraint that each latent variable connects to exactly four filters (factors).

**Learning a Simple Cross-correlation Model** We trained an instantiation of the basic model containing no pooling. One question we wanted to address is whether the probabilistic model learns within-camera quadrature pairs. Figure 3 shows a representative set of learned left and right camera filters. While filters tend to come in within-camera quadrature pairs and mainly position-shifts across cameras, there is an additional type of variability, which we noted is stable across multiple runs: There is within-camera variability in the direction *orthogonal* to the motion direction. In this case, this is the vertical direction, since motion direction is horizontal in this type of data. The variability affects both position and frequency. An example of a within-camera position shift is the lower-left group, an example of a within-camera frequency change the lower-right group.

**Learning on Rectified Camera Setups** Figure 2 (two left most columns) shows as subset of $25$ learned input and output filters $f_{Lk}$ and $f_{Rk}$, respectively, based on data set 1 (rectified camera setup) using model and training parameters as described before. A representative subset of filters has been chosen randomly, using the same indices for input and output filters. It can be seen that our model extracts Gabor like filters at different frequencies and orientations, overall covering the whole image planes fairly evenly. Note that, while the orientation and frequency of pairs of input and output are very similar, pairs of filters differ by a shift, which is due to the baseline of the camera setup. Filters at different frequencies occur due to the (varying) scene depth (see also Fig. 1).

**Learning on Rotated Cameras** Figure 2 (two right most columns) shows as subset of $25$ learned input and output filters $f_{Lk}$ and $f_{Lk}$, respectively, based on data set 2 using the same model and training parameters as described before. Again, the subset of filters has been chosen randomly. Remember that for data set 2, the right camera has been rotated about $45^\circ$ around the $z$ axis. It can be seen that our model extracts Gabor like filters as well, with the difference that pairs of filters not only differ in a shift but are also related by a rotation when compared to the results obtained for data set 1.

## 4 Experiments

**Predicting depth** As the latent variables $d_k$ encode disparity indirectly, we need to learn to relate the model generated codes $\boldsymbol{d}$ with the given ground truth disparities of our data set. There are various approaches to extracting disparity or depth from binocular filter responses [5, 19, 22]. Here we use the simplest one, which is linear regression trained using depth $\boldsymbol{d}$ inferred for each patch pair with Eq. (4) and the available ground-truth on the training data. Performing linear regression on $\boldsymbol{d}$ is equivalent to adding an extra layer to the network. Thus, the overall network is a three-layer network, that can be trained with Hebbian-like learning, and whose intermediate layer contains multiplicative interactions. Figure 4 shows an example of depth inference for a random dot stereogram based on our cross-correlation model incorporating pooling. The images are of size $100 \times 100$ pixels. To infer depth across the whole image, we apply the model convolutionally: We predict disparity at

| | NORB TRAINING SUBSET: | | | | NORB TESTSET: | | |
|---|---|---|---|---|---|---|---|
| METHOD | RBMMON | RBMBIN | CC | CC+BIN | RBMBIN | CC | CC+BIN |
| ERROR RATE (%) | 73.65 | 60.43 | 34.85 | 31.48 | 63.28 | 38.91 | 36.80 |

Table 1: Classification accuracy on NORB. See text for details.

the center pixel of every $17 \times 17$ sub-patch. The figure shows that the model correctly infers the depth profile encoded in the image pair. We predict depth at each site independently, and we do not perform any smoothing or MRF inference to clean up the predictions.

Figure 3 (right) shows the Pearson correlation on hold-out data as a function of the number of filters used. A particularly interesting conclusion in line with [5] is that pooling has a distinctly positive effect on the predictability of disparity (`basic-pooled` in the plot). We included the "quadrature" model (=basic cross-correlation) in the comparison, where each latent variable is connected to exactly two filters in each camera (for a total of $4$ filters).

We also trained a model on the concatenation of left/right patches for comparison. Due to the conceptual similarity to the models discussed, we used a restricted Boltzmann machine (`rbm` in the plot) with parameter settings comparable to those of the other models used in this experiment. In particular, we used a number of hidden units equal to the number of latent variables in the corresponding pooled cross-correlation model. This shows that multiplicative interactions are a key ingredient to encode disparity. Finally, for the basic-pooled model, we experimented with sparsifying penalties to sparsify (a) the disparities (using the approach described in [12], `sparse-latent` in the plot), and (b) the filter responses (by adding a small multiple of the average $l_1$-norm of the filter responses to the objective function, `sparse-filter` in the plot). Interestingly, in neither case there was a noticeable effect on the prediction performance, which is in contrast to recent findings on object recognition (for example, [1]).

