Contact Author: Tomaso Poggio

Learning to See

Tomaso Poggio
Center for Biological and Computational Learning
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: tp@ai.mit.edu

Abstract

Learning is becoming the central problem in trying to understand intelligence and in trying to develop intelligent machines. It may even be argued that Computer Science and Artificial Intelligence need a new Turing test which should clearly position the problem of learning at the core of any attempt at understanding and synthesizing intelligence. After all, we are just beginning to realize how "intelligent" lower animals are and how complex are the problems that our senses routinely solve. We also realize how intractable the problem of producing software is and how much of it would be needed in order to replicate even the simplest aspects of intelligence.

It is not surprising then that the hot new frontier of computer vision is learning. The challenge is to develop machines that learn to perform visual tasks such as visual inspection and visual recognition from a set of training examples. This reflects an overall trend – to make intelligent systems that do not need to be fully and painfully programmed. It is the only way to develop vision systems that are robust and easy to use in many different tasks.

In other words, computers will have to be much more like our brain, learning to see rather than being programmed to see. Thus, learning is also the frontier where computer vision will meet the science of biological vision. Biological visual system are more robust and flexible than machine vision mainly because they continuously adapt and learn from experience. In fact, not only computer vision but also neuroscience is just about to realize the key role of learning. I venture to say that a new paradigm is emerging in visual neuroscience: plasticity of cortical neurons and their connectivity is *becoming the key to understanding how visual cortex works. Thus the problem of learning is now at the very core of the two fields of computer vision and of visual neuroscience.*

In both areas, progress in understanding the role and the mechanisms of learning has been very rapid. What has happened, however, is just the beginning of a veritable revolution in the development of novel artificial vision systems and in our understanding of human vision. The issues at stake are engineering issues as well as scientific ones. On the engineering side the possibility to build vision systems that can adapt to different tasks can have enormous impact in many areas such as automatic inspection, image processing, video editing, virtual reality, multimedia databases, computer graphics and man-machines interfaces. On the biological side, our present understanding of how the cortex works may radically change if adaption and learning turn out to play a key role. Instead of the hardwired cortical structures implied by the classical Hubel and Wiesel work, we may be confronted with neuron properties and connectivities that change as a function of visual experience over time scales of a few minutes or seconds. Psychophysical and physiological data point already in this direction. This would be a fundamental paradigm shift which may finally unlock the secrets of the functions and the mechanisms of the cortex.

The article will start by showing how several vision and graphics tasks can be formulated in terms of training a learning box with appropriate sets of examples – that is input-output vector pairs. The article will then sketch the underlying theory which connects function approximation techniques, neural network architectures and statistical methods. Next, a few examples will be described. At MIT we have developed systems that learn to recognize objects, in particular faces, systems that learn to find specific objects in cluttered scenes, software that learns to draw cartoon characters from an artist's drawings and algorithms that learn to synthesize novel image sequences from a few real pictures and thereby promise to achieve extremely high compression in applications such as video conference and video e-mail. A novel image representation based on pixelwise correspondence which is the key to all these successfull learning systems will be briefly introduced. Finally, the article will speculate on the implications of this research for how the brain ought to work. In particular, it will describe some very recent physiological data which support the learning models we have developed and provide a glimpse of how 3D objects are represented in the visual cortex.

1 Introduction

Seeing is intelligence. From a computational point of view the problem of vision is one of the most complex in the whole problem of understanding the mind, which is itself one of the four greatest problems confronting science today. From a biological

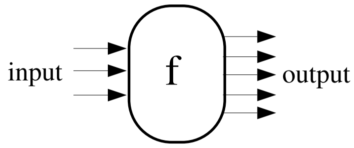


Figure 1: *In the learning-from-examples paradigm, we learn a function f from input-output pairs $(\mathbf{x}_i, \mathbf{y}_i)$ called the training set.*

point of view the visual system is one of the most amazing parts of the brain. A large part of the cortex is dedicated to vision, proportionally more than to other aspects of intelligence such as language.

Just in the last four years or so both computer vision – which attempts to build machines that see – and visual neuroscience – which aims to understand how our visual system works – are witnessing a paradigm change in their approaches. Instead of trying to understand all the details of the visual cortex, visual neuroscience is beginning to focus on the mechanisms which allow the cortex to adapt its circuitry and learn a new task. Instead of building a hardwired machine or program to solve a specific visual task, computer vision is trying to develop systems that can be trained with examples of any of a number of visual tasks.

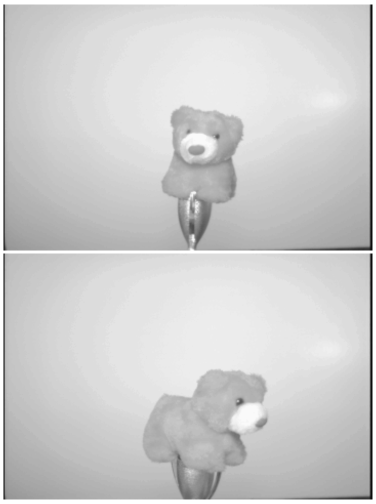
In the following I will explain more precisely the meaning for vision of the powerful metaphor of learning-from-examples.

2 Visual tasks and learning boxes

In this chapter we will concentrate on one aspect of learning: *supervised learning*. Supervised learning – or *learning-from-examples* – refers to a system that is trained, instead of programmed – by a set of examples, that is input-output pairs $(\mathbf{x}_i, \mathbf{y}_i)$ as sketched in Fig. 1. At run-time the trained system – our "learning box – will hopefully provide a correct output for a new input not contained in the training set.

What does this learning-from-example framework imply for vision tasks? Typically, in computer vision tasks the input to the learning box of figure 1 is an image or an image sequence and the output is a set of parametrs estimated from the image. For instance, in the ALVIN system, developed by Dean Pomerleau at CMU for the task of driving a car, the input is a series of images of the road and the output is degrees of steering (figure?). These computer vision tasks are typically *image analysis* tasks.

In the learning-from-examples approach to the problem of image analysis, a learning module is trained to associate input images to output parameters – such as the classification label for the object in the image and possibly pose and expression parameters associated with it (see figure 2). After training with a set of images and corresponding parameters, the network is expected to generalize correctly; that is, to classify new views of the same object, recognize another object of the same class, or estimate its pose/expression parameters.



