

Image Segmentation by a Network of Cortical Macrocolumns with Learned Connection Weights

Markus Lessmann, Rolf P. Würtz

Abstract Image understanding in the brain or a computer requires segmentation of observed images, i.e., their partition into different semantically-connected parts that each constitute one physical object. This task is fundamental for further processing and analysis of visual information and seems to be accomplished by the brain very easily. Nevertheless it is a very demanding challenge for computer algorithms. In this article, we present a network of neuronal macrocolumns, which processes contour information by favoring closed contours. The connecting weights have been learned from real image sequences before. Then, segmentation is achieved on the basis of color, texture, and contour information.

1 Introduction

The task of computer vision as well as human vision is to extract semantic information from images or image sequences. This means that the values of a priori unrelated pixels on a camera chip or on the retina of the eye must be organized into larger entities, which can then be identified as objects. A subtask is the division of images into object candidates, a process called segmentation. Psychologically, the properties of this process have been studied for a long time.

Based on experiments in human vision Gestalt psychologists like Max Wertheimer, Wolfgang Köhler and Kurt Koffka formulated eight *Gestalt principles*, formal rules that guide this process [1]. Later this list was extended by Stephen Palmer who

Markus Lessmann
Institute for Neuroinformatics, Ruhr-University Bochum
e-mail: Markus.Lessmann@neuroinformatik.rub.de

Rolf P. Würtz
Institute for Neuroinformatics, Ruhr-University Bochum
e-mail: Rolf.Wuertz@neuroinformatik.rub.de

added 3 further rules [2]. Many of these principles take into account the spatial or temporal context in which different patterns appear.

The details of how the process of segmentation is carried out in the brain, i.e., how networks of neural agents can actually perform this task, are still unknown. On the technical side, the organizational principles that must be applied for meaningful segmentation are intense investigation.

In our project, we use a biological model of basic neuronal ensembles in the brain to perform segmentation by using only non-semantic bottom-up criteria that are easily extractable from input data by application of mathematical filters like Gabor wavelets and Gaussians, both known to be applied by the visual cortex. This means we only rely on a few of the Gestalt rules, namely the *law of proximity*, the *law of similarity*, and the *law of collinearity and curvilinearity*. The first one postulates the vicinity of pixels representing the same object. Distant, unconnected parts of an image are unlikely to belong to the same object. The second rule states the resemblance of areas of the input picture for belonging together. Adjacent points with very similar color and texture are highly likely to belong to the same item. The last rule concerns the contours of an object. In the ideal situation a closed contour is detected by filter functions and separates it from the background. During perception contour segments are grouped by the visual system in a way that smooth curves occur without any acute angles.

To apply these rules contour information is extracted from all 3 channels of colored input images using a biologically-inspired convolution with Gabor wavelets of 8 different orientations, which collect evidence for oriented contour elements, and Gaussians, which calculate mean color values for each pixel. The contour information is preprocessed using non-maximum suppression to keep as much edge information as necessary but delete as many unimportant edges as possible.

The responses of the Gabor and Gaussian filters provide the input to the minicolumns of a macrocolumn network, which will be introduced in the third chapter. The network calculates for each pixel the presence or absence of an edge and, in case of presence, its orientation. This data is combined with information about the similarity of color and texture to decide on the connectedness of neighboring points, finally yielding image segments.

2 Input pictures, filter functions and preprocessing

The program works on color pictures transformed into *CIELab* color space for further processing. This is done because of the resemblance of *Lab* data to the output of retinal neurons and its construction following human perception [3]. By applying Gabor filters to *a*- and *b*-channel of an image it is easy to simulate color-coding simple cells of the primary visual cortex that compare red stimuli with green ones and the blue components of the input with yellow ones [4].

