

# Visual Classification by a Hierarchy of Extended Fragments

Shimon Ullman and Boris Epshtein

Weizmann Institute of Science,  
67100, Rehovot, Israel  
{shimon.ullman,boris.epshtein}@weizmann.ac.il

**Abstract.** The chapter describes visual classification by a hierarchy of semantic fragments. In fragment-based classification, objects within a class are represented by common sub-structures selected during training. The chapter describes two extensions to the basic fragment-based scheme. The first extension is the extraction and use of feature hierarchies. We describe a method that automatically constructs complete feature hierarchies from image examples, and show that features constructed hierarchically are significantly more informative and better for classification compared with similar non-hierarchical features. The second extension is the use of so-called semantic fragments to represent object parts. The goal of a semantic fragment is to represent the different possible appearances of a given object part. The visual appearance of such object parts can differ substantially, and therefore traditional image similarity-based methods are inappropriate for the task. We show how the method can automatically learn the part structure of a new domain, identify the main parts, and how their appearance changes across objects in the class. We discuss the implications of these extensions to object classification and recognition.

## Introduction

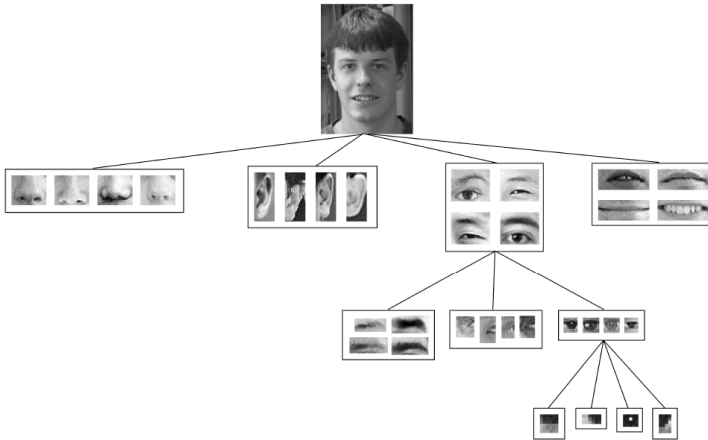
Object classification involves two main stages: feature extraction, and then using these features to classify a novel image. Many different features have been proposed in the past, ranging from simple local ones such as Wavelets or Gabor filters [23], to complex features such as geons [3],[13] which are view-invariant 3-D primitive shapes. Most of the features used in the past, from the simple to the more complex, were usually generic in the sense that the same limited set of features was used for all objects and object classes.

In several recent classification schemes, objects are represented as a combination of informative image parts [1],[6],[9],[21]. This approach was shown to be effective for various classification problems. Unlike previous schemes, these features are class-specific: different features are extracted automatically for different classification tasks from the training data. The present work extends this approach in two directions: the use of hierarchical features, and the representation of object parts by equivalence classes of features, called ‘semantic features’.

The idea of representing objects in a class by their informative parts can be extended recursively: the informative object parts can themselves be represented as an arrangement of informative sub-parts, the sub-parts can then be split into smaller parts and so on. This general scheme raises a number of questions, related to the creation of such a hierarchy: a method for selecting the best parts and sub-parts, a stopping rule for decomposing the features, and the optimal selection of parameters such as the size of search region for each part. There are also questions related to the use of the hierarchy: how to perform classification using this structure, and how to best detect sub-parts of the object using their context. We discuss these questions in Part 2 of the chapter.

Each object part and sub-part (say, an eye) can be represented not just by a single representative image fragment, but by a collection of semantically equivalent fragments, representing different appearances of the part, such as an open eye, closed eye, or eyes of different shapes. Questions related to this issue include: how to extract such sets of semantically equivalent fragments, and how to use them for classification. These issues are discussed in Part 3.

The two components discussed above, hierarchical representation and semantic features, can be used independently, but can also be used naturally in a combined manner. Taken together, they give rise to the following feature organization: an object or a class are represented by a hierarchy of parts and sub-parts. This hierarchy can be represented as a tree, with semantic fragments at each node, as is illustrated schematically in Figure 1. In the remaining of this chapter we will discuss how this hierarchy of semantic fragments is constructed and used. The chapter is divided into three parts. The first briefly summarizes the extraction of informative features, the second describes the construction of



**Fig. 1.** Representing a class by a hierarchy of semantic fragments. A face is represented as an arrangement of parts such as nose, eyes, ear and mouth. Each of these parts is represented as a semantic equivalence set. The parts are represented in turn in terms of their sub-parts. For simplicity, only the sub-parts of the eye part are shown.

feature hierarchies, and the third describes the extraction and use of semantically equivalent parts. We conclude with a discussion of using the feature hierarchies and semantic equivalence sets together.

## 1 Informative Classification Fragments

In this section we describe the algorithm for extracting informative images fragments and learning their associated parameters, such as the detection threshold for each fragment. This family of features proved to be highly effective for classification. An empirical comparisons with other classification features can be found in [22].

Fragments are selected from the training data using the the procedure in [21]. The process proceeds by identifying fragments that deliver the maximal amount of information about the class. A large number (tens of thousands) of candidate fragments are extracted from the training images at multiple locations and sizes. For each fragment, the optimal detection threshold is computed as explained below. This detection threshold indicates the minimal visual similarity that a fragment must have within an image, to be detected. Normalized cross-correlation was used in the past as a similarity measure, but other similarity measures, such as SIFT [12], can also be used. A binary variable can then be associated with each fragment depending on its presence in the image  $I$ :

$$f_i(I, \theta_i) = \begin{cases} 1, & \text{if } S(I, f_i) > \theta_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$S(I, f_i)$  is the maximal visual similarity between fragment  $f_i$  and image  $I$ ,  $\theta_i$  is the threshold associated with  $f_i$ . The class variable variable  $C(I)$  is defined as 1 if the image belongs to the class being detected, and 0 otherwise. We can then derive the mutual information between the two binary variables:

$$MI(f_i(\theta_i); C) = \sum_{f_i, C} p(f_i, C) \log \frac{p(f_i, C)}{p(f_i)p(C)} \quad (2)$$

The mutual information in this expression depends on the detection threshold  $\theta_i$ . If the threshold is too low, the information delivered by the fragment about the class will be low, because the fragment will be detected with high frequency in both the class and non-class images. A high threshold will also yield low mutual information, since the fragment will be seldom detected in both the class and non-class images. At some intermediate value of threshold, the mutual information reaches a maximum. The value  $\theta_i$  of threshold yielding maximal information for the fragment  $f_i$  is therefore associated with the fragment. The most informative fragments are selected successively, using the following max-min procedure. After finding the first fragment with the highest mutual information score, the search identified the next fragment that delivered the maximal amount of additional information. At iteration  $i$  the fragment  $f_i$  is selected to increase the

mutual information of the fragment set by maximizing the minimal addition in mutual information with respect to each of the first  $i-1$  fragments.

$$f_i = \arg \max_{f_k \in K_i} \min_{f_j \in S_i} (MI(f_k, f_j; C) - MI(f_j; C)) \quad (3)$$

