CHAPTER 1

OBJECT RECOGNITION WITH DEFORMABLE FEATURE GRAPHS: FACES, HANDS, AND CLUTTERED SCENES

Jochen Triesch
Department of Cognitive Science, UC San Diego
9500 Gilman Drive, La Jolla, California, 92093-0315, USA
E-mail: triesch@uscd.edu

Christian Eckes
Fraunhofer IMK NetMedia
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
E-mail: Christian.Eckes@imk.fhg.de

A fundamental question in invariant object recognition is that of representation. This chapter reviews object representations based on deformable feature graphs that describe particular views of an object as a spatial constellation of image features. These representations are particularly useful in situations of high clutter and partial occlusions. We demonstrate the benefits of these representations in three recognition applications: face analysis, hand gesture recognition, and the interpretation of cluttered scenes composed of multiple partly occluded objects.
We conclude by discussing current trends and open challenges.

1. Introduction

1.1. The Problem of Invariant Object Recognition

Object recognition is perhaps the most fundamental problem in computer vision. Humans can recognize objects without effort despite background clutter, partial occlusions, or changes in object location, view point, lighting situation, etc. This capability is called invariant recognition because the recognition process has to be invariant to these and other environmental changes that can dramatically change all the pixels of an image while the depicted object remains its identity (Fig. 1).

Despite considerable effort it has not been possible to endow computer vision systems with levels of competence that are anywhere close to human performance. What makes the task so difficult? A part of the answer is that invariant recognition is a computationally very demanding task and today’s computers are no match for the processing power of the human visual system. Another part of the answer may be

that we have not found the right way of representing objects that facilitates efficient solutions to the invariant recognition problem. This chapter reviews approaches that
Fig. 1. Left: invariant recognition. Humans can easily recognize a person or an object despite great variability in the appearance due to pose, lighting, occlusion, clutter, facial expression and other factors. Right: segmenting an object from the background can be quite difficult and may require prior knowledge of the object's appearance — in this case the dalmation dog in R.C. James's famous picture.

represent particular views of an object as a deformable constellation of certain image features.

1.2. Recognition and Segmentation

A problem closely related to the invariant recognition problem is the segmentation problem. Segmentation refers to the process of grouping together all pixels in an image that belong to the same object. The way in which researchers view the relation between recognition and segmentation is a topic of substantial controversy. The traditional view is that segmentation should happen independent of and prior to object recognition. Image elements are grouped together solely based on low level properties of individual pixels or small image patches such as brightness, color, motion, stereo disparity, or texture. Once an object has been segmented recognition proceeds by extracting relevant features from the segmented object and subsequent classification. This approach may be called "segmentation, feature extraction, recognition (SFR)". It is the method of choice in many application domains such as industrial inspection, where a high degree of control over the imaging process is possible. In this case, the proper choice of background and lighting as well as a possibly quite restricted set of objects can make segmentation relatively easy. When it comes to vision in more natural and less controlled environments, however, the SFR approach is severely limited. Despite decades of intensive research, it has been impossible to find robust solutions to the segmentation problem that can handle the complexities and ambiguities of everyday visual scenes1. The necessary information for correctly grouping all the pixels of an image to the proper objects may simply not be present in the low level pixel properties, as illustrated by R.C. James's famous picture of a dalmation dog (Fig. 1, right).

There are two alternatives to the SFR scheme. The first one simply ignores the segmentation problem altogether and tries to recognize objects directly from features extracted from the entire image. We will refer to this approach as "direct
Object Recognition with Deformable Feature Graphs

recognition (DR)". This is the viewpoint we will focus on in this chapter. The second alternative tries to couple segmentation and recognition in a closed feedback loop such that partial or tentative segmentation results aid recognition while provisional recognition results try to improve segmentation. This idea could be called "integrated segmentation and recognition (ISR)". This second approach has hardly been studied in the literature but is likely to become an important topic for computer vision research over the next years.

1.3. Object Representations for Direct Recognition

In the following we will contrast two kinds of methods that try to recognize objects on the basis of features extracted directly from the raw unsegmented input image: feature constellation techniques and invariant feature techniques. Our goal is not to provide a comprehensive review but rather to just highlight some important ideas.