**Learning implicit depth cues** The reason that depth inference is a common task in computer vision is that depth is useful in a large number of applications, including recognition, obstacle detection and segmentation. In this section, we show that, in contrast to traditional disparity estimation pipelines, depth features may be used as *cues* along with additional information in order to improve 3-D object recognition performance. We use the NORB-cluttered-jittered dataset [11], that was introduced for the purpose of evaluating 3-D recognition models. In this data, depth cues are extremely valuable, because they make it much easier to distinguish between true objects and background clutter than monocular image features [11]. The dataset consists of $291600$ training image pairs with a spatial resolution of $108 \times 108$ pixels showing objects of $5$ different classes on cluttered background. There is a sixth class, *no-object*, showing just background clutter. Object position and orientation vary across examples (see Figure 5 (left) for some example image pairs). The NORB test-set consists of $58200$ image pairs showing different instances of the same object *types* (not the same exact objects). So this is a (mild) form of transfer learning.

Figure 5 (right) shows filter pairs learned from the NORB training dataset. We used $1024$ depth features (= filter pairs) and $256$ pooling units (=$d_k$'s), and we PCA-whitened the images before training (using a single whitening basis for left and right views, and retaining $1364$ features, which amounts to $95\%$ of the variance). The figure shows that filters are highly localized and account for the differences in camera orientation through across-image phase-shifts. We also trained a "binocular" Restricted Boltzmann Machine (RBM) with the same number of hidden units on the concatenated image pairs.

Table 1 (left) compares the classification performance of the monocular RBM (`RBMMON`), binocular RBM (`RBMBIN`), the proposed pooled cross-correlation model (`CC`) and the proposed pooled cross-correlation model combined with the binocular RBM (`CC+BIN`) on a (hold-out) subset comprising $29100$ cases from the training data set ("no transfer learning"). We used a simple logistic regression model with $l_2$-regularization penalty ($\lambda \|W\|_F$ where $W$ are the linear logistic regression parameters) and chose $\lambda$ from the set $\{0.1^a\}_{a=1,\ldots,5}$ using a validation set. Both training and validation set consist of $29100$ randomly chosen subset of the NORB training set. The monocular model performs badly (on this data-set $83\%$ correct would be the performance of random guessing). This is in line

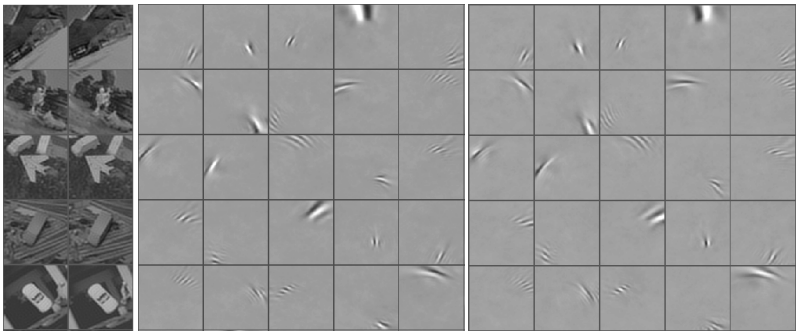

Figure 5: **Left:** Example image pairs from the NORB-cluttered-jittered dataset. **Right:** Learned binocular filter pairs.

with [11] who found that monocular features are not very useful for this difficult data-set. However, the table shows also that the RBM trained on the concatenation of views performs only slightly better. Including multiplicative interactions, in contrast, yields much better performance, and so does the combination of features from both types of model (right-most figure in the table).

Table 1 (right) shows the classification performance on the test set ("transfer learning"). We trained the classifier on a subset of 10000 training cases and we show the performance on the whole test-set. The table shows that the plain RBM features, again, perform much worse than the depth-features. It also shows that combining the two types of features yields better performance than using just one type of feature, showing that the two different models do in fact encode different types of structure in the data. It is interesting to note that the proposed model combined with simple linear classification on a few hundred dimensional feature vector can achieve recognition rates better than a large, monocular convolutional network (see [11] for details).

## 5 Discussion

We described a way to cast binocular disparity estimation as a deep belief network, and we showed how learning on data-bases of multi-camera views of a scene leads to shifted Gabor filter pairs, that are well suited to extract depth information from image pairs. This shows that depth inference can emerge naturally in a three-layer network with multiplicative interactions which is trained with simple Hebbian learning.