$K_i$  is the set of candidate fragments,  $S_i$  is the set of already selected fragments at iteration  $i$ ,  $f_i$  is the new fragment to be selected at iteration  $i$ . The update rule for the fragment sets is:

$$\begin{aligned} K_{i+1} &= K_i \setminus \{f_i\} \\ S_{i+1} &= S_i \cup \{f_i\} \end{aligned} \quad (4)$$

The initial  $K_0$  is the set of all candidate fragments;  $S_0$  is the set containing a single fragment with the highest mutual information with the class. The iterations end when adding new fragment to the set  $S$  makes only a small increment to the mutual information, less than some small threshold  $\varepsilon$ . Once the set of informative fragments is determined, the optimal size of the region of interest (ROI) for each selected fragment is computed. The ROI defines the area in novel images where the fragment is searched for. For each fragment  $f$ , the amount of information it delivers about the class depends on the size of its ROI. When the ROI is too small, the information is low, because in many class images the fragment will fall outside the search region, and therefore will not be detected. If the size of the ROI is too large, the number of false detections will increase. At some intermediate size of the ROI, the mutual information reaches a maximum (Figure 6). The algorithm therefore evaluates different ROI sizes from zero to half the size of the full search window, and identifies the size that brings the MI to its maximum. The full search window is a fixed region within the input image, where the algorithm looks for the entire object. This window was set in the experiments described in this chapter to size 200x200 pixels. To detect an object within a larger image, the search window can either scan the image, or move only to selected salient locations [10]. The locations of the ROIs of the informative fragments are defined relative to the center of the search window.

## 2 Feature Hierarchies for Object Classification

In this part we describe a method for extracting complete feature hierarchies from training examples. The method includes the construction of the feature hierarchies, and learning the required parameters, such as the combination weight for each part. We briefly discuss a method of using the feature hierarchy for classification. Experimental comparisons with other classification features illustrate the advantages offered by the use of feature hierarchies compared with non-hierarchical features.

### 2.1 Construction of Hierarchical Features

The search for useful sub-fragments is similar to the search of useful top-level classification features. The top-level features are selected based on their usefulness for detected class examples. In an analogous manner, useful sub-fragments

should appear with high frequency in regions containing the ‘parent’ feature, but infrequently elsewhere. As for the top-level fragments, a useful selection criterion is the mutual information between the sub-fragment and its parent fragment. To evaluate this information, we need for each ‘parent’ fragment  $f$  a set of positive examples, namely, image regions containing the fragment  $f$ , and a set of negative examples, where the detection of  $f$  should be avoided. The positive examples for the fragment  $f$  are provided by identifying all the locations in the class images where the fragment  $f$  was detected. This set is then increased, since the goal of the fragment decomposition is to successfully detect additional examples, which were not captured by the fragment  $f$  alone. The positive set is increased by lowering the detection threshold of the fragment  $f$ , yielding examples where  $f$  is either detected or almost detected. The reduced threshold was determined to increase the positive set by 20%. This amount of increase was chosen to add a significant number of almost-detected examples, and avoid examples that are dissimilar to  $f$ . A set of negative examples was similarly derived from the non-class images. Negative examples are selected from non-class images that give “false alarms”, and therefore supply negative instances which lie close to the boundary between class and non-class instances. The reduced detection threshold used for the positive examples is applied here as well, to obtain non-class examples where the feature was incorrectly detected, or almost detected.

In terms of the positions of the fragment examples within the training images, examples come from regions in class images where the parent feature was detected or almost detected within its ROI, and negative examples come from regions in the non-class images where the feature was detected. In this case, the feature position in the training images was determined by the computation of optimal positions of all the hierarchy nodes together (Part 2.2) so that at most one example was taken from each training image.

Once the positive and negative examples of the feature  $f$  are established, sub-fragments are selected by exactly the same information maximization procedure used at the first level. The candidate sub-fragments in this case are the sub-images with their center point within the parent fragment, and having an area up to 1/4 of the parent’s area. Sub-features are added to the tree, until the additional information falls below a threshold (0.08). Experimentally, fragments with smaller contributions did not improve significantly the detection of the parent feature. If the decomposition of  $f$  into simpler features increased the information delivered by the entire hierarchy, the same decomposition was also applied to  $f$ ’s sub-features. Each of the sub-fragments is considered in turn a parent fragment, positive and negative examples are found and the set of its informative sub-fragments is selected. Otherwise, the decomposition is terminated, with  $f$  considered an atomic fragment. Atomic fragments were usually simple, typically containing edges, corners or lines. Hierarchy examples are shown in Figure 4. Examples of atomic fragments are shown in Figure 5.

During the classification stage, only the atomic features are directly correlated with the input image, and their responses are combined using weights learned at the training stage (Part 2.2).

## 2.2 Optimizing the Parameters of the Hierarchy

For each hierarchy node (fragment or sub-fragment), a region of positional tolerance is extracted, which is the feature's region of interest (ROI) (as in Part 1). The locations of the ROIs of sub-fragments in every image are determined relative to the detected position of their parent fragment. The dimensions of the ROI for all the sub-fragments are adjusted during learning to maximize the information delivered by the feature hierarchy. During the hierarchy construction, the initial ROI size of a sub-fragment is set to the size of its parent. After the hierarchy is completed, additional optimization of the ROI sizes is performed in a top-down manner: first, the ROI of the uppermost node is optimized to maximize the mutual information between the class variable and hierarchy's detection variable, while all other ROIs are fixed. A similar process is then applied to its sub-fragments, and the optimization proceeds down the hierarchy, where at each stage the ROIs of the higher levels are kept fixed.

An additional set of hierarchy parameters used for classification is the combination weights of the sub-features responses. The optimization of the combination weights is described below together with the use of these weights in the classification process.

The classification performance of the hierarchy was evaluated using a network model similar to HMAX [16], with layers performing maximization and weighted sum operations. For a given feature, the maximal response of each sub-feature is taken over the sub-feature's ROI, and then the responses of all sub-features are combined linearly:

$$r = w_0 + \sum_{i=1}^n w_i s_i \quad (5)$$

where  $r$  is the combined response,  $s_i$  the maximal response of sub-feature  $i$  within its ROI,  $w_i$  are the weights of the combination, and  $n$  the number of sub-features. For the atomic sub-features, the response was equal to the maximal normalized cross-correlation between the sub-feature and the image within the ROI. The final response  $s_p$  of the parent feature was obtained by a sigmoid function,

$$s_p = \frac{2}{1 + e^{-r}} - 1 \quad (6)$$

which normalizes  $s_p$  to the range  $[-1,1]$ .

The response of the topmost node of the hierarchy, which determines the presence or absence of the entire object, is then compared to a detection threshold. The amount of information about the class carried by the hierarchy is defined as the mutual information between the class variable  $C$  and the hierarchy detection variable  $H$ , which is equal to 1 when the response of the topmost node is higher than threshold and 0 otherwise.

The combination weights are adjusted during training using iterative optimization that alternates between optimizing positions and weights, as described below. First, the weights are initialized randomly in the range  $(0,1)$ . The scheme then alternates between the following two steps.