Feature constellation techniques represent a particular view of an object as a spatial constellation of localized image features\(^2\).\(^3\).\(^4\).\(^5\).\(^6\).\(^7\).\(^8\).\(^9\).\(^10\).\(^11\).\(^12\).\(^13\).\(^14\).\(^15\). During recognition, a similar constellation of image features is searched for in a novel input image. A goodness of fit criterion evaluates the match. These techniques vary in a number of aspects including the kinds of image features used in the representation and the algorithm for matching. In order to reach invariance with respect to pose changes, appearance changes, articulation, etc. these methods have to do one or several of the following: 1) represent very different views of an object with separate models, 2) model the variability in the appearance of individual features, 3) model variability in the spatial arrangement of the features. Depending on the number of degrees of freedom in the models, the matching procedure can be come very expensive to compute.

An early example of such a technique was proposed by Fischler and Elschlager\(^2\) who proposed a method similar to dynamic programming to match object models to the input image. The matching procedure seeks to minimize a cost function that is a sum of two terms. The first punishes dissimilarities of image features to the stored model features while the second penalizes geometric distortions. It has been shown that such a cost function can be derived from a statistical approach for modeling object appearance under a number of assumptions.

A widely used feature constellation technique is Elastic Graph Matching\(^4\).\(^6\).\(^13\) (EGM), which will be discussed in more detail below. There, the image features stored in an object model are typically convolution results of Gabor wavelets\(^6\) of different orientations and spatial scales. Matching is usually done in a heuristic coarse to fine strategy to make the optimization more tractable. EGM has been very successful in a number of difficult recognition applications, ranging from face detection and recognition\(^4\).\(^6\).\(^17\), over gesture recognition against cluttered backgrounds\(^18\).\(^13\), to the analysis of scenes with partially occluded objects\(^19\).\(^8\), just to name a few.

There has also been significant progress in employing probabilistic
frameworks\textsuperscript{7,10,12,20}. For example Lanitis \textit{et al.} achieve compact descriptions of face shape variations using a point distribution model and use discriminant analysis techniques to aid recognition\textsuperscript{7}.

Nelson and Selinger\textsuperscript{9,21} have used a manually constructed "perceptual grouping hierarchy" in a 3D object recognition system. At the lowest level, local edge elements are grouped to curve segments. These are in turn grouped to so-called context patches that describe a local arrangement of curve segments. An arrangement of these context patches then describes a particular view of an object. Finally, several such views are grouped together to obtain a full 3D object description. Their approach is very interesting because of the use of a hierarchical object description and it gives very good results for a 3D object recognition task for uniform backgrounds. However, it is not very robust with respect to cluttered backgrounds.

\textbf{Invariant feature techniques.} The origins of these approaches\textsuperscript{22,23,24} can be traced back to the work of Swain and Ballard\textsuperscript{25} on using color histograms for object recognition. In a sense, they represent an extreme stance for reaching invariant recognition. The idea is to extract a number of invariant features, i.e. features whose responses are supposed to stay the same under a number of object transformations like translation, scaling, rotation, and change of aspect. Early work in this direction focused on color histograms\textsuperscript{25} and ways to make them more robust with respect to illumination changes\textsuperscript{26,27}. More recently, researchers have also used a number of other more complex features. To this end, the responses of a potentially large number of filters of different scales and orientations are applied to the entire image and the responses are then pooled together.

Mel's SeeMore system\textsuperscript{22} simply adds all the responses of the same filter type for different scales, and orientations applied to the entire image. It evaluates 102 different feature channels, whose responses signal how frequent features of one type are observed in the image regardless of their position, orientation, or scale. Recognition is performed by comparing the 102 dimensional vectors using a nearest neighbor classifier. Similarly, Schiele and Crowely\textsuperscript{24} use a histogram representation. The responses of filters of one type but different scales, orientations, and positions are all combined into the same histogram bin. Recognition is done by comparing these histograms with statistical techniques.

These approaches have given impressive results for recognition on large object data bases. However, they do not perform very well for cluttered scenes. Typically, researchers use data bases with objects on homogeneous backgrounds, and experiments have shown that heavily cluttered backgrounds can have a strong negative impact on performance. The problem is that as soon as there is cluttered background or there are multiple objects in the image, the invariant feature representation will throw all local image features "into one pot" resulting in a super-position catastrophe. For example, the histogram representation is a super-position of contributions from the background and all objects in the scene from which it is no longer possible
to tell the individual sources if several objects can give rise to the same features. There is no way of “grouping” or “binding” together the features stemming from the same object.

