International Journal of Neural Systems, Vol. 8, No. 1 (February, 1997) 27–39
Special Issue on Neural Networks for Computer Vision Applications
© World Scientific Publishing Company

DYNAMIC CELL STRUCTURES FOR THE EVALUATION OF KEYPOINTS IN FACIAL IMAGES*

R. HERPERS[†], L. WITTA[‡], J. BRUSKE[§] and G. SOMMER[§]
GSF – National Research Center for Environment and Health,
Institute for Medical Informatics and Health Services Research, MEDIS,
Ingolstädter Landstr. 1, 85764 Oberschleißheim, Germany
[‡]*Lehrstuhl für Mensch-Maschine-Kommunikation, TU-München,*
Arcisstr. 12, 80290 München, Germany
[§]*Institut für Informatik, Christian-Albrechts-Universität,*
Preußerstr. 1-9, 24105 Kiel, Germany

In this contribution **D**ynamic **C**ell **S**tructures (DCS network) are applied to classify local image structures at particular facial landmarks. The facial landmarks such as the corners of the eyes or intersections of the iris with the eyelid are computed in advance by a combined model and data driven sequential search strategy. To reduce the detection error after the processing of the sequential search strategy, the computed image positions are verified applying a DCS network. The DCS network is trained by supervised learning with feature vectors which encode spatially arranged edge and structural information at the keypoint position considered. The model driven localization as well as the data driven verification are based on steerable filters, which build a representation comparable with one provided by a receptive field in the human visual system. We apply a DCS based classifier because of its ability to grasp the topological structure of complex input spaces and because it has proved successful in a number of other classification tasks. In our experiments the average error resulting from false positive classifications is less than 1%.

1. Introduction

The detection and exact localization of certain characteristic keypoints in face images, such as the corners of the eyes or the mouth, are relevant issues for many applications in face recognition. In general, these keypoints cannot be detected by purely data-driven methods. For example, the definition of an eye corner is more a semantic or high level definition than a low level one, i.e. it is not based solely on the local image structure. Moreover, the local image

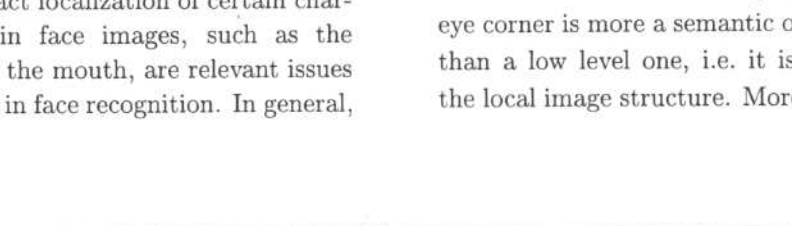

Fig. 1. Examples of inner eye corners to be verified. The high degree of variability between different subjects is shown.

*This work is partially supported by the DFG grants So 320\2-1 and Ei 322\1-1.

[†]E-mail: herpers@gsf.de

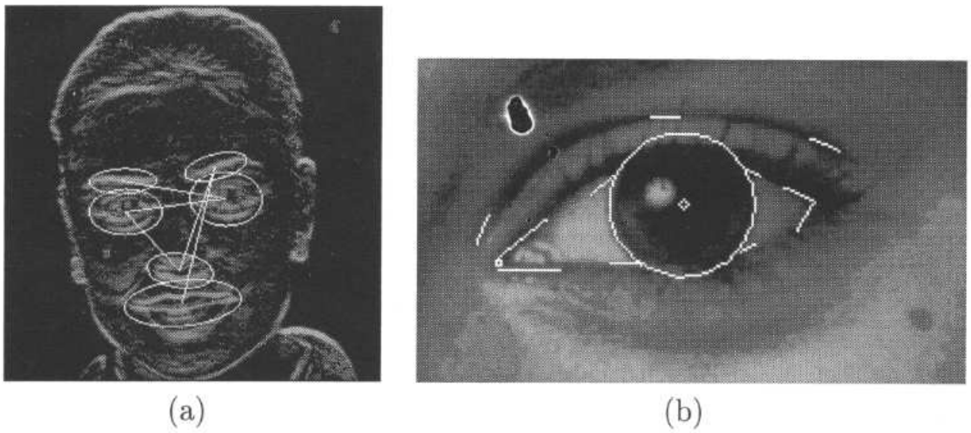

Fig. 2. Attentive localization of prominent facial regions based on a saliency representation [Ref. 11] (a). Result of the detailed investigation of an eye region applying a combined model and data driven sequential search strategy [Ref. 14] (b).

structure of an eye corner and its local neighborhood shows large variations between individuals and even between different images of the same individual (Fig. 1).

Sequential search strategy

In previous work we have developed a sequential search strategy which integrates appropriate model knowledge into a local data driven processing strategy.[14,15] This algorithm is based on a very flexible and powerful filtering scheme[17] and consists of two steps:

1. Selection of the region of interest (ROI) using an attentive localization algorithm.
2. Detailed investigation of the selected local region applying a sequential search strategy to detect the included keypoints.

In the first step, motivated by the behavior of the human visual system, the most salient regions in facial images, such as the eye, nose, and mouth region, are selected, applying an attentive localization mechanism [Fig. 2(a)]. To this end, a saliency representation is established in which the most salient cues of the facial image are encoded. In grey level face images these are prominent edge and line structures or generally image regions with a high degree of local energy.[18] By evaluating a multiscale representation of the saliency representation a sequential order of spatially well restricted facial regions is obtained. A complete description of the attentive localization of the facial regions is beyond the scope of this paper but we refer to Ref. 11.

In the second step the selected image regions are investigated using more sophisticated and expensive image processing methods. The result of the detailed analysis is the detection of a set of important facial landmarks. In the case of the eye regions these are the corners of the eyes, the shape and the center of the iris, the intersections of the iris with the eyelids and the line of the eyelid wrinkle [Fig. 2(b)]. It is based on a sequential search strategy which uses model knowledge about the considered scene to enable a robust and accurate keypoint detection.[14,15] This sequential search strategy relies on line and edge information for the detection and tracking operations. The included model knowledge allows a stepwise verification of each state of processing.