$\Rightarrow$ (bear, 0° view)

$\Rightarrow$ (bear, 45° view)

Figure 2: *The problem of computer vision from a learning-from examples perspective: given an image, the goal is to recognize the object and to estimate its pose. This is representative for the class of image analysis problems. The learning-from-examples framework suggests immediately a view-based approach to the problem: the training stage relies on views and not on 3D models. From Girosi (unpublished).*

(bear, 0° view) $\Rightarrow$

(bear, 45° view) $\Rightarrow$

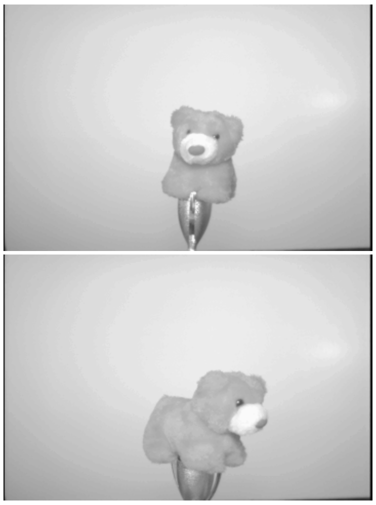


Figure 3: *The problem of computer graphics from a learning-from examples perspective: given a pose, the goal is to generate the corresponding image. From Girosi (unpublished).*

The analysis problems of estimating object labels as well as other parametrs from images is the problem of vision. It is the *inverse* of the classical problem of classical optics and modern computer graphics: how to synthesize images corresponding to given 3D surfaces as a function of parameters such as direction of illuminant and position of the camera. In our learning-from examples framework a learning module is used to associate input parameters to output images (see figure 3). This module synthesizes new images and represents a rather unconventional approach to computer graphics. Traditional 3D computer graphics simulates the physics of the world by building 3D models, transforming them in 3D space, simulating their physical properties and finally rendering them – by simulating geometrical optics. The learning-from-examples paradigm suggests a rather different approach: take several real images of a 3D object – possibly a non rigid one – and create new images by generalizing from those views, under the control of appropriate pose and expression parameters, assigned by the user during the training phase. Our previous discussion should make clear that this metaphor – of learning-from-examples – is equivalent to performing multidimensional interpolation or approximation, in this case a form of multidimensional morphing.

Clearly these view-based approaches to image analysis – that is the traditional domain for computer vision – and to image synthesis – the domain of computer graphics – cannot be seen as substitutes for the more traditional and physically more "correct" approaches. However, the learning-from-examples approach to vision and graphics may represent an effective shortcut to several problems and, furthermore, has a profound appeal in terms of describing what biological visual systems may do.

In the following, I will sketch projects in our group using the learning-from-examples approach for the analysis and for the synthesis of face images. I will also mention how the combination of the analysis and synthesis modules may provide a technique potentially useful for videoconferencing or video e-mail systems that aim to achieve very-low bit rates.

3 Learning to see: examples

I start by describing the basic learning module used in our approach. This can be regarded as a general purpose learning network. It may be called a neural network, though its true roots are in the theory of multivariate function approximation.

3.1 The learning module

One way to set the problem of learning-from-examples in a mathematically well-founded framework – and one which is familiar to computer vision – is the following. Supervised learning can be regarded as the regression problem of interpolating or approximating a multivariate function from sparse data. The data are the examples. Generalization means estimating the value of the function for points in the input space in which data are not available.

Once the ill-posed problem of learning-from-examples has been formulated as a problem of function approximation, a modern mathematical approach to solving it is *regularization*. Regularization imposes a constraint of smoothness on the space of approximating functions. The functional regularization approach can also be formulated within a statistical and Bayesian framework.

A key result, first derived in the context of learning by Poggio and Girosi at MIT, is that under rather general conditions the solution of the regularization formulation of the approximation problem can be expressed as the linear combination of basis functions, centered on the data points and depending on the input $\mathbf{x}$. The form of the basis function G depends on the specific smoothness prior:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i) + p(\mathbf{x}), \tag{1}$$

where G is the basis function and $p(\mathbf{x})$ is a polynomial term that is often not needed.

As observed by Poggio and Girosi and for the special case of Radial Basis Functions by Broomhead and Lowe in England, the solution provided by the equation can always be rewritten as a network with one hidden layer containing as many units as examples in the training set (see figure 4). We call these networks Regularization Networks (RN). The coefficients c_i that represent the "weights" of the connections to the output are "learned" by minimizing the training error over the training set The scalar output case described so far can be easily generalized to the multi-output case, that is to the approximation of vector fields (see figure 5).

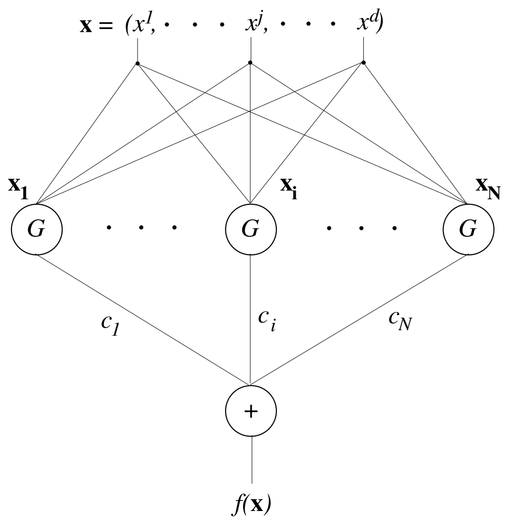


Figure 4: *A one-output Regularization Network. The input vector* $\mathbf{x}$ *is d-dimensional, there are* N *hidden units, one for each example* $\mathbf{x}_i$*, and the output is a scalar function* $f(\mathbf{x})$*.*

An interesting special case arises when the basis functions are radial. Legal basis functions G include the Gaussian, the multiquadric, thin-plate splines and others. In the Gaussian case, these RBF networks consist of units each tuned to one of the

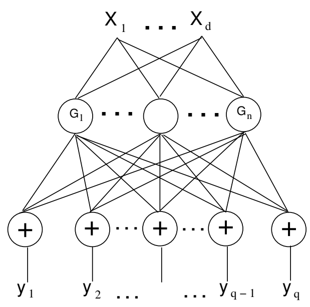


Figure 5: *The most general Regularization Network with one hidden layer and vector output. In the analysis case the network will have images as inputs (represented as ordered values of shape (e.g. x, y) and texture (e.g. color) values associated with each pixel) and pose-expression parameters as outputs; in the synthesis case inputs are pose and other parameters and outputs are images.*

examples with a bell-shaped activation curve. In the limit of very small variance of the Gaussian basis functions, RBF networks become look-up tables. Thus

- Each "unit" computes the distance $||\mathbf{x} - \mathbf{x}_i||$ of the input vector $\mathbf{x}$ from its center $\mathbf{x}_i$ (that is the dissimilarity of the input and the example stored in that unit) and
- applies the function G to the distance value, i.e. it computes the function $G(||\mathbf{x} - \mathbf{x}_i||)$
- in the limiting case of G being a very narrow Gaussian, the network becomes a *look-up* table
- centers are like *templates*

"Vanilla" RBF networks can be generalized to the case in which there are fewer units than data and the dissimilarity measure is more general than euclidean distance. These generalized RBF networks have sometimes been called HyperBF networks.