Several attempts have been made to remedy this problem. The first restricts histogram computation to localized image regions. This approach works reasonably well if there are few objects in an otherwise empty scene but it is unclear whether this approach is sufficient to overcome the challenge of unfamiliar cluttered backgrounds without any prior assumptions about object sizes and shapes. The second is the use of highly specific feature detectors that are only triggered for very few objects. The idea is that if several such features are “triggered” this provides very strong evidence for the presence of the object. At the same time false positives are thought to be avoided through the high specificity of the feature detectors. To this date, it remains an open question how feature detectors with the needed specificity and the desired invariance characteristics can be found. A third approach is to introduce semi-local constraints between the invariant features extracted at interest points. This approach is quite related to the feature constellation techniques discussed above.

Considering that some feature constellation techniques use somewhat invariant features and that some invariant feature techniques consider spatial constraints among features it becomes clear that these approaches really lie on a continuum. The features extracted can have varying degrees of invariance to specific transformations (translation, rotation, scale, etc.) and there can be more or less tight constraints regarding their arrangement in the image. Just how much invariance of what kind is optimal for a specific recognition problem and how proper features can be learned is an open question. Similarly, the right level of complexity and specificity of features is an important unsolved problem. Inspiration regarding the answers to these difficult questions may come from the study of biological vision systems, that seem to use a hierarchy of feature detectors of increasing complexity and invariance properties.

2. Object Recognition with Labeled Feature Graphs

In the remainder of this chapter we will focus on a specific kind of feature constellation technique — Elastic Graph Matching (EGM). Elastic Graph Matching (EGM) is an object recognition architecture, which has been successfully applied to object recognition, face and gesture recognition, and the analysis of cluttered scenes. Although being motivated by a theory of neural information processing, EGM is similar to other elastic matching approaches. In EGM views of objects are represented as labeled graphs with an underlying two-dimensional topology. The nodes of a graph are labeled with a local image description. Edges of a graph are labeled with a distance vector. Sometimes, the graphs are chosen to have a simple grid-like topology, while sometimes nodes are placed on fiducial points (interest points, landmarks) on the object of interest as illustrated in Fig. 2.
Fig. 2. Representation of faces with labeled graphs. Left image: simple graph with grid-like topology. Right two images: Graphs with flexible topology where nodes are placed on fiducial points on the face. The first image shows a graph manually placed on the face during training. The second image shows the graph matched to a novel input image where the face has a slightly different pose and expression. Each node of the graph is labeled with a local image description, edges between nodes are labeled with distance vectors.

Elastic matching of a model graph to an image means to search for a set of node positions such that 1) the local image description attached to each node matches the image region around the position where the node is placed and 2) the graph is not distorted too much. This is expressed as the optimization of a goodness-of-match function with a term for the similarity of the features and a cost term punishing graph distortions.

2.1. Gabor Wavelet Transform

As a local image description for labeling the nodes of a graph a Gabor jet is usually used. This is a vector of responses of Gabor wavelets (Fig. 3):

$$
\psi_k(x) = \frac{|k|^2}{\sigma^2} \exp \left(-\frac{|k|^2 |x|^2}{2\sigma^2}\right) \left[ \exp \left(ik^T x\right) - \exp \left(-\frac{\sigma^2}{2}\right) \right].
$$

These wavelets represent a plane wave with wave vector $k$ restricted by a Gaussian envelope function of width $\sigma$; $x$ denotes the two-dimensional image location. The responses of several such filters with different sizes and orientations, parameterized by different $k$, constitute a jet:

$$
k_{\nu \mu} = k_{\nu} \frac{\cos(\alpha_{\mu})}{\sin(\alpha_{\mu})} \quad \text{with} \quad k_{\nu} = k_{max}\sqrt{f^\nu}, \quad \alpha_{\mu} = \frac{\mu\pi}{D}.
$$

Here, the index $\nu \in \{0, \ldots, L - 1\}$ labels the $L$ different spatial scales or frequency levels, and $\mu \in \{0, \ldots, D - 1\}$ labels the $D$ different orientations. The value $f$ is the spacing factor between kernels in the frequency domain and $k_{max}$ is the maximum wave number. A jet is a complex vector composed of the $L \times D$ complex filter responses $c_j$, where the index $j$ is a double index running over different orientations and scales. The $c_j$ are represented as absolute value $a_j$ and phase $\phi_j$: $c_j = a_j e^{i\phi_j}$. 
Interestingly, the visual system of primates seems to extract Gabor-like feature at its early processing stages, which may be related to the statistical structure of natural images. The absolute responses of such filters are quite robust to small rotations and translations which introduces a certain amount of invariance into the representation.