Detection of dysmorphic facial signs

Our primary application is the detection of dysmorphic facial signs. Dysmorphic signs in face images are minor anomalies which, by definition, do not lead to functional disturbances (Fig. 3, [Ref. 24, p. 42]). The ratios of distances between certain facial keypoints are statistically significant for discriminating between the faces of normal children and

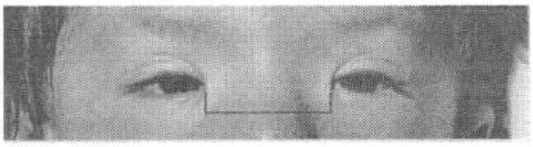

Fig. 3. Example of a very enlarged intercanthal distance which is a typical dysmorphic facial sign (from [Ref. 24, p. 42]).

different classes of dysmorphic syndromes.[23,24] Therefore, the detection of particular keypoint positions in dysmorphic facial images is of high diagnostic value. The localization of the keypoints needs to be very accurate and reproducible, and should correspond to the anatomical definition of the keypoint position. To this end, the main aim of this investigation is to decrease the false positive classification error as much as possible. From a medical point of view, it is better not to detect a keypoint position than to detect one at an incorrect location.

Description of problem

Due to artifacts, occluded keypoints or unpredictable edge elements, the sequential search sometimes fails and terminates at non-keypoint positions (Fig. 4). In addition, every edge element considered during the detailed processing has to be interpreted correctly to ensure that the edge and line tracking is not misled. Such sources of error, compounded by the general complexity of the search task lead to a variety of different failures of the sequential search strategy. Hence, we have decided to augment our system with a third processing stage, a verification component, which is the topic of this contribution. Its task is to classify the image position at which the processing of the sequential search strategy has terminated. In the worst case this may be any position inside the considered facial region. Examples of the many different appearances of keypoint positions to be verified are demonstrated in Fig. 1 taking the inner eye corner as an example.

In summary, the trained DCS network is applied to verify computed keypoint positions and to discriminate them from non keypoints. The probability of taking a non-keypoint as a valid keypoint should be reduced as much as possible. The combination of the additional verification stage together with the sequential search strategy is based on Bayes's theorem.

Outline

In the next section we will introduce the neural verification stage and its combination with the sequential search strategy. First the basic principles

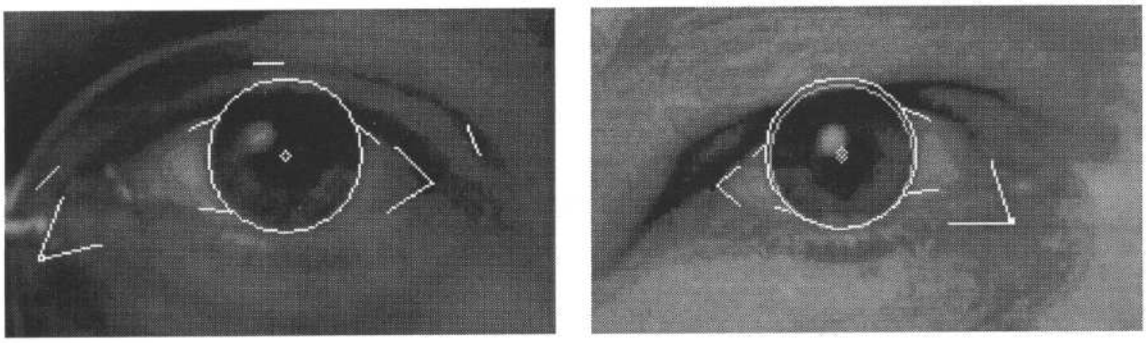

Fig. 4. Examples of detection results after applying the sequential search strategy which have terminated at non keypoint positions for both inner eye corners. In the first case the failure is caused by the glasses and in the other case by too diffuse edge structures in the area of the inner eye corner. These false detected keypoint positions should be excluded applying the DCS based verification step.

of the feature generation will be presented. Subsequently the fundamental principles of dynamic cell structures will be presented together with some modifications to the learning rule and the insertion and deletion of neurons. In addition, the computation of the classifications and the applied Bayesian framework will be described. In Sec. 3 we will present experimental results from the application of the DCS network, taking the eye region as an example, and finally in Sec. 4 we will end with some concluding remarks.

2. The DCS Based Verification Stage

In this section we briefly introduce DCS networks together with some modifications developed for this application. First we will explain the generation of feature vectors serving as input to our classifier and subsequently describe the design of the actual DCS based classifier. Finally we give the theoretical justification for the presented verification stage within a Bayesian framework.

2.1. *Feature generation*

The feature vectors are obtained by applying a particular set of linear filters to the considered image position and its local 27 × 27 neighborhood (see Fig. 5). The projection coefficients of this particular set of filters are taken as components of our feature vectors. The applied filters are based on the concept of steerable filters.[17] They are essentially used for the sequential search strategy to detect the keypoint positions because they provide a high degree of flexibility for the edge and line detection. In addition, they are also used to select the facial regions in the images.[11,12] Hence, the convolutions needed have already been calculated. No additional computing time is spent for the calculation of the feature vectors. In the following we will briefly

