Fast Dynamic Link Matching by Communicating Synapses

Christoph von der Malsburg *
Department of Computer Science
University of Southern California
Los Angeles, CA 90089
malsburg@selforg.usc.edu

Junmei Zhu
Department of Computer Science
University of Southern California
Los Angeles, CA 90089
junmeizh@selforg.usc.edu

Abstract

This work aims to provide a framework for finding continuous mappings between patterns. One important application is the correspondence problem in vision. We concentrate here on object recognition. Two patterns, identified as image and model, can have variations in size and orientation. Dynamic link matching (DLM) developed in our lab has the advantage that it can cope with translation and other variations, including deformation, but it is too slow to be the mechanism for object recognition in adults. Our new system improves the speed of DLM, by coupling synapses directly, and by thus storing and activating previously acquired mappings. Additional mechanisms are introduced for dealing with invariance in scale and orientation.

1 Introduction

1.1 Mappings

The ability to map one domain to another is essential for animals to interact with their environment. Animals perceive an object in the outside world via different modalities that project to different brain areas. Within each modality, information flows through different levels, such as V1, V2, V4 and IT in the ventral visual pathway. Mappings thus need to be created dynamically between different areas. In pattern theory [5], this is called domain warping. The importance of mapping can best be demonstrated in one of its subproblems, the visual correspondence problem, including stereo correspondence, correspondence between consecutive images in time for motion estimation, and correspondence between image and model for object recognition. All these examples show that mappings must be dynamical objects, created in an image-dependent way.

In this paper, we will use object recognition as an example to illustrate mapping creation. The two domains between which we want to create a mapping are called image and model. Models have the same representation type as the image, which means they are two-dimensional (2D) instead of 3D. An important property we would like mappings to have is continuity, i.e., we want to find a homeomorphism between two domains. Continuity is required for some mappings and it has the good property that the composition of continuous mappings is still continuous. Because our domains are two dimensional patterns, we can simply use the usual topology in $\mathbb{R}^2$ to define continuity.

1.2 Why is object recognition difficult

Object recognition is not an easy task because the appearance of an object on the retina varies tremendously. Various factors contribute to this: translation, different aspect because of different pose of the object, deformation because of nonrigid motion, different illumination, different background, and occlusion. There is also noise in the process of image formation.

*Also at Institut für Neuroinformatik, Ruhr-University Bochum, Germany.
To deal with these variations it is not practical to store every possible appearance of the object in memory to ensure that a close match to the input can always be found. In addition to the limitations in space and time, exhaustive sample would not permit generalization. In fact, one does not need to see every aspect of an object before he can recognize it [2].

1.3 Why is object recognition possible

Despite all these difficulties, our visual system solves the task readily in our everyday life. Technically, object recognition is possible because of the regularities in the world. Objects in the world are not random, they occupy a certain region in space. Transformations are of several restricted types, each of which has its own regularities. For instance, 3D rigid motion is controlled by 6 parameters; nonrigid motion is also constrained, such as facial expressions are constrained by muscles and bones. It is these regularities that made object recognition possible.

From the point of view of information theory, the information contained in mappings is actually very limited. Mappings can be seen as controlled by some hidden variables, which should ultimately be learned from experience, if not genetically determined.

1.4 How to recognize

The ability to identify different appearances of an object indicates that there must be a metric space, in which instances from one object are closer than those from different objects. The space and metric are not unique. Various approaches in object recognition span a whole spectrum of choices of space and metric.

One end of the spectrum is formed by feature-space methods. A feature space is composed of more or less complex invariant features, such as various moments that are invariant to transformations. The metric may simply be Euclidean distance. Because spatial relationship is lost in the feature space (due to their invariance), some of the features have to be global, otherwise rearrangement ambiguities will occur. So far, this method has not been successful except for limited application domains in which invariant features can be defined.

At the other end of the spectrum the space is identified with the input directly, and a global, complicated metric deals with all types of transformations explicitly. This variety is often called template matching. This is obviously difficult to implement.

More feasible approaches are in between. In these approaches, the space is formed by local features that are robust against some variations (such as illumination or deformation). As long as a mapping, or a correspondence, exists, the similarity metric can be computed readily and locally, being just the summation of the similarity metric of each corresponding feature pair. The definition of metric for a pair of features depends on feature types. Roughly speaking there are two types of features, hard and soft. Hard features, such as salient points, edges, etc, are obtained after a hard decision. Their similarity metric is the Euclidean distance between feature positions. Soft features are usually the response of some filters, and there is no label assigned to them by a hard decision. Their similarity metric is computed as the difference of the continuous feature responses at corresponding positions. Overall, metric can also be defined as probability, where one can use maximum a posteriori estimation.

What remains is mapping creation. Mappings have to be able to deal with transformations and spatial relationships that are not taken care of already by feature encoding. We will concentrate on mapping creation in the next section.

2 Mechanisms for mapping creation

Several important questions regarding mapping creation are: what transformations can this mapping handle; what are the control variables; how to set the value of the control variables; how is spatial relationship preserved.