Two similarity functions are commonly used for comparing Gabor jets:

\[ S_{\text{abs}}(J, J') = \frac{\sum_j a_j a_j'}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}}, \]  
\[ S_{\text{pha}}(J, J') = \frac{1}{2} \left( 1 + \frac{\sum_j a_j a_j' \cos(\phi_j - \phi_j')}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}} \right). \]

\( S_{\text{abs}} \) uses only magnitudes of the complex filter responses. It has the form of a normalized scalar product. \( S_{\text{pha}} \) uses the magnitude and phase of the complex filter responses. Both functions yield similarity values between zero and one. While \( S_{\text{abs}}(J, J') \) slowly changes when \( J' \) is “moved” across the image, \( S_{\text{pha}}(J, J') \) varies very rapidly because the phases of filter responses change significantly on a spatial scale corresponding to the wave-vector \( \mathbf{k} \) of strongly responding kernels. This difference between \( S_{\text{abs}} \) and \( S_{\text{pha}} \) can be understood by comparing the real part of the convolution result with the absolute value of the convolution result in Fig. 3. The absolute response varies much more smoothly.

2.2. Elastic Matching Process and Recognition

Let us consider the case of having a set of objects we may want to recognize and we have created one model graph for each object. For recognition, i.e. finding the best model for a given input image, all object models are matched sequentially. For each we need to find the minimum of the similarity or goodness-of-fit function. The similarity of a graph \( G \) matched to the image \( I \) at node positions \( \mathbf{x}_n \) may be defined
as simply the average of the similarities of the individual jets:

\[ S^G(G, I) = \frac{1}{N} \sum_{n=1}^{N} S_{abs}(J^n, J(x_n)) . \]

Often it is useful to introduce an additional cost term to punish geometric distortions of the graph. Also, allowing for different weightings of different nodes in the evaluation of a match can be beneficial.

After matching all models to the image, the model obtaining the highest similarity represents the classification result. For object detection, thresholds for all object models must be learned. An object is detected if the matching process yields a score that is above threshold.

During matching we need to simultaneously optimize all degrees of freedom of the deformable object model. If object models with many degrees of freedom are allowed (such as shifting across the image, independent scaling in x- and y-direction, rotation in plane, various non-rigid deformations) then matching time can become an issue making an exhaustive search over the parameter space unfeasible. To combat the adverse effect of many degrees of freedom in the object model, a coarse to fine matching scheme can be used. For example, a strategy may be to first try to determine the position, orientation, and scale only very coarsely, and then to refine the estimate in subsequent matching steps. In a final step, “diffusion” of individual nodes may be allowed to compensate for small “irregular” deformations of the object in the current input image.

2.3. Bunch Graphs

The bunch graph idea\(^6\) was developed to model variability in feature appearance. The natural variability in the jets of corresponding points in several images of the same object or class of objects (e.g., several left eyes of different persons) is captured by labeling each node of a graph with a set or “bunch” of jets (short: bunch jet) extracted from corresponding points in different example images rather than only with a single jet. This method can also be used to model variability due to different backgrounds\(^10\). For the matching process, we need a similarity function comparing the set of jets attached to each node of the bunch graph with local image information. The similarity of a bunch jet \(B^n\) comprising \(K\) jets at a node \(n\) to a single jet \(J(x)\) taken at a point \(x\) in an image is defined to be the maximum of the similarities of the \(K\) individual jets:

\[ S^B_{abs}(B^n, J(x)) = \max_k \{S_{abs}(J^n(k), J(x)) , k = 1, \ldots, K\} . \]

\(S^B_{abs}\) is defined analogously. When a graph \(G\) with \(N\) nodes is matched to an image \(I\) at the node positions \(x_n\), its total similarity is given by the average of the node similarities:

\[ S^G_{abs}(G, I) = \frac{1}{N} \sum_{n=1}^{N} S^B_{abs}(B^n, J(x_n)) . \]
Again, we define $S_{\text{pha}}^G$ in an analogous manner.