*Positions Step:* fix the weights, optimize feature positions. For every position of the parent fragment within its ROI the positions of sub-fragments (within their relative ROIs) that maximize the responses of the sub-fragments are found. Then, the position of the parent fragment that maximizes its response  $s_p$  is chosen. This routine can be implemented efficiently using Dynamic Programming.

*Weights Step:* fix feature positions, optimize weights. The combination weights of the features are optimized using the standard Back-Propagation algorithm with batch training protocol. The algorithm ends when no feature changes its position during the Positions Step.

This weight selection procedure can be shown to converge to a local minimum of classification error. Experimentally, we found that the algorithm converged in less than 10 iterations, average just 3 iterations. The obtained optimum was found to be stable, since starting from multiple random initial weights the algorithm terminated with similar performance.

## 2.3 Experiments

Empirical testing was used to test two main aspects of the hierarchical scheme. First, we compared the classification performance of the hierarchical features with similar features used in a holistic, non-hierarchical manner. Second, we compared the use of adaptive against a uniform hierarchy. The adaptive hierarchy adjusted the center positions and individual ROI for all the features as described above. The uniform hierarchy used instead a hierarchy where both ROI sizes and the sub-fragments were chosen in a fixed manner on a uniform grid.

In comparing the adaptive with a fixed grid hierarchy, the fixed ROI size was set at each hierarchy level to the average size of the units in the adaptive scheme, which simulations showed to be a good average size. Comparisons were averaged for all units with more than a single hierarchical level. To compare a fixed-grid hierarchy with the adaptable scheme above, each parent feature was divided into  $k$  sub-features, where  $k$  was set to the average number of sub-features in the adaptive hierarchy (6 for faces, airplanes, 9 for cows). The horizontal and vertical dimensions of the sub-features were similarly set at each level to the average dimensions in the adaptive hierarchy, shown by simulations to be a good average size.

Training images for features extraction contained 200 faces, 95 cows, 320 airplanes. The images were grey-level, 120-210 pixels in each dimension. Non-class images included a random collection of landscape, fruits, toys, etc., with a similar grey-level range. Feature detection experiments were performed on a new set of 1770 images (800 faces, 220 cows, 750 airplanes), repeated by randomly partitioning the full set into training and test images.

In computing the ROC curves [8] of a feature, the hits and false alarms were defined by using the feature as a single feature classifier. That is, test images were classified based on the feature in question; hits corresponded to class image identified correctly, false alarms to non-class images identified incorrectly. By varying the classification threshold, the complete ROC curves were obtained.

## 2.4 Summary of the Results

We first compared the non-hierarchical top level fragments with the same fragments detected in a hierarchical manner, in terms of information supplied and classification performance. The information supplied by the first-level hierarchical features increased in the test set for all fragments ( $n=150$ , 3 classes), and was significantly higher compared with the corresponding holistic features (average increase 46.6%, s.d. 30.5%,  $p < 10^{-9}$  one-tailed paired t-test). The holistic and hierarchical features were also compared using their complete ROC curves, showing a significant advantage of the hierarchical detection over the entire range, (0-90% false alarm,  $n=150$ ,  $p < 0.000001$ , Figure 3b). These comparisons clearly show that hierarchical features are more informative and produce better classification.

Further decomposition into a multi-level hierarchy provided additional significant gain in information ( $n=97$  features, average increase 10.0%, s.d. 10.7%  $p < 10^{-9}$  one-tailed paired t-test). The ROC detection curves also improved significantly (example in Figure 3a).

The full hierarchy also proved considerably more robust than holistic features. This is of interest particularly when the feature hierarchies are considered as a possible biological model for object processing. A biological system cannot be expected to converge to the exact optimal parameters, but we found that introducing size and position errors (13%, 25% of feature size) reduced the MI on average by 10.8% for the full hierarchy, compared with 35.3% for holistic features ( $n=41$ ,  $p < 10^{-10}$ , paired t-test).

Using the optimal ROI sizes adds significantly to the MI compared with a fixed ROI size, that was optimized for each level separately (average 8.1% s.d. 13.7%  $p < 0.0055$ ), and different subunits had different optimal ROI size. Adapting the relative positions of the subunits is also significant: if the subunits' centers were arranged on a uniform grid, rather than selecting their optimal locations during training, the MI decreases ( $N=153$ , average 43% s.d. = 35%  $p < 10^{-10}$  paired t-test), and the detection performance of the units decreases (Figure 3a).

These results can be used to compare the use of hierarchical and holistic features in both computer vision and biological modelling. Most computational models of recognition and classification in the past did not use hierarchical features. This is in contrast to the primate visual system where objects are analyzed by a hierarchy of features. Our analysis and testing shows that hierarchical features are significantly more informative and better for classification than holistic features. It also shows that this improvement requires the learning of positions and sizes; without this the hierarchical scheme is not significantly better than a single layer of top-level features.

Some previous biological models ([11],[16]) used a hierarchy of features, to simulate the cortical structure. However, these models used fixed uniform architecture in contrast with the adaptive scheme used here, and which proved valuable to the construction of a successful hierarchy. The method of selecting the features, based on the information they contribute, also proved to produce better results than either fixed features [16], or features extracted by back-propagation





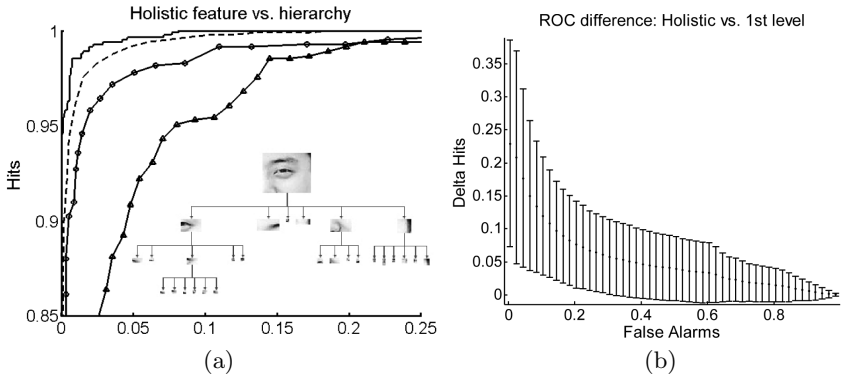
**Fig. 2.** Informative fragments (examples on the left) and their optimal sub-fragments (right), selected automatically from three object classes by maximizing mutual information

neural network model [11]. See [5] for more details on experimental comparisons with other types of features.