Filter functions As mentioned before Gabor wavelets are used throughout this program as well as Gaussians. The Gaussians have a width of 1.0 pixel, the Gabor functions are elliptic with widths $\sigma = 1.25$ and $\tau = 1.0$. Eight equally spaced orientations and five frequencies with a ratio of 1.025. As a model of complex cells in the visual cortex, the *magnitude* of the complex-valued Gabor responses is used.

Preprocessing After the extraction of contour information a lot of the edges perceivable by a human are detected, but also a lot more contours parallel to these meaningful outlines. The reason for this is that a Gabor wavelet placed next to a contour also yields a non-zero-response. To sort these out non-maximum suppression is used. Thus, the orientation of the biggest Gabor filter response at every pixel is determined. Then this biggest value is compared with the best filter responses of pixels on a straight line perpendicular to the orientation of the maximum value, provided that these responses have the same filter alignment. The maximum of these values is labeled by incrementing a counter. If only the pixels with counter values bigger than one are used a lot of unnecessary edges have been erased. The edge elements extracted by the edge detectors and the preprocessing step are shown in figure 1.

3 The macrocolumn network

3.1 Fundamentals of the macrocolumn network

The macrocolumn network is based on findings about biological nerve cell clusters in the mammalian cortex and was first introduced in [5, 6]. It consists of a set of macrocolumns each composed of k different minicolumns, which in turn are a bunch of highly interconnected neurons. These minicolumns are represented by their activity $p_\alpha(t)$, which is the fraction of its neurons firing at time t . α is the index of the respective minicolumn. Minicolumns get inputs from the input data and minicolumns of other macrocolumns they are connected with. All minicolumns of the same macrocolumn are inhibited proportional to the maximum of their activity and a modifiable parameter v .

$$\frac{d}{dt}p_\alpha(t) = ap_\alpha(p_\alpha - v \max_{\beta=1..k} p_\beta - p_\alpha^2) \quad (1)$$

The dynamics of the network reveal a bifurcation depending on the value of v . For $v < 0.5$ a macrocolumn can have arbitrarily many active minicolumns, but for $v > 0.5$ only states with at most one active minicolumn are stable. This puts the minicolumns into competition with each other when v is raised from start values below 0.5 to values above 0.5. At the end of this process only the minicolumn with the biggest input can still be active, all others are turned off. That means the system makes a decision on which column received the strongest input.