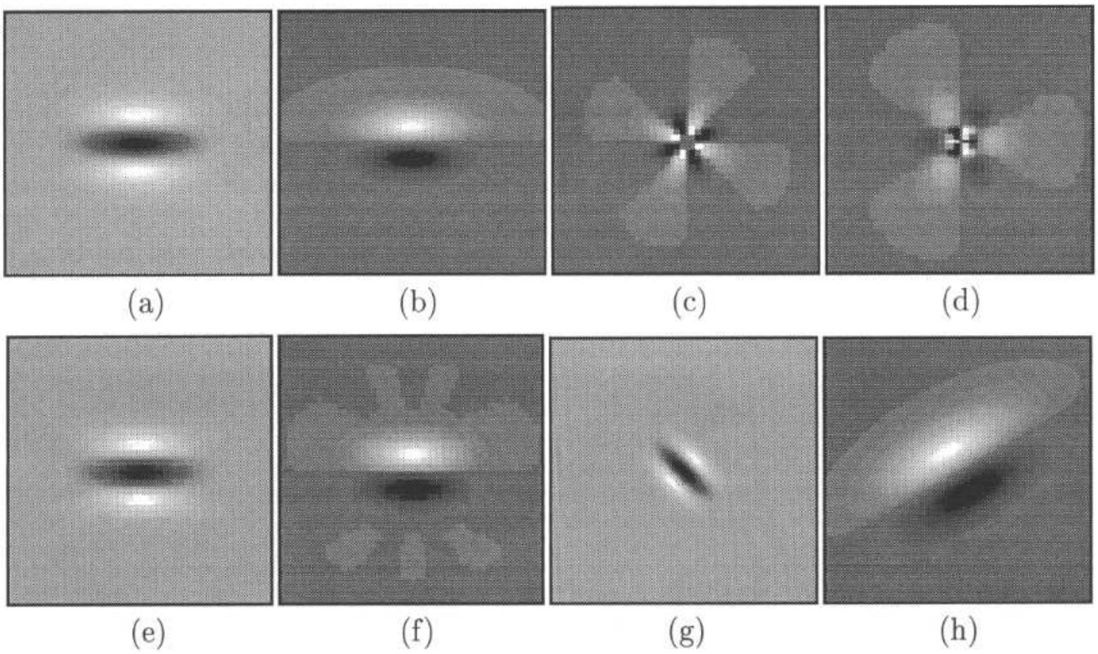

Fig. 5. Second (a) and first (b) derivative of Gaussian filter for line and edge detection. Examples of the basis functions [(c),(d)] to steer the line (a) and edge detection filter (b) in scale and orientation. Each considered image position is projected to 70 of these basis functions (30 for the edge detector and 40 for the line detector). The resulting projection coefficients form the components of the feature vector. Reconstruction of the filters with 40 basis functions (e) for a line detector or 30 basis functions for a edge detector (f). The line and edge detection filters are steered in orientation and scale [(g),(h)]. All rotated and scaled filters are computed from a small number of basis functions [(c),(d)] which optimally approximate any desired deformed filter.

introduce steerable filters, for a more detailed discussion of the filtering scheme we refer the reader to Refs. 17, 7, and 21. Examples of applications of steerable filters can be found in Refs. 13, 14, 15, and 18.

2.1.1. *Steerable filters*

Steerable filters were introduced to efficiently calculate the responses of filters in arbitrary orientations, scales, and other deformations.[17,7,21] The reconstruction of all deformed filters F_α is calculated by a superposition formula of the following type:

$$F_\alpha(\mathbf{x}) = \sum_{k=1}^{N} b_k(\alpha)\, A_k(\mathbf{x}) \tag{1}$$

Applying this technique it is possible to generate an infinite number of different 'deformed' filters (e.g. edge and line detectors steered in scale and orientation within a given interval) as a weighted sum of only a few so called basis functions [Figs. 5(c) and 5(d)]. The number N of the orthogonal basis functions $A_k, (k = 1, \dots, N)$ is assumed to be small compared to the number of deformed filters. Typically N will be small (30 or 40 for example), while the deformation parameter α can theoretically assume an infinite number of values and many thousands in practice (for orientation and scale). For example, 100 orientation steps and 17 distinct scales would result in 1700 different edge or line filters in a classical filtering scheme. Two examples of deformed filters (rotated and scaled) are shown in Figs. 5(g) and 5(h). With steerable filters, however, this huge number of filters is generated only by 30 (for the edge detector) or 40 (for the line detector) basis functions with an average approximation error of about 3% with respect to a kernel size of 27 × 27 pixels [Figs. 5(e) and 5(f)].

2.1.2. *Feature vector*

The projection coefficients of a considered image position within its 27 × 27 neighborhood relative to the basis functions establish our feature vector. The underlying assumption is that all the relevant and necessary line and edge information is encoded by the 70 projection coefficients of the basis functions. The application of this set of basis functions 'to reconstruct' or represent the included edge and line information results in a high reconstruction quality in the center and a stepwise decreased resolution in the surrounding region dependent on the distance from the center [Fig. 6]. The overall reconstruction quality can be enhanced by adding more basis functions. This is one essential property of the employed orthogonal basis functions which enhances the flexibility and performance of the processing.

2.1.3. *Properties of the representation*

The projection coefficients computed at the considered keypoint positions can be compared with the responses of receptive fields in the human visual system. The sensitivity of the applied filters is restricted to particular structural information from the considered image part within a limited spatial extension. Hence, the application of the filters to particular image positions as demonstrated in Fig. 6 can be viewed

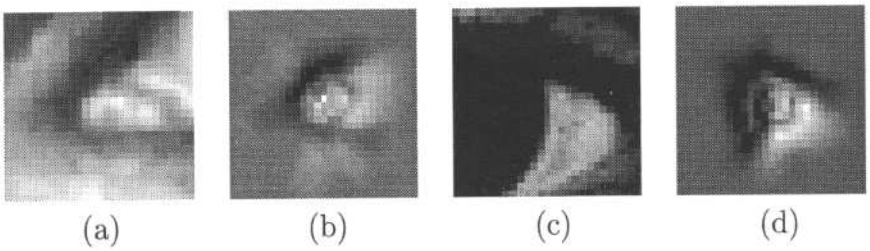

Fig. 6. Demonstration of the reconstruction properties of the filtering set at particular image positions. Two original image parts are presented [(a), (c)] along with the reconstructions [(b), (d)] using 70 basis functions. In the left image pair an inner eye corner is shown, in the right image pair the edge between the iris and the bulbus. The filter set is positioned exactly at the center of the image parts (image and filter kernel size is 27 × 27 pixels). In the central regions of the images the reconstruction is nearly complete, in the close surrounding regions it is quite acceptable (7 × 7) while the quality decreases drastically or is reduced to an average grey value towards the image borders.

