Visual System based on a Chorus of Shape Categorization Modules

George Valentin Voina

Katholieke Universiteit Leuven, Center for Processing Speech and Images (ESAT/PSI), Kasteelpark Arenberg 10, B-3001 Leuven (Heverlee), BELGIUM, *Phone:* +32-16-32.1095, *Fax: +32-16-32.1723, E-mail:* George.Voina@esat.kuleuven.ac.be

Abstract. The paper describes a Visual System based on a Chorus of Shape Categorization Modules (C-SCM) inspired by the biological visual system. As in *"Representation and Recognition in Vision"* by Shimon Edelman, the shape recognition problem is tackled by specially trained SRMs (shape recognition modules), able to express the similarity of a given shape with the fundamental shape of the SRM. The categorization decision in a given context is based on a SCM (shape categorization module), consisting of the chorus of SRMs of the shapes defined in that context. Finally, a classifier of the C-SCM Visual System can be constructed based on the chorus of SCMs of all the system's known contexts. The power of the C-SCM visual system is given by its ability to define on its own, using as input only basic features, optimal complex features, and based on them to create abstract models of shapes or shape categories. The behavior of such systems, having different training algorithms for the SRM's, and different designs for the SCM's and C-SCM, are tested on the COIL-100 library.

1 Introduction

1.1 About object categorization

Object recognition is about recognizing specific objects like "my yellow car". In the case of object categorization, the the goal is no longer to match an object which was already encountered, to a model stored internally by the system. The new goal is to be able to make pertinent decisions regarding a new perceived object, based on the previous knowledge acquired by the system, by giving a more abstract response like "this is a car". Such a system requires higher level cognitive reasoning, in other words the system should posses the necessary "intelligence" which allows him to distinguish visual object classes not only individual objects.

In the quest for a reliable model of an artificial visual system, able to categorize the objects from its environment, several classes of theories have been developed:

1. *Structural decomposition based theories* (eg.[2]). An object is represented by a set of generic standard components and the relationships between them.

This theories are suitable for simple tasks, in which the environment where the objects are situated is not so complex. The problems of this methods are the need to find the best standard components which are to be detected, and the standard component's detection reliability. Another big problem is the instability of description in terms of standard components, problem which stems from the possibility of decomposing a shape in elementary parts in a number of different ways.

2. *Geometric constraints based theories* (eg.[3]). An object is represented by a geometrical model. The problems of this methods are linked to the reliability of the geometric feature extraction and of the feature correspondence algorithms. The poor performance of this steps can make a system based on this kind of approaches completely unreliable.
3. *Multidimensional feature spaces based theories* (eg.[4]). An object is represented by a point or a cloud in a multidimensional feature space. This theories are more general and also more close to the representations that appear in the real biological visual systems.

The theory presented in this paper is part of the class of *multidimensional feature spaces based theories.* The main idea of the theory is to try to model the way the biological visual systems behave in the context of object categorization.

1.2 An abstract model of the biological visual system and possible parallels with artificial vision systems

Nature offers us a model, the biological visual system, always able to answer to a recognition or categorization problem, learning from its experience and adapting its knowledge continuously. So, it is natural to consider the design of the biological visual system [5], appropriate to follow in the development of an artificial general purpose visual system. In the process of formulating such a theory, a range of problems appear. The most important one is linked to the fact that nobody knows exactly how the biological visual system actually works. All the data we have at our disposal are experimental data, and the correlations between all the acquired knowledge in this domain is not so easy to achieve.

In the biological visual systems (Fig.1), as a concrete example the human visual system is used, in the first stage the light entering the eye is detected by the retina. Retina, an array of photo-receptors on the rear surface of the eye, encodes this detected light levels in electric signals. At the next step this electric signals are transmitted through the optic tract to the lateral geniculate nucleus (LGN) of the thalamus, situated at the base of each side of the brain. From the LGN the signals travels to the primary visual cortex (V1) at the rear of the brain. Here the first stage of the cortical processing of vision takes place. The output generated from this area goes to many different higher cortical areas of the brain where further higher complexity processing will take place.

A visual system which mimics the biological visual systems must implement at least the following processing steps:

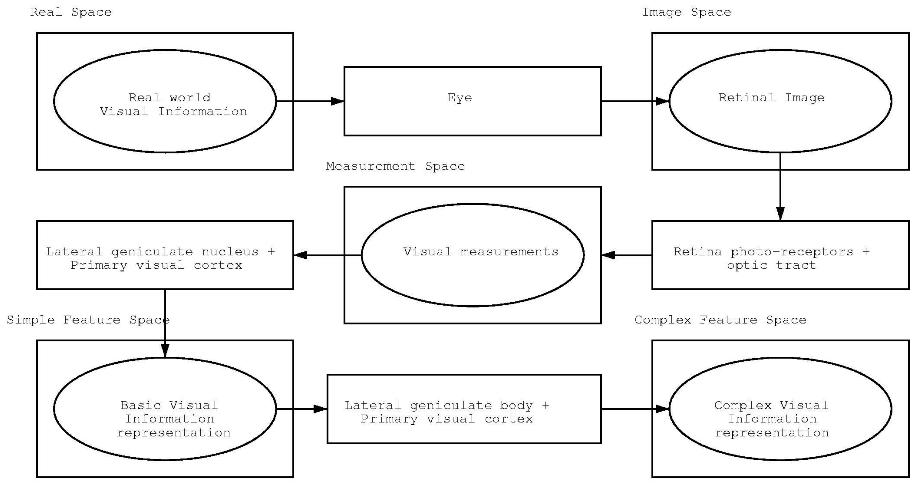

Real Space
Real world Visual Information
Eye
Image Space
Retinal Image
Measurement Space
Visual measurements
Lateral geniculate nucleus + Primary visual cortex
Retina photo-receptors + optic tract
Simple Feature Space
Complex Feature Space
Basic Visual Information representation
Lateral geniculate body + Primary visual cortex
Complex Visual Information representation

Fig. 1. Abstract representation of the biological visual system

1. *real space* to *image space* projection of the visual information, received at the input by the system (eyes optical system functionality).
2. *image space* to *measurement space* mapping, detecting *raw features* (retina functionality)
3. *measurement space* to *simple feature space* mapping, which based on *raw features* defines and detects *simple features* (primary visual cortex V1 functionality)
4. *simple feature space* to *complex feature space* mapping (higher cortical areas functionality).

The system should exhibit the same hierarchical organization as the biological system: in V1, the neurons are organized in *neuron columns* which react to the same *raw features* received from the *measurement space,* and in the rest of the cortical area, one can identify *cortical regions* which give similar reactions to the same *simple features,* received from V1.

In a simplified model, the *cortical area* of the brain can be modeled as a *complex feature space,* defined by *active landmark modules*, corresponding to specific *cortical regions*. An *active landmark module* exhibits a high response to a specific *shape*, and when faced with another shape, a lower response, proportional with the *similarity* between the *active landmark's* specific shape and the input shape. Different *complex feature spaces* are assigned to different levels of abstraction. So, in a *complex feature space* a point can represent a view of a shape, a shape, a shape class or a group of shape classes. Like the grouping of cortical regions (a group of cortical regions responding to more general shape models), the points and manifolds from a *complex feature space* combine hierarchically.

As in [1] we define the *distal shape space* as the external world of shapes, the *proximal shape space* as a context dependent *complex feature space* and the notion of *similarity* as a proximity metric in the *proximal shape space.* To this end, the *proximal shape space* is locally approximated as metric space. When defining *similarity, the problem of arbitrariness* [6](similarity is observer biased), *the problem of context* [7](similarity is context biased) and *the problem of asymmetry*[8] should be addressed. A context dependent similarity notion defined as a metric function deals with this problems.

1.3 Overview of the paper

Section **2** describes the architecture and the construction of a general shape categorization module SCM the building block of our Visual System. Section **3** describes the C-SCM Visual System constructed using SCMs defined on different contexts. In section **4** the behavior of a C-SCM Visual System is tested on the COIL-100 [9] library. Section 5 concludes the paper.

2 Architecture of a SCM

The first stage of a SCM, consists in a mapping F from a *distal shape space* to a *proximal shape space*, with the properties of *preservation of distinctness* (distinct points from *distal shape space* project to distinct points in *proximal shape space*) and *full similarity spectrum preservation* (neighboring points in the *distal shape space* project to neighboring points in the *proximal shape space*). The SCM assumes that the instances of a shape form a manifold in the *distal shape space*, so the F mapping will project a manifold from the *distal shape space,* to a manifold (an *active landmark*) in the proximal shape space, maintaining its topological properties. Due to the discrete nature of the data acquisition process, the SCM must be able to generalize from sparse data, so a manifold (*active landmark*) in the *proximal shape space* is approximated by interpolation with high dimensional Gaussian basis functions.

In the second stage of the SCM, based on the manifolds (*active landmarks*) obtained in the *proximal shape space* and the similarity measure defined on the *proximal shape space,* categorization decisions can be inferred by the SCM.

2.1 The *distal shape space* to *proximal shape space* mapping

Until now, the concepts of distal shape space and proximal shape space were introduced, and a similarity measure was defined on them. When the goal is to construct an artificial visual system the main focus is on the proximal shape space, or how the system should internally represent the world. In order for the artificial visual system to be able to work with concepts in the proximal shape space, first a projection of the distal shape space objects into its internal, or proximal shape space is necessary to be performed by the system.

The minimal requirements of this mapping are:

1. *preservation of distinctness*: the mapping should be a one-to-one mapping, distinct points in the distal shape space are mapped to distinct points in the proximal shape space
2. *full similarity spectrum preservation:* the mapping should preserve the structure of the distal shape space, two points which are nearest neighbor in the distal shape space must remain the same also in the proximal shape space

Formally, can be defined as:

$$F : D \to P$$

$$F = f_4 \circ f_3 \circ f_2 \circ f_1$$

where: D - distal shape space; P - proximal shape space;

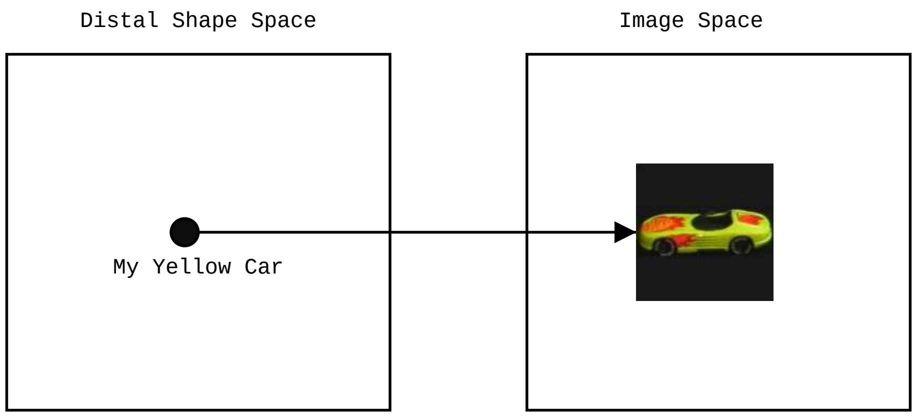


Fig. 2. The imaging component $f_1 : D \to I$

The imaging component f_1 (Fig.2) has the same functionality as the eye in the biological optical system. The image space I is the space on which the image views of the real shape are defined. This component is in fact a projection of the current view, as is perceived by the observer, of a real object (distal space object) in a 2 dimensional Euclidean space. In an artificial system this step is performed by the optical device used for the image acquisition. Depending on the type of image accepted by the next processing level, different filters or image preprocessing modules can be part of this step.