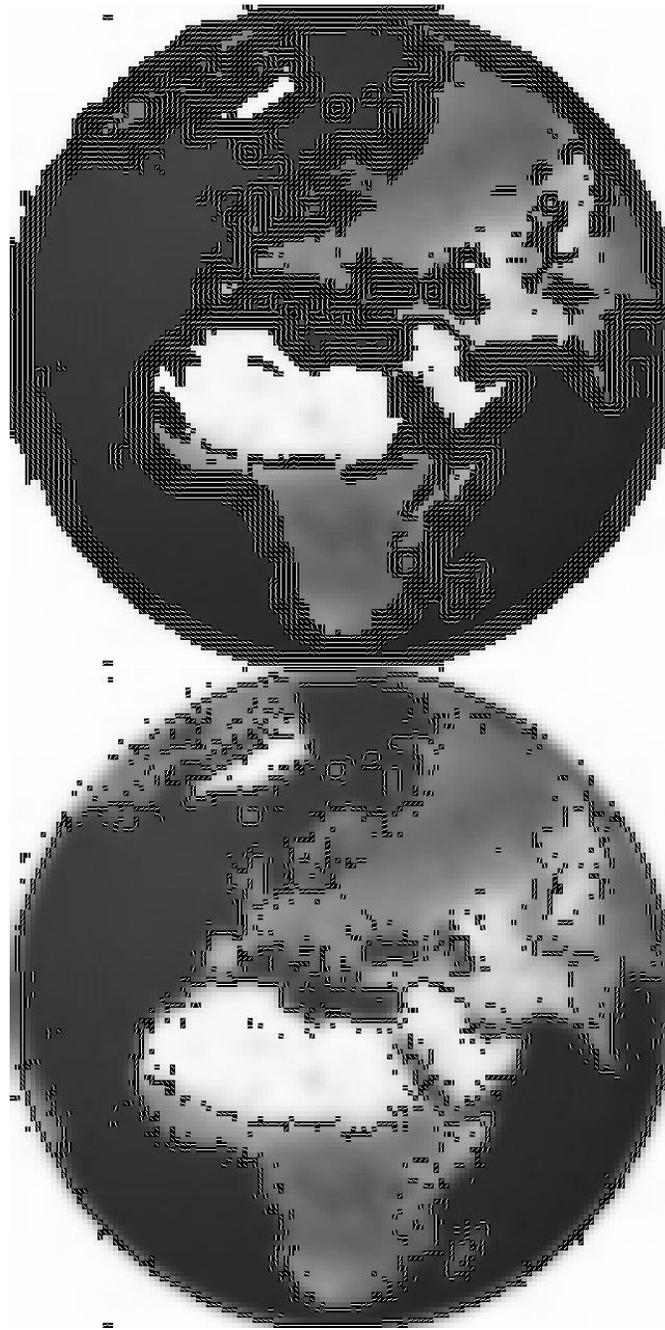


Fig. 1 Preprocessing of a globe image. Top: edges from edge detectors, bottom: edges after non-maximum suppression.

To use information about the local environment the minicolumn i can be connected to other columns in its vicinity. Every connection is defined by the index of the associated minicolumn j and the weight of the connection. These weights are given in a matrix R_{ij} . They are multiplied with the current activity of column j and added to the differential equation.

$$\frac{d}{dt}p_{\alpha}(t) = ap_{\alpha}(p_{\alpha} - v \max_{\beta=1..k} p_{\beta} - p_{\alpha}^2) + \kappa \sum_{j=1}^N R_{\alpha j} p_j^E + \eta \sigma_t \quad (2)$$

Gaussian noise ($\eta \sigma_t$) is added to distinguish between equally sized inputs. The factor κ defines the relative importance of the connection inputs.

3.2 Use of the macrocolumn network

For the task of contour completion, the macrocolumns must decide whether a pixel of the input picture is part of a contour with a specific orientation. To fulfill this function, preprocessed and normalized responses of Gabor filters constitute the starting values of the minicolumns. Since every picture consists of 3 different color channels and the number of minicolumns should be kept at a minimum due to computational costs one of them is picked to represent this part of the image. This is done by computing the variance of the Gabor filter responses for each channel. That with biggest variance includes the filter response that differs most from responses for other orientations and is highly likely to contain the most distinctive edge. Therefore, its filter values are selected as input to the network. By setting a threshold for the variance, weak edges can be filtered out.

Thus, every minicolumn represents a contour element with a specific orientation. If the minicolumn wins against all others in the macrocolumn then it obtains strong inputs from the filter values and minicolumns of surrounding macrocolumns. That means the picture contains an intense luminance or color contrast at this point or an edge of this orientation fits well to the adjacent contours. An additional 9th minicolumn becomes active if the image does not contain an edge at this pixel. The starting value for this background minicolumn decides how many further contours are detected during the simulation of the network and has to be chosen manually. The higher the start value the smaller is the chance of other minicolumns to compete with the 9th column. It is set to 0.0 for all edges labeled by the preprocessing to be meaningful and to values in the range of 10.0 for all else. That way meaningful contours need not compete with the background and can win easily. All other edges can be suppressed by the background column but still have a chance to win if they get strong inputs via their connections which means they fit well to the surrounding edges.