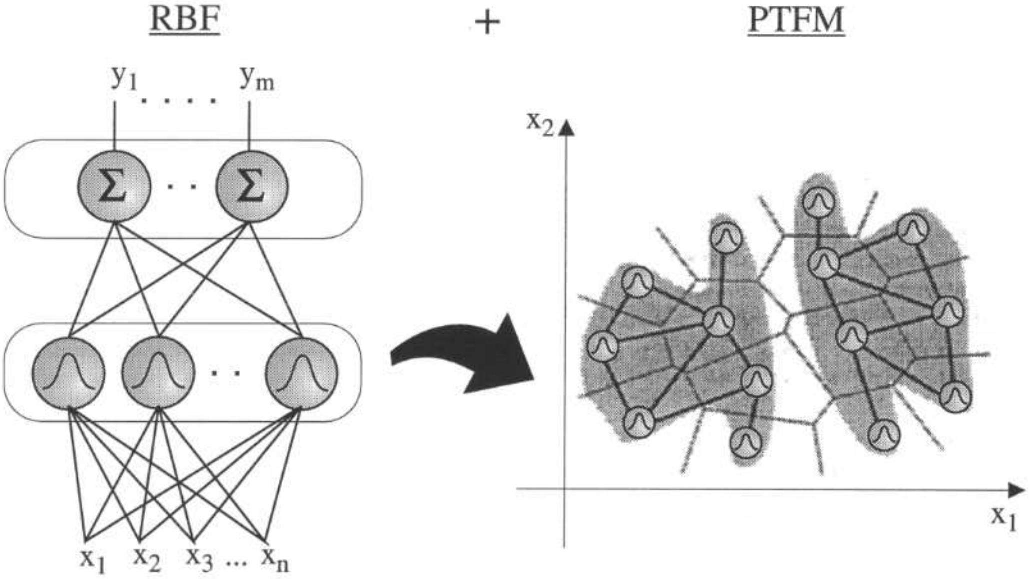

Fig. 7. DCS are RBF networks (left) plus an additional lateral connection structure between the RBF units formed by competitive Hebbian learning and approximately Perfect Topology preserving Feature Mapping (PTFM) (right). In the latter, two neural units are connected if their masked Voronoi polyhedrons have a common border.

as a joint focus and analysis process restricted to only the necessary visual cues.

However, the generated representation is not complete, i.e. the representation is not sufficient to reconstruct an image area or the grey value distribution completely and error-free. Only the most important structural information is encoded and most of the irrelevant information is not considered. Furthermore, this representation of the projection coefficients allows an effective and significant reduction in the amount of data, by a factor of more than 10, compared to the original grey level information of an image area with a size of 27 × 27 pixels.

2.2. *Dynamic cell structures*

The recently introduced **D**ynamic **C**ell **S**tructures (**DCS**)[6] represent the class of RBF-based approximation schemes which attempt to concurrently learn and utilize **P**erfect **T**opology preserving **F**eature **M**aps (**PTFMs**). DCS networks are a subclass of Martinetz's **T**opology **R**epresenting **N**etworks (**TRN**),[19] defined as containing any network using competitive Hebbian learning for building PTFMs.

The architectural characteristics of a DCS network are (see Fig. 7):

1. one hidden layer of radial basis functions (possibly growing/shrinking)
2. a dynamic lateral connection structure between these units and
3. one layer of (usually linear) output units.

Training algorithms for DCS adapt the lateral connection structure to a PTFM by employing a competitive Hebbian learning rule and activate and adapt RBF units in the neighborhood of the current stimulus, where "neighborhood" relates to the simultaneously learned topology.

2.2.1. *Growing DCS*

The **G**rowing **DCS** (**GDCS**)[6] is a DCS network which grows by inserting units according to a local error measure and the emerging PTFM. It is similar to Fritzke's GCS network[8] except that it uses a PTFM instead of a fixed hypertetrahedrical connection structure. GDCS thus addresses the major problems of Kohonen-type SOMS, i.e. the fixed number of neural units (GDCS incrementally grow and shrink), the fixed and usually imperfect topology preserving neighborhood relation (DCS learn and utilize PTFMs instead) and a distribution of neural units that depends solely on the input probability distribution (in GDCS the distribution of neural units also depends on the local error measure which can be chosen according to the task in hand). Finally, DCS networks can aid in automatic class/cluster separation by searching for connected components in the PTFM.

These properties of GDCS and their simplicity, efficiency and superior classification performance on a number of CMU classification benchmarks[6] made them the premier choice for our classifier. Using learning PTFMs, DCS are able to grasp part of the topological structure of high dimensional input spaces, and by exploiting this knowledge, they usually yield better classification results than simple nearest neighbor or other conventional classifiers.

We use a normalized RBF interpolation scheme and hence the output of the DCS calculates as

$$\mathbf{y}(\mathbf{x}) = \frac{\displaystyle\sum_{i \in Nh^+(bmu)} \mathbf{o}_i \, rbf_i(\|\mathbf{x} - \boldsymbol{\mu}_i\|)}{\displaystyle\sum_{i \in Nh^+(bmu)} rbf_i(\|\mathbf{x} - \boldsymbol{\mu}_i\|)} \tag{2}$$

where $rbf_i(\|\mathbf{x} - \boldsymbol{\mu}_i\|)$ denotes a radial basis function i with center $\boldsymbol{\mu}_i$ and input stimulus $\mathbf{x}$. The vectors $\mathbf{o}_i$ can be thought of as output weight vectors attached to each *rbf* unit i. The radial basis functions are strictly monotonically decreasing with $rbf_i(0) = 1.0$ and $rbf_i(\infty) = 0$. In the experiments reported in this article the radial basis functions have been realized by Gaussians:

$$rbf_i(\mathbf{x}) = e^{-\frac{1}{2\sigma^2} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2} \tag{3}$$

The neighborhood $Nh^+(i)$ of a neural unit i is defined as the unit itself together with its direct neighbors $Nh(i)$ (w.r.t. the PTFM)

$$\begin{aligned} Nh^+(i) &= Nh(i) \cup \{i\} \\ &= \{j | C_{i,j} \neq 0, 1 \leq j \leq N\} \cup \{i\}, \end{aligned} \tag{4}$$

where N denotes the current number of neural units in the network and C is the adjacency matrix explained in the following paragraph. The best matching unit *bmu* is given by

$$\|\boldsymbol{\mu}_{bmu} - \mathbf{x}\| \ \leq \ \|\boldsymbol{\mu}_i - \mathbf{x}\| \quad , \quad (1 \leq i \leq N). \tag{5}$$

2.2.2. *Individual learning rates for each unit*

For the application reported in this article we modified some of the learning rules suggested in Ref. 6 to better meet the demands of our application. The adjacency matrix C represents the lateral connection structure between the RBF units i, j and is adapted by a competitive Hebbian learning rule such as

$$C_{i,j} = \begin{cases} 2T : rbf_i * rbf_j \geq rbf_k * rbf_l\,, \\ \qquad (1 \leq k, l \leq N)\,, \\ C_{i,j} - 1 : i,j \neq bmu, \text{ second best}\,, \\ \qquad \text{and } C_{i,j} \geq 0\,. \end{cases} \tag{6}$$

This rule connects and subsequently enhances (refreshes) two units if their masked Voronoi polyhedrons have a common border, giving rise to a PTFM.[19,20] Furthermore, it may break the connection, if it is not refreshed after a maximum of 2 training cycles (T is the number of the feature vectors in the training set).