3.2 A computer vision task: estimating pose and expression

Given a set of example images labeled in pose and expression space (figure 6), a learning network as described above can be trained to estimate the pose-expression parameters of new images of the same person. The input to the network, or our image representation, is not the image itself but the associated *shape vector* that is computed from the image. We will explain it later. For the moment, let us think of it as a representation of (x, y) locations of facial features. Output are in this case three parameters, representing the amount of smile and rotation (left-right and up-down) of the face. The network is constructed is trained from eight examples images, shown in figure 6. After "training", the network can "generalize" to new images of the same person, estimating the associated rotation and smile parameters. Examples of input images and their estimated parameters are shown in figure 7.

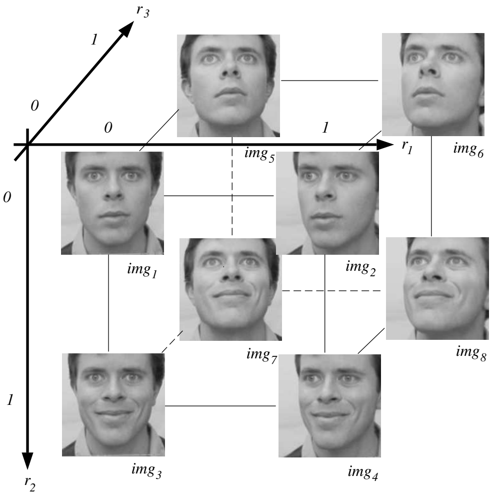


Figure 6: *In our demonstrations of the example-based approach to image analysis and synthesis, the example images* img_1 *through* img_8 *are placed in a 3D rotation-expression parameter space* (r_1, r_2, r_3)*, here at the corners of the unit cube. For analysis, the network learns the mapping from images (inputs) to parameter space (output). For synthesis, we synthesize a network that learns the inverse mapping, that is the mapping from the parameter space to images. The three axes of the parameter space correspond to smile, rotation left-to-right and rotation up-down.*

.

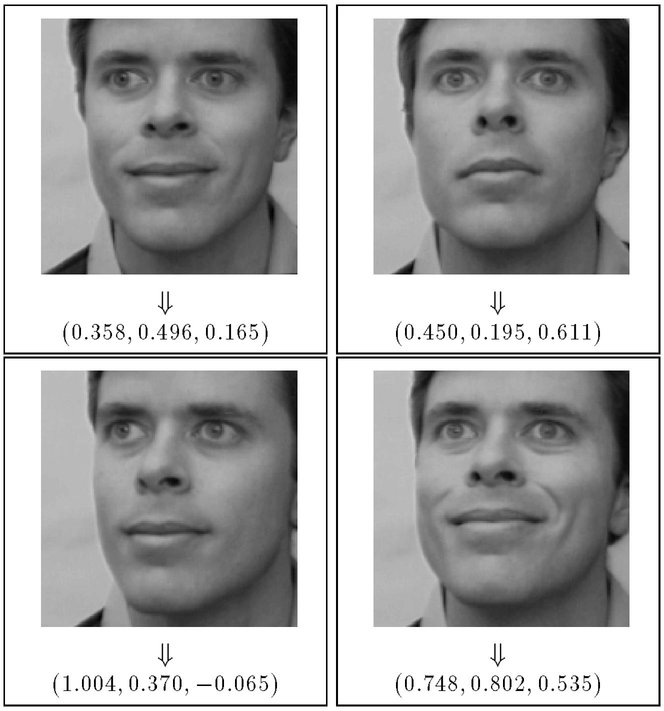


Figure 7: *The analysis network trained on the eight examples of figure 6 estimates the two rotation and expression parameters for a novel image. The figure shows the network output* (r_1, r_2, r_3) *for four novel images that are different from the examples in the training set.*

3.3 A computer graphics task: synthesizing novel images

In this section we discuss an example-based technique for generating images of non-rigid 3D objects as a function of input parameters. This image synthesis task – a computer graphics task – is the inverse of the problem of image analysis that we have discussed in the previous section.

The input space is hand-crafted by the user to represent desired degrees of control over the images of the 3D object. After assigning example images of the object to positions in parameter space, novel images are synthesized by using learning networks of the same type we have described for the analysis problem. Here we will demonstrate the technique on images of faces with input parameters such as pose, facial expression, and person identity, but other parameters may also be used.

The first step is to find pixelwise correspondence, that is *vectorize* the images, as we had described above for the analysis case. Once the example images are vectorized, the learning stage can take place. Inputs are the pose-expression parameters, while the outputs are the vectorized images, that is the shape and texture vectors associated to each image.

To train the synthesis network, the user

1. associates each example image to an appropriate position in the parameter space,
2. uses a multidimensional Regularization Network to synthesize the mapping between inputs and outputs, and
3. displays, that is renders, the vectorized image as a grey-level or color image.

After the learning stage the "trained" network can synthesize a new image for any reasonable pose input vector given to the network. The synthesis network has typically a small number of inputs – the pose parameters – and a large number of outputs, each corresponding to one attribute of either shape or texture of a pixel in the image.

Figure 6 shows eight examples of a face and their assigned pose parameters in a 3-dimensional input space. Figure 8 shows some pairs of (r_1, r_2, r_3) parameters and the color image produced by the synthesis module.