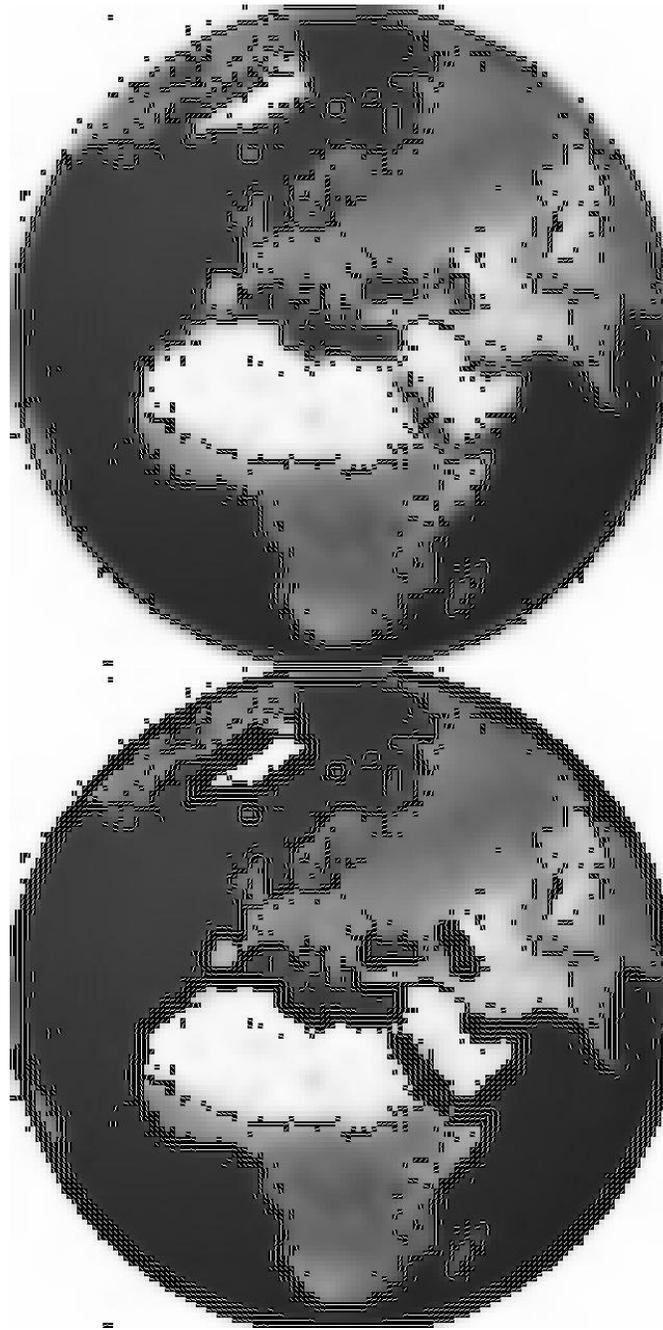


Fig. 2 Processing of a globe image by the macrocolumn network. Top: input to the macrocolumn network, bottom: contours after network processing.

3.3 Learning of the macrocolumn network

In order to enhance contours the network must rely on lateral connections, which support these, even in the case of missing edge segments. So the rule would be that roughly collinear edge detectors should excite each other, while orthogonal ones should inhibit each other. We have made several attempts to specify the connecting weights on a theoretical basis, but the results were not satisfactory. Therefore, we turned to learning them from examples.

3.3.1 Learning rule and normalization

To improve the edge information of input pictures the network has to learn typical spatial relations between different edge orientations from sample pictures (see also [7]). This is done by processing of the sample images with the network and modification of the connection weights using the activities of the minicolumns after each network simulation in conjunction with a Hebbian learning rule as described by Singer [8]:

$$\begin{aligned}\Delta w_{ij}(t) &= \varepsilon \cdot A_i(t) \cdot A_j(t) \\ w_{ij}(t) &= w_{ij}(t-1) + \Delta w_{ij}(t) \quad \text{if } A_i(t) > \Theta_u \text{ and } A_j > \Theta_u \\ w_{ij}(t) &= w_{ij}(t-1) - \Delta w_{ij}(t) \quad \text{if } A_i(t) > \Theta_u \text{ and } A_j < \Theta_l\end{aligned}$$