We have also modified the Kohonen-type learning rule suggested in Ref. 6 for training the centers of the DCS neurons. The main difficulty encountered with this learning rule was in choosing the correct learning factor for the best matching unit, ϵ_{bmu}, and for the direct neighbor units, ϵ_{nb}, as well as for the decreasing of ϵ_{bmu} and of ϵ_{nb} during the learning process.

We solved this problem by using *frequency*[a] modulated individual learning rates for each *bmu* $\epsilon_{bmu,i}$ and for each neighbor unit of the *bmu* $\epsilon_{nb,i}$: If the learning rate $\epsilon_{bmu,i}$ is substituted by $\frac{1}{n_{bmu,i}}$ and if the counter $n_{bmu,i}$ is increased by 1 each time when neuron i is *bmu*, then the resulting learning rule for the *bmu* (first case of Eq. 7) computes the exact (non-floating) average of all feature vectors which have fallen into the Voronoi polyhedron of neuron i since the beginning of the training:

$$\Delta\boldsymbol{\mu}_i = \begin{cases} \dfrac{1}{n_{bmu,i}} * (\mathbf{x} - \boldsymbol{\mu}_i) & \text{for } i = bmu\,, \\ \dfrac{1}{n_{nb,i}} * (\mathbf{x} - \boldsymbol{\mu}_i) & \text{for } i \in Nh(bmu) \quad \text{and}\,, \\ 0, \quad \text{otherwise}\,. \end{cases} \tag{7}$$

Applying these individual learning rates, the DCS is observed to rapidly converge during the first training periods. To avoid a "freezing" of the network caused by too slow learning rates $\frac{1}{n_{bmu,i}}$, an increase in $n_{bmu,i}$ is stopped if $n_{bmu,i}$ reaches a threshold calculated by $k_{bmu} \times z_{bmu,i}$. Here k_{bmu} is a constant and $z_{bmu,i}$ counts the number of feature

[a] The term 'frequency' refers to the frequency of a unit to become *bmu*.

vectors that have fallen into the Voronoi polyhedron of neuron i during the training steps of the last training cycle. Hence the learning rate does not decrease any further and the individual learning rule for the neuron i starts to behave in the same way as the original update rule (i.e. calculation of a floating average) after approximately k_{bmu} training cycles. The learning rule for the neighbors of the *bmu* (second case of Eq. 7) is similar to the *bmu* case just discussed. The individual learning rate is $\frac{1}{n_{nb,i}}$, where $n_{nb,i}$ counts the number of times the neuron i has been a neighbor of a *bmu*. The factor k_{nb} restricts the growth of $n_{nb,i}$ and is chosen to be about 5 to 10 times greater than k_{bmu}.

2.2.3. *Insertion and deletion of neurons*

The faster convergence of the centers and the improved stability in regions of high data density support the rapid formation of PTFMs in these regions. Other modifications of the original DCS learning rules (detailed in Ref. 25) concern the Hebbian learning rule, shrinking the variances of the Gaussians and an additional deletion strategy for excess neural units.

The learning factors of adjacent neurons also increase, because the inserted neuron should represent some properties of the feature vectors which were previously represented by its neighbors.

The insertion of neurons is guided by a local error measure attached to each neuron which is computed as the cumulative output error of that neuron. A new neuron is inserted between the unit with the highest error and its neighbor with the second highest error measurement. An additional effect of the learning rule is that a recently inserted unit always starts training with a high learning factor. The learning factors of its neighbors are also increased to adapt to the changed situation because the number of feature vectors which fall into the Voronoi polyhedron may have changed. As the network grows, a situation may develop where a neuron has no neighbors. Such isolated units occur where a unit has been neither a *bmu* nor a neighbor of a *bmu* during the last two training cycles. A second group of units are dead end units which are only neighbors to a *bmu* for one cycle but have never been a *bmu* themselves during the entire last training cycle. Both groups of neurons contribute very little or nothing at all to the output of the DCS network and therefore become subjects for deletion.

With the insertion or deletion of neurons, no special care is taken to maintain the adjacency matrix apart from inserting or deleting the appropriate row and column of the matrix. The correct neighborhood relationships in the affected part of the DCS network are established during the next training cycle by the competitive Hebbian learning rule (see formula 6).

2.2.4. *Relation to previous work*

Further applications of DCS utilizing different learning rules can be found in Ref. 5 employing error feedback learning for adaptive saccade control of a binocular head, in Ref. 2 employing real valued reinforcement learning for learning collision avoidance with a mobile robot, in Ref. 3 employing Q-Learning for learning discrete control policies, in Ref. 4 employing supervised learning for pole balancing and in Ref. 22 using unsupervised learning for incremental category learning of KHEPERA robots. Supervised DCS has been successfully applied to a series of classification tasks, as in Ref. 6, as well as to regression tasks and time series prediction, as in Ref. 10.

Finally, it should be mentioned that independently of us, B. Fritzke has further developed his GCS and created his Growing Neural Gas (GNG) which is virtually identical to our GDCS (compare Refs. 1 and 9).

2.3. *Class and activity map*

In the application presented here, we have considered 9 different classes of facial landmarks: Inner eye corner, outer eye corner (for both the left and the right eye), up to 4 intersections of the iris with the upper and lower eyelid and a point on the eyelid wrinkle. It should be pointed out, that the intersections of the iris with the lower eyelid may not exist in every eye region. Although most of the pixels in an eye region are not keypoints we do not introduce an explicit rest class, i.e. a class of non-keypoint positions. The reason is that the network should not allocate neural units for a class we are not interested in. Instead we implicitly define the rest class by all pixels with an activity below a relative class dependent threshold. By presenting examples from the 9 keypoint classes, only keypoints are finally represented by the network.