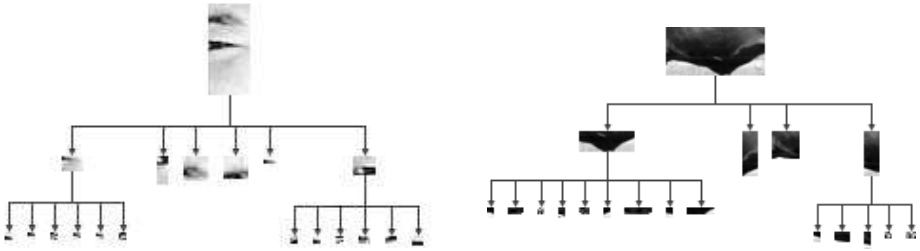
### 3 Semantically Equivalent Features

In this part we consider the problem of detecting semantically equivalent parts of objects belonging to the same class. By ‘semantic’ equivalence we mean parts of the same type in similar objects, often having the same part name, such as a nose in a face, an animal’s tail, a car’s headlight and the like. The aim is to identify such parts, although their visual appearance can be highly dissimilar. The input to the algorithm is a set of images belonging to the same object class, together with an image patch (called below a “root fragment”), depicting a part of an object. The output is a set of image patches from the input images, containing object parts which are semantically equivalent to the one depicted in the root fragment. Examples of semantically equivalent fragments are shown in Figure 7. In each row, the leftmost image contains the root fragment, the other images are semantically equivalent fragments discovered by the algorithm. The identification of equivalent object parts has two main goals. First, the correct detection and identification of object parts is important on its own right, and can be useful for various applications that depend on identifying parts, such as recognizing facial expressions, visual aid for speech recognition, visual inspection, surveillance and so on. Second, the correct identification of semantically equivalent object parts improves the performance of object recognition algorithms. In several recent object recognition schemes [1],[6],[9],[21] image fragments depicting object components are used as classification features. Our results show that the performance of such schemes can be improved when an object component is represented not by a single fragment, but by a set of semantically equivalent fragments.

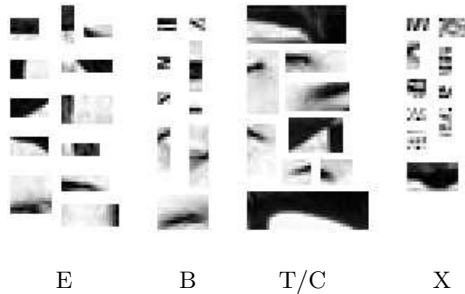
The general idea behind our approach is to use common context to identify equivalent parts. Given an image fragment  $F$  depicting a part of an object, we



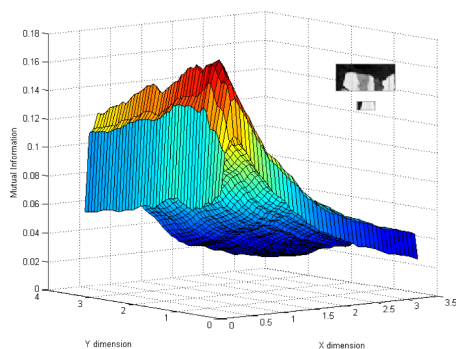
**Fig. 3.** Comparing recognition by hierarchical and holistic features. (a) ROC of a single fragment comparing detection by a holistic feature (third from top), optimal decomposition into sub-features (second from top), full hierarchical decomposition (top curve), and decomposition on a fixed grid (lowest curve). y-axis: percent correct identification of class images by the fragment (hits), x-axis: percent incorrect identification of non-class images (false alarms). (b) Average gain in ROC, vertical axis: increase in hit rate, horizontal: false alarm rate ( $n = 150$  fragments). See text for further details.



**Fig. 4.** Examples of full feature hierarchies, (bottom nodes are atomic features)



**Fig. 5.** Atomic features, derived from three classes. Most are tuned to oriented edges (E), bars (B), terminations/corners (T/C); some are more complex (X).



**Fig. 6.** Increase in mutual information as a function of search region size, for one sub-feature (inset). Color code: increase in mutual information, horizontal axes: ROI size, (size of parent feature is taken as ‘1’). Optimal x-size: 0.27, y-size: 0.43.



**Fig. 7.** Examples of semantically equivalent fragments, extracted by the algorithm. The leftmost image in each set is the input root fragment, the others are equivalent parts identified by the algorithm (horse torso with forelegs, car wheels). The algorithm identifies semantically similar parts that can have markedly different appearance in the image.

look for a context  $C$ , defined as a collection of image fragments that co-occur with  $F$  consistently and in a stable geometric configuration. When such context is found, we look for all images where the context fragments are detected, and infer from their positions the location of fragments that are likely to be semantically equivalent to  $F$  (Figure 8).

### 3.1 Description of the Algorithm

In this section, we describe the algorithm for the detection of semantically equivalent image fragments. The main stages of the algorithm are the identification of common context (3.1) and the use of context to extract equivalent parts (3.1). We begin with describing visual similarity matching used as a pre-processing step.

**Visual Similarity Matching.** The input to the algorithm consists of a set of images of different objects within a class,  $I_k$ , and a single fixed fragment  $F$  (the “root fragment”). We first identify in each of the input images  $I_k$  the image patch with the maximal similarity to  $F$ . We used the value of normalized cross-correlation as a similarity measure, but other image-based similarity measure can be used as well. To improve the performance of visual similarity-based matching, the images are filtered with Difference of Gaussians (DoG) filter [12] before computing the NCC. This filter emphasizes the gradients in images and removes small noise. The combination of DoG filtering with computation of NCC is called below DNCC.

Image patches at all locations in  $I_k$  are examined, and the patch  $P(I_k, F)$  with highest DNCC score is selected. If the cross-correlation between  $P(I_k, F)$  and  $F$  exceeds a pre-defined threshold, then  $F$  is detected in  $I_k$ , and  $P(I_k, F)$  is called the patch corresponding to  $F$  in image  $I_k$ . The set of all the images  $I_k$  where corresponding patches  $P(I_k, F)$  are detected is denoted by  $D(F)$ . The detection threshold for candidate context patches was chosen automatically as explained in Part 1.

**Context Retrieval.** After determining the set  $D(F)$ , containing the images where  $F$  was detected, the next goal is to identify context fragments that consistently co-occur with  $F$  and its corresponding patches  $P(I_k, F)$ . Reliable context fragments should meet two criteria: the context fragment  $f$  and root fragment  $F$  should have high probability of co-occurrence, and their spatial relations should be stable. We next describe the selection based on these criteria.

The search for good context fragment starts by pairing the root  $F$  with patches  $f_i$  in each image in  $D(F)$  at multiple sizes and positions. These patches are the candidate context patches for  $F$ . In practice, we limited the search to patch sizes ranging from 50% of  $F$  size up to 150% in each dimension, with scaling step of 1.5. For each patch size, we examine patches in positions placed on a regular grid with step equal to 1/4 of the size of a patch. The exact position and size of a context patch is eventually optimized as described later in this section. For every candidate patch  $f$ , we find the set  $D(f)$  of images containing patches visually similar to  $f$ , as described in Part 3.1.

The first context condition above was high co-occurrence, that is, a good context fragments should satisfy  $p(F|f) > p(F)$ . We also want to focus on context fragments that appear together with  $F$  at least some minimal number of times, and therefore require:

$$P(f|F) > \theta_p \quad p(F|f) > p(F) \tag{7}$$

The value of  $\theta_p$  was computed automatically by sampling a set of candidate patches from  $D(f)$ , computing their probabilities of co-occurrence with  $F$ , and setting the threshold to average co-occurrence probability plus one standard deviation.