3.4 Image compression: video e-mail at 3 bytes per frame

The analysis and the synthesis network can each be used independently for a variety of applications. The analysis network is not limited to face images and can be regarded as a *trainable man-machine interface* that could perform customizable gesture recognition and thereby drive a computer interface. The synthesis network is an alternative way to do graphics and animation that extends to real images the approach already demonstrated for line drawings by Poggio and Brunelli at IRST and by Librande of nFX Corporation in Santa Clara, an Internet company

A particularly interesting application of the analysis and the synthesis network is very-low bandwidth video e-mail and videoconferencing. Figure 9 shows a demonstration of the basic idea. An analysis network trained on a few views of the specific

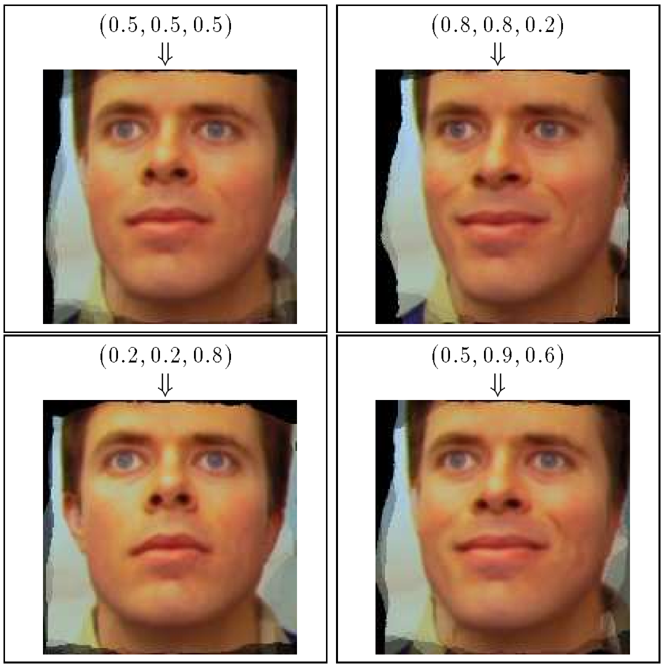


Figure 8: *The synthesis network trained on the eight examples of figure 6 synthesizes new color images from the input rotation and expression parameters* (r_1, r_2, r_3).

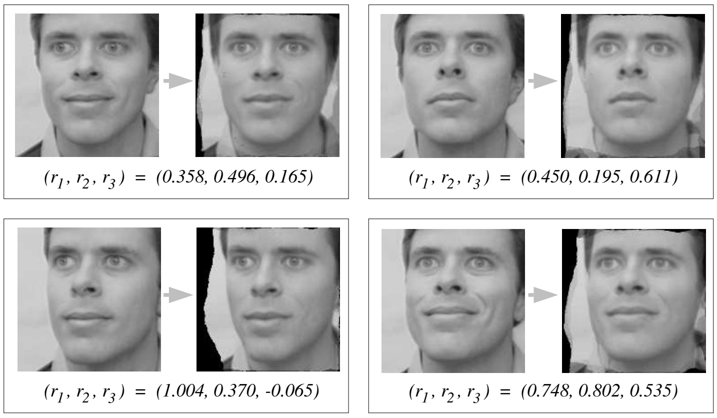


Figure 9: *A demonstration of very-low-bandwidth teleconferencing or video e-mail. In the boxed image pairs, the novel input real image (left) is fed into an analysis network to estimate left-right rotation* r_1, *expression* r_2, *and up-down rotation* r_3. *These parameters are then fed into the synthesis module which synthesizes the image shown on the right. This example-based approach can achieve in principle very high compression. Here each frame is compressed to about 3 bytes, though it is very limited in term of the range of pose and expressions. A more realistic estimate consists of around* $10-20$ *parameters to be estimated and sent, plus a few others defining the image plane transformations, for a total of around 20 bytes per frame.*

user analyzes new images in terms of pose and expression parameters. These parameters are sent electronically for each frame and used at the receiver site by a synthesis network previously trained on example views of the same user to synthesize appropriate new views. This example-based approach can achieve in principle very high compression. The example shown in the figure probably needs in the order of 3 bytes per frame at run time. In the meantime, we have experimented with additional control parameters such as the degree of open-closed mouth, open-closed eyes, direction of gaze and so on. A realistic estimate consists of around 10 parameters to be estimated and sent, plus a few others defining the image plane transformation for a total of perhaps $10-20$ bytes per frame. Of course, the practical feasibility of such a scheme is still an open question, even as a component in more complex compression architectures that incorporate standard compression schemes such as JPEG and MPEG.

3.5 Object recognition and face recognition

An obvious application of our learning-from-examples framework is object recognition. The idea is simply to train the learning module with a few views of the object to be recognized and the corresponding label (as output). This is a specific case of the analysis problem of figure 2. One interesting class of objects for recognition are human faces. Over the last several years several computer vision groups have used this view-based paradigm to develop and demonstrate successfull face recongition systems. In our own group at MIT we have developed four different view-based face recognition systems that rely on several example views per person.

3.5.1 A frontal face recognition system

Following the face recognition work of Baron, Brunelli and Poggio at IRST in Italy, have used a template-based strategy to recognize frontal views of faces. From an example model view, faces are represented using templates of the eyes, nose, mouth, and entire face. A normalized correlation metric is used to compare the model templates against the input. Input images are geometrically registered with the model views by aligning the eyes, which normalizes the input for the effects of translation, scale, and image-plane rotation. This step provides an approximative correspondence. The eyes are automatically located in a separate initial step using eye templates. On a data base of 47 people, 4 views per person, the recognition performance is 100% when two views are used as model views and the remaining two views are used for testing. the best results.

3.5.2 A PC-based face recognition system

Using a very similar algorithm, Gilbert and Yang at Harvard developed a fast PC-based face recognition system that uses custom VLSI chips for fast similarity measurements. The correlation chip performs an absolute-difference correlation of a 68×68 template with a 64×64 template, allowing for translational shifts of the latter template from -2 to +2 in both x and y. One correlation takes 0.9 ms, and going through a library of 173 model images takes only 0.3 sec. A PC implementation that uses the chip in a special accelerator board takes only 2 to 3 seconds to go from frame grabbing to recognition (above 93% correct in their experiments).

3.5.3 A pose-invariant face recognition system

In a view-based extension of these systems, MIT's Beymer developed a pose-invariant face recognizer that uses 15 views per person, views that cover different out-of-plane rotations (see the 5 different left-right rotations and 3 different up-down rotations in Fig. 10). Similarly to the previous two systems, translation, scale, and image-plane rotation are factored out by first detecting eyes and nose features and then using these features to register the input with model views. The feature finder must be person and pose independent, so it uses a simple view-based approach that employs a bank of eyes-nose templates covering a variety of "exemplar" people and poses. A new input is matched against all model views of all people and the best match is reported. The matching step consists of a geometrical registration step driven by the eyes and nose features followed by normalized correlation using eyes, nose, and mouth templates. On a data base of 62 people and 10 test views per person, the recognition rate is 98%. The test images were at a variety of image rotations, both in and out of the image plane.

