

# Second-Order Statistics of Natural Images

Hauke Bartsch and Klaus Obermayer

*Dept. of Electrical Engineering and Computer Science,  
Technical University Berlin, Germany, E-mail: hauke@cs.tu-berlin.de*

---

**Abstract.** Assuming adaptation of the visual cortex to its environment, we analyse the invariance structure found in natural images to explain the selective response of neurons to different modalities.

We argue that the invariant structure of images can be formally expressed by dot-product kernels. After learning these kernels using natural images we show, how they can be analysed by an eigen-sub-space analysis. In addition to localized edge detectors we found neurons that respond to changes in texture and cells that detect edge curvature. The analysis suggests new types of features neurons in primary visual cortex may be selective to.

*Key words:* symmetry, edges, texture, curvature

---

## 1 Introduction

Under the assumption that the visual cortex adapts to its environment, the invariances found in natural images offer a likely explanation for the emergence of cortical wiring patterns and the operations performed therein [1, 3].

By referring to an object as invariant or symmetric we usually indicate that the object appears unchanged if observed from different perspectives. Symmetry of an object must therefore be defined with respect to an operation or transformation.

A high similarity of an object with its transformed counterpart will indicate high symmetry of this object given the transformation.

## 2 One-dimensional Symmetry Detectors

Transformations will be expressed by a square symmetric matrix  $A$  applied onto a data vector  $\mathbf{x}$  ( $A\mathbf{x} = \mathbf{y}$ ). Whereas  $x$  represents an image patch, throughout this paper we will assume that  $\mathbf{x}$  is a vector, appending successive columns. If the transformation  $A$  leaves the structure in  $\mathbf{x}$  unchanged (with respect to its position, form and orientation), we will claim  $A$  as coding the symmetry found in  $\mathbf{x}$ . A well chosen  $A$  with respect to symmetry is indicated by  $A\mathbf{x} \approx x$ . If we restrict  $A$  to leave the length of the vector  $\mathbf{x}$  unchanged, e.g. if  $A$  is a permutation matrix, also a large scalar product of  $A\mathbf{x}$  and  $\mathbf{x}$  also highlights symmetry. Contrarily a small product will indicate an operation for which most of the structure in the image is destroyed by  $A$ . Following from that observation our goal is to find a matrix  $A$  for which its quadratic form gets extremal:

$$\mathbf{x}^T A \mathbf{x} \stackrel{!}{=} \text{extremal.} \quad (1)$$

### 2.1 Analysis

Assuming an ensemble  $X$  of gray level images ( $\mathbf{x} \in X$ ), in a first step we maximize and then we minimize the quadratic form in order to analyse the model.

Using the observation that

$$A\mathbf{x} = \sum_i^N \lambda_i \mathbf{n}_i (\mathbf{n}_i^T \mathbf{x}), \quad (2)$$

where  $\lambda_i$  and  $\mathbf{n}_i$  are the eigenvalues respective normalized eigenvectors of  $A$  we can re-write the quadratic form:

$$\mathbf{x}^T A \mathbf{x} = \mathbf{x}^T \sum_i^N \lambda_i \mathbf{n}_i (\mathbf{n}_i^T \mathbf{x}) = \sum_i^N \lambda_i (\mathbf{n}_i^T \mathbf{x}) (\mathbf{x}^T \mathbf{n}_i). \quad (3)$$

Calculating the expectation of the quadratic form for all data points  $\mathbf{x}$  yields according to above equation

$$\langle \mathbf{x}^T A \mathbf{x} \rangle_x = \left\langle \sum_i^N \lambda_i \mathbf{n}_i^T \mathbf{x} \mathbf{x}^T \mathbf{n}_i \right\rangle_x = \sum_i^N \lambda_i \mathbf{n}_i^T \langle \mathbf{x} \mathbf{x}^T \rangle_x \mathbf{n}_i = \sum_i^N \lambda_i \mathbf{n}_i^T C \mathbf{n}_i. \quad (4)$$

Only the correlation matrix  $C$  of the data will be of importance for defining  $A$  (statistics of second order). More precisely, only the direction of the largest eigenvalues eigenvector of  $C$  has to be taken into account to **maximize**  $A$  (setting  $\lambda_0$  to 1 and  $\lambda_{1..N}$  to 0).  $A$  can be further specified by assuming that also in the space orthogonal to the direction of largest symmetry we want to maximize the quadratic form. Using a successive projection into orthogonal spaces all eigenvectors of  $A$  equal corresponding eigenvectors of  $C$ . Assuming further limited eigenvalues  $\lambda_i$  an obvious solution will be  $A = C$ . Thus choosing  $A$  to be the correlation matrix of the data maximizes the quadratic form.