Second,  $F$  and  $f$  should appear in a stable spatial configuration. If the variations in scale and orientation between the images are assumed to be small, then the relative location of  $F$  and  $f$  when they are detected together should be similar. We therefore test the variance of coordinate differences:

$$Var(F_x - f_x) < \theta_{VarX} \quad Var(F_y - f_y) < \theta_{VarY} \tag{8}$$

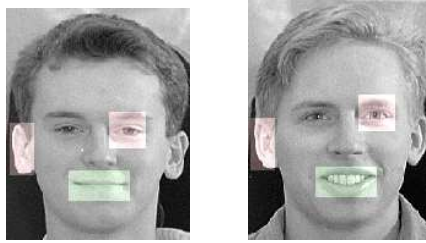
Here  $F_x$  and  $f_x$  are vectors of  $x$ -coordinates of the centers of image patches corresponding to  $F$  and  $f$ , respectively, in images from  $D(F) \cap D(f)$ , similarly for  $F_y$  and  $f_y$ . The thresholds  $\theta_{VarX}$  and  $\theta_{VarY}$  determine the flexibility of the geometric model of the object. These thresholds are also set automatically by computing the values of  $Var(F_x - f_x)$  and  $Var(F_y - f_y)$  for the sampled fragments, for which  $P(f|F) > \theta_p$  and setting the thresholds to the average of these values plus one standard deviation.

To identify the best context fragments, we first remove from the set of candidates all fragments that do not meet requirements (7) and (8). We next select from the remaining set the fragments with the highest probability of co-occurrence with  $F$ , and smallest variances of coordinate differences (indicating a stable geometric relation with the root  $F$ ). To combine these criteria, we compute a ‘consistency weight’,  $w_f$ :

$$w_f = P(f|F) \cdot \frac{1}{1 + \sqrt{\max(Var(F_x - f_x), Var(F_y - f_y))}} \tag{9}$$

The fragment with the highest  $w_f$  is then selected as a context fragment. Since the initial search for context fragments was limited to a fixed grid, we refine the optimal position and size of the context fragment by searching locally for the best fragment position and size that maximize  $w_f$ . We add the optimized fragment to the set of context fragments.

To avoid redundancy, and prefer conditionally independent context fragments (see Part 3.1 for details), we remove from the set of remaining candidates all the fragments that intersect the selected one by more than 25% of their area, and repeat the process until no candidates are left. The final context set contains fragments  $f_i$  that have high co-occurrence with  $F$ , and with stable relative positions. Typically this set contains between 6 and 12 fragments.



**Fig. 8.** Left: a root fragment (mouth) together with context fragments (ear and eye). Right: the same context detected in another image; a semantically equivalent part is identified by the context.

**Identifying Semantically Equivalent Parts.** After the set of context fragments has been selected, they are used to infer the positions of fragments that are semantically equivalent to the root fragment  $F$ . Using a probabilistic model, we identify for each image  $I_k$ , in which at least one context fragment has been detected, the most likely position of  $F_k$ , a semantically equivalent fragment to  $F$ .

Assume for simplicity first that our context set consists of a single fragment  $C$ . Our modelling assumption is that if  $C$  is detected in some image  $I_k$  at coordinates  $(x_c, y_c)$ ; then the probability density of  $F$  being found at coordinates  $(x, y)$  is 2D Gaussian centered at  $(\hat{x}_c, \hat{y}_c)$ , where  $\hat{x}_c$  and  $\hat{y}_c$  are the expected coordinates of the root fragment's center, predicted by context fragment  $C$ . The values of  $\hat{x}_c$  and  $\hat{y}_c$  are computed as:

$$\hat{x}_c = x_c + \overline{\Delta x_c} \quad \hat{y}_c = y_c + \overline{\Delta y_c} \quad (10)$$

where  $\overline{\Delta x_c}$  and  $\overline{\Delta y_c}$  are the mean coordinate differences between the centers of  $F$  and  $C$ , estimated during training.

$$P(F(x, y)|C) = P(F|C) \cdot N(x - \hat{x}_c, y - \hat{y}_c; \Sigma_c) \quad (11)$$

where  $\Sigma_c$  is the covariance matrix of coordinate differences between the centers of fragments  $F$  and  $C$ , estimated during training.

If the context fragment  $C$  is not detected in the image  $I_k$ , we assume 2D uniform probability density of  $F$  being found at coordinates  $(x, y)$ :

$$P(F(x, y)|\bar{C}) = P(F|\bar{C}) \cdot U(W, H) \quad (12)$$

here the distribution bounds  $W$  and  $H$  are set to the width and height of the image.

When the context consists of several fragments, we assume conditional independence between them given the detection of  $F$  at position  $(x, y)$ :

$$P(C_1, \dots, C_N|F(x, y)) = \prod_{i=1}^N P(C_i|F(x, y)) \quad (13)$$

The modelling assumption of the conditional independence is motivated by the observation that if geometric relation between fragments is stable, the positions of the context fragments are determined by the position of the root fragment. The fluctuations of the positions are due to noise, which is assumed to be independent for the context fragments. Modelling of higher-order geometric relations between fragments is also possible, but we found in testing that it did not make a significant contribution. Applying Bayes rule to (13):

$$P(F(x, y)|C_1, \dots, C_N) = \frac{P(F(x, y))}{P(C_1, \dots, C_N)} \prod_{i=1}^N P(C_i|F(x, y)) \tag{14}$$

We assume the prior probability  $P(F(x, y))$  of finding  $F$  at the coordinates  $(x, y)$  to be uniform, consequently not depending on  $x$  and  $y$ . It is also straightforward to use non-uniform prior. The probability  $P(C_1, \dots, C_N)$  similarly does not depend on  $(x, y)$ . Therefore, we can write:

$$P(F(x, y)|C_1, \dots, C_N) \propto \prod_{i=1}^N P(C_i|F(x, y)) \tag{15}$$

For the individual factors  $P(C_i|F(x, y))$  we use equations (11) or (12), depending on whether or not the context fragment  $C_i$  was detected in the image. Applying the Bayes rule again, if  $C_i$  was detected in the image:

$$P(C_i|F(x, y)) = \frac{P(C_i) \cdot P(F(x, y)|C_i)}{P(F(x, y))} = \frac{P(C_i) \cdot P(F|C_i) \cdot N(x - \hat{x}_{ci}, y - \hat{y}_{ci}; \Sigma_{C_i})}{P(F(x, y))} \tag{16}$$

If  $C_i$  was not detected in the image:

$$P(\bar{C}_i|F(x, y)) = \frac{(1 - P(C_i)) \cdot P(F(x, y)|\bar{C}_i)}{P(F(x, y))} = \frac{(1 - P(C_i)) \cdot P(F|\bar{C}_i) \cdot U(W, H)}{P(F(x, y))} \tag{17}$$