3.5.4 Face verification for secure login

A fourth system (developed by Romano, Beymer and Poggio at MIT) addresses the issue of verifying the identity of a person instead of identifying a person in a data

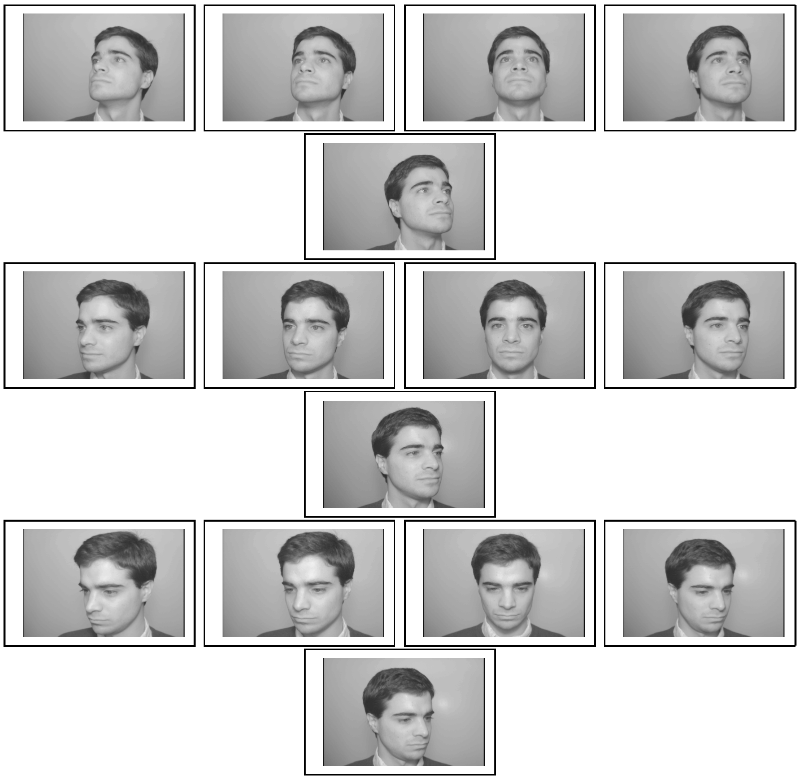


Figure 10: *The pose-invariant, view-based face recognizer uses 15 views to model a person's face. From Beymer .*

base. The system is a real-time face verification system on a low-end SGI workstation which reliably accepts a human face image labeled with its correct identity and rejects a falsely labeled image. It is motivated by security system applications in which access to an electronic network or secured building may be controlled by automatic authentication of users. We have tested the basic verification system on preliminary data sets of up to 25 users and found an estimated false acceptance rate of less than 1%.

3.6 Object detection: finding people in cluttered images

Even more difficult than recognition of an isolated specific object is detection of an object of a certain class in a cluttered image. Compare for instance the task of recognizing a specific face with the task of finding whether a cluttered image contains a face and where.

Sung and Poggio at MIT have developed an example-based learning techniques for oject detection. The system can be trained on images of a class of objects such as faces. We have demostrated successful training for finding faces, eyes and specific digits.

Figure 11 shows an example of the performance of the system after it has been trained to locate frontal images of faces. Key to it is the large number of virtual examples used to train the system (see later) : there are 3000 positive virtual examples in addition to the 1000 real ones and the 50,000 negative examples in the training set. A very similar system was also implemented by Modaggan and Pentland at the Media Lab.

3.7 The Curse of Dimensionality and Virtual Views

The key problem for the practical use of *most* learning-from-examples schemes – and for non-parametric techniques in general – is often the insufficient size of the training set. Because of the typically high dimensionality of the input vectors the required number of training examples is very high. This is the so-called *curse of dimensionality*. We have been developing the idea of exploiting prior information about the specific domain to generate additional *virtual* examples from a small set of real example images. In recent work we have shown how prior knowledge of symmetry properties of the class of 3D objects allows the synthesis of additional examples. For instance, since faces are approximatevely bilaterally symmetric, a new image can be created from a single image by mirroring it. Psychophysical evidence in favor of the use of this constraint by biological visual systems was found by Vetter at the Max Planck Institute in Tuebingen. More generally, we had suggested learning class-specific transformations from examples drawn from objects of the same class in order to synthesize new virtual views of a new object of the same class from a single view.

Figure 12, shows one way to generate virtual views by applying the transformation "learned" from a prototype of the same class (a more powerful technique developed by Poggio, Vetter and Jones uses several protoype images). Figure 13 shows the result of applying this technique to produce several rotated *virtual views* of the same face from a single real image.

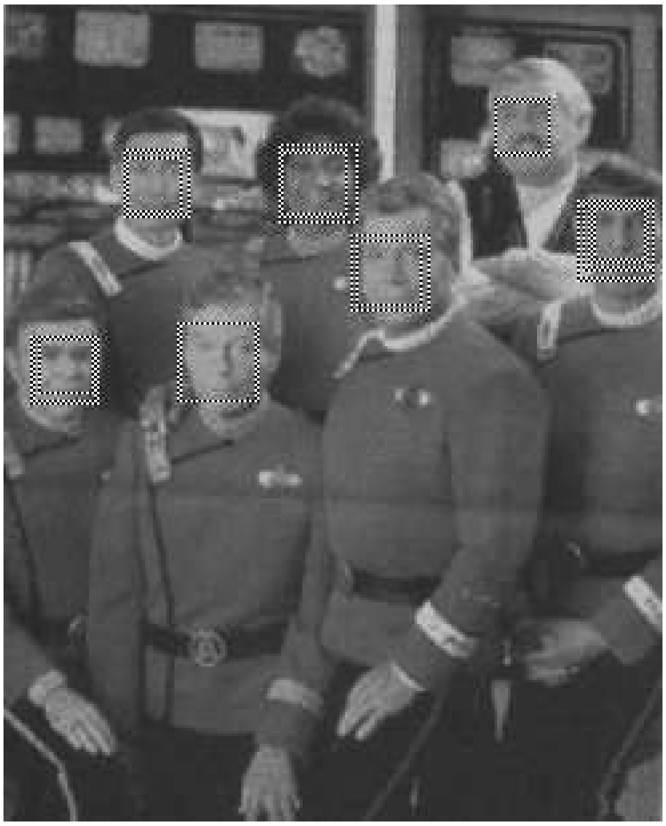


Figure 11: *Some face detection results by Sung-Poggio system. From Sung and Poggio (Sung and Poggio, 1994) .*