A similar reasoning can be used to find a transformation  $A$  that **minimizes** the quadratic form. Assuming the eigenvalues of  $A$  to be negative and using an eigenvalue decomposition of  $A$  we easily see that  $-A = -(U^T D U) = U^T (-D) U$  which shows that  $C = -A$  is indeed a minimum of the quadratic form.

Whereas maximizing the quadratic form can be identified with detecting symmetric structure in the data by minimizing the quadratic form we detect the opposite direction – anti-symmetry. Because we are interested in finding both, symmetric and anti-symmetric structures,  $A$  will not be restricted to being either positive or negative definite. We will allow it to simultaneously explore both directions. Thus

matrix  $A$  can be indefinite with positive as well as negative eigenvalues pointing in the respective directions of either symmetry or anti-symmetry.

## 2.2 Rotation, Scaling and Translation by Dot-Product Kernels

Restricting the matrix  $A$  to operations on the index-numbers of  $\mathbf{x}_{1\dots N}$ ,  $A\mathbf{x}$  codes for permutations of the pixel values in  $\mathbf{x}$ . Three matrices  $A^{\text{rot}}$ ,  $A^{\text{sca}}$  and  $A^{\text{shift}}$  are designed now which code for the invariance transformations rotation, scaling and shift.

[Fig. 1 about here.]

In the following, the coordinates of a pixel  $i$  will be denoted by ( $\pi$ -periodic) polar coordinates  $(\theta_i, r_i)$ . To make the analysis simple we will not try to transform the raw image into polar coordinates. This would involve an ambiguous prior on the image structure (for example smoothness) for the re-calculation of the coordinate grid and certainly corrupt our data. Instead, we rely on the equally spaced (cartesian) grid of gray values and express pixel coordinates in terms of their counterpart in polar coordinates. This of course also introduces discretisation errors that work as prior structure.

**Rotation.** The goal is to let  $A^{\text{rot}}$  encode rotations of the image pixels around their mean position. We have to set the matrix  $A$  to the covariance matrix  $C$  obtained from a data set  $X$  that contains a rotationally symmetric figure. Instead of really sampling from such an image ensemble, we incorporate the respective covariance structure directly into the matrix.  $a_{ij}^{\text{rot}}$  ( $i \neq j$ ) will be set to one if  $|r_i - r_j| < \varepsilon$  ( $i \neq j$ ) and otherwise to zero.  $\varepsilon$  was arbitrarily chosen to be the distance of two neighboring pixels on the outer perimeter of the circular receptive field (see fig. 1, left).

[Fig. 2 about here.]

**Scaling.** In a similar way  $A^{\text{sca}}$  is constructed to encode scaling. Entries of matrix  $a_{ij}$  will be set to one only if the angular distance between  $i$  and  $j$  is sufficiently small, i.e.  $|\theta_i - \theta_j| < \varepsilon$  ( $i \neq j$ ) (see fig. 1, middle). We do not take into account that by this we also code for inversion symmetries: true scaling would not allow for pixels to cross the origin. The choice of a specific pixel pair subset does not uniquely define a single symmetry operation.

[Fig. 3 about here.]

**Shift.**  $A^{\text{shift}}$  is constructed to encode the pixel pairs that occur for translations of pixels along one direction (here  $90^\circ$ ).  $a_{ij}$  ( $i \neq j$ ) is set to one only if  $(x_i = x_j)$  and otherwise to zero.

[Fig. 4 about here.]

All prototype pattern found by the analysis (see figures 2, 3 and 4) show maximal responses for images with respective symmetries, circles for rotational symmetries, crosses and fan-like figures for scale invariances and lines for shift invariance.