Approaches differ in their answers to these questions. There is a tradeoff between the number of control variables and the transformations one can handle. Generally speaking, the more control variables one has, the more degrees of freedom and transformation types one can deal with. The transformations that cannot be accounted for will have to be considered as noise. In this section, we review some approaches in mapping creation.
2.1 Alignment [7]

Assuming models are at standard position, alignment is the procedure to put the image (or rather, image segment) into the corresponding position. Even though alignment can be loosely called mapping, we here refer to the method as proposed by Ullman [7]. Its control variables are rigid motion parameters. Nonrigid motions are also possible by implementing local rigid motions. These control variables are computed directly from alignment keys, such as salient points, unless the orientation of the object is independently known.

The problem with this approach is that alignment keys must be extracted first independently in the image and model, and correspondences be established between them before control variables can be computed. This proved to be difficult.

2.2 Deformable template [12, 13]

Deformable templates are simple parameterized shapes that interact with an image to obtain the best fit of their parameters. These parameters are the control variables. They are predefined by hand, and so are the corresponding allowed transformations. The value of parameters is usually computed by following the gradient of a predefined energy function.

The problem with this approach is that the definition of template as well as energy function often is done by hand and is very clumsy and thus lacks the ability to generalize.

2.3 Shifter circuits [1, 6]

Shifter circuit is a general strategy for dynamic mapping control. It relies on a set of control neurons to modify the synaptic strengths of inter-area connections so as to create a mapping. Each control unit modulates a local group of synapses. The allowed transformations are scale and translation. Actually other transformations are possible but the constraints on the control variables will be much more complicated. The value of control units is computed by following the gradient of an energy function, which says how valid the control states are and how well the output resembles the stored patterns. Spatial relationship is ensured by the constraints for control units. In [5], Mumford provided some proposals as to how the shifter circuits can be implemented in the brain.

Anderson proposes several levels in between the two domains, such that each level has smaller fan-in and simpler control. However, when using patterns stored in the associative memory to guide the control variables, i.e., top-down control, there is the problem of how to back propagate this information down to the control units at lower levels.

Shifter circuits provide a good routing model that preserves spatial relationships, but there are still some problems. It is not sure how this system can learn from experience. Moreover, in the neurobiological implementation, it requires a third party, pulvinar, to provide the control signals, and the question arises how that third party gains access to information on correspondences between the two domains to be connected.

2.4 Optical flow [3]

Optical flow is to create a correspondence between two consecutive images. It is not for object recognition, but is listed here because it gives an example of mapping creation. Control variables are the velocity at each pixel, which are computed locally. However, extra constraints, such as smoothness constraints, are required, since only normal velocity can be computed because of the aperture problem. In some sense, the smoothness constraint helps to preserve the spatial relationships.

The problem with optical flow is that the change between the two images has to be small, which makes it difficult or impossible to exploit the mechanism for object recognition. And the illumination cannot change too much as the mapping depends on the pixel intensity value [8].

2.5 Dynamic Link Matching (DLM) [11, 9]

DLM is based on rapid synaptic plasticity. It is performed by network self-organization, in which mapping between two domains is controlled by signal correlations, which are shaped by neighboring connections and by dynamic links themselves. Each connection weight is a control variable, but the actual degree of freedom
is much smaller because of the constraints on weight growth. DLM is intrinsically invariant to translation, and is very robust to deformations. Spatial relationships are preserved due to cooperativity between connections.

Weight dynamics can also be derived from minimizing an energy function [10]. The difference to the deformable template is that DLM has more degrees of freedom, and greater ability to generalize.

However, DLM [11] is very slow, because the connections are updated in a sequential manner, by running blobs in the image and model. This makes it unlikely to be the mechanism for recognition in adults. In a fast algorithmic version of DLM [9], only restricted moves are allowed, and this causes the reduced ability to deal with deformations. Also, the neural implementation of that mechanism is not clear.

3 Our system

In this section we present a system for creating a mapping between two patterns for object recognition. The idea is to improve the speed of DLM by letting synapses talk to synapses, and by thus activating stored mappings directly. Moreover, additional mechanisms are introduced for dealing with invariance in scale and orientation.

3.1 Input and Output

The input is two grey level patterns, applied to the image domain and model domain, respectively. The goal is to create output in the form of a continuous mapping between correspondence points in these two patterns. The mapping is represented as a connection matrix, which is 4D for 2D patterns, and 2D for 1D patterns.

3.2 Features

The features we use are Gabor wavelet responses, typical soft features as were used in [11, 9]. In each position in the image and the model we compute the Gabor responses with different orientation and scale, forming a jet. Here we only use magnitude of the Gabor responses.

3.3 Initial connection weights

Initially the connection pattern is all-to-all. Each connection weight is computed as the similarity of the two jets, extracted in image and model at the points that are connected. Because the similarity function depends on the relative scale and orientation between the image and the model, we have separate connection matrices for a number of discrete scales and orientations, covering the whole range of interest.