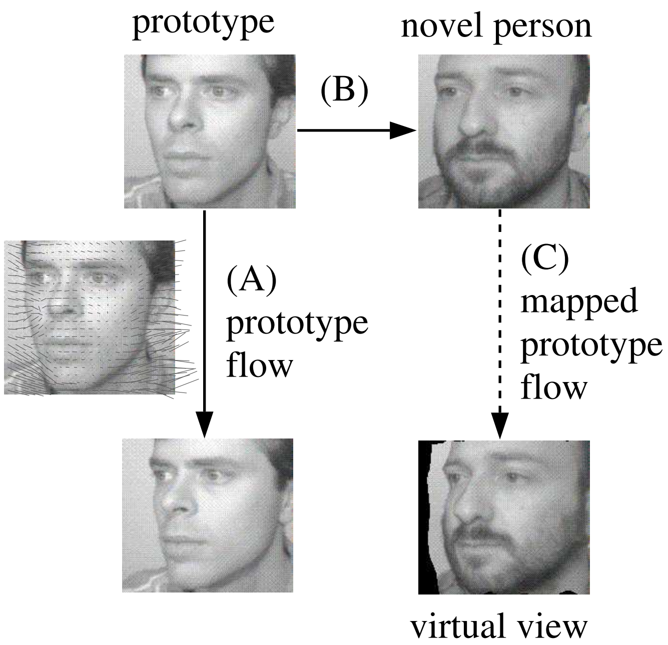


Figure 12: *(A) A 2D deformation representing a transformation is "learned" by finding correspondence among prototype images. In this example, the transformation is rotation and optical flow was used to find a dense set of correspondences. Next, in (B), the transformation is mapped onto the novel face, and (C) the novel face is transformed to a "virtual" view. From Beymer and Poggio (1995b) .*

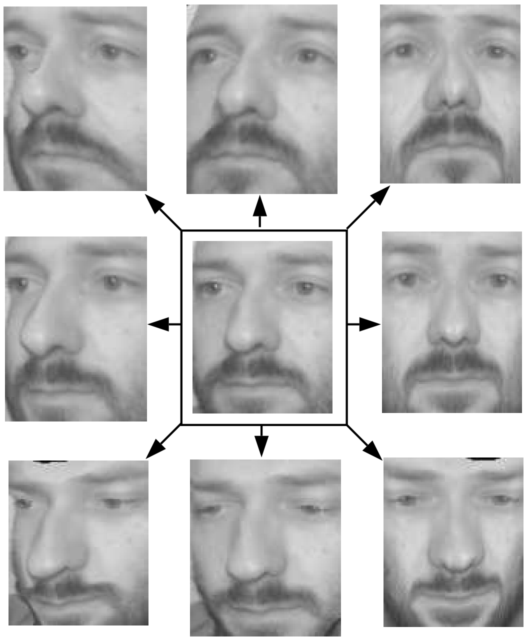


Figure 13: *A real view (center) surrounded by virtual views derived from it using parallel deformation. From Beymer and Poggio (1995b) .*

The virtual views generated in this way have been used as model views in the pose-invariant recognizer described earlier to achieve recognition rates that are likely to compare well to human performance in the same situation of relying on a single real example view.

4 Image Representations for Learning

The applications described above depend on a representation of images that induces a linear vector space structure and in principle requires dense correspondence. This image representation allows the use of learning techniques for both the analysis and synthesis of images, that is for both computer vision and computer graphics.

Since this is a key observation, I will try to describe it in more detail.

All the approaches to image analysis and synthesis described in the previous section depend on an implicit *key* assumption: *image representation must have a linear vector space structure.* All the image processing we described – applications of RN modules for analysis and synthesis, eigenfunctions decomposition, Mahalanobis distance – require that images can be treated as vectors. It is important to observe that this requirement holds true for many recent recognition systems (for instance for the eigenvectors based systems of Murase and Nayar at Columbia and of Turk and Pentland at the Media Lab). Ironically, however, the need of a linear vector space structure has not been explicitely recognized in the literature and in the systems mentioned above, though many researchers have attempted to achieve an approximate correspondence by aligning images and models using a small number of anchor points such as the eyes in the case of face recognition. The approach we have taken for *vectorizing* images is to establish pixelwise between an image and a reference image. In this way, an image is represented as a *shape* vector of x, y positions of labeled features (or pixels) and an equally ordered *texture* vector of the "color" of each feature (or pixel). Correspondence induces a linear vector space structure in which each component has a definite meaning. We believe that this representation is key to the correct use of vector space techniques such as eigenvectors and learning networks,

In order to vectorize an image we have used optical flow techniques. In the quest for more robust ways to vectorize a new image we have developed also a new example-based technique which fully exploits the idea of a linear space structure and opens up the way to build powerful learning systems in vision. We will not describe the technique here.

5 How the Brain works

Very recent data suggest that neurons in infratemporal cortex may become tuned to specific views of specific 3D objects that the monkey was trained to recognize. The representation is the same suggested by the learning box approach outlined earlier.

As shown by our various implementations described in the previous sections the example-based approach is successful in practical problems of object recognition, image analysis and image synthesis. It is not surprising therefore to ask whether a similar approach may be used by our brain. Networks that learn from examples have an obvious appeal from the point of view of neural mechanisms and available neural data. Over the last four years psychophysical experiments have indeed supported the example-based and view-based schemes that we suggested as one of the mechanisms of object recognition. Very recently, physiological experiments have provided a suggestive glimpse on how neurons in IT cortex may represent objects for recognition. The experimental results seem to agree to a surprising extent with our view-based models.

Here I will review briefly our model and then the physiological evidence for it.

Figure 14 shows our basic module for object recognition. Classification or identification of a visual stimulus is accomplished by a network of units. Each unit is broadly tuned to a particular view of the object, for instance though tuning to visible features in the view. We refer to this optimal view as the center of the unit. One can think of it as a template (or a set of *template components*) to which the input is compared. The unit is maximally excited when the stimulus exactly matches its template but also responds proportionately less to similar stimuli. The weighted sum of activities of all the units represents the output of the network.