The measurement component f_2 (Fig.3) mimics the functionality of the retina photo-receptors. In the same way in which the retina photo-receptors detect raw features, and transmits them as electric signals to the next stages, the measurement component extracts raw basic image features and the resulting measurement vector is passed to the next level for further processing. The dimensionality

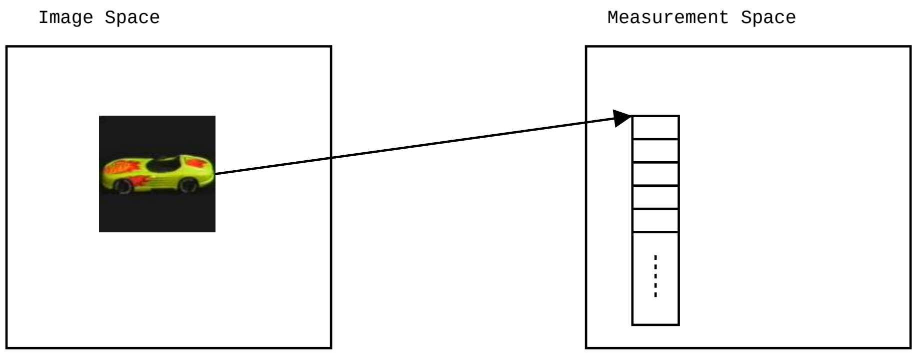


Fig. 3. The measurement component $f_2 : I \rightarrow M$

of the *measurement space* M space is given by the number of raw features extracted by the measurement component, from the image representation of a perceived view of a distal shape space object. The dimensionality of the measurement space depends on the accuracy of the measurement component, and also on the type of input required by the next higher level components of the F mapping, and usually is high-dimensional.

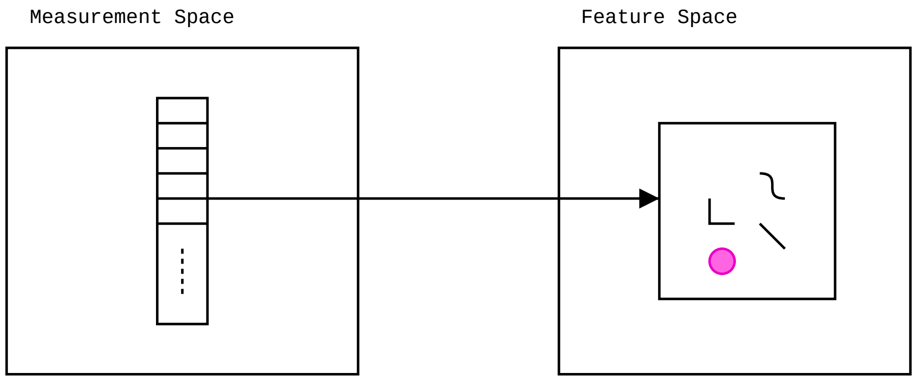


Fig. 4. The feature detection component $f_3 : M \rightarrow F$

The feature detection component f_3 (Fig.4), mimics the processing which occurs within the lateral geniculate nucleus and the primary visual cortex of a biological visual system. At this stage from the raw measures obtained by the measurement component, through specific processing simple features are detected. The features detected by the system at this stage are basic visual features like edges, corners, color structures etc. So, the features space F is similar to the simple feature space of a biological visual system.

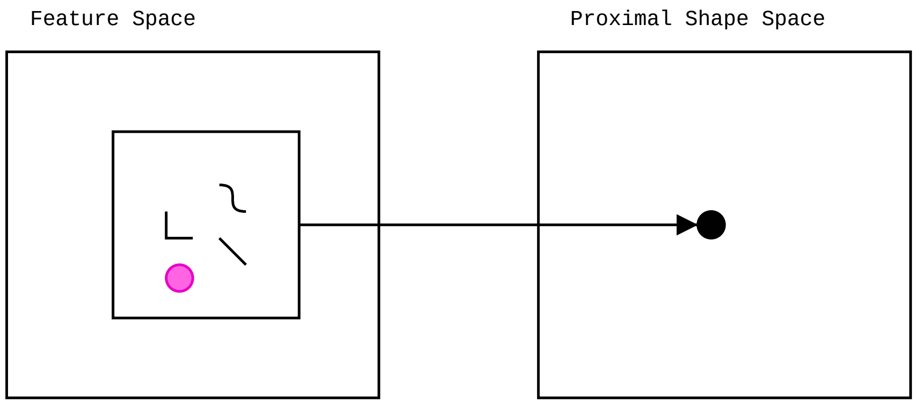


Fig. 5. The cognitive component $f_4 : F \to P$

The cognitive component f_4 (Fig.5), mimics the operations which take place in the higher cortical areas. At this stage using the simple features detected at the previous level, the system infer complex features based on which is able to recognize or categorizes a perceived instance of a shape, as being a representation of that real shape. The cognitive component can be viewed at its turn as a composition of several components which map shape representations from lower level proximal shape spaces to higher level proximal shape spaces (Fig.6). With each new component the abstraction level of the representation is increased.

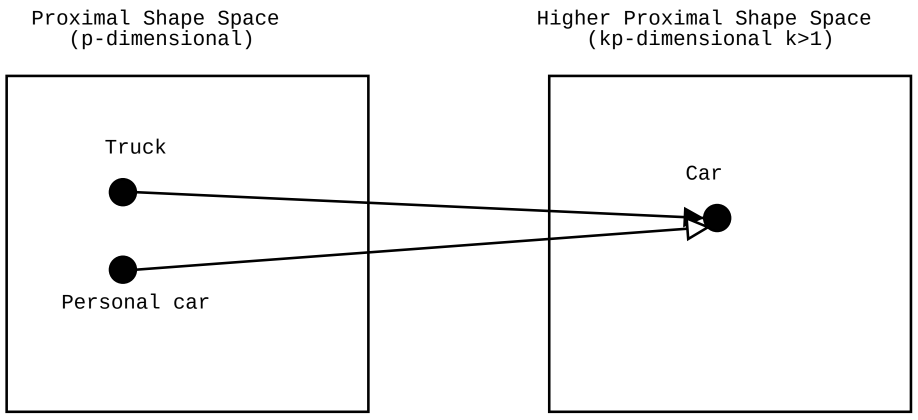


Fig. 6. Lower level proximal shape space to higher level proximal shape space mapping

2.2 Implementation of the *distal shape space* to *proximal shape space* mapping

In our artificial visual system, the f_1 component is implemented by the imaging device.

The f_2 component is a raw feature detection module. This component is implemented by a set of converting and filtering blocks arranged in a pipeline. In most of the cases the image obtained by an optical device (camera) is in RGB format. Because this format is far from the representation format of the image in a biological system, a format more close to the biological visual system, the HSV representation, is used as the input for the filtering blocks.

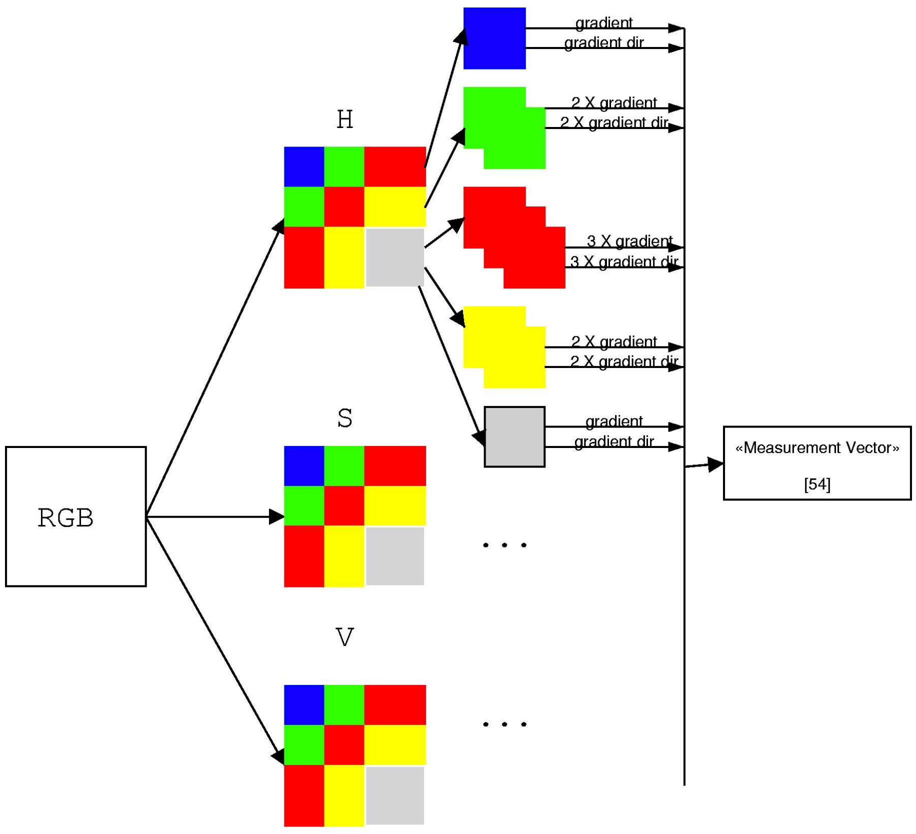


Fig. 7. RGB to HSV and patching operations

After the hue (H), saturation (S) and value (V) components of the initial image are obtained, each component image is decomposed in superposing tiles(patches) (Fig.7). The superposition of the patches is necessary, in order to eliminate some errors that may occur in the output of the measurements extraction modules. The errors are generated at the border regions in the case of considering a disjoint patch decomposition.

In the next step, on each image patch, a sobel type filter is used to obtain 2 measurements per patch, one expressing the magnitude of the "variation" (global gradient value) and the other the predominant direction of "variation" (predominant direction of gradient). Depending on the type of the patch the notion of "variation" refers to : hue variation, saturation variation or value (intensity) variation.