Modeling the variability in feature appearance by storing a set of prototypes is obviously not the only plausible method and a number of other approaches, e.g. based on principal component analysis, can be used as well.

2.4. Application to Face Recognition

Automatic face recognition is an important problem with applications ranging from human computer interaction to surveillance and security. A number of studies have explored the use of EGM for face finding and pose estimation\cite{17}, identification\cite{4,6,33,34}, facial expression recognition, gender estimation and more. Figure 2 shows how a face graph may be constructed from a training image so that the graph can then be used to recognize the face in a novel view. While early systems used simple graphs with a grid topology, substantial improvements can be obtained by using graphs whose nodes are placed on specific fiducial points (or landmarks) on the face.

The bunch graph technique can been used to represent a generic face model in a single graph. To this end, jets from corresponding points (say, several left eyes) of several training images from different persons are attached to a single node. The underlying assumption is that feature variation, e.g. variability in the appearance of left eyes, is approximately independent of relational variability, i.e., variability in the relative position of left eyes with respect to other features of the face. The great advantage of this method is that it allows to separate the matching process, where the facial landmarks are found, from the identification process, where features from the found facial landmarks are compared to a gallery of known persons. A number of other extensions have been made to the basic method described so far that are beyond the scope of this chapter\cite{6,33,34}.

3. Combining Multiple Feature Types

While early versions of EGM only used a single shape or texture feature type such as Gabor or Mallat filters, it has also been extended for handling multiple feature types\cite{8,13}. There are several ways of extending EGM for multiple feature types (Fig. 4). First, the feature types used may differ from node to node. For example, a model graph describing a face could employ edge features at the outline of the face and texture features at the inside. Second, all nodes may be labeled with a combination of different feature types which is identical for all nodes. Third, nodes may be labeled with combinations of feature types which may differ from node to node. Here we focus on the second scheme.


In order to allow for multiple feature types at a node we introduce the notion of a compound jet. In a compound jet, several local image descriptions are simply con-
Fig. 4. Combining different feature types in graph-based object representations can be done in a number of ways. All nodes may be labeled with just a single (a) or multiple (c) features that are identical for all nodes. Or all nodes may be allowed to be labeled with a single (b) or multiple (d) feature types that are different for all nodes.

catenated. When similarities between two compound jets $J$ and $J'$ are computed, first the similarities of their corresponding constituents are considered. These are then combined by computing a weighted average:

$$S(J, J') = \sum_{\mathcal{F}} w_{\mathcal{F}} S_{\mathcal{F}}(J_{\mathcal{F}}, J'_{\mathcal{F}}), \quad \sum_{\mathcal{F}} w_{\mathcal{F}} = 1.$$

(8)

The $S_{\mathcal{F}}$ are similarity functions for comparing jets of a particular feature type $\mathcal{F}$. Figure 5 compares different kinds of features for finding image point correspondences. Combining a number of complementary features into a compound jet can improve the overall performance.

3.2. Compound Bunch Graphs.

One can easily extend the bunch graph idea to graphs with compound jets as well. Compound bunch graphs are constructed in the following manner:

- For every node $n$ of a particular graph of a view of an object the jets of the same feature type $\mathcal{F}$ extracted from $K$ different training images are integrated into a bunch jet $B_n^\mathcal{F} = \{J_n^\mathcal{F}(1), \ldots, J_n^\mathcal{F}(K)\}$. With these bunch jets as node labels one bunch graph is created for every feature type, whose geometry is averaged from the constituent graphs.
- The bunch jets $B_n^\mathcal{F}$ of different feature types but from the same node $n$ of the same view are concatenated to a compound bunch jet $B_n^\mathcal{G}$. With these compound bunch jets as node labels, a compound bunch graph is created, whose geometry is identical to that of the constituent bunch graphs of different feature types.
For evaluating the similarity $S^B(B^n, J(x))$ between a compound bunch jet $B^n$ and a compound jet $J(x)$ extracted at a particular location $x$ a weighted average of contributions $S_F^B$ stemming from different feature types is used:

$$S^B(B^n, J(x)) = \sum_{F} w_F S_F^B(B^n_F, J_F(x)), \quad \sum_{F} w_F = 1.$$  \hspace{1cm} \text{(9)}$$