Consider how the network "learns" to recognize views of the object shown in figure 15. In this example the inputs of the network are the x, y positions of the vertices of the wireframe object in the image. Four training views are used. After training, the network consists of four units, each one tuned to one of the four views as in figure 15. The weights of the output connections are determined by minimizing misclassification errors on the four views and using as negative examples views of other similar objects ("distractors").

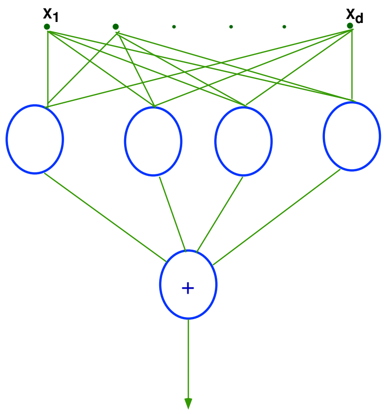


Figure 14: *A RBF network with four units each tuned to one of the four training views shown in the next figure. The tuning curve of each of the unit is also shown in the next figure. The units are view-dependent and selective, relative to distractor objects of the same type.*

The figure shows the tuning of the four units for images of the "correct" object. The tuning is broad and centered on the center of the unit, that is the training view. Somewhat surprisingly, the tuning is also very selective: the dotted line shows the average response of each of the unit to 300 similar distractors (for further details see Edelman and Bülthoff, 1992). Even the maximum response to the best distractor is in this case always less than the response to the optimal view. The output of the network, being a linear combination of the activities of the four units, is essentially view-invariant and still very selective. The activity of the unit measures the global similarity of the input vector to the center: for optimal tuning all features have to be closed to the optimum value. Even the mismatch of a single component of the template may set to zero the activity of the unit. Thus the rough rule implemented by a view-tuned unit is the conjunction of a set of predicates, one for each input feature, measuring the match with the template.

This example is clearly a caricature of a view-based recognition module but it helps making the main points of the argument. Of course, biologically plausible features are different from the coordinates of the corners used by the toy network described above. We (Bricolo, Logothetis and Poggio at Baylor College of Medicine and at MIT) have recently performed simulations of a biologically more plausible network. We first filter the image through a bank of directional filters of various orders and scale, similar to V1 neurons. Then at each point in the visual field a learning module compares the vector of filter values with a center that represent the response of the filters to a patch of the target object. We found that two centers, each tuned to a different patch of the image, are sufficient for recognition and show the properties found in psychophysical and physiological experiments.

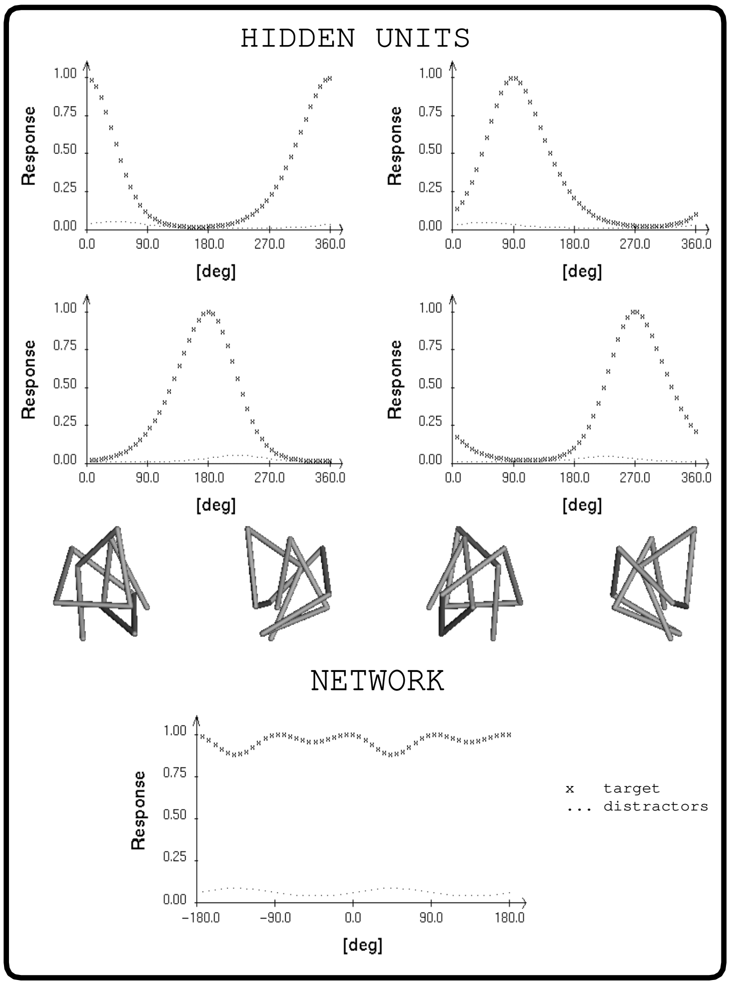


Figure 15: *Tuning of each of the four hidden units of the network of the previous figure for images of the "correct" 3D objects. The tuning is broad and selective: the dotted lines indicate the average response to* 300 *distractor objects of the same type. The bottom graphs show the tuning of the output of the network after learning (that is computation of the weights c): it is view-invariant and object specific. Again the dotted curve indicates the average response of the network to the same* 300 *distractors.*

Despite its gross oversimplification, our model manages to capture some of the basic psychophysical and physiological findings, in particular the existence of view-tuned and view-invariant units and the shape of psychophysically measured recognition fields. Let me summarize now the recent physiological findings (by Logothetis and Pauls at Baylor College of Medicine).

Three monkeys were trained to recognize computer-rendered objects irrespective of position or orientation, two of which were used in the physiology experiments. The monkeys were first allowed to inspect an object, the *target*, presented from a given viewpoint, and subsequently were tested for recognizing views of the same object generated by ± 10 to ± 180 degree rotations around the vertical axis. In some experiments the animals were tested for recognizing views around either the vertical or the horizontal axis, and in some others the animals were tested for views around all four axes. The images were presented sequentially, with the target views dispersed among a large number of other objects, the *distractors*. Two levers were attached to the front panel of the chair, and reinforcement was contingent upon pressing the right lever each time the target was presented. Pressing the left lever was required upon presentation of a distractor. Correct responses were rewarded with fruit-juice.