The quality of the measurement information extracted from an image is direct proportional with the number of patches in which the image is divided. As a consequence the number of raw features detected from an input image of an instance of a shape is high. This situation is similar to the biological visual system where the number of raw features can be considered close to the number of axons in the optic nerve, around 1 million. Due to this big number of raw features, the higher stages of the visual information processing must deal with the curse of dimensionality yielded by this. Because of the course of dimensionality the number of patches considered in the decomposition of an image is very important and we should derive a method to find an optimum of this number or to find a system topology which is able to compensate for sparse input data.

The implementation of the feature detection (f_3) and of the cognitive (f_4) mappings are usually treated as two separate problems. The first decision which needs to be made is what kind of simple features are going to be detected by the feature detection module: edges, corners, invariants, etc. . After that based on this features a model of the observed shape is created by the cognitive component, by recording the relations between the simple features observed at the input. At the recognition or categorization step the system searches for a specific set of features and tries to match the topology of the detected features to the topology of an already learned shape models. This kind of approach to the problem is more prone to erroneous results , mainly because of the wrong choice of the set of simple features that are used to model a shape.

A new approach is used in this paper, approach based on the idea that, the feature detection step is not explicitly implemented as a separate processing step. This is done by using as the base element for implementation of such a system a neural network. Contrary to the traditional approach functionality, this neural network based system defines its own set of simple features which will be used to model a complex shape. An external observer of the system has no idea, or is not interested, what simple features the system is using to model the observed shapes. This can be an advantage, because the system will be able to define simple features, which may not be so obvious for an external user, but in the same time being the best possible set of features which can be used to model a specific shape. So, the feature detection mapping becomes a hidden internal process of the system.

We make the assumption that all the possible instances (different views under different conditions and transformations) of a real world shape form a shape manifold in a shape space. In this case, the main goal of the distal shape space to proximal shape space mapping F is to project all the shape manifolds from the distal shape space to the proximal shape space, maintaining in the same time the topological properties of the manifolds. Because our system will get at the input sparse information about the distal space manifold of a shape (due to the discrete nature of the data acquisition process), a method must be found to ensure that the corresponding manifold constructed in the proximal shape space emulates "the true complete manifold" of that shape. This is necessary because the goal is to create a system able to categorize novel instances of a shape correctly, so a system able to do generalization.

The easiest way to construct a complete manifold of a shape (or try to do that), if we have only sparse information available, is to construct it by means of interpolation with high dimensional Gaussian basis functions. A manifold of a shape in the proximal shape space will look something like in (Fig. 8), where the color range varying from blue (value 0) to red (value 1) correspond to the intensity of the basis Gaussian functions. The manifold has the same characteristics as an energy cloud, with several "hot" spots and with a decreasing energy value towards the border regions.

This basis functions, also define a subspace of the proximal shape space, the topological space of that shape, in which each dimension is specified by a basis function.

Now, talking about the proximal shape space in the context of a categorization problem, this shape space can be seen as a space which embeds a collection of topological subspaces corresponding to the different shapes (in this context the notion of shape refers to a class of real world shapes, ex "box like shape"), which are used in the categorization decision process.

Because the shape manifolds embedded in the proximal shape space aren't discrete shape manifolds, like the shape manifolds constructed by the discrete measurement data in the distal shape space, but rather border-less topological continues subspaces (same as the energy clouds), the proximal shape space can be viewed as an energy field map described by active landmarks.

In this model, each shape manifold embedded in the proximal shape space is seen as an energy emitting landmark. A landmark is defined as having a unitary uniform energy value in a kernel region, and emitting a radial energy signal, who's intensity decreases in an exponential fashion when it propagates through the embedding space (Fig.9). Because the characteristics of the emitted signal of each landmark and the signal diffusion properties of the embedding space are known, the position of an arbitrary point in the embedding space can be fully specified by the vector of the signal intensities of each landmark at that point. The metric defined on the proximal shape space has the same behavior in the proximal shape space as the metric defined by the signal intensity of a landmark in the energy map. The metric defined in the proximal shape space is in fact

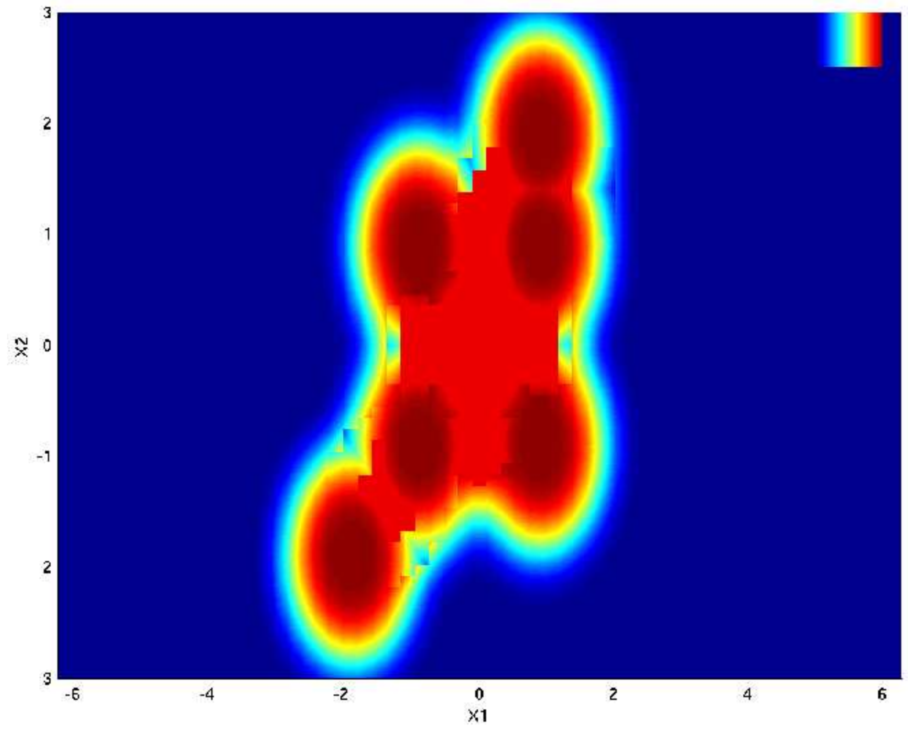


Fig. 8. Example of a 2D shape manifold spanned by 6 HBF basis functions

the result of the F mapping of the similarity metric defined on the distal shape space.

The f_3 and f_4 components are implemented by SRMs (shape recognition module), trained for specific shapes, designed as 3 layer HBFs (Hyper Basis Function [10]) neural networks, with Gaussian basis functions. Trained to mimic the behavior of an *active landmark,* a SRM responds with 1 to an instance of its associated shape, or with a value in [0,1) proportional with the *similarity* distance between the input shape instance and the shape associated with the SRM.

A HBF architecture (Fig.11), described by the equations from (Fig.10) is chosen to implement the SRM, instead of a simpler RBF architecture used in [1], because in this case the basis functions are no longer radial, so the SRM is able to model more complex shape manifolds. Also, in [10] is proved that an approximator based on a HBF architecture oscillates less, so it presents a smoother behavior. The input weights of an HBF type SRM perform an implicit normalization of the response and an input noise filtering, resulting in a shorter convergence time of the SRM tuning. The use of exponential basis functions,

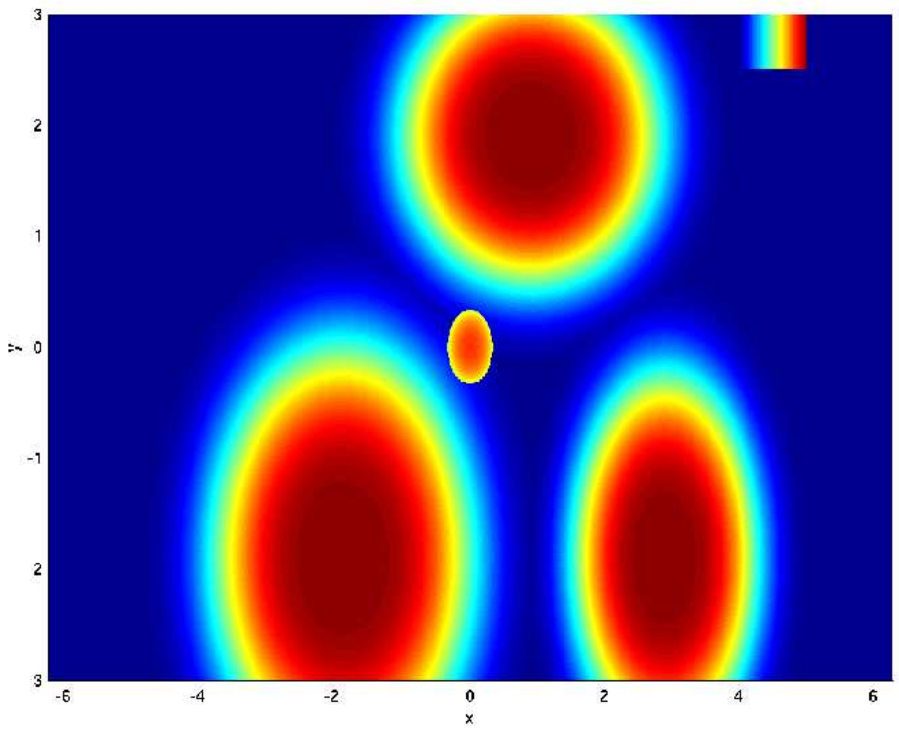


Fig. 9. 4 shapes in a proximal space

down-sizes the *curse of dimensionality,* but is not enough to drop it as a problem of the system.

An HBF neural network is completely specified by choosing the following parameters:

1. The number H of radial basis functions;
2. The centers and the position of the basis functions;
3. The input layer weights and the output layer weights;

The number H of radial functions is a critical choice and depending on the approach can be made a priori or determined incrementally. We will call a learning algorithm that starts with a fixed number H of radial functions determined a priori 'static' , and an algorithm that during the computation is able to add or delete one or more basis functions 'dynamic'.

A static learning algorithm is also parametric, because the search for the optimal neural network which models the SRM, corresponds to a search in the parameter space defined by the fixed number of radial basis functions. A dynamic learning algorithms changes the parameter space in which it operates, while adding or deleting radial basis functions.