The functions $S_F^B$ compare a bunch jet $B^n_F$ to a simple jet $J_F$ of the same feature type $F$ in analogy to (6):

$$S_F^B(B^n_F, J_F(x)) = \max_k \{S_F(J^n_F(k), J_F(x)), \quad k = 1, \ldots, K\}.$$  \hspace{1cm} \text{(10)}$$

The similarity functions $S_F$ directly compare two jets of a particular feature type $F$. The similarity of a compound bunch graph $G$ with node positions $x_n$ to an image $I$ is defined as the average of the similarities of all $N$ nodes, in analogy to the previous section:

$$S^G(G, I) = \frac{1}{N} \sum_n S^B(B^n, J(x_n)).$$  \hspace{1cm} \text{(11)}$$

Again, an additional topological cost term can be introduced to punish large geometric distortions, and nodes can be given different weights.
Fig. 6. Person-independent hand posture recognition against complex backgrounds is a challenging problem. Due to differences in hand anatomy and individual performance of a gesture there is substantial variability in the appearance of the hand. In addition, rings, etc. may be present and long sleeves may introduce partial occlusions.

3.3. Application to Hand Gesture Recognition

Gestures are a powerful means of communication among humans. In fact, gesturing is so deeply rooted in our communication that people often continue gesturing when speaking on the telephone. Consequently, the automatic recognition of human gestures is an important topic in human computer interaction. The recognition of hand postures is an important ingredient for building gesture-based interfaces. Although there have been attempts to recognize gestures and sign language without recognizing the specific posture of the gesturing hand it is clear that these approaches are limited. For example, in American Sign Language the same movement can have very different meanings depending on the posture of the gesturing hand.

A number of requirements make hand posture recognition a particularly challenging task. First, if a system is intended to be person independent it must cope with shape variability due to different hand anatomy or different performance of the same gesture by different persons. Second, if the system is intended to work in unconstrained environments it must be able to deal with cluttered backgrounds, making bottom-up segmentation of the gesturing hand difficult. Along similar lines, it should not be necessary for subjects to take off rings etc. before interacting with the system but they should be able to just “come as they are”. Other desiderata are real-time performance and robustness to varying lighting conditions.

Using a deformable feature graph approach such as EGM is a very good starting point for this challenging recognition task because 1) it has an inherent ability to handle geometric deformations, 2) it does not require prior segmentation of the gesturing hand, and 3) it can elegantly handle variability in feature appearance with the bunch graph method.

We have used EGM with multiple feature types to address this difficult problem\textsuperscript{13}. In particular, we added two kinds of color features to the local im-
age descriptions at the graph nodes. The first additional feature was a local average of the color in the vicinity of the node (in HSI color space), the second additional feature was a Gabor jet extracted from a preprocessed version of the image that emphasized skin tones. The image database consisted of more than 1000 color images of twelve hand postures performed by 19 persons against simple and complex backgrounds with varying amount of skin color (Fig. 6). The images of three subjects signing against uniform light and dark backgrounds formed the training set, giving 6 training images per posture, the remaining images formed the test set. For the images in the training set, we constructed graphs of 15 nodes. The nodes were manually placed at anatomically significant points. Model graphs extracted from the training images were combined into 12 compound bunch graphs (one for every posture) as described above.

The matching procedure had 4 steps: coarse positioning of the graph by scanning it across the image in steps of five pixels; estimating rotation in plane (+/- 15 degrees) while refining the position estimate; estimating scale changes (up to 20 percent); local diffusion of individual nodes to fine tune the matching. An example of a successful match is given in Fig. 7. The nodes usually find their proper positions during the matching, even if the background is very cluttered or contains large regions of skin color.

3.4. Evaluation

Cross-runs on a test set of 604 images taken against uniform light or dark background and 338 images against complex backgrounds were performed. The results are summarized in Tab. 1. The weighting factors $w_F$ between the three different feature types were systematically varied in an exhaustive manner considering all relative weightings of the types 1:0:0, 4:1:0, 3:2:0, 3:1:1, 2:2:1, 2:1:1, and 1:1:1, and system performance on the test set was measured. The best result was obtained for $w_{Gabor} = w_{ColorGabor} = 25\%$, $w_{Color} = 50\%$. Since the weighting of cues only intro-
roduces 2 free parameters, we decided not to use a separate validation set. It turns out that a proper combination of the three feature types outperforms any of them alone. For example, the error for the best combination found is less than half as big as that for using Gabor features alone. Performance does not depend critically on the precise weighting between the features if all are being used. The recognition rates vary smoothly with the weighting and there is a large plateau of weightings yielding recognition rates higher than any of the feature types alone could account for. If only Gabor and colorGabor features but not color features are used, performance is still very good, suggesting that the color features are not essential for the system's performance.