### 3 Multi-dimensional Symmetry Detection

An object may be symmetric with respect to more than one transformation. Thus, we expect to find different instantiations of transformations to be mixed in an ensemble of example patches from natural images. Therefore, a more appropriate model is to find a number  $i$  of symmetries each one expressed by its own matrix  $A^i$ . Let's define  $\Phi$  as a vector valued function mapping the data into the space of different symmetry operators:

$$\Phi(\mathbf{x}) = (x^T A^1 x, x^T A^2 x, \dots, x^T A^N x)^T \quad (5)$$

Additional assumptions can be made for the distribution of  $\Phi$  incorporating for example sparseness, independence or uncorrelateness. But before doing this, we first cast the problem into the notion of dot-products. This will give us the vantage of analysing a linear model in which assumptions can be justified more easily.

Any quadratic form can be written as a linear weighting in the space of dot-products:

$$\mathbf{x}^T A^i \mathbf{x} = (a_{11}^i x_1 x_1) + \dots + (a_{kk}^i x_n x_m) = (a_{11}^i, \dots, a_{kk}^i)(x_1 x_1, \dots, x_n x_m)^T \quad (6)$$

$$= \mathbf{w}^i \mathbf{s}. \quad (7)$$

Adding more dimensions to the model  $\Phi$ , e.g. more quadratic forms  $x^T A^i x$ , we arrive at

$$\Phi = W \mathbf{s}. \quad (8)$$

Here,  $W$  is a matrix with as many rows as  $\Phi$  has dimensions. It can be viewed as a linear mixture expressed by the mixing matrix  $W$  of the signals  $\mathbf{s}$  in the space of dot-products.

It may not be desirable to choose a square matrix  $W$  because there are dependencies expressed in the dot-products, data may be projected mainly to a lower dimensional manifold. This would leave the matrix  $W$  under-determined. To avoid this and to simplify the computation we reduce the high dimensional dot-product space by an eigenvalue decomposition and keep only the directions in which the variance of the data is largest.

There is no a priori choice for a specific algorithm solving equation 8. The system is ill-posed, thus additional assumptions about the underlying sources are required. Assuming statistical independence of the transformations in space of dot-products lead us in the next section to methods of ICA.

### 3.1 Results

To perform an ICA we used the FastICA Matlab package described in [2] (symmetric approach with  $\tanh$  as non-linearity). 100,000 example image patches  $\tilde{y}$  of size  $7 \times 7$  were extracted at random positions from images depicting natural scenes. We performed no pre-processing of the images except a shift to zero mean for each image separately. We assumed that  $\tilde{y}_i \tilde{y}_j = \tilde{y}_j \tilde{y}_i$  and  $\tilde{y}_i \tilde{y}_i = 0$  together with a circular rather than rectangular receptive field which reduces the number of valid pixel combinations to  $7\pi(7\pi - 1)/2 \approx 700$ . A dimension reduction was performed by doing an eigenvalue decomposition of  $\Phi$  and keeping only the eigenvectors belonging to the 49 largest eigenvalues (this keeps 94% of the variance).

Doing ICA in the space of dot products results in finding independent sources in dot product space. To visualize single components, we aim to find a transformation from the dot product space to the more intuitive image space. We do this by basically repeating the reasoning of section 2.1. There we constructed  $A$  from given data  $\mathbf{x} \in X$  to maximize (or minimize) the quadratic form, now given  $A$ , we search for data  $\mathbf{x}$  that either maximizes or minimizes the quadratic form.

[Fig. 5 about here.]

By looking at the eigenvectors of the matrix  $A^i$  we can decide for which patterns in the image the quadratic form responds. Neglecting eigenvectors with small corresponding eigenvalues in figure 5 each independent component found can be represented by its maximal (columns A) and minimal (columns B) eigenvector. Among others, edges with different orientation, spatial frequency and position are prominent structures found by the model (see figure 5). A further analysis of the found filters is given in the discussion.

## 4 Discussion

It is worthwhile to show that *independent components in image space not solemnly define the independent components in the space of dot-products*. Whereas standard ICA assumes a linear mixing model  $x = As$ , we calculate a linear model in dot-product space:

$$xx^T = Ass^TA^T = \sum_{kl} \underbrace{a_{ik}a_{jl}}_{M_{ijkl}} \underbrace{s_k s_l}_{c_{kl}} = M_{ijkl}c_{kl}. \quad (9)$$

Because the elements of  $c_{kl}$  are not independent from each other, as assumed by ICA, we cannot expect the ICA in dot-product space to find this solution.