Layer	Input	Activation	Size
Input	-	a_k	M
Hidden	$h_i = \left\Vert \{v_{ki}\}_{k=1}^{M} \otimes \left(\{a_k\}_{k=1}^{M} - \{b_{ki}\}_{k=1}^{M} \right) \right\Vert_2^2$	$y_i = e^{-\frac{h_i^2}{\sigma_i^2}}$	H
Output	$o = \sum_{i=1}^{H} w_i y_i$	$SRM = o$	1

w_i=hidden layer weights; v_{ki} =input layer weights; a_k =k-th input feature; b_{ki} =k-th feature the i-th key view; σ_i =width of the i-th basis function; $\Vert\overrightarrow{x}\Vert_2 = L_2$norm of $\overrightarrow{x}$; $\otimes$= element-wise matrix multiplication.

$$\begin{aligned}
h_i &= \left\Vert \{v_{ki}\}_{k=1}^{M} \otimes \left(\{a_k\}_{k=1}^{M} - \{b_{ki}\}_{k=1}^{M} \right) \right\Vert_2^2 \\
&= \left\Vert \{v_{ki}a_k\}_{k=1}^{M} - \{v_{ki}b_{ki}\}_{k=1}^{M} \right\Vert_2^2 \\
&= \left(\{v_{ki}a_k\}_{k=1}^{M} - \{v_{ki}b_{ki}\}_{k=1}^{M} \right)^T \left(\{v_{ki}a_k\}_{k=1}^{M} - \{v_{ki}b_{ki}\}_{k=1}^{M} \right) \\
&= \left(\{v_{ki}a_k - v_{ki}b_{ki}\}_{k=1}^{M} \right)^T \left(\{v_{ki}a_k - v_{ki}b_{ki}\}_{k=1}^{M} \right) \\
&= \sum_{k=1}^{M} (v_{ki}a_k - v_{ki}b_{ki})^2 \qquad (5)
\end{aligned}$$

Fig. 10. Equations of the SRM neural network

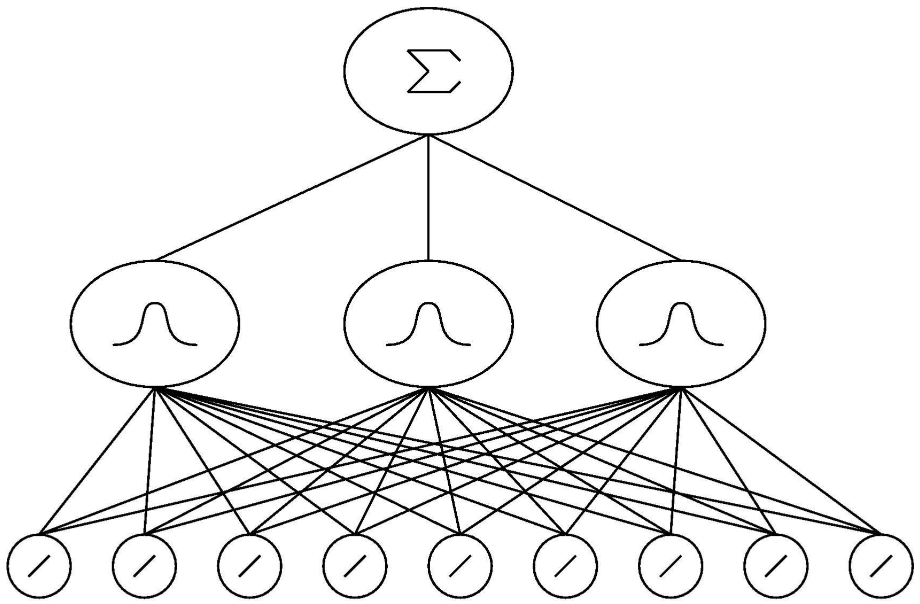


Fig. 11. Example of a SRM defined on a 9 dim input space and spanned by 3 HBFs

```
1. set global generalization error $Err = \infty$, key views set $I_c = \{view_1\}$,
   $keyview_{next} = view_y$; $y \in [2, \ldots, T]$, $view_i \in S_{train}$
2.1 $S_{train}$ isn't optimally described by $I_c$, so add $keyview_{next}$ to $I_c$
2.2 compute the new basis function width $\sigma$ corresponding to the new $I_c$,
    as the square of the average $L_2$ norm distance between the elements of $I_c$
    $\sigma_i = \frac{(n-k)\cdot k!}{n!} \cdot \sum_{i=1}^{H-1} \sum_{j=i+1}^{H} \|keyview_i - keyview_j\|_2^2, i \in [1 \ldots H]$
2.3 construct a SRM with $\sigma$ wide basis functions having elements of $I_c$ as
    key views and with $v_{ki} = 1$, $w_i = 1, \forall i, k$
2.4 Generalized Lloyd Iteration (K-medoid clustering)
2.4.1 let $O_c$ and $OSRM$ two empty sets
2.4.2 $\forall view \in S_{train}$ if $view \in I_c$ then add $SRM(view)$ to $O_c$ else add
      $SRM(view)$ to $OSRM$.
2.4.3 $\forall p_k \in O_c$ define $CL_k = (p_k, A_k)$ a cluster centered in $p_k$, where $A_k$ the
      set of all the elements of the cluster except $p_k$, initially empty
2.4.4 $\forall q \in OSRM$ if $CL_x$ is the cluster who's centroid is the closest to $q$ then
      add $q$ to $CL_x$
2.4.5 $\forall CL_k$ clusters, find the new centroid $p_k^{new}$ as the k-medoid of $CL_x$
2.4.6 define the improved $O_c = \{p_k^{new} \mid k \in [1 \ldots H]\}$ and define the im-
      proved $I_c = SRM^{-1}(O_c)$ .
2.4.7 compute $E_{Lloyd} = \sum_{k=1}^{H} \sum_{j=1}^{card(A_k)+1} \|p_k^{new} - d_{kj}\|_2^2$, $d_{kj} \in CL_k$.
2.4.8 if step<MaxStep and $E_{Lloyd}$ >suitable threshold then goto 2.4
2.5 if $E_{Lloyd} < Err$ then $Err = E_{Lloyd}$ and select $keyview_{next} \in O_c$ with
    the smallest sum of square distances to the elements of $O_c$, goto 2.1
3. construct the improved SRM as in step 2.3
4. set the limits of the basis function widths, $\sigma_{min}$ and $\sigma_{max}$, the squares of
   the min. and max. $L_2$ norm distance between the elements of $I_c$.
```

Fig. 12. ***Network Initialization Algorithm***

The learning algorithms are also very different depending on whether the sample set S is completely available before the learning process or if it is given, sample by sample, during it. In the former case, off-line learning is possible while in the latter, an on-line learning approach is needed. While some of the static algorithms can be adapted for both learning type, the application of the dynamic one makes sense only in the case of on-line learning.

2.3 Training SRMs with static learning algorithms

A SRM associated to a shape is trained using 2 sets of shape instances S_{train} and S_{test}. First the centers (key views) and the initial widths of the basis functions are determined, then the weights and the basis function widths are tuned.

The number of basis functions of the SRM and the associated shape instances (key views) are determined by applying the algorithm in (Fig.12). The algorithm, based on an idea from [1] , is a modified generalized Lloyd vector quantization [11], applied on the SRM projection of S_{train} in the *proximal shape space.*

General forms of the update rules:

$$w_i(n+1) = w_i(n) - \eta \cdot \frac{\partial E}{\partial w_i(n)}$$

$$v_{ki}(n+1) = v_{ki}(n) - \eta \cdot \frac{\partial E}{\partial v_{ki}(n)}$$

$$\sigma_i(n+1) = \sigma_i(n) - \eta \cdot \frac{\partial E}{\partial \sigma_i(n)}$$

are reduced in case of (1) **as:**

$$w_i(n+1) = w_i(n) - \eta \cdot (SRM(n) - 1) \cdot y_i(n) - \frac{\eta \cdot w_i(n)}{\omega}$$

$$v_{ki}(n+1) = v_{ki}(n) + \frac{4\eta \cdot (a_k - b_{ki})^2}{\sigma_i^2(n)} \cdot v_{ki}(n) \cdot w_i(n) \cdot (SRM(n) - 1) \cdot h_i(n) \cdot e^{-\frac{h_i(n)^2}{\sigma_i^2(n)}} - \frac{\eta \cdot v_{ki}(n)}{\theta}$$

$$\sigma_i(n+1) = \sigma_i(n) - 2\eta \cdot h_i^2(n) \cdot \sigma_i(n) \cdot (SRM(n) - 1) \cdot w_i(n) \cdot e^{-\frac{h_i^2(n)}{\sigma_i^2(n)}}$$

Fig. 13. ***Gradient descent update rules***

The SRM's weights and basis functions widths, that maximize the degree of fit to an element of S_{train}, hence minimize the error

$$E = \frac{1}{2} \cdot (1 - SRM(view))^2 + \frac{1}{2\omega} \cdot \sum_{i=1}^{H} w_i^2 + \frac{1}{2\theta} \cdot \sum_{i=1}^{H} \sum_{k=1}^{M} v_{ki}^2 \qquad (1)$$

are found by gradient descent based update rules (Fig.13). The 2nd and 3rd terms (weight decay terms) of (1) are always equal, because the SRM behaves like a homogeneous object cooling in time. Hence, at the initial settings ($w_i = 1$, $v_{ik} = 1$, $\forall i, k$) we obtain $\theta = card(S_{train}) \cdot \omega$.

To train an SRM, a static, offline learning algorithm (Fig.19) with no negative examples is used. The static learning algorithm described here is also an offline algorithm, assumed that the training set S_{train} is completely known at the beginning of the learning and no new additions will take place.

After the training the SRM is ready to face the real world. By presenting to the trained SRM, a new image of a shape, the module will react with an output expressing the similarity of this new image with the shape for which the SRM was trained.

The SRM will be able to discriminate the images of shapes which are detected at the input, but it will be unable to use directly and immediately this

```
1.1 call Network Initialization Algorithm (Fig.12) on $S_{train}$, set $Eg_{old} = \infty$
2.1 $\forall view \in S_{train}$ update all $w_i$, $v_{ki}$, $\sigma_i$ using the update rules from (Fig.13)
    with the constraints: $w_i > 0$, $v_{ki} > 0$, $\sigma_{min} < \sigma_i < \sigma_{max}$
2.2 set $Eg = 0$, $\forall view_i \in S_{test}$, compute $E$ given by (1). Set $Eg = \sum_{i=1}^{card(S_{test})} E(view_i)$
2.3 if (step<MaxStep and $Eg > Err + \frac{H}{\omega}$ and $Eg - Eg_{old} < 0$) then $Eg_{old} = Eg$, goto 2.1
```

Fig. 14. ***Static Visual System Training Algorithm***

acquired knowledge to improve its performance. In order to use the knew acquired knowledge a user intervention is necessary. After an important amount of new knowledge is obtained by the system the user has to restart the system, in order to create the new improved SRMs by executing the ***Static Visual System Training Algorithm*** (Fig.19) on the new S_{train} sets. A new SRM (representing a class of shapes), can be created only as a result of a user intervention. The user has to specify the new class or classes of shapes, containing the examples which were classified by the system constructed with the old SRMs, as being images of unknown shapes.

It is obvious, that this type of system is not the best choice, when someone wants to construct an independent and evolutionary system for shape categorization.