Now we can find the values of coordinates  $x$  and  $y$  that maximize (15), i.e. find a Maximum Likelihood solution for the coordinates of the center of the fragment  $F$ :

$$(x, y) = \arg \max \prod_i N(x - \hat{x}_{ci}, y - \hat{y}_{ci}; \Sigma_{C_i}) \tag{18}$$

where each 2D Gaussian can be explicitly written in terms of its parameters: mean position and covariance matrix. Note that the product is taken over only the detected context fragments. Taking the log of the product, differentiating with respect to  $x$  and  $y$ , and setting the derivatives to zero, yields a system of equation of the form:

$$xA - yB + C = 0 \quad yD - xB + E = 0 \tag{19}$$

where

$$\begin{aligned}
 A &= \sum_i \frac{1}{(1-\rho_{xyi}^2)\sigma_{xi}^2} \\
 B &= \sum_i \frac{\rho_{xyi}}{\sigma_{xi}\sigma_{yi}} \\
 C &= \sum_i \left( \frac{\rho_{xyi}(y_{ci}+\overline{\Delta y_{ci}})}{\sigma_{xi}\sigma_{yi}} - \frac{x_{ci}+\overline{\Delta x_{ci}}}{(1-\rho_{xyi}^2)\sigma_{xi}^2} \right) \\
 D &= \sum_i \frac{1}{(1-\rho_{xyi}^2)\sigma_{yi}^2} \\
 E &= \sum_i \left( \frac{\rho_{xyi}(x_{ci}+\overline{\Delta x_{ci}})}{\sigma_{xi}\sigma_{yi}} - \frac{y_{ci}+\overline{\Delta y_{ci}}}{(1-\rho_{xyi}^2)\sigma_{yi}^2} \right)
 \end{aligned} \tag{20}$$

$$\sigma_{xi} = \sqrt{Var(x - x_{ci})}, \quad \sigma_{yi} = \sqrt{Var(y - y_{ci})}, \quad \rho_{xyi} = \frac{CoVar((x - x_{ci}), (y - y_{ci}))}{\sigma_{xi}\sigma_{yi}}$$

Solving (19), we obtain:

$$y = \frac{AE + BC}{B^2 - AD} \quad x = \frac{By - C}{A} \tag{21}$$

After obtaining the maximal likelihood solution for the coordinates  $(x, y)$ , we extract a fragment centered at  $(x, y)$  with size equal to the size of  $F$ , and add it to the set of fragments semantically equivalent to  $F$ .

The set of semantically equivalent fragments constructed in this manner is called the “equivalence set” of the part. We next sort it by measuring the strength of the evidence used to select the fragments. This is obtained by setting the optimal values found for  $(x, y)$  into (15) and taking the log. The resulting quantity is equal to the log-likelihood of the optimal solution plus a constant factor. This value is then used to sort the equivalence set: the log-likelihood will be smaller when only a few context fragments are detected in a particular image, or when their evidence was inconsistent, i.e. they predict different locations of a semantic fragment. The decision regarding the number of fragments from the equivalence set to be used is application-dependent. For the object recognition experiments we used the upper 30% of the sorted equivalence set. For the part detection experiments we used the entire set and counted the number of errors.

The section above describes the main computation; its accuracy can be improved by incorporating a number of additional steps. We used in particular simple criteria to reject outliers, based on the fact that they will be detected at highly variable image locations. We therefore computed the average value of coordinate differences between the detected positions of  $F$  and  $f$ , and removed the farthest outliers, until the variance of coordinate differences is below threshold. The same procedure for outlier rejection is used when performing the Maximum Likelihood estimation, since some of the context fragments can correspond to false detections.

### 3.2 Experimental Results

**Object Parts Detection.** We selected first for testing 7 root fragments depicting different parts of the human face (shown in Table 1), and applied the



algorithm described in Section 3.1 to detect semantically equivalent parts in new face images independently for each root fragment. For comparison, we applied the algorithm for detecting face parts based on their visual similarity to the root fragment, as described in Section 3.1, using the same input image set and root fragments. The visual similarity for testing was computed using two different measures - DNCC and SIFT [12]. We applied both algorithms to a database of 1000 face images (about 150x200 pixels in size, roughly the same scale and orientation) and counted the number of images where all the parts were simultaneously detected correctly. The numbers of face images where all 7 fragments were simultaneously detected correctly were 379 using semantic equivalence, 5 using DNCC visual similarity and 7 using SIFT visual similarity. As can be seen, the method is successful in recovering a large number of correct part configurations, that cannot be identified by their visual similarity. The percentage of correctly identified matches, verified by humans, for semantic equivalence and DNCC visual similarity was also computed for each individual part, yielding the results in Table 1. Using the SIFT similarity measure produced similar results to DNCC. See [4] for the details of the experiments on other object classes.

**Object Recognition.** The classifier we used for the experiments is an extension of a classifier described in [21]. Briefly, an object from a general class is represented by a collection of object parts. A set of fragments (either visually similar or semantically equivalent) selected automatically, is used to represent each part. An object part is detected in the image if one of the fragments representing it is detected within a detection window. Each fragment is assigned a weight  $w_i$  determined by the log-likelihood ratio:

$$w_i = \log \frac{P(F_i|C = 1)}{P(F_i|C = 0)} \quad (22)$$

where  $C$  is a class variable (1 in images containing an object, 0 otherwise) and  $F_i$  is a fragment variable (1 when the fragment was detected, 0 otherwise). Final detection is based on a Bayesian decision,

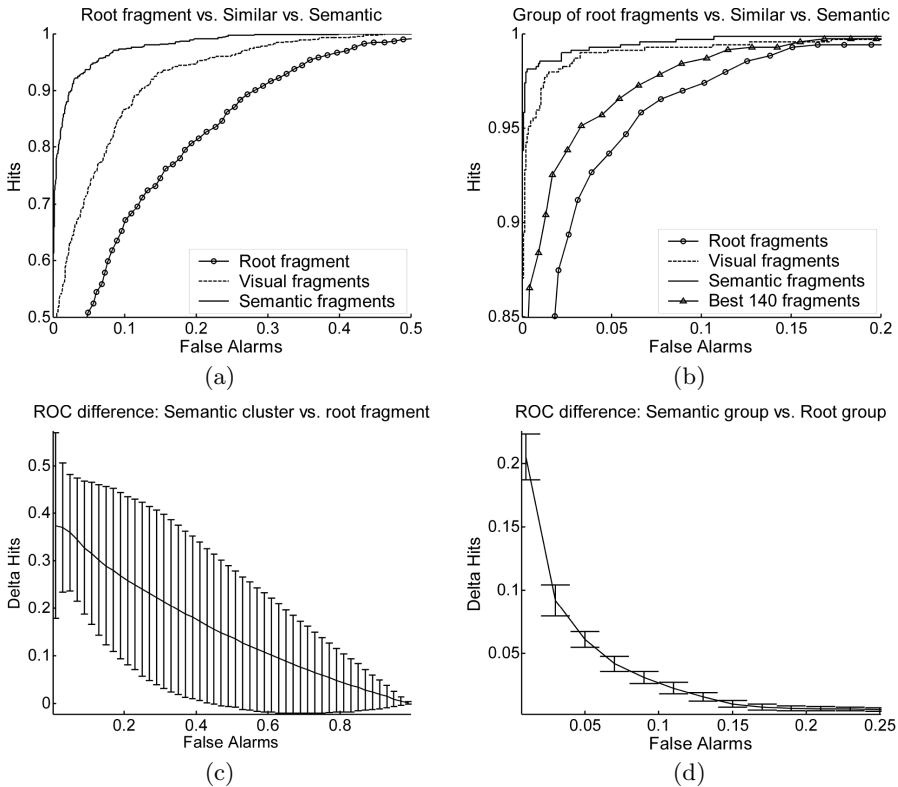
$$\sum_i w_i F_i > \theta \quad (23)$$

where  $\theta$  is decision threshold; by varying  $\theta$  complete ROC curves are obtained (Figure 9).