*Coding edge detector units with only second-order statistics.* Whereas coding of edges is trivially implemented by a linear filter, our model relies solely on statistics of second order, e.g. products of pixel values. Edges are implemented by a simple mechanism of 'be the same' and 'be different'. The pixels on each side of the edge are pooled together by assigning only positive weights between the pixels. Together they code for the same coherent region of space. Contrarily, the weights between the two groups are negative, thus forcing the two regions to code for different structure.

*Coding of texture properties:* So far we have only looked onto the space of the two most prominent eigenvectors. Figure 6 shows some more eigenvectors of a selected filter.

[Fig. 6 about here.]

As one can see, the two sub-fields of the filter stay largely separated. The filter distinguishes in its response non-textured regions (objects) from textured regions (background) or vice versa.

*Detecting curvature.* More difficult to explain are the filters that are best described as showing a *yin-yang* type pattern. They are stable features in our simulations. We do not expect these patters to be present in natural images but we can give an interpretation to their function in terms of forming a curvature detection unit. For this we look onto the zero-point solution or the eigen-sub-space that is spanned by the largest positive and negative eigenvector  $\gamma \mathbf{n}_0 + (1 - \gamma) \mathbf{n}_N$ ,  $\gamma = [0 \dots 1]$ . As one can see in figure 7, maximum (left) and minimum (right) together constitute an edge detector. Moving from the center ( $\gamma \approx 0.5$ ) to either left or right bends the edge in direction of positive or negative curvature.

[Fig. 7 about here.]

## References

- [1] P. C. Bressloff, J. D. Cowan, M. Golubitsky, P. J. Thomas, and M. C. Wiener. Geometric visual hallucinations, euclidean symmetry, and the functional architecture of striate cortex. *Phil. Trans. Royal Soc. London B*, 356:299–330, 2001.
- [2] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10:626–634, 1999.
- [3] Konrad P. Körding and Peter König. Neurons with two sites of synaptic integration learn invariant representations. *Neural Computation*, 13:2823–2849, 2001.

## List of Figures

1	Connection structure	11
2	Eigenvectors of $A^{\text{rot}}$	12
3	Eigenvectors of $A^{\text{sca}}$	13
4	Eigenvectors of $A^{\text{shift}}$	14
5	Learned symmetry detectors by FastICA	15
6	Texture dependent sub-fields	16
7	Filter that codes for curvature	17

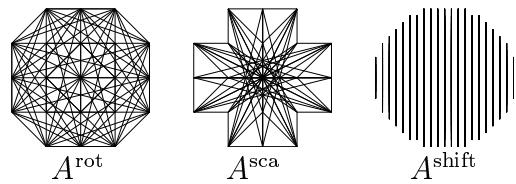


Fig. 1. Connection structure. For every matrix element  $a_{ij} = 1$  the corresponding pixel pair  $(i, j)$  is connected by a line.

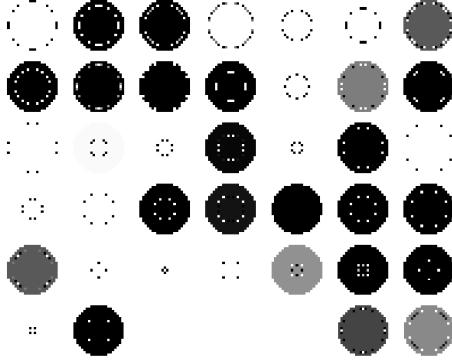


Fig. 2. Eigenvectors of  $A^{\text{rot}}$  sorted in decreasing order from left to right, top to bottom. They explain 51% of the variance in  $A^{\text{rot}}$ . The two eigenvectors after the gap have the largest negative eigenvalues. The ratio of the largest to the smallest eigenvalue is  $-5.1$ .