2.4 Training SRMs with dynamic learning algorithms

The dynamic learning algorithms modify the number of basis functions of the network, integrating the actions occurring in the initialization and in the refining phases in an incremental learning algorithm. In this case the topology of the network is no longer constant and so the final goal of the trained network can be faster achieved.

The main problem which appears in the process of on-line network structure extension, is to decide when and where a new basis function must be inserted.

Let S_{train} be an off-line acquired training set of feature vectors which describe a shape. Considering that the SRM associated to the shape described by S_{train} has a general topology of the type described in (Fig.10) and (Fig.11). Then for all basis functions $y_i, i \in [1 \dots H]$ of the SRM, the following error measures can be defined:

$$E_i = \sum_{k=1}^{T} (1 - SRM(view_k)) \cdot y_i \qquad (2)$$

$$E_i^{(abs)} = \sum_{k=1}^{T} |1 - SRM(view_k)| \cdot y_i \qquad (3)$$

where:

E_i and $E_i^{(abs)}$ are the contributions to the global error of the network (error over S_{train}) of the input feature vector $view_k$ due to the basis function y_i.

When all the errors E_i associated with the i-th basis function are all positive or all negative the condition $\frac{|E_i|}{E_i^{(abs)}} = 1$ (4) holds. For a different behavior of the error terms E_i inequality $\rho_i = \frac{|E_i|}{E_i^{(abs)}} \ll 1$ (5) holds.

When the relation (5) holds for all the basis functions currently defined on S_{train}, then together with the condition $E_i^{(abs)} \gg 0$ the necessary condition for inserting new basis functions in the network is found. This condition can be expressed by the inequality:

$$E_i^{(abs)} - |E_i| = E_i^{(abs)} (1 - \rho_i) \gg 0 \qquad (6)$$

This expression is not suitable for the online learning algorithms, but can be used by considering not a fixed learning set S_{train}, but a window on the least T learning events, as is done in some incremental learning algorithms. A second technique consists of using an iterative evaluation of E_i and $E_i^{(abs)}$, where the contribution due to events in the past, decays exponentially with time:

$$E_i (t+1) = (1-\alpha) \cdot E_i (t) + \alpha \cdot y_i (t) \cdot (1 - SRM (t)) \qquad (7)$$

$$E_i^{(abs)} (t+1) = (1-\alpha) \cdot E_i^{(abs)} (t) + \alpha \cdot y_i (t) \cdot |(1 - SRM (t))| \qquad (8)$$

In what follows based on the previous concepts two dynamic learning algorithms will be presented. Note that both algorithms consider that, the system was initialized by performing a static learning procedure on an initial training set S_{train}.

Dynamic Competitive Learning (DCL) based training (Fig.15)

DCL algorithm presented in [12] is for the most part, an improvement of the GCS algorithm [13].The algorithm iterates through a cycle in which the network is first activated, feed-forward on a new input vector *view*, then the weights w_i,v_{ki} and the radial basis functions widths σ_i are updated using the gradient descent method. The position of the basis radial functions are updated by the on-line clustering algorithm and the neighborhood topology is updated by a competitive hebbian learning procedure. The algorithm for extending the network size by adding new basis functions is activated, with a longer period in order to grant the time for adjusting the network parameters between two insertion phases.

The insertion procedure considers only a subset of the basis functions, satisfying the criterion (6), basis functions which contribute to the 85% of the global $E^{(abs)}$ of the network. The initialization of the weights and width associated with the new basis function are chosen in order to reduce as much as possible the disturbance caused by the new insertion.

Dynamic Regression Trees (DRT) based training (Fig.16)

This algorithm first presented in [12] was inspired by Breiman's CART algorithm [14] for inducing regression trees. The basic idea consists in recursively splitting the basis function activation area until a granularity sufficient to fit the target function is obtained. In our case, the activation area of a basis function, constructed on the view p_k is in fact the cluster $CL_k = (p_k, A_k)$ where $p_k \in I_c$ and $A_k \subseteq S_t$. The main idea of the method is that, an DRT based algorithm is trying to split the clusters when the gradient descent reaches a local minimum.

Adjacent radial functions strongly interact because of overlapping. Therefore, splitting two adjacent basis functions at the same time could be unnecessary, because a single split could be sufficient to reduce the error on both basis functions.The strategy of DRT is to split simultaneously, only the basis functions

```
Let $cycle = 0$, $EdgeGraph = \{(kv_x, kv_y, age) \mid kv_x, kv_y \in I_c, age \in \aleph\}$, $ind(kv)$ = the
    index of the basis function which is constructed based on $kv$
1. Activate the network on a new input vector $view$ , add it to $S_{train}$ , $cycle++$
2. Update the neighborhood topology
2.1 Determine the two nearest key views of the SRM, $kv_1$ , $kv_2 \in I_c$ for which the SRM's
    response is the closest to $SRM\,(view)$.
2.2 If $(kv_1, kv_2, a) \in EdgeGraph$ then $a = 0$, else $EdgeGraph = EdgeGraph \cup \{(kv_1, kv_2, 0)\}$
2.3 Age the links: $\forall\,(kv_i, kv_x, a_x) \in EdgeGraph$, $i = \{1,2\}$ , do $a_x++$
2.4 Eliminate the old links: $\forall\,(kv_x, kv_y, a_x) \in EdgeGraph$, if $a_x > MAXAGE$ then
    $EdgeGraph = EdgeGraph \setminus \{(kv_x, kv_y, a_x)\}$
3. Update the positions of the basis functions using, the Generalized Lloyd
    Iteration (K-medoid) algorithm on the outputs of the SRM
3.1 let $O_c$ and $OSRM$ two empty sets
3.2 $\forall view \in S_{train}$ if $view \in I_c$ then add $SRM(view)$ to $O_c$ else add $SRM(view)$ to
    $OSRM$.
3.3 $\forall p_k \in O_c$ define $CL_k = (p_k, A_k)$ a cluster centered in $p_k$, where $A_k$ the set of all the
    elements of the cluster except $p_k$, initially empty
3.4 $\forall q \in OSRM$ if $CL_x$ is the cluster who's centroid is the closest to $q$ then add $q$ to
    $CL_x$
3.5 $\forall CL_k$ clusters, find the new centroid $p_k^{new}$ as the k-medoid of $CL_x$
3.6 define the improved $O_c = \{p_k^{new} \mid k \in [1 \ldots H]\}$ and define the improved $I_c =$
    $SRM^{-1}\,(O_c)$ .
4. Update the SRM weights and widths
4.1 $\forall view \in S_{train}$ compute the error E given by (1), update all $w_i$, $v_{ki}$, $\sigma_i$ using
    the update rules from (Fig.13)
4.2 If $currentstep < MAXSTEPS$ and $E > ERRORTHRESHOLD$ and
    $abs\,(E\,(currentstep) - E\,(currentstep - 1)) > MINVARIATION$ goto 4.1
5.1 $\forall kv \in I_c$, update the values of $E_{ind(kv)}$ and $E^{(abs)}_{ind(kv)}$ using the update rules (7), (8)
5.2 If $cycle = MAXCYCLES$ then goto 6. else goto 1.
6.1 Let $E_c = 0$, $I_c^E = \left\{kv \in I_c \mid E^{(abs)}_{ind(kv)} - \left|E_{ind(kv)}\right| = E^{(abs)}_{ind(kv)}\left(1 - \rho_{ind(kv)}\right) \gg 0\right\}$ ,
    where the condition on which the set is constructed is defined by equation (6)
6.2 Let $kv_1 \in I_c^E$ with the property $E^{(abs)}_{ind(kv_1)}\left(1 - \rho_{ind(kv_1)}\right) =$
    $max\left\{E^{(abs)}_{ind(kv)}\left(1 - \rho_{ind(kv)}\right) \mid kv \in I_c^E\right\}$ and $kv_2 \in I_c^E$ with the property
$E^{(abs)}_{ind(kv_2)}\left(1 - \rho_{ind(kv_2)}\right) = max\left\{E^{(abs)}_{ind(kv)}\left(1 - \rho_{ind(kv)}\right) \mid kv \in I_c, (kv_1, kv) \in EdgeGraph\right\}$
6.3 Select $view_x \in S_{train}$ situated between $kv_1$and $kv_2$ and add it to $I_c$.
6.4 Initialize the weights and widths associated with the new $kv = view_x$ as $w_{ind(view_x)} =$
    $\frac{1}{2} \cdot \left(w_{ind(kv_1)} + w_{ind(kv_2)}\right)$, $v_{k,ind(view_x)} = \frac{1}{2} \cdot \left(v_{k,ind(kv_1)} + v_{k,ind(kv_2)}\right)$ , $k \in [1 \ldots M]$
6.5 $\forall v \in \{view_x, kv_1, kv_2\}$ compute $\sigma_{ind(v)} = \frac{\sum_{i=1}^{SizeOf(Ng(v))} \|v - keyview_i\|_2^2}{SizeOf(Ng(v))}$ where:
    $keyview_i \in Ng\,(v)$, $Ng\,(v) = \{kv \mid (v, kv, a) \in EdgeGraph\}$ neighborhood of $v$
6.6 Let $E_c = E_c + E^{(abs)}_{ind(kv_1)}$, $I_c^E = I_c^E \setminus \{kv_1\}$
6.7 If $E_C > 0.85 \sum_{j=1}^{H} E_i^{(abs)}$ and $I_c^E \neq \emptyset$ goto 6.2
7. go to 1.
```