A total of 970 single neurons in a part of the cortex called infratemporal cortex (figure??) were recorded from two monkeys during combined psychophysical-electrophysiological experiments. Logothetis and coworkers found a significant number of units that showed a remarkable selectivity for individual views of wire objects that the monkey was trained to recognize. View-tuned neurons responded briskly to all views around the training view. Views generated by a 36 degree rotation in either direction elicited 41% of the maximum response, while 50 degree views failed to stimulate the neuron.

For three objects more than one neuron was found to be tuned to different views of the same object. Figure 16 shows the responses of three units that were found to respond selectively to four different views of an wire object (Wire 71). The animal had been exposed repeatedly to this object, and its psychophysical performance remains above 95% for all tested views, as can be seen in the lower plot of figure 16). Notice that one of the 3 neurons is tuned to a view and its mirror image, consistently with other theoretical and psychophysical work. The figure is surprisingly similar to figure 15 showing the response of the view-tuned hidden units of the model of figure (14).

A small percentage of cells (8 out of 773) responded to wire-like objects presented from any viewpoint, thereby showing view-invariant response characteristics, superficially similarly to the output unit of the model of figure 1. An example of such a neuron is shown in figure (17). The upper plot shows the monkey's hit rate and middle plot the neuron's average spike rate. The cell fires with a rate of about 40Hz for all target's views. The lower plot shows the responses of the same cell to 120 distractors. With four exceptions activity was uniformly low for all distractor objects presented. In all cases, even the best response to a distractor, however, remains about one half of the worst response to a target view. This neuron seems to behave as the output of the model of figure (14).

The main finding of this study is that there are neurons in IT cortex with properties intriguingly similar to the "cartoon" model of figure (14), which is itself supported by psychophysical experiments in humans and primates. Several neurons showed a remarkable selectivity for specific views of a novel, computer-rendered object that the monkey had learned to recognize. A smaller number of neurons were object-specific but view-invariant, as expected in a network in which "complex"-like view-invariant cells are fed by view-centered "simple"-like units. Furthermore, we believe that our results reflect experience dependent plasticity in IT neurons. From our simulations and computatonal arguments we expect the view-centered neurons to be effectively tuned to (a few) components or features of a view and only in some cases to require the full view for optimal activation: experiments presently planned will check this idea so far supported by the more recent simulations which also give useful hints on the nature of the "features" that are input to the view tuned neurons, a kind of visual vocabulary extracted by the brain from the visual world.

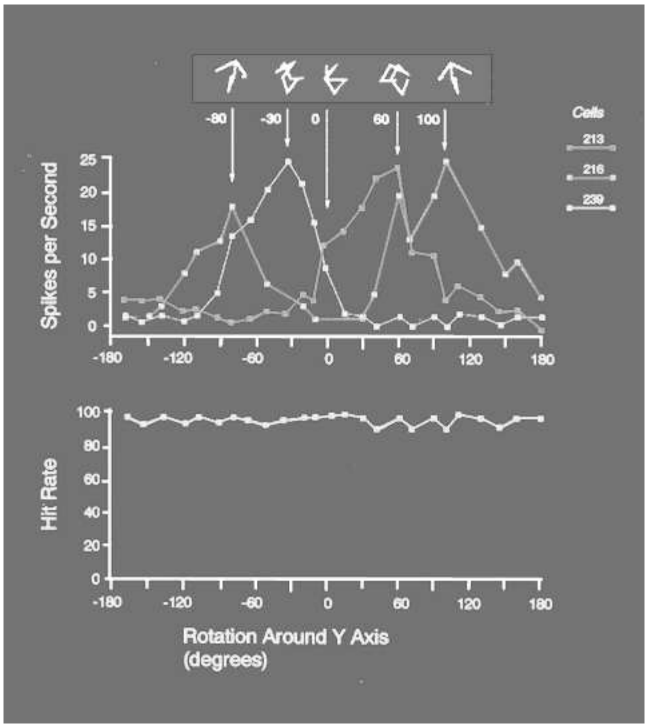


Figure 16: *The top graph shows the activity of three units in IT cortex, as a function of the angle of the stimulus view. The four neurons are tuned to four different views of the same object. One of the units shows two peaks for two mirror symmetric views. The neurons firing rate was significantly lower for all distractors (not shown here). The bottom graph represents the almost perfect, view-invariant behavioural performance of the monkey for this particular object to which he was extensively trained.(From Logothetis and Pauls, 1995).*

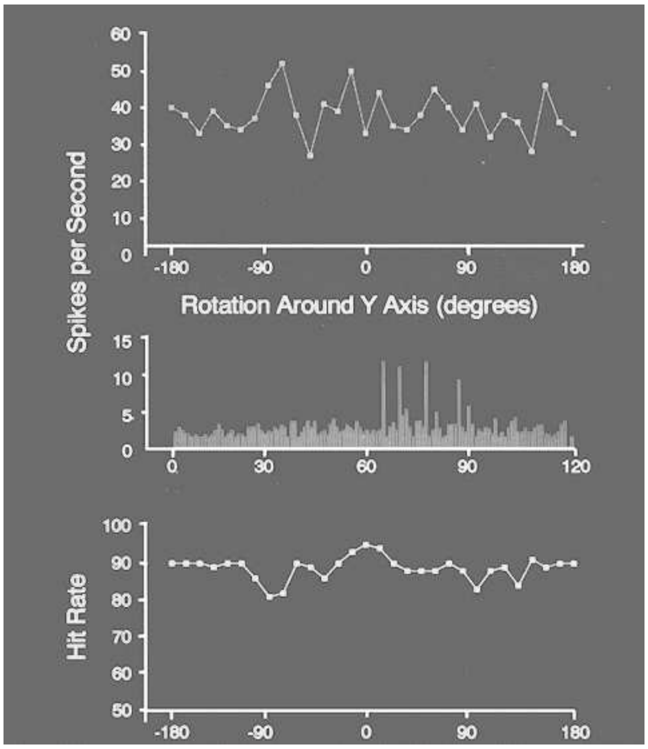


Figure 17: *The monkey performed quite well on this particular wire after extensive training (bottom graph). A neuron in IT was found that shows a view-invariant response, with about* 30 *spikes/sec to any view of the wire object (top). The response of the cell to any of the* 120 *distractors is lower as shown in the middle graph. (From Logothetis and Pauls, 1995).*