In order to obtain a class map of an image [(Fig. 8(a)], all pixels of the image region are classified by presenting their feature vectors to the DCS network. Since we have not explicitly defined a rest class, each pixel — even at non-keypoint positions — is labeled as belonging to a certain keypoint class

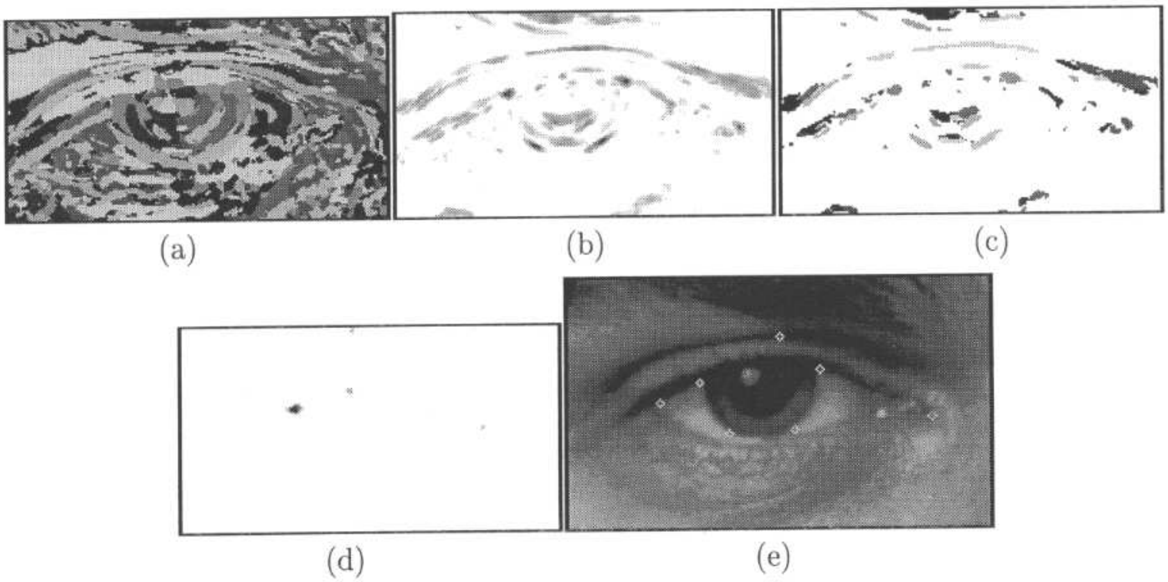

Fig. 8. Classification results of the DCS network after presentation of an eye region. (a) Class map for the entire considered image part without threshold; each pixel is assigned to a particular keypoint class encoded in distinct grey-levels. (b) Activity map of the classification result; high activity is encoded by dark grey and low activity by bright values. The activity is increased only at potential keypoint positions and in their direct neighborhood. (c) Resulting classification map of all keypoint classes after application of class dependent activity thresholds. All white pixels are assigned to the rest class and each grey value encodes a distinct keypoint class. (d) Activity map related to the keypoint class 'upper left intersection of the iris with the upper eyelid'; the biggest black blob indicates the position of the expected keypoint position. The other smaller grey points mark positions, at which false positive classifications may occur, if the sequential search strategy terminates there. (e) Original eye region in which the keypoints are marked. The image size of the computed classification results is smaller than of the original facial region because of the 27×27 kernel size of the applied filters.

[(Fig. 8(a)]. Additionally, a second map, the activity map, is computed [(Fig. 8(b)] which encodes the activity induced by each pixel. The activity of a pixel is defined as the activation of the *bmu* on presentation of the feature vector $\mathbf{x}$ of the considered pixel position (Eq. 3). Combining the class map and the activity map we obtain the resulting classification map as depicted in Fig. 8(c) where all the pixels with an activity below the class average have been assigned to the rest class.

Taking the 'upper left intersection of the iris with the upper eyelid' keypoint class as an example, Fig. 8(d) shows that only very few pixels belong to a particular keypoint class. Among them there are image positions in the direct neighborhood of the keypoint position and some pixels at positions which are totally unrelated to the original keypoint position. The latter pixels will contribute to the very low false positive decision probability (1% on average) (see Sec. 2.4 ff). Obviously, this probability strongly depends on the selection of the class dependent activity threshold because the threshold is responsible for the assignment to the rest class.

2.4. *Bayesian framework for verification*

Within a Bayesian framework the introduction of a verification stage for keypoint verification can be justified as follows:

First, the sequential search for a keypoint of class C_i is modeled as a random experiment. The position at which the sequential search terminates is assigned to the random variable X and hence $P(X = C_i)$ denotes the probability that the position at which the search terminated really belongs to the keypoint class for which we have searched. $P(X \neq C_i)$ is the error probability that the sequential search has failed, i.e. that we have terminated at a position not belonging to the desired keypoint class C_i.

In order to further reduce the error probability of taking a non-keypoint as a keypoint of class C_i we introduce a second stage, the verification stage.

This verification stage analyzes the position at which the search terminated and classifies it as belonging to one of the n keypoint classes. Modeling the verification stage as a second random experiment and assigning the output of the classifier to the random variable Y, it decides for keypoint class C_i with conditional probability $P(Y = C_i|X = C_i)$ and $P(Y = C_i|X \neq C_i)$. The conditional classification error is determined by $P(Y = C_i|X \neq C_i)$ for false positive classification, and by $P(Y \neq C_i|X = C_i)$ for false negative classification. For the combined system the *a posteriori* probability of deciding on a keypoint of class C_i but not having terminated at a position of that keypoint class is $P(X \neq C_i|Y = C_i)$. Using Bayes's theorem this probability calculates as

$$\begin{aligned} P(X \neq C_i|Y = C_i) &= \frac{P(Y = C_i|X \neq C_i)\ P(X \neq C_i)}{P(Y = C_i)} \\ &= \frac{P(Y = C_i|X \neq C_i)\ P(X \neq C_i)}{P(Y = C_i|X \neq C_i)\ P(X \neq C_i) + P(Y = C_i|X = C_i)\ P(X = C_i)}\,. \end{aligned} \tag{8}$$

Obviously, the verification stage only makes sense if the *a posteriori* error probabilities $P(X \neq C_i|Y = C_i)$ turn out to be smaller than the *a priori* probabilities $P(X \neq C_i)$. In Sec. 8 we will show that due to the small conditional error probabilities of the DCS based verification stage they are indeed *much* smaller.