Most current practical work on stereopsis focusses on extracting dense depth-maps using MRFs. Potential reasons for biology to take a different route might be that (a) depth via deep learning makes it possible to use the exact same learning algorithm for depth inference that is also used to recognize objects (and probably also motion); (b) often, a simple depth cue, as given in some population code, $\boldsymbol{d}$, is entirely sufficient to take swift vital decisions, such as to dodge an approaching object; (c) learning depth inference from data allows for feed-forward depth perception, and thus to avoid the need for a complicated and brittle pipeline, which involves rectification, hypothesis generation, and robustification using RANSAC.

In this paper we showed how unsupervised feature learning may be used to mimick this way of extracting depth maps from image pairs, and that it allows us to compute implicit depth cues from binocular data for use in object recognition tasks, showing that biologically consistent depth inference is achievable with the right types of non-linearity.

## References


[1] Coates A., H. Lee, and A. Ng. An analysis of single-layer networks in unsupervised feature learning. In *Artificial Intelligence and Statistics*, 2011.

[2] P. A. Arndt, H.A. Mallot, and H.H. Bulthoff. Human stereovision without localized image features. *Biological Cybernetics*, 1995.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge. *IJCV*, 2010.

[4] D. Fleet, H. Wagner, and D. Heeger. Neural encoding of binocular disparity: Energy models, position shifts and phase shifts. *Vision Research*, 36(12):1839–1857, June 1996.

[5] D. J. Fleet, A. D. Jepson, and M. R. M. Jenkin. Phase-based disparity measurement. *Image Underst.*, pages 198–210, 1991.

[6] D. Grimes and R. Rao. Bilinear sparse coding for invariant vision. *Neural Comp.*, 2005.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[8] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comp.*, 2000.

[9] A. Hyvärinen and P. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Comp.*, pages 1705–1720, 2000.

[10] A. Hyvarinen, J. Hurri, and Patrik O. Hoyer. *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*. Springer Verlag, 2009.

[11] Y. LeCunn, F.-J. Huang, and L. Bottou. Learning Methods for generic object recognition with invariance to pose and lighting. In *CVPR*, 2004.

[12] H. Lee, C. Ekanadham, and A. Ng. Sparse deep belief net model for visual area v2. *NIPS*, 20:873–880, 2007.

[13] Y. Liu, L. K. Cormack, and A. C. Bovik. Natural scene statistics at stereo fixations. In *Symposium on Eye-Tracking Research and Applications*, pages 161–164, 2010.

[14] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*. IEEE Computer Society, 2001.

[15] B. W. Mel, D. L. Ruderman, and K. A. Archie. Toward a single-cell account for binocular disparity tuning: an energy model may be hiding in your dendrites. In *NIPS*, 1997.

[16] R. Memisevic and G. E. Hinton. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Comp.*, pages 1473–1492, 2010.

[17] I. Ohzawa, G. C. Deangelis, and R. D. Freeman. Stereoscopic Depth Discrimination in the Visual Cortex: Neurons Ideally Suited as Disparity Detectors. *Science*, 1990.

[18] B. Olshausen, C. Cadieu, J. Culpepper, and D. Warland. Bilinear models of natural images. In *Human Vision Electronic Imaging*, 2007.

[19] N. Qian. Computing stereo disparity and motion with known binocular cell properties. *Neural Computation*, 6:390–404, May 1994.

[20] M. Ranzato and G. E. Hinton. Modeling Pixel Means and Covariances Using Factorized Third-Order Boltzmann Machines. In *CVPR*, 2010.

[21] N. C. Rust, O. Schwartz, J. A. Movshon, and E. P. Simoncelli. Spatiotemporal Elements of Macaque V1 Receptive Fields. *Neuron*, pages 945–956, 2005.

[22] T. Sanger. Stereo disparity computation using Gabor filters. *Biological Cybernetics*, 59:405–418, 1988. 10.1007/BF00336114.

[23] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47, 2002.

[24] J. Susskind, R. Memisevic, G. Hinton, and M. Pollefeys. Modeling the joint density of two images under a variety of transformations. In *CVPR 2011*, 2011.

[25] K. Swersky, M. Ranzato, D. Buchman, B. Marlin, and N. Freitas. On autoencoders and score matching for energy based models. In *ICML*, 2011.

[26] I. Tosic and P. Frossard. Dictionary learning in stereo imaging. *IEEE Transactions on Image Processing*, 20(4), 2011.

[27] I. Tosic, B. A. Olshausen, and B. J. Culpepper. Learning sparse representations of depth. *CoRR*, 2010.

[28] C. M. Wang, J. Sohl-Dickstein, I. Tosic, and B. A. Olshausen. Lie group transformation models for predictive video coding. In *DCC*, 2011.