Fig. 15. Dynamic Competitive Learning (DCL) based training

```
Let $E_{th}$ =error split threshold, $WS_t$ =the training set window. The last N
    elements perceived by the system.
Let $E_g = 0$ and $t = 0$ and $WS_t = S_t$
1. Activate the network on a new input vector $view$ and add it to $S_{train}$
2.1 $\forall view \in S_{train}$ compute the error E given by (1), update all $w_i$, $v_{ki}$,
    $\sigma_i$ using the update rules from (Fig.13)
2.2 If $currentstep < MAXSTEPS$ and $E > ERRORTHRESHOLD$ and
    $abs\left(E\left(currentstep\right) - E\left(currentstep - 1\right)\right) > MINVARIATION$ goto
    2.1
3.1 Compute the current global error of the system $E = \frac{1}{2} \cdot \sum_{i=1}^{T}\left(1 - SRM\left(view_i\right)\right)^2$, where $view_i \in S_{train}$
3.2 Let $E_g\left(t+1\right) = E_g\left(t\right)\beta + \left(1-\beta\right)\left|E\right|$ and $t = t + 1$
3.3 If $E_g > E_{th}$ then goto 4.1 else goto 1.
4.1 let $O_c$ and $OSRM$ two empty sets
4.2 $\forall view \in S_{train}$ if $view \in I_c$ then add $SRM(view)$ to $O_c$ else add
    $SRM(view)$ to $OSRM$.
4.3 $\forall p_k \in O_c$ define $CL_k = \left(p_k, A_k\right)$ a cluster centered in $p_k$, where $A_k$ the
    set of all the elements of the cluster except $p_k$, initially empty
4.4 $\forall q \in OSRM$ if $CL_x$ is the cluster who's centroid is the closest to $q$ then
    add $q$ to $CL_x$
4.5 Select the set $I_c^R \subseteq I_c$ corresponding to the basis functions candidate
    for split. The set is constructed such as $\forall kv_x, kv_y \in I_c^R$ the condition
    $S\left(y_{ind(kv_x)}, y_{ind(kv_x)}\right) = 0$ holds.
5. Split all the clusters corresponding to the selected basis functions from $I_c^R$
5.1 $\forall CL_k$,$CL_k = \left(p_k, A_k\right), p_k \in I_c^R$, sort $A_k$ in increasing order using the
    measure $\left\|q_l - p_k\right\|_2^2, q_l \in A_k$
5.2 Let $\left(q_x, q_{x+1}\right)$ the pair of consecutive elements of sorted $A_k$ for which
    $\left\|q_x - q_{x+1}\right\|_2^2$ has the smallest value, and select $CL_{q_x} = \left(q_x, B_x\right)$ and
    $CL_{q_{x+1}} = \left(q_{x+1}, B_{x+1}\right)$ where $B_x = B_{x+1} = \emptyset$, $\forall q \in A_k$
5.3 If $\left\|q - q_x\right\|_2^2 < \left\|q - q_{x+1}\right\|_2^2$ then $CL_{q_x} = \left(q_x, B_x \cup \{q\}\right)$ else $CL_{q_{x+1}} =$
    $\left(q_{x+1}, B_x \cup \{q\}\right)$
5.4 Let the new output codebook $O_c = \left(O_c \setminus \{p_k\}\right) \cup \{q_x, q_{x+1}\}$
5.5 Initialize the weights and widths of the new $keyview$'s as
    $w_{ind(SRM^{-1}(q_x))} = w_{ind\left(SRM^{-1}\left(q_{x+1}\right)\right)} = \frac{1}{2} w_{ind(SRM^{-1}(p_k))}$,
    $v_{k,ind(SRM^{-1}(q_x))} = v_{k,ind\left(SRM^{-1}\left(q_{x+1}\right)\right)} = \frac{1}{2} v_{k,ind(SRM^{-1}(p_k))}, k \in$
    $[1 \ldots M]$, $\sigma_{ind(SRM^{-1}(q_x))} = \sigma_{ind\left(SRM^{-1}\left(q_{x+1}\right)\right)} = \frac{1}{2}\sigma_{ind(SRM^{-1}(p_k))}$
4.5 Construct the new input codebook as $I_c = SRM^{-1}\left(O_c\right)$
goto 1.
```

Fig. 16. ***Dynamic Regression Trees (DTR) algorithm***

which do not interact among themselves, so only the basis functions for which the superposition (the scalar product of the basis functions) $S(y_i, y_j) = 0$.

The superposition measure can be reduced as:

$$
\begin{aligned}
S(y_i, y_j) &= \int_{\Re^M} y_i y_j dview \\
&= \int_{\Re^M} \left(e^{-\frac{h_i^2}{\sigma_i^2}} \cdot e^{-\frac{h_j^2}{\sigma_j^2}} \right) dview \\
&= \int_{\Re^M} e^{\left(\frac{h_i h_j}{\sigma_i \sigma_j}\right)^2} dview \\
&= \int_{\Re^M} e^{\left(\frac{\left(\Sigma_{k=1}^{M} (v_{ki} a_k - v_{ki} b_{ki})^2\right) \left(\Sigma_{k=1}^{M} \left(v_{kj} a_k - v_{kj} b_{kj}\right)^2\right)}{\sigma_i \sigma_j} \right)^2} d\{a_k\}_{k=1}^{M} \qquad (9)
\end{aligned}
$$

2.5 Categorization classifier

The second stage of a SCM, performing the actual categorization, is implemented as the chorus of all the known SRMs defined in a context. In a context with S known SRMs, the reaction of the SCM defined on that context, when presented at the input with the view x of a shape is the vector:

$$SCM(x) = [Tr_1 \cdot (1 - |1 - SRM_1(x)|), \cdots, Tr_S \cdot (1 - |1 - SRM_S(x)|)] \quad (10)$$

The Tr_S factor from the expression (10) is a trust level assigned to a SRM, directly proportional with the compactness level (expressed by H) of the *proximal shape space* manifold of the modeled shape. In general this should depend also on the context. If the averange minimum number of submanifolds needed to represent the manifold corresponding to an SRM, in a given context, is $kvno_{min}$ than the trust level assign to each SRM can be given by the expression $Tr_S \frac{\log(kvno_{min})}{\log(H_S)}$.

3 Architecture of the C-SCM Visual System

If the categorization problem is posed in a context associated with a limited region of the *distal shape space,* the classifier can be further improved, by eliminating the influence of the SRMs corresponding to shapes situated outside the context (eg. Fig.17). To do this, we define the *virtual distal shape space* of the context, as a subspace of the *proximal shape space,* defined by the SRMs of the context. A shape manifold in the *distal shape space* is mapped to a manifold in the *virtual distal shape space* of the context, by projecting each of its points to a point in the *virtual distal shape space.* The coordinates of the projected point are given by the outputs of the SRMs of the context, when presented at the input with the corresponding shape instance. For each of the manifolds from the *virtual distal shape space* a new SRM is defined and trained. Based on them a new, context limited SCM, is defined.

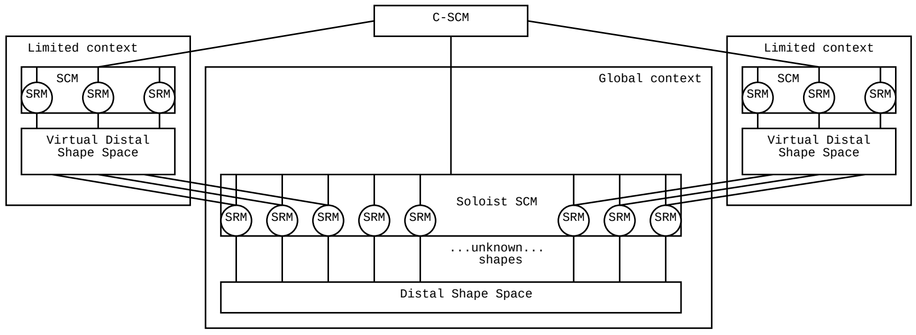


Fig. 17. A C-SCM Visual System Topology with 2 sub-contexts

C-SCM Visual System's global classifier is defined as the chorus of the SCMs associated with all the known contexts, having as a *soloist* the SCM of the *global context* (the context associated with the *distal shape space*), eg.(Fig.17). If besides the global context, the C-SCM Visual System defines K sub-contexts, than a simple form of the global classifier used to classify a shape view x can be:

$$C - SCM\left(x\right) = \left\{SCM\left(x\right), SCM_1\left(x\right), \cdots, SCM_K\left(x\right)\right\} \quad (11)$$

When the C-SCM Visual System is presented at the input with a set of T views of the same object X ($x_i \in ViewOf(X)$) our simple shape classifier is:

$$C{-}SCM\left(X\right) = \left\{\frac{1}{T} \cdot \sum_{i=1}^{T} SCM\left(x_i\right), \frac{1}{T} \cdot \sum_{i=1}^{T} SCM_1\left(x_i\right), \cdots, \frac{1}{T} \cdot \sum_{i=1}^{T} SCM_K\left(x_i\right)\right\} \quad (12)$$

3.1 Simple C-SCM Visual System

The most simple architecture of a C-SCM Visual System can be defined as the system associated with the the *global context* and the context of all the objects known by the system (Fig.18), having a classifier of the form (13).

$$C - SCM\left(x\right) = \left\{SCM_{gc}\left(x\right), SCM_{fkgc}\left(x\right)\right\} \quad (13)$$

where: $SCM_{gc} = SCM$ of the global context, $SCM_{fkgc} = SCM$ of the finite system known global context and x is a object view.

When the Simple C-SCM Visual System is presented at the input with a set of T views of the same object X ($x_i \in ViewOf(X)$) the response of the shape classifier looks like:

$$C - SCM\left(X\right) = \left\{ \frac{1}{T} \cdot \sum_{i=1}^{T} SCM_{gc}\left(x_i\right), \frac{1}{T} \cdot \sum_{i=1}^{T} SCM_{fkgc}\left(x_i\right) \right\} \quad (14)$$

This minimal architecture gives just a minimal amount of information, as a reaction to the external stimuli received by such a C-SCM Visual System. The quality of the response of the C-SCM Visual System is direct proportional with the number of different and specialized contexts known by the system, so the need to have more contexts appear.