An analysis of the system's errors reveals that strong shape variations due to either differences in performance of a posture, different hand anatomy, strong rotation in depth, or combinations of these account for practically all errors. Interestingly, it was found that a greater flexibility of the graph during the final matching step was not effective in reducing the number of errors. The reason seems to be that the geometric distortions are not of a random nature which would be well modeled by a diffusion process, but instead show high correlations. This is due to the hand's kinematics which only allow for certain correlated movements of nodes. A more refined modeling of likely graph distortions thus seems like a good avenue for future research. In its extreme form, this would amount to a full kinematic model of the hand, but this may not always be necessary.

4. Analysis of Cluttered Scenes with Partial Occlusions

The analysis of cluttered scenes such as the one depicted in Fig. 8 is one of the most challenging problems in computer vision. Massive background clutter makes it virtually impossible to properly segment the scene into regions corresponding to the objects in a bottom-up fashion, i.e. without using explicit object models. Furthermore, occlusions can render large portions of an object invisible. The ease and efficiency with which the human visual system solves this problem belies the fundamental computational challenges that this task poses. It turns out that for this kind of task, the use of deformable feature graphs as the object representation is again a very good choice. In particular, the use of local image features allows to explicitly discover and model partial occlusions by other objects\textsuperscript{19,8}. In this Section
Object Recognition with Deformable Feature Graphs

Fig. 8. Analysis of cluttered scenes. Given a number of object models in the form of deformable feature graphs (left), the goal is to analyze a cluttered scene with partially occluded objects (middle) and to return a description of the identities, locations, and occlusion relations of all objects in the scene (right).

we will outline an extension of the methods discussed so far to stereo image pairs and apply them to the analysis of cluttered scenes of multiple objects occluding one another.

4.1. Extension of EGM to Stereo Image Pairs

Most work on stereo vision has as its goal the estimation of 3-D structure from 2-D images. In contrast, we are interested in improving object recognition through the use of stereo processing. The essence of stereo vision is solving the correspondence problem, i.e. determining which point in the left and right image correspond to the same physical point. Most stereo algorithms consider fairly simple image features when trying to establish the correspondence relations, but in principle features of arbitrary complexity can be used for this purpose — or even entire objects. The basic ideas in our case is that correct object matches in the left and right image are not independent but related through the epipolar geometry of the stereo system. This induces a constraint that can be used to discard false matches if they fall on inconsistent locations in the left and right image (Fig 9). As a side product, the distance of correct matches can be estimated. If absolute size information about the objects is available, then the distance information must in turn be related to the apparent size of the object in the images — an additional constraint that can be used to reject false matches.

4.2. Scene Analysis

Our goal for object recognition in the presence of partial occlusions is to explicitly detect and discount occlusions when evaluating the quality of a match. One idea for doing so works in a front-to-back fashion. Object models are matched to the image and the object models obtaining good matches enter a set of candidates. Of these candidates, the front-most object, i.e. the one closest to the camera is determined
Fig. 9. Simultaneous recognition in stereo image pairs. The exploitation of epipolar constraints can aid recognition in difficult situations with clutter and partial occlusions. a: Stereo image pair of cluttered scene with partially occluded object “duck”. b: Simultaneous matching of object model duck in both images subject to epipolar constraints manages to correctly locate the object despite the partial occlusion. c: In contrast, independent matching in the left and right image is more prone to errors. In this case, it fails for the right image.

on the basis of stereo disparity, match similarity and possibly other cues. The frontmost candidate is then accepted and removed from the candidate list. Since this object is the closest candidate, it cannot be occluded by other objects other than those recognized on previous iterations. However, it may itself be occluding some objects. To take this into account, the image area covered by the accepted object is labeled as occluded. Now, the matches for other objects are re-evaluated by discounting any nodes that fall into the occluded area. This may lead to additional objects entering the set of candidates. The algorithm now repeats the selection of the front-most candidate until all no more candidates remain.