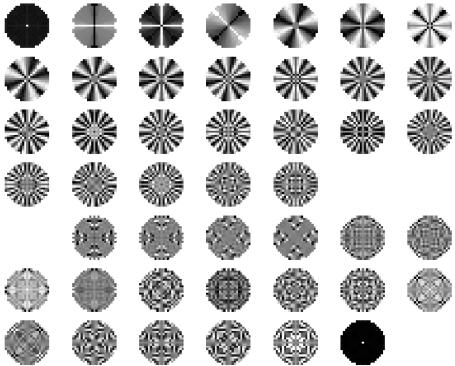


Fig. 3. Eigenvectors of  $A^{\text{sca}}$ . They explain 65% of the variance in  $A^{\text{sca}}$ . All eigenvectors after the gap have negative eigenvalues. The ratio of the largest to the smallest eigenvalue is  $-2.9$ .

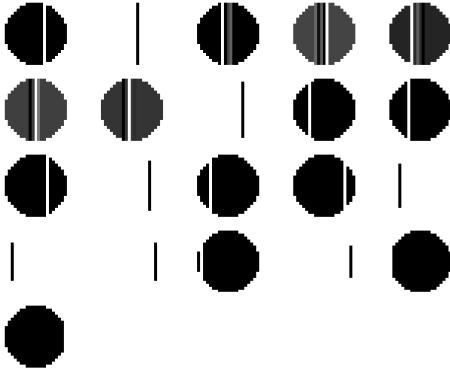


Fig. 4. Eigenvectors of  $A^{\text{shift}}$ . They explain 100% of the variance in  $A^{\text{shift}}$ . All eigenvectors shown have positive eigenvalues.

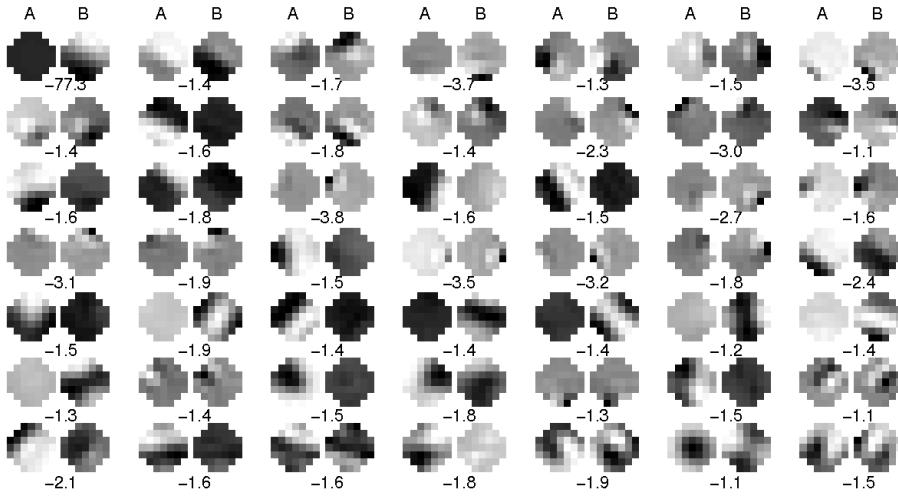


Fig. 5. Learned symmetry detectors by FastICA. For each of 49 found independent component two images (**A**, **B**) are shown. **A** maximizes the output of the filter **B** minimizes the output. The number below each independent component codes for the ratio of the largest to the second largest eigenvalue.



Fig. 6. Texture dependent sub-fields. Eigenvectors with negative eigenvalues concentrate onto the lower right part of the receptive field, eigenvectors with positive eigenvalues onto the upper left part.

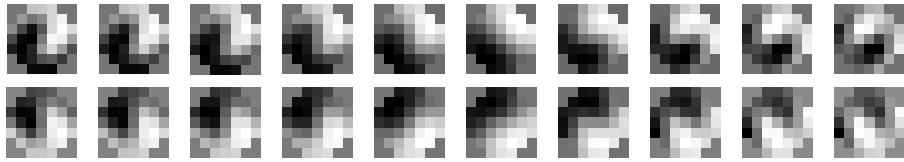


Fig. 7. A dot-product filter that codes for curvature. The **top row** shows the line-space that is spanned by the largest and smallest eigenvector of the ICA-component nr. 47 shown in figure 5. In the **bottom row** the ICA-component nr. 49 is shown. Both components match at their zero-point solution edge detector units.