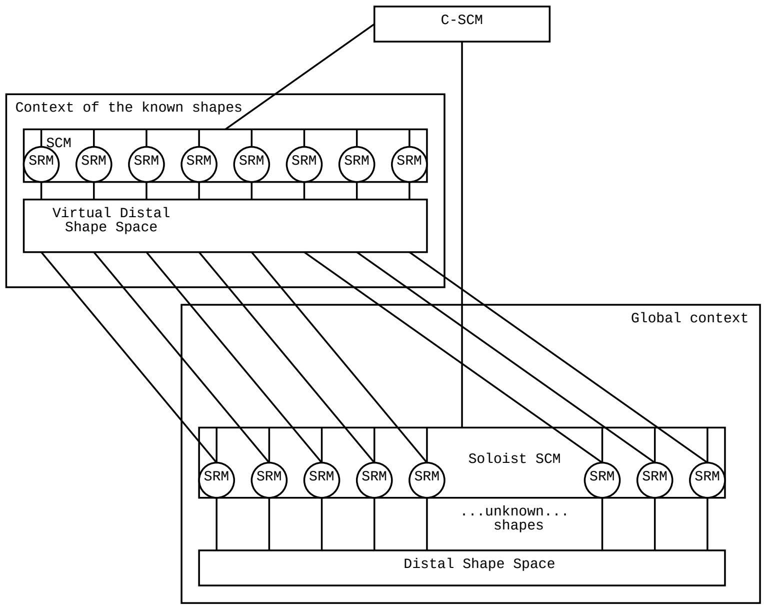


Fig. 18. Simple C-SCM Visual System

3.2 C-SCM Visual System based on shape category contexts

The next step in developing a more complex system is to create a method through which the system defines when necessary new contexts. The easiest way is to create contexts based on outside intervention of a human user. In this case the outside user asks the system to give a more detailed answer by providing to it informations necessary to delimit a context in which the outside user is interested. To have an autonomous C-SCM Visual System, the process of defining new contexts, should be an internal process of the system, with no outside intervention.

```
1. $\forall i \in [1 \ldots S]$ where $S$ is the number of known objects
begin
   1.1 compute the vector $SCM_{fkgc}(i) = \frac{1}{A} \cdot \sum_{a=1}^{A} SCM(i_a)$ where
   $A$ =number of views of object $i$
   1.2 construct the set of pairs
$PSCM_i = \{(j, SCM_{fkgc}(i)[j]) \mid SCM_{fkgc}(i)[j] > similarityth, j \in [1 \ldots S]\}$
end
2. Define the set of categories $SCAT = \emptyset$ and $CAT_{sing} = \emptyset$ the "category" of
   singular objects
3. Initialize $SCAT$
   3.1 create $CAT_1 = \{1\}$
   3.2 $\forall (j, x) \in PSCM_1$
     begin
       If $\exists (1, y) \in PSCM_j$ then $CAT_1 = CAT_1 \cup \{j\}$
     end
   3.3 $SCAT = SCAT \cup \{CAT_1\}$
4. $\forall i \in [2 \ldots S]$
begin
   4.1 If $\exists k$ such that $i \in CAT_k$ , $CAT_k \in SCAT$
     then
       4.1.1 $\forall (j, x) \in PSCM_i$
         begin
           If $\exists (i, y) \in PSCM_j$ then $CAT_k = CAT_k \cup \{j\}$
         end
     else
       4.1.2 create $CAT_i = \{i\}$
       4.1.3 $\forall (j, x) \in PSCM_i$
         begin
           If $\exists (i, y) \in PSCM_j$ then $CAT_i = CAT_i \cup \{j\}$
         end
       4.1.4 $SCAT = SCAT \cup \{CAT_i\}$
end
5. $\forall CAT_k \in SCAT$ If $SizeOf(CAT_k) = 1$ then $SCAT = SCAT \setminus \{CAT_k\}$,
   $CAT_{sing} = CAT_{sing} \bigcup \{k\}$
```

Fig. 19. ***Shape Categories Detection Algorithm***

```
1. $\forall k \in CAT_{sing}$ create $TempCAT_k = \{k\}$
2. Given $X$ if $\exists k$ for which $catdist(X, TempCAT_k) = \frac{1}{T} \cdot \sum_{i=1}^{T} SCM_{TempCAT_k}(x_i) = \frac{1}{T} \cdot \sum_{i=1}^{T} [Tr_k \cdot (1 - |1 - SRM_k(x_i)|)] > similarityth$ and $catdist(X, TempCAT_k)$ is the maximum value for all $TempCAT_a, a \in CAT_{sing}$ then
    2.1 $TempCAT_k = TempCAT_k \bigcup \{w\}$
    2.2. If the previously defined $catdist(X, TempCAT_k) > identityth$ then add the views of the new observed object, to the set of views ($S_{train}$) of the object corresponding to $SRM_k$, else define $SRM_w$ corresponding to the new observed object.
    2.3. $SCAT = SCAT \bigcup \{TempCAT_k\}$ and $CAT_{sing} = CAT_{sing} \setminus \{k\}$
```

Fig. 20. ***New Category Detection Algorithm***

First a Simple C-SCM Visual System is constructed having the architecture described in (Fig.18) an a classifier like in (14). Then using the algorithm from (Fig.19) the known objects are classified in categories. The generality of the categories is controlled by a similarity threshold (sets the limit under which the similarity between 2 objects starts to be vague). As a result a number of categories containing at least 2 objects are defined, and the rest of the objects which were not categorized form the set of singular objects.

The last step consist in defining a context for each of the determined shape categories and for the set of singular objects, and create the more complex C-SCM Visual System based on the *global context,* the context of all the objects known by the system and the newly created contexts.

$$C - SCM(x) = \{SCM_{gc}(x), SCM_{fkgc}(x)\} \bigcup \left\{ \underbrace{SCM_{CAT_y}(x) \cdots, SCM_{CAT_z}(x)}_{SizeOf(SCAT)}, SCM_{sing}(x) \right\} \quad (15)$$

where: $SCM_{gc} = SCM$ of the global context, $SCM_{fkgc} = SCM$ of the finite system known global context, $\underbrace{SCAT_{CAT_y}(x) \cdots, SCM_{CAT_z}(x)}_{SizeOf(SCAT)} = SCMs$ of the contexts defined on the elements of $SCAT$, $SCM_{sing} = SCM$ of the set of singular objects and x is a view of an object.

When the C-SCM Visual System is presented at the input with a set of T views of the same object X ($x_i \in ViewOf(X)$) the response of the shape classifier looks like:

$$C - SCM\left(X\right) = \left\{\frac{1}{T}\cdot\sum_{i=1}^{T} SCM_{gc}\left(x_i\right), \frac{1}{T}\cdot\sum_{i=1}^{T} SCM_{fkgc}\left(x_i\right)\right\} \bigcup \left\{\underbrace{\frac{1}{T}\cdot\sum_{i=1}^{T} SCM_{CAT_y}\left(x_i\right)\cdots, \frac{1}{T}\cdot\sum_{i=1}^{T} SCM_{CAT_z}\left(x_i\right)}_{SizeOf(SCAT)}\right\} \bigcup \left\{\frac{1}{T}\cdot\sum_{i=1}^{T} SCM_{sing}\left(x_i\right)\right\} \quad (16)$$

Based on the classifier (16), the C-SCM Visual System will have one of the next behaviors when presented at the input with the new observed object X represented by T views having the ID w:

(1) The new observed object is classified as part of a known object category CAT_z if $catdist(X, CAT_z) > similarityth$ and $catdist(X, CAT_z)$ is the maximum value $\forall CAT \in SCAT$.

$$catdist(X, CAT_z) = \frac{1}{T}\cdot\sum_{i=1}^{T}\left[\frac{1}{SizeOf\left(CAT_z\right)}\cdot\sum_{j=1}^{SizeOf(CAT_z)} SCM_{CAT_z}\left(x_i\right)[j]\right] \quad (17)$$

If $\exists j$ such that $\frac{1}{T}\cdot\sum_{i=1}^{T} SCM_{CAT_z}\left(x_i\right)[j] > identityth$ then add the views of the new observed object, to the set of views (S_{train}) of the object corresponding to the $j - th$ term of the SCM_{CAT_z}vector. If no j was found to satisfy the previous condition then create in the CAT_z context a new SRM corresponding to the new observed object.

(2) If no category from $SCAT$ was found to satisfy the condition $catdist(X, CAT_z) > similarityth$ given in (17) then, try to create a new category which contains the new observed object and an object from CAT_{sing}, using an algorithm like in (Fig.20).

(3) If no category from $SCAT$ was found to satisfy the condition $catdist(X, CAT_z) > similarityth$ given in (17) and no new category can be created then $CAT_{sing} = CAT_{sing} \bigcup \{w\}$

4 Results

4.1 Test of the behavior of the Simple C-SCM Visual System on the COIL-100 database

The Simple C-SCM Visual System was tested on the COIL-100 [9] library (Fig. 27), a database of 100 shapes, each described by a view sphere of 72 images. The values of the input features (108 features per image) are obtained by splitting

Fig. 21. Reaction of the global context SCM to shape 23

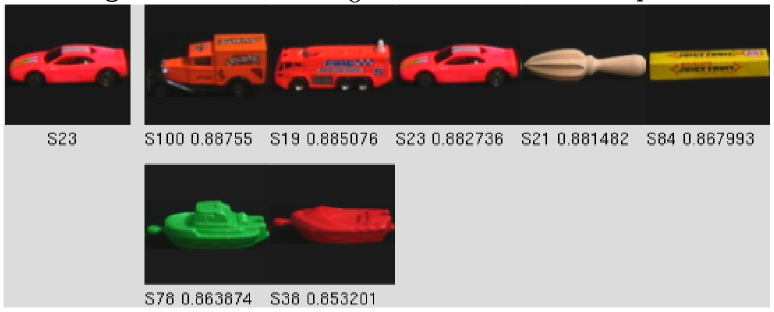


Fig. 22. Reaction of the COIL-100 context SCM to shape 23

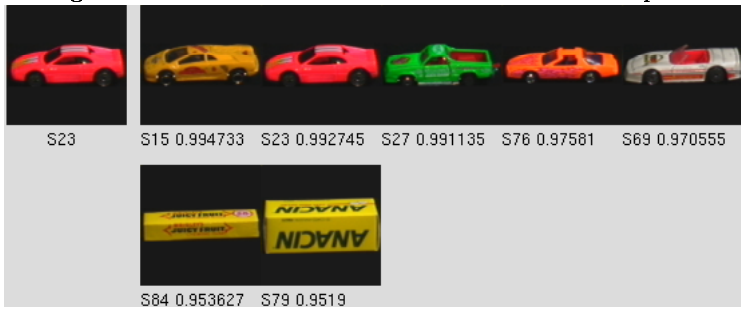


Fig. 23. Reaction of the global context SCM to shape 57

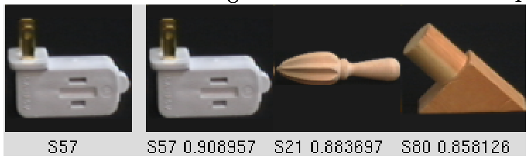


Fig. 24. Reaction of the COIL-100 context SCM to shape 57

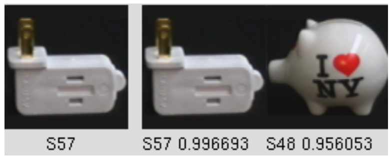


Fig. 25. Reaction of the global context SCM to shape 41

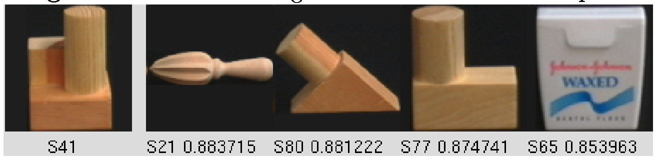
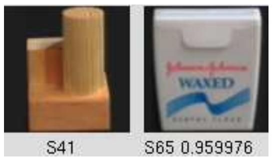


Fig. 26. Reaction of the COIL-100 context SCM to shape 41



Legend: In each figure the first picture represent the input view sphere, and the next pictures represent the SRMs of the SCM response vector in a decreasing order of their response to the input shape.

Fig. 27. COIL-100 Shape Database

the input image (128x128 pixel) in 36 superposing patches (23x23 pixels), and computing the mean gradient value, on the hue, saturation and value components of each patch. The input features were chosen to be as simple as possible, avoiding user biased prior knowledge. The number of the input features is also limited to 108 input features, to avoid as much as possible the *curse of dimensionality.* For each shape from COIL-100 a SRM is constructed, using 48 images of the view-sphere as training data and the rest of 24 images as test data. Then the soloist SCM of the Visual System, corresponding to the global context, and the SCM corresponding to the COIL-100 context are constructed. The classifier of the C-SCM Visual System is defined as the system (3) with the constructed SCMs.