4.3. Evaluation

To test the method we have recorded a data base of stereo image pairs of cluttered scenes composed of 18 different objects in front of uniform and complex backgrounds. There is considerable variation in object shape and size, but there are also objects with nearly identical shapes differing only in color and surface texture. The training images comprise one stereo image pair of each object in front of a uniform blue cloth background. The test images comprise scenes composed of 2-5 objects in front of one simple background (80 scenes) and four complex backgrounds (160 scenes). In systematic experiments evaluating the above scene analysis methods we found that the extension to recognition in stereo image pairs and the use of multiple feature types in the form of compound jets can lead to quite dramatic performance
improvements. In particular, for a fixed level of false positives, a version of the system using both extensions reached roughly twice as many correctly recognized objects as a monocular version of the system employing Gabor features only.

An important lesson from these experiments is that the representation of objects as constellations of localized features allows to explicitly discount partial occlusions — a property that is quite appealing. Representations based on “larger” features covering big parts of the object appear less suited for this task because they are more prone to be corrupted by partial occlusions. It may be possible to develop hierarchical architectures that integrate features at different scales and combine the advantages of more localized and more global features.

Progress in the area of scene analysis has been comparatively slow and we feel that a reason for this is that researchers have usually tried to decompose the problem in the wrong way. A specific problem may have been a frequent insistence on treating segmentation and recognition as separate problems, to be solved sequentially. We believe that this approach is inherently problematic and this work is to be seen as an initial step towards finding an integrated solution for both problems. In the method described above, which is an extension to the work by Wiskott, segmentation is purely model driven, i.e. if an object has been recognized, the corresponding region of the image is labeled as a segment. This need not be the case and current work explores ways of also integrating bottom-up segmentation cues.

5. Discussion

This chapter discussed object recognition approaches based on deformable feature graphs — a special case of what we have called direct recognition approaches, that try to recognize objects on the basis of features directly extracted from unsegmented images. Deformable feature graphs represent particular views of objects as spatial constellations of certain image features. Such an object representation has a number of benefits. First, it allows recognition without prior segmentation of the object of interest. Second, robustness to small variations in appearance tends to be very good
if features are chosen properly, i.e. there tends to be good generalization. Third, also depending on the features used and the type of recognition scheme employed, it can be quite fast, lending itself to real time implementations with current off-the-shelf hardware. Fourth, it allows for explicitly modeling and discounting of partial occlusions. Fifth, the methods described in here have already proved their potential in very different applications ranging from face detection and recognition, over hand gesture recognition, to the analysis of cluttered scenes of partly occluded objects.

Overall, these benefits ensure continued interest in these methods. However, there are many fundamental challenges still lying ahead.

First, it is desirable to automate the learning process as much as possible. In the gesture recognition example it was necessary to manually label the anatomical landmarks in the training images. How can object models be built without precise feature correspondences but only a coarse localization of the object in the training images? Even better would be a system for which only a label (object is present/not present) has to specified for each image. Finally, it should even be possible to discover object categories without any explicit supervision. Human infants spontaneously form categories of visual objects without explicit instruction or supervision. A machine vision system that simply observes its visual environment and learns to cluster objects into the right categories in a completely unsupervised fashion would represent a quantum leap for the field.

Second, how can large object data bases be searched efficiently? Methods that engage in elaborate matching procedures for thousands of objects in a sequential fashion seem rather inappropriate. An idea that could lead to benefits in recognition speed is to share partial object descriptions among several models and to search for objects in a bottom-up fashion, where the detection of individual features triggers a more elaborate analysis for all matching candidate objects. In this context, how can partial object descriptions be learned that lend themselves to efficient sharing between objects while still allowing effective discrimination? There may be large benefits in employing hierarchical object representations using features at several levels of complexity. Fairly unspecific lower level features can be shared between object models most effectively while more specific higher level features give the necessary discriminative power for recognition. What are good frameworks for learning such representations?

Third, how can recognition and segmentation be integrated within a closed feedback loop to allow for efficient recognition in the presence of clutter and occlusions? These questions are likely to remain of great interest for years to come. Some inspiration may come from the study of biological vision systems, which have mastered object recognition better than any machine vision system that man has ever created.

Acknowledgments

This work was supported by the National Science Foundation under grants NSF-0208451 and NSF-0233200. We thank Christoph von der Malsburg for comments
on parts of this chapter.

References