3. Results

3.1. *Verification of keypoints with the DCS*

To train the network, we use feature vectors calculated from keypoint positions and their near neighborhood. The overall number of examples was 668 keypoint positions obtained from 110 different images of left and right eyes. Feature vectors from the keypoints of 87 of the 110 eye images formed the training set. The remaining 23 eye regions contributed to the validation set. The false negative classification error averaged over all keypoint classes for the training and the validation set as a function of the number of neural units inserted in the DCS network is shown in (Table 1). Since the (false negative) classification error for the validation set does not fall below 4.9% for more than 127 neural units this was the point chosen to stop the growing of the DCS network. Otherwise generalization capabilities can be expected to decrease.

Table 1. Classification error related to the size of the DCS network.

Network Size						
Neurons	23	32	59	87	115	127
Training Set						
Error (%)	9.7	8.2	5.5	3.4	1.5	0
Validation Set						
Error (%)	8.5	7.8	7.0	6.3	5.6	4.9

Table 2 shows the results for the combined search and verification system for four related keypoint classes. In this table,

- $\hat{P}(X \neq C_i)$ denotes the *a priori* probability of the sequential search strategy terminating at a position not belonging to the expected keypoint class C_i,
- $\hat{P}(Y = C_i|X \neq C_i)$ the conditional probability for a false positive classification of the DCS network,
- $\hat{P}(Y \neq C_i|X = C_i)$ the conditional probability for a false negative classification of the DCS network and, finally,
- $\hat{P}(X \neq C_i|Y = C_i)$ the *a posteriori* probability of the combined system.

The latter are much lower than the prior error probabilities, due to the low conditional classification error of the DCS.

All the items in Table 2 are based upon examinations of an extended face database (more than 110 face images).[14] Since we have considered only image positions where the computation of the sequential search strategy has terminated, there are differences in the amount of test data for each class of keypoints. Only isolated false positive and false negative detections were

Table 2. Empirical error probabilities (%).

Keypoint Class C_i	$\hat{P}(X \neq C_i)$	$\hat{P}(Y = C_i \mid X \neq C_i)$	$\hat{P}(Y \neq C_i \mid X = C_i)$	$\hat{P}(X \neq C_i \mid Y = C_i)$
Inner eye corner	14.4	6.7	2.2	1.1
Outer eye corner	8.3	11.1	1.0	1.0
Int.s. iris/upper lid	3.2	14.3	0.9	0.5
Int.s. iris/lower lid	4.1	0.0	0.9	0.0
Overall	6.5	8.3	1.2	0.2

observed. The results of the class 'eyelid wrinkle' are not included in the table because strictly speaking the eyelid wrinkle cannot really be viewed as a single keypoint position. There is no exact and reliable definition for the location of this keypoint, a range of values for the location of the contour or the dark line on the eyelid must be taken into account. Therefore, an evaluation comparable to the other 'real' keypoints is impossible. The results presented in previous work are obtained based on different and much smaller datasets.[16]

3.2. *DCS based keypoint detection*

In an additional experiment we tried to detect the keypoints directly with the DCS network, i.e. without the sequential search strategy. To this end we generated class maps and searched for that pixel with the maximum activity for each keypoint class. The results for this purely data driven keypoint detection are summarized in Table 3.

The relatively high error rate for the keypoint detection in particular for the detection of the eye corner keypoints and the eyelid keypoint is not due to the DCS network failing to assign the correct class to keypoint positions but rather to other image structures somewhere in the eye region inducing more activation in the network than the exact keypoint position. The reason is that in complex images there are a lot of image structures with comparable or very similar structure to that of the keypoint under investigation and that, by training the network on keypoints only, we did not teach the network to discriminate between keypoints and these similar structures.

Table 3. Purely data driven detection error.

	Inner Eye Corner	Outer Eye Corner	Intersection Iris/Lids	Eyelid Wrinkle
Error (%)	56.5	47.8	11.9	61.9

For illustration, the keypoints detected by the purely data-driven keypoint search are depicted in Fig. 9 taking three images from the validation set as an example. In the eye images the keypoints of the eyelid wrinkle and of the eye corner are sometimes misclassified because of the reasons already mentioned, but the keypoints "upper and lower intersection points of the iris with the eyelids", are detected reliably without any preliminary sequential search because of their characteristic image structure.

4. Concluding Remarks

In this paper we have presented the application of a DCS network for the verification of keypoint positions in facial images which have been computed in advance by a model and data driven sequential search strategy. We have shown that the DCS network is able to verify keypoint locations suggested by the sequential search strategy with high reliability and that the results for the combined search and classification scheme as expressed in the last column of (Table 2) are very promising.

The results given in Table 3 concerning the direct DCS based keypoint detection indicate that the purely data driven approach is unlikely to lead to satisfactory results for keypoint detection in complex real world images. The reason is that no relationships between the important structures in the facial region have been considered. The integration of model knowledge is essential for a successful and reliable detection of real world keypoints because of

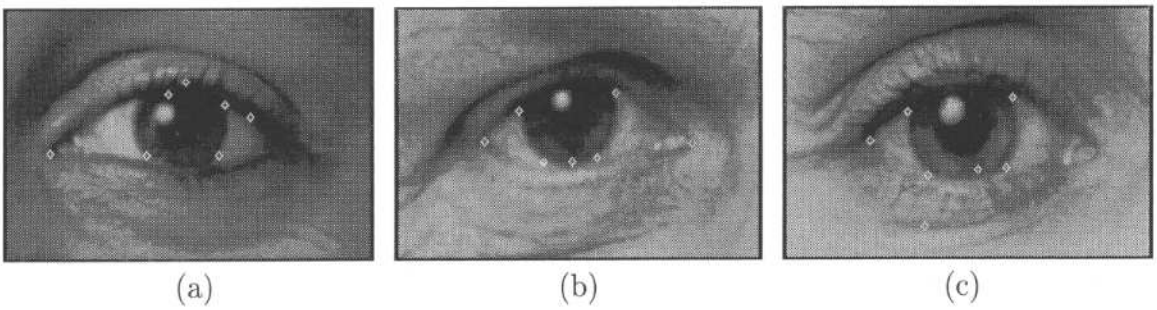

Fig. 9. Examples of detected keypoints in three eye regions from the validation set (a)–(c). The depicted results are computed by direct classification of each image position of an eye region with the DCS network but without sequential search of the keypoints in advance. Since the network has been trained with keypoints only (no negative examples) not all keypoints in an eye region can be classified correctly. In particular, the keypoint on the eyelid wrinkle is often misclassified because its location cannot be predicted as reliably as for an eye corner. The underlying structural information is not as straightforward to compute as for obvious classifications.

the large variation in the grey value distribution of the keypoints to be searched. The eye region is a good example because its multiple structures caused by eyelashes and a lot of small wrinkles on the upper and lower eyelids. The discrimination between keypoints and non keypoint positions may be impossible if no context information is used because too many image structures may be present with comparable structural grey value distributions.