Face detection performance was compared using 7 face parts, shown in Table 1. Each part was then represented by 20 representative image fragments selected to optimize performance. The two schemes we compared used an identical classifier, but differed in the selection of the image fragments representing each part. In the ‘semantic’ scheme, each part was represented by a set of 20 semantically equivalent fragments, selected by the algorithm described in Part 3.1. In the ‘visual similarity’ scheme, each part was represented by 20 representative image fragments, selected from the set of visually similar fragments so as to optimize performance.













The selection of representative fragments for each face part was done in a greedy fashion, using a mutual information criterion: the fragment delivering the highest information between the classifier response and the true class was selected first. Next, all the remaining fragments were examined, to determine the fragment contributing the largest amount of additional information when added to the first one. The process was repeated until 20 fragments have been selected. An identical selection procedure was used to select the best representatives from the set of visually similar fragments.

The image set was divided randomly into a training set (300 images) and test set (700 images), and the computation was repeated 50 times. The results are presented in Figure 9. Figure 9a shows the comparison of ROC curves of a single root fragment (the mouth in Table 1): the ROC curve of the classifier based on this fragment alone (line with circles), the ROC curve of the classifier based on visually similar fragments (dashed line) and the ROC curve of the classifier based



**Fig. 9.** Comparing recognition by semantic equivalence and visual similarity. (a) ROC curves for a single part (mouth); (b) classification by 7 parts; (c) average gain in ROC between semantic and visual similarity for single parts; (d) average gain in ROC for classification by 7 parts, using semantic vs. visual similarity. See text for further details.

**Table 1.** Percentage of correctly identified fragments representing object parts in three classes: faces, car rear views and toy cars. See text for further details.

Root fragment	Semantic equivalence	Visual similarity (DNCC)
	94%	33%
	92%	39%
	92%	71%
	89%	20%
	90%	29%
	88%	51%
	84%	26%
	65%	41%
	55%	18%
	64%	25%
	71%	59%
	42%	32%

on semantic equivalence class (solid line). Figure 9b shows the performance of 7 root fragments, compared to the performance of visually similar and semantic fragments, where each of 7 parts was represented by a set of 20 fragments. The graph also shows the performance based on the selection of 140 individual fragments. Figure 9c shows the mean difference between the ROC curves of the classifier based on visually similar fragments, and the classifier that uses semantic equivalence classes for single parts. Figure 9d shows the mean difference between the ROC curves of the classifier based on the group of 7 parts, represented by semantically equivalent fragments, compared with the performance of 7 root fragments used together.

Image similarity was based in the scheme above on normalized cross-correlation. Other, more robust image comparison measures can be used to compensate for scale changes, affine transformations, and small local distortions (see [15] for a review). Comparisons in [14] have shown that in the absence of scale changes and affine transformations, the performance of normalized cross-correlation is comparable to the performance of the SIFT descriptor [12] and better than the results

obtained with other measures. Since we tested the algorithm under these conditions, the use of DNCC was appropriate. We also compared the performance of DNCC and SIFT, in the following way. For each face image, the fragment semantically equivalent to the root and the fragment most visually similar to the root were determined by the algorithm (only the images where both fragments were found by the algorithms were considered). The images where the computed semantic fragment was correct (as determined by an observer), but the fragment selected by visual similarity was incorrect, were chosen for comparison. For each image, we then normalized the three fragments (the root, the semantically equivalent and the most visually similar) by an affine transform to a normal form [15], and compared the SIFT distance between the root and semantic fragment, to the SIFT distance between the root and the visually similar fragment. In 74.6% of the cases, the SIFT made the incorrect selection: the visually similar fragment was closer to the root fragment than the semantic fragment. We conclude that the SIFT distance did not overcome the incorrect choice of the visually similar fragment made by the DNCC.

### 3.3 Other Methods of Obtaining Equivalent Fragments

The scheme described above identifies sets of semantically equivalent fragments in the training images. These semantically equivalent fragments depict corresponding parts in different objects of the same class, such as different hairlines, aircraft wings, car wheels and the like. They can also identify different views of the same object part under different conditions, such as a smiling vs. neutral mouth, or open vs. closed eye. Other methods have been developed in the past for identifying the same object part under changes in viewing conditions, in particular, changes in viewing direction and illumination conditions. These equivalence relations then play a crucial part in identifying specific objects under different conditions. We briefly review in this section past methods for identifying such equivalent fragments, and comment in the final discussion on their use in object identification.

**Motion-Based Fragment Equivalence.** Motion can serve as a powerful cue for identifying the same object part under different viewing conditions. If an image region transforms in a smooth continuous manner over time, then its image at different times are likely to represent the same part under different conditions. In particular, when the object moves rigidly in space, such motion-based equivalence can be used to identify different appearances of an object part from different viewing directions. Motion-based equivalence has been used to deal with the problem of position invariance [7] as well as more complex transformations [17],[19],[20].

In [19], the problem of obtaining fragment sets representing the same object part under different viewing angles was considered, and a method for identifying informative equivalent parts was developed based on motion correspondence. The method first extracts a large pool of so-called extended fragments, which are sets of fragments representing the same object part under different viewing conditions, in this case different viewing directions. The correspondence between

fragments in different views is established using motion tracking [18]. From this initial pool of motion-related fragments, informative extended fragments are extracted based on the mutual information supplied by the extended fragments for view-invariant recognition. The selection of informative fragments is similar to the algorithm described in Part 1, but applied to extended fragments rather than to individual fragments. The selection process is initialized by selecting the extended fragment with the highest mutual information. Extended fragments are then added one by one by the max-min procedure described above, until the gain in mutual information is small, or a pre-selected size of fragment bank is reached. Fragment detection is done by computing the maximal similarity between the fragment and underlying image patch over the entire image, and comparing it to a pre-determined threshold. The optimal thresholds were computed automatically, by a procedure similar to the one described in Part 1.

In the recognition stage, the system was given a single image of a novel object from the learned class, for example, a face in frontal view. The task was then to identify the same object from a side view, from a large set of both frontal and side-view faces. The identification was based on the activation of the identified extended fragments. The main underlying assumption is that after learning, a frontal face  $F$  and a corresponding side view  $F'$  share the same extended fragment. If a particular fragment  $f$  is found in the frontal view, then its corresponding counterpart  $f'$  should be present in the side view. In order to identify the corresponding side view, the activation pattern for the query frontal view was computed. The activation pattern is a binary vector containing 1 in  $n$ -th position if the  $n$ -th fragment in the object representation was active in the image, and 0 otherwise. Similarly, the activation patterns were computed for all the images in the test set. The test image, whose activation pattern was the closest to the one of the query image, was then selected as the corresponding side view.