In these equations, ε is the learning rate, $w_{ij}(t)$ the connection weight between minicolumns i and j at time t , and Θ_u and Θ_l are appropriately chosen upper and lower thresholds for the activities $A_i(t)$ and $A_j(t)$ of the concerned columns ($\Theta_u = 0.4$ and $\Theta_l = 0.1$). If both minicolumns have activities bigger than Θ_u their correlation is significant and their connection is strengthened. If minicolumn i is very active ($A_i(t) > \Theta_u$) but minicolumn j is not ($A_j(t) < \Theta_l$) it means that j is not of much importance for i and the weight is reduced. In all other cases it is not modified.

Of all connections to the same minicolumn this learning rule favors those with high temporal correlation over those with low correlation. Further competition between them is introduced by normalization over index j . One occurring problem is, that the influence of the weights can completely overcome the filter values of the input image. This is especially unfavorable in the beginning of the learning process, when they are learned from only a few examples and not very reliable. Therefore, they are scaled to 0.1 to keep their influence small enough.

3.3.2 Learning data

Pictures for learning of edge relations should fulfill some criteria:

1. Edges of one single object should be learned to be sure that they have a semantic context and don't belong to neighboring items.

2. The object should be moving so that many possible relations can be observed.
3. The object must not be too inflexible. A very inelastic object like a metal-ball would not reveal many different edge-relations during movement.
4. Only parts of the moving object should be allowed for learning because static background would disturb the statistics of the learning samples.

To match these requirements 2 different sets of training pictures have been chosen. The first one are several photo sequences of a person moving his arm in front of a static background. The other set are the first 8 objects of the COIL100-database [9]. This collection consists of photos of 100 different objects that are rotated about 5° between 2 shots. Therefore they also contain a movement of the item and should be appropriate.

3.3.3 Selection of pixels for learning

During learning, one picture after another of the training set is processed by the network. Then pixels have to be chosen that contain valuable information about edge relations in the training picture. The activities of the macrocolumns at these pixels and of the macrocolumns in an 7×7 environment are read out and used for Hebbian learning. To select the specific pixels a more fundamental segmentation method is used which relies on the movement of objects in the picture (Gestalt law of common fate, see also [7] for discussion). Therefore, movement vectors are calculated for every part of the picture by comparing Gabor responses of a certain pixel in one picture of the sequence with the responses in a 3×3 environment of the current picture element in the following picture. By selecting the pixel with the highest similarity a translation vector can be determined that can be compared with vectors of surrounding pixels. This method does not yield very precise movement information, but it is good enough to separate a moving object from a static background. Each pixel of the area labeled this way containing an edge is chosen for learning purposes.

4 Segmentation

The contours improved by the network are shown in figure 2 and are subsequently used for segmentation with the following procedure.

Color similarity To calculate the similarity of color values of 2 adjacent pixels the difference vector in Lab space and its magnitude are determined. It is one option not to use the L-channel so that only color and no luminance differences are considered. The range from 0 to the maximum difference vector magnitude in the picture is first projected onto the range from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ and then onto $-\infty$ to ∞ using the tangent. At last the Fermi function $\frac{1}{1+e^{c \cdot y}}$ with $c = 0.1$ transforms the interval into a similarity measure with 0.0 for the maximum difference vector and 1.0 for no distance in color space. Now a threshold decides about the connectedness of 2 neighboring pixels.