The performance of the direct DCS based detection may be improved for particular very well defined keypoint positions by a modified or enhanced training scheme. The intersection of the iris with the eyelids may serve as an example. In this case the image structure is so characteristic and straightforward that a reliable detection is possible, but no guarantee against false detections can be given. Indeed we have observed some remarkable failures, where the border of a highlight was misclassified as an intersection keypoint because the structural information resembled very closely that of the ordinary keypoint under examination.

The presented *hybrid* approach of a model based search strategy followed by a data driven neural network verification component outperforms each of its constituents taken alone. It is an excellent example of how methods for image analysis can be evaluated and enhanced using state of the art neural networks.

Acknowledgments

We are grateful to Dr. M. Michaelis from the Medis-Institute at the GSF Research Center for providing the steerability scheme and for many helpful discussions. The work has been partially supported by DFG grants So 320/2-1 and Ei 322/1-1.

References

1. J. Bruske and G. Sommer 1995, "Dynamic cell structures," in *Proc. NIPS 7*, pp. 497–504.
2. J. Bruske, I. Ahrns and G. Sommer 1996, "An Integrated Architecture for Learning of reactive behaviors based on dynamic cell structures," Technical Report 9604, Inst. f. Inf. u. Prakt. Math., CAU zu Kiel.
3. J. Bruske, I. Ahrns and G. Sommer 1996, "Practicing Q-learning," in *Proc. ESANN'96*, pp. 25–30.
4. J. Bruske, I. Ahrns and G. Sommer 1995, "On-line learning with dynamic cell structures," in *Proc. ICANN'95* Vol. 2, pp. 141–146.
5. J. Bruske, L. Riehn, M. Hansen and G. Sommer 1996, "Dynamic cell structures for calibration-free adaptive saccade control of a four-degrees-of-freedom binocular head," Technical Report 9608, Inst. f. Inf. u. Prakt. Math., CAU zu Kiel.
6. J. Bruske and G. Sommer 1995, "Dynamic cell structure learns perfectly topology preserving map," *Neural Comput.* **7**(4), 845–865.
7. W. T. Freeman and E. H. Adelson 1991, "The design and use of steerable filters for image analysis," *IEEE Trans. PAMI* **13**(9), 891–906.
8. B. Fritzke 1993, "Growing cell structures — A self organizing network for unsupervised and supervised training," ICSI Berkeley, Tech.-Rep., tr-93-026.
9. B. Fritzke 1995, "A growing neural gas network learns topologies," in *Proc. NIPS 7*, pp. 497–504.
10. A. Giordana and P. Katenkamp 1995, "Growing radial basis function networks," in *Proc. EWLR-4*, Karlsruhe.

11. R. Herpers, H. Kattner, H. Rodax and G. Sommer 1995, "GAZE: An attentional processing strategy to detect and analyze the prominent facial regions," in *Proc. Int. Workshop Autom. Face and Gesture Rec.* ed. M. Bichsel, Zurich, Switzerland, pp. 214–220.
12. R. Herpers, M. Michaelis and G. Sommer 1995, "GAZE: Detection and analysis of facial regions applying an attentive processing scheme," GSF-Bericht, 23/95, D-85764 Oberschleissheim, Germany, 1995.
13. R. Herpers, M. Michaelis, G. Sommer and L. Witta 1995, "Detection of keypoints in face images," Tech.-Rep. GSF-Bericht 24/95, D-85764 Oberschleissheim, Germany.
14. R. Herpers, M. Michaelis, K. H. Lichtenauer and G. Sommer 1996, "Edge and keypoint detection in facial regions," in *Proc. 2nd Int. Conf. on Automatic Face and Gesture Recognition FG'96*, Killington, Vermont, (IEEE Computer Society Press), pp. 212–217.
15. R. Herpers, M. Michaelis, L. Witta and G. Sommer 1996, "Context based detection of keypoints and features in eye regions," in *Proc. 13th Int. Conf. Pattern Recognition, 13-ICPR*, Wien, (IEEE Computer Society Press), Vol. B, pp. 23–28.
16. R. Herpers, L. Witta, J. Bruske and G. Sommer 1996, "Evaluation of local image structures applying a DCS network," in *Solving Engineering Problems with Neural Networks*, Proc. of the 2nd Int. Conf. EANN96, London, eds. A. B. Bulsari, S. Kallio and D. Tsaptsinos, pp. 305–312.
17. M. Michaelis 1995, "Low level image processing using steerable filters," PhD thesis, Christian-Albrechts-Universität, D-24105 Kiel, Germany.
18. M. Michaelis, R. Herpers and G. Sommer 1995, "A common framework for preattentive and attentive vision using steerable filters," in *Proc. CAIP'95 Prague*, eds. V. Hlaváč and R. Šava, pp. 912–919.
19. T. Martinetz 1993, "Competitive Hebbian learning rule forms perfectly topology preserving maps," in *Proc. ICANN 93* pp. 426–438.
20. T. Martinetz and K. Schulten 1994, "Topology Representing Networks," *Neural Networks* **7**, 505–522.
21. P. Perona 1992, "Steerable-scalable kernels for edge detection and junction analysis," in *Proc. ECCV'92*, ed. G. Sandini, pp. 3–18.
22. C. Scheier 1996, "Incremental category learning in a real world artifact using growing dynamic cell structures," in *Proc. ESANN'96*, pp. 117–122.
23. S. Stengel-Rutkowski, P. Schimanek and A. Wernheimer 1984, "Anthropometric definitions of dysmorphic facial signs," *Hum. Genet.* **67**, 272–295.
24. S. Stengel-Rutkowski and P. Schimanek 1985, *Chromosomale und nicht-chromosomale Dysmorphiesyndrome* (Enke Verlag, Stuttgart).
25. L. Witta 1995, "Entwicklung von steuerbaren Filtern zur Ableitung von robusten Merkmalen," Diploma-thesis, Institut für Informationstechnik, Technical University Munich, September 1995.