**Equivalence Under Arbitrary Changes in Viewing Conditions.** The motion-based correspondence of object parts proved useful for dealing with view-invariance under changes in viewing direction. However, motion-based correspondence is not always applicable for identifying the same object part under different conditions. For example, views of the same object under different illumination conditions are usually not related by continuous motion. In [2], a different criterion for obtaining fragment equivalence sets was therefore employed, without relying on motion correspondence. Fragment equivalence was established instead based on the consistency in parts appearance in different objects in which these parts are present. If two fragments,  $F_1$  and  $F_2$  represent the same object part under different viewing conditions (such as different illuminations, or also different viewing directions),  $C_1$  and  $C_2$ , then their detections should be consistent – namely, if  $F_1$  is detected in an image of some object  $O$  under viewing condition  $C_1$ , then  $F_2$  should also be detected in an image of  $O$  under condition  $C_2$ . In contrast, two unrelated fragments will be in general significantly less consistent. Therefore, this consistency criterion can be used for identifying equivalent object fragments.

Given a set of images  $I_{11} \dots I_{1N}$  of  $N$  objects taken under condition  $C_1$  and a set of images  $I_{21} \dots I_{2N}$  taken under condition  $C_2$ , the activation patterns  $A_1$  and  $A_2$  of fragments  $F_1$  and  $F_2$  respectively can be computed. Their consistency can be derived, for example by the simple score:

$$S(F_1, F_2) = \frac{NCC(A_1, A_2) + 1}{2}$$

This is just the correlation of the activation patterns, but re-normalized to lie between 0 and 1. To make the scheme robust to noise and to within-object redundancy, this consistency measure was augmented with a measure based on geometric consistency, which used a simple proximity assumption: if two object parts are located nearby, their matching parts also should lie close to each other. This constraint was implemented using a hierarchical representation of proximity relations. The scheme was shown to deal effectively with changes in illumination and pose without relying on motion correspondence.

## 4 Summary and Discussion

In this chapter we have presented two extensions of the fragment-based object recognition scheme. The basic scheme uses informative image fragments as classification features. Here we proposed a hierarchical decomposition of the features into parts and sub-parts at multiple levels. The second extension was to use semantic equivalence sets of features, depicting different appearances of the same object part. We have shown that hierarchical features are more informative and better for classification compared with the same features used non-hierarchically. For semantic features, we have shown how the method can automatically learn the part structure of a new domain and extract sets of semantically equivalent fragments. Semantic features are an example of the more general concept of extended features, which are sets of fragments representing the same or similar object parts under different viewing conditions. Different methods were described above for extracting extended fragments based on common context, motion, and consistency across transformations. Extended features are used in the proposed scheme as the basis for making broad generalizations in object recognition, at the level of general classification as well as specific object identification. For example, a particular object can be recognized across changes in pose, illumination, and complex local shape changes, based on the representation of its components in terms of extended features. The capacity of the recognition system to deal with large variability in appearance at the objects level is inherited in this scheme from learning the variability at the level of common informative components.

The two aspects described above, hierarchical representation and the use of extended fragments, can be combined into a representation using a hierarchy of sub-parts, where each sub-part is represented by extended fragments. This representation can be extended in several directions. For example, in terms of the classification algorithm using this representation, instead of the bottom-up computation described above, we have also used a full Bayesian network

which produced a significantly better interpretation of the constituent parts. A second general direction is the extension of the hierarchy from a single class to a multi-class representation. The issues here include, for example, the optimal construction of a feature hierarchy for multiple classes simultaneously, extracting semantically equivalent fragments across different classes, sharing features across classes at multiple levels in the hierarchy, and using the hierarchy to make fine distinctions between similar classes. Finally, given the hierarchical nature of objects representation in the primate visual cortex, it will be of interest to use the computational studies of feature hierarchies and extended features to model aspects of the human visual system.

## Acknowledgements

This work was supported by grant no. 3-992 from the Israeli Ministry of Science and Technology, and conducted at the Moross Laboratory for Vision and Motor Control.

## References

1. S. Agarwal, A. Awan, D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE TPAMI*, Vol. 26(11). (2004) 1475–1490
2. E. Bart, S. Ullman. Class-based matching of object parts, *Proc. CVPR Workshop on Image and Video Registration*, (2004)
3. I. Biederman. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, Vol. 94(2) (1987) 115–147.
4. B. Epshtein, S. Ullman. Identifying Semantically Equivalent Object Fragments. *CVPR*, (2005) 2–9
5. B. Epshtein, S. Ullman. Feature Hierarchies for Object Classification. *ICCV*, (2005), to appear.
6. R. Fergus, P. Perona, A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. *CVPR*, (2003) 264–271
7. P. Foldiak. Learning invariance from transformation sequences. *Neural Computation*, Vol. 3(2). (1991) 194–200
8. D. Green, J. Swets. *Signal Detection Theory and Psychophysics*. Wiley, NY, (1966)
9. B. Heisele, T. Serre, M. Pontil, T. Vetter, T. Poggio. Categorization by learning and combining object parts. *NIPS*, (2001)
10. L. Itti, C. Kosh, E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, Vol. 20(11) (1998) 1254–1259
11. Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, Vol. 1(4) (1989) 541–551
12. D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vis*, Vol. 60(2) (2004) 91–100
13. D. Marr, H. Nishihara. Representation and recognition of the spatial organization of three dimensional structure. *Proceedings of the Royal Society of London B*, Vol. 200 (1978) 269–294.

14. Mikolajczyk, K., Schmidt, C.: A performance evaluation of local descriptors. CVPR, (2003) 257–264.
15. K. Mikolajczyk, C. Schmidt. Scale and affine invariant point detectors. *Int. J. Comp. Vis.*, Vol. 60(1) (2004) 63–86
16. M. Riesenhuber, T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, Vol. 2(11) (1999) 1019–1025
17. S. Stringer, E. Rolls. Invariant object recognition in the visual system with novel view of 3D objects. *Neural Computation*, Vol. 14. (2002) 2585–2596
18. C. Tomasi, T. Kanade. Detecting and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University (1991)
19. S. Ullman, E. Bart. Recognition invariance obtained by extended and invariant features. *Neural Networks*, Vol. 17. (2004) 833–848
20. S. Ullman, S. Soloviev. Computation of pattern invariance in brain-like structures. *Neural Networks*, Vol. 12. (1999) 1021–1036
21. S. Ullman, M. Vidal-Naquet, E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, Vol. 5(7) (2002) 1–6
22. M. Vidal-Naquet, S. Ullman. Object Recognition with Informative Features and Linear Classification. ICCV, (2003) 281–288
23. L. Wiskott, J. Fellous, N. Kruger, C. von der Malsburg. Face Recognition by Elastic Bunch Graph Matching, *IEEE TPAMI*, Vol. 19(7), (1997) 775–779