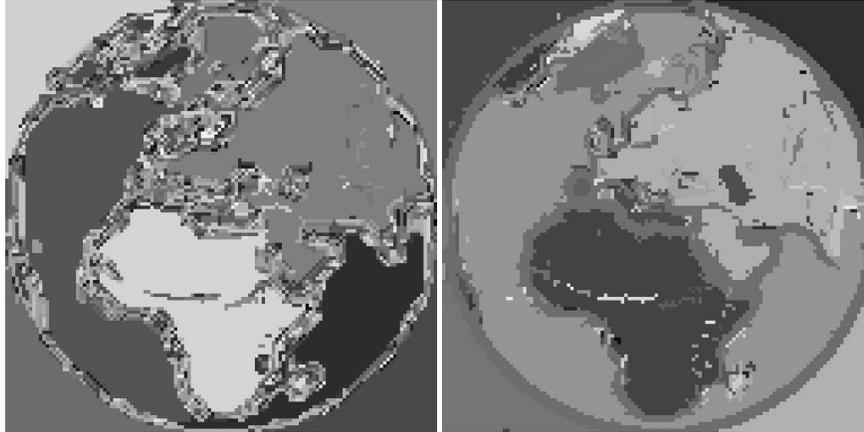


Fig. 3 Segmentation of sample picture before (left) and after processing by the network (right). All adjacent pixels with the same grey value are connected.

Texture similarity Vectors of Gabor responses are now used as texture descriptors, the parameters are as before, but the scales are spread wider. When comparing the similarity of texture features the normalized dot-product of the vectors of Gabor responses is calculated and then mapped onto the range from 0.0 to 1.0 by adding 1.0 and multiplication with 0.5. After doing this the mean value of color and texture similarity is computed and evaluated by means of a threshold.

Integration of contour information Contour information is taken into account by connecting all neighboring pixels containing edges and canceling all connections between edge and non-edge pixels originating from color and texture similarity.

5 Discussion

We have presented a biologically inspired network that organizes image segmentation. This is a collaborative computation, which follows biological inspiration. The required parameters are learned from visual examples using a neurobiologically plausible learning rule. We could show that these learned parameters worked better than the ones we could come up with by theoretical inspection. This is evidence for the correctness of the assumption that the wiring of the visual system reflects the statistics of natural images. The images processed by the contour network were much better suited for segmentation (see figure 3). Full details can be found in [10]. Future developments aim at integrating high-level knowledge of previously seen objects into the segmentation process.

Acknowledgements

Partial funding by the Deutsche Forschungsgemeinschaft (MA 697/5-1, WU 314/5-2) is gratefully acknowledged.

References

- [1] M. Wertheimer. Untersuchungen zur Lehre von der Gestalt II. *Psychologische Forschung*, 4:301–350, 1923.
- [2] Steven E. Palmer. *Vision Science*. MIT Press, 1999.
- [3] David Falk, Dieter Brill, and David Stork. *Seeing the Light: Optics in Nature, Photography, Color, Vision, and Holography*. John Wiley & Sons, New York, 1986.
- [4] David H. Hubel and Margaret S. Livingstone. Anatomy and the physiology of a color system in the primate visual cortex. *Journal of Neuroscience*, 4(1):309–356, 1985.
- [5] Jörg Lücke, Christoph von der Malsburg, and Rolf P. Würtz. Macrocolumns as decision units. In José R. Dorronsoro, editor, *Artificial Neural Networks – ICANN 2002, Madrid*, volume 2415 of *LNCS*, pages 57–62. Springer, 2002.
- [6] J. Lücke and C. von der Malsburg. Rapid processing and unsupervised learning in a model of the cortical macrocolumn. *Neural Computation*, 16(3):501 – 533, 2004.
- [7] Carsten Prodöhl, Rolf P. Würtz, and Christoph von der Malsburg. Learning the gestalt rule of collinearity from object motion. *Neural Computation*, 15(8):1865–1896, 2003.
- [8] A. Artola, S. Bröcher, and W. Singer. Different voltage-dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature*, 347:69–72, 1990.
- [9] S.A. Nene, S.K. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical Report CUCS-006-96, Columbia University, 1996.
- [10] Markus Lessmann. Konturenerkennung mit einem Modell kortikaler Makrokolumnen. Master’s thesis, Physics Dept., Univ. of Dortmund, Germany, January 2008.