The SCM response vector associated to an input shape in the global context, is constructed by the SRMs of the global context, which give a bigger than 0.85 average answer to the elements of the input shape view sphere. In the case of the COIL-100 limited context the threshold of the average answer is considered to be 0.95. The following behaviors of the C-SCM Visual System were observed:

(1)Good categorization in the global context (Fig.21) and very good categorization in the COIL-100 context (Fig.22), when the input view sphere form a representative model of the input shape and the input shape is similar to many shapes from the given context. For instance, the response vector of the global context SCM, to the view sphere of the shape **S23** (Fig.21), already gives some hints about the shapes similar to **S23**. Because the scores of the SRMs are only around 0.88 the categorization is still not 100% trustworthy. The response vector of the COIL-100 context SCM, to the view sphere of the shape **S23** (Fig.22), accurately finds the shapes similar to **S23**. By combining the responses of the COIL-100 context SCM, to the view sphere of the shapes **S23**, **S15**, **S27**, **S76**, **S69** the *small car* shape category can be easily identified.

(2)Good recognition but no categorization in both the global and COIL-100 context (Fig.23 and Fig.24), when the input view sphere form a representative model of the input shape, but the shape is singular in the given context. The response vector of the global context SCM, to the view sphere of the shape **S57** (Fig.23), already give a hint to the recognition the the shape **S57**, and to the fact that **S57** is a singular shape in the knowledge base of the system. The response vector of the COIL-100 context SCM, to the view sphere of the shape **S57** (Fig.24), transforms the hints derived in the global context in certain facts.

(3)Bad recognition and vague categorization in both the global and COIL-100 context (Fig.25 and Fig.26), when the input view sphere is not representative for the input shape and the shape is vague similar to other shapes from the given context. The response vectors of the global context and of the COIL-100 context SCMs, to the view sphere of the shape **S41** (Fig.25 and Fig.26), don't give any hints, about what shapes are similar to **S23**. This is because the scores are only just above the thresholds defined in each context.

4.2 Test of the C-SCM Visual System based on shape category contexts on the COIL-100

The same experimental settings from **4.1** are used but the classifier of the system is defined as in (16). The first 80 shapes are used to initialize the system and the last 20 shapes as testing set for the categorization of shapes.

After the system is initialized the COIL-80 (COIL-100 minus the test shapes) database is split in the categories. Depending on the choise of the *similarityth* the system finds less or more categories containing less or more objects. Actually the choise of the threshold, even in the biological systems is very subjective depending on the context and on the experience of the observer. In this experiment the choosen values were *similarityth* = 0.65 and *identityth* = 0.9 and the obtained categories are given in (Fig. 28), (Fig. 29), (Fig. 30), (Fig. 31), (Fig. 32), (Fig. 33), (Fig. 34), (Fig. 35) . It can be seen that the categories are not mutually exclusive (some objects are members of more of one category). This is maybe because the *similarityth* was set up to a lower than optimal value, but is also natural for an object to have features that place it in more than one category.

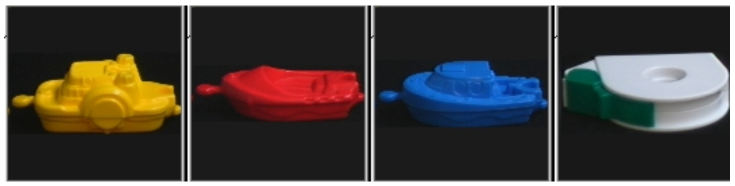


Fig. 28. Category A

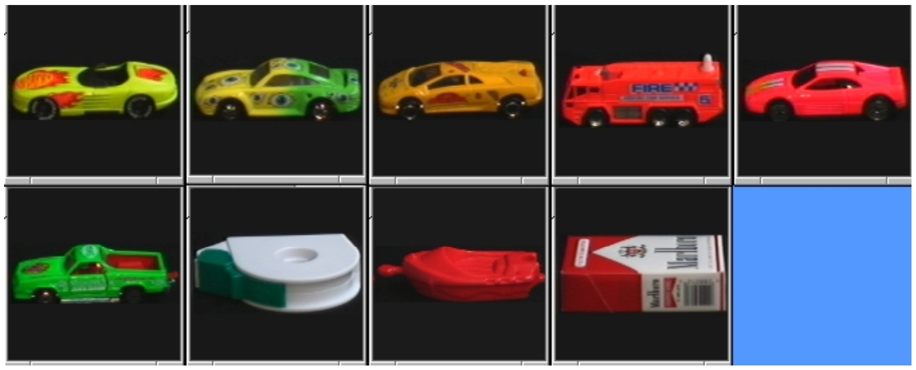


Fig. 29. Category B

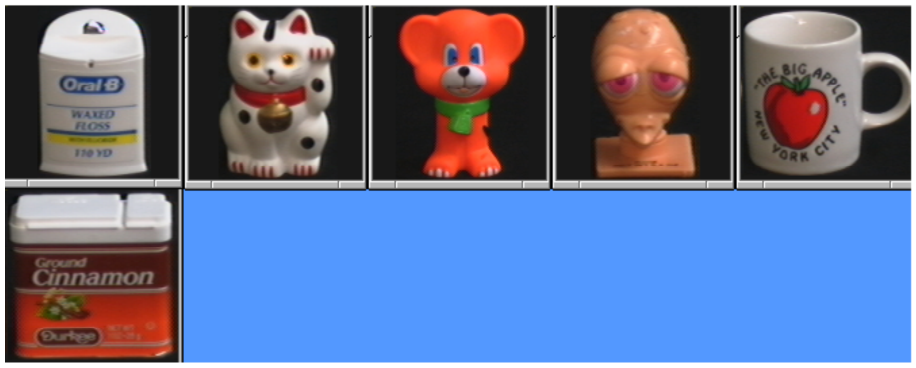


Fig. 30. Category C

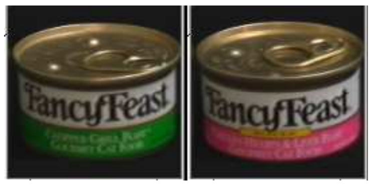


Fig. 31. Category D

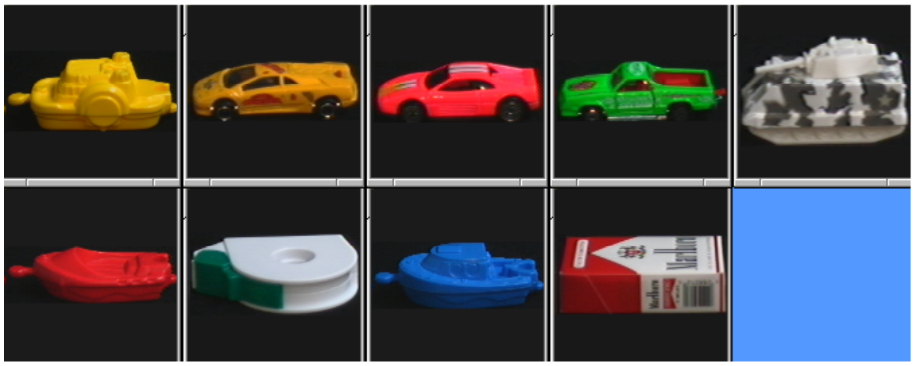


Fig. 32. Category E

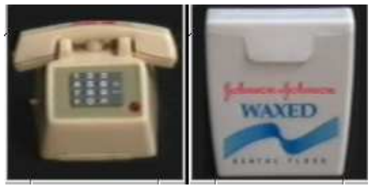


Fig. 33. Category F

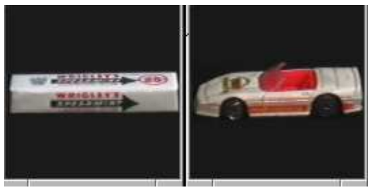


Fig. 34. Category G

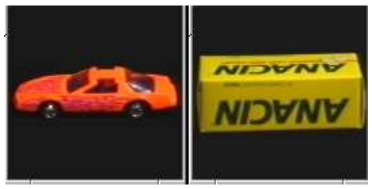


Fig. 35. Category H

5 Conclusion

The C-SCM Visual System inspired by the biological visual system, and by *"Representation and Recognition in Vision"* of Shimon Edelman proves to be a reliable system even in this early stage of implementation. Its power is given by the ability to define on its own, using as input only basic raw features, optimal complex features, and based on them to create abstract models of shapes. The system is also able to deal with the concept of "context", taking into account the fact that the process of defining categories is highly dependent on the environment and on the expectations of the observer. As a result a categorization decisions is always made in a specific context and is based on a SCM (shape categorization module), consisting of the chorus of SRMs (shape recognition modules associated with the model of a object defined in the given context) of the shapes defined in that context. Finally, a classifier of the C-SCM Visual System can be constructed based on the chorus of SCMs of all the system's known contexts.

References

1. Shimon Edelman, *Representation and Recognition in Vision*, The MIT Press, 1999
2. Biederman I., *Recognition by components: A theory human image understanding*, Psychological Review 94:115-147, 1987
3. Ullman S., *Aligning pictorial descriptions:An approach to object recognition*, Cognition 32:193-254, 1989
4. Duda R. O., Hart P. E., *Pattern classification and scene analysyss*, NewYork: Wiley, 1973
5. David H. Hubel, *Eye,Brain and Vision*, Scientific American Library, 1988
6. Watanabe S., *Pattern Recognition: Human and mechanical*, New York:Wiley, 1985
7. Tversky A., *Features of similarity*, Psychological Review 84:327-352, 1977
8. Mumford D., *Mathematical theories of shape: Do they model perception?*, Geometric methods in computer vision. Volume 1570. Bellingham, WA: SPIE. 2-10, 1991
9. www.cs.columbia.edu/CAVE/research/softlib/coil-100.html
10. Girosi F., Jones M., Pogio T., *Regularization theory and neural networks architecture.* Neural Computation 7:219-269, 1995
11. Linde Y., Buzo A., Gray R., *An algorithm for vector quantizer design.* , IEEE Transactions and Communications COM-28:84-95 , 1980
12. Blanzieri, E. and Katenkamp, P. Learning radial basis function networks on-line, In 13-th International Conference on Machine Learning, pages 37-45, Bari, 1996
13. Fritzke, B. Growing cell structure - a self-organizing network for unsupervised and supervised learning. Neural Networks, 7(9):1441-1460, 1994
14. Breiman, L., Friedman, J., Ohlsen, R., and Stone, C.,Classification And Regression Trees, Wadsworth and Brooks, Pacific Grove, CA., 1984