For scale and orientation index \( p \), the initial weight, \( w_{t,r}^p \), between two jets \( J = (a_1, a_2, \ldots, a_n)^T \) at position \( r \) in the image domain, and \( J' = (a'_1, a'_2, \ldots, a'_n)^T \), at position \( t \) in the model domain (where \( T \) denotes transposition) is defined as the similarity

\[
    w_{t,r}^p(0) = S_{t,r}^p = \frac{\sum a_i d_{p}^f(i)}{\sqrt{\sum a_i^2 \sum d_{p}^2 f^2(i)}}
\]  

where \( f_p(i) \) is the function that maps corresponding elements in two jets with scale and orientation difference \( p \). Because for different scale some elements in one jet do not find a correspondence in the other jet, the summation is only over the elements of a jet that have correspondence.

3.4 System dynamics

Connection growth is controlled by cooperation and competition. Cooperation for a synapse comes from synapses that connect neighboring cells in the image and neighboring cells in the model, and from the same scale and orientation only. The region of cooperativity is specific to the scale and orientation. Competition is between synapses diverging from one point in the image, as well as between synapses converging to one point in the model. There is global competition between different scales and orientations.

Weight dynamics is similar to that in the retinotopy model [4].
\[ \dot{w}_{t,r}^p = f_{t,r}^p(W^p) - w_{t,r}^p \frac{1}{NP} \left( \frac{1}{\lambda_p} + \frac{1}{\lambda_p + 1} \sum_{r',p'} f_{t',r'}^p(W^{p'}) + \frac{\lambda_p}{\lambda_p + 1} \sum_{r',p'} f_{t',r'}^p(W^{p'}) \right) \]  

(2)

where the growth term

\[ f_{t,r}^p(W^p) = \alpha + \beta w_{t,r}^p C_{t,r}^p(W^p) \]  

(3)

\[ C_{t,r}^p(W^p) = \sum_{r',p'} \frac{c^p(t, t', r, r')}{N^p} w_{t',r'}^p \]  

(4)

and \( W^p = (w_{t,r}^p) \) is the weight matrix. \( N \) is the number of sampling points in the image (for simplicity, model has the same number of sampling points, and image and model have the same number of pixels). \( P \) is the total number of scales and orientations. \( \alpha \) and \( \beta \) are real numbers. \( \alpha \) is the formation rate of synapses, and \( \beta \) is the rate for cooperativity. \( c^p(t, t', r, r') \) is the cooperation coefficient.

The negative term in equation 2 is the competition term. The factor of \( w_{t,r}^p \) is the average of divergent competition (synapses diverging from one point in the image, over all scale and orientation indices \( p \)) and convergent competition (synapses converging to one point in the model, also over all \( p \)). \( \frac{\lambda_p}{\lambda_p + 1} \) and \( \frac{\lambda_p}{\lambda_p + 1} \) are the normalization factors for convergent and divergent competition, respectively. They are so defined that their summation is 1, and their ratio is \( \lambda_p \), where \( \lambda_p \) is a function of \( p \) (or more precisely, the scale component of \( p \)). This issue arises from discrete sampling, and that we have the same sampling rate in the image and model. When image and model are of different scale, we have to allow different contributions from divergence and convergence. In the case of 1D image and model, \( \lambda_p \) is simply the size ratio of model to image for scale index \( p \).

Equation 4 indicates direct synaptic communication. It assumes zero boundaries, i.e., all the weights outside the boundaries are assumed to be zero, and there is no periodicity assumption.

Matrix \( c^p \) is specific to the scale and orientation \( p \), and this is where past knowledge is incorporated to speed up the system. It should be learned from experiences, but we will not study how it is learned in this paper. In the following we give an example of \( c^p \) for 1D patterns, which is easier to illustrate than that for 2D patterns. \( c^p \) depends only on \( |t - t'| \) and \( |r - r'| \). In 1D case it is, for instance, a rotation of the 2D Gaussian:

\[ G(x, y) = \frac{1}{2\pi \sigma_1 \sigma_2} e^{-\frac{x^2}{2\sigma_1^2} - \frac{y^2}{2\sigma_2^2}} \]  

(5)

where \( \sigma_1 > \sigma_2 \). The rotation angle depends on the relative size of image and model. If horizontal direction \( (x) \) indicates image and vertical \( (y) \) indicates model, and if the size ratio of model to image is \( \lambda \), then the Gaussian should be rotated \( \arctan(\lambda) \) clockwise. Figure 1 shows such an example. The size of \( c \) matrices are \( 9 \times 9 \), \( \sigma_1 = 2, \sigma_2 = 1 \).

![Figure 1: Specific C Matrices. From left to right, the relative size of model to image is 1/2, 1/\sqrt{2}, 1, \sqrt{2}, 2.](image)

4 Experiments

Input. Here we use 1D patterns because the result is easier to demonstrate. Figure 2 is an example of 1D patterns, where the size of image and model differs by a factor of \( \sqrt{2} \), which corresponds to 0.5 octave. The
patterns are extended in the vertical direction, for the convenience of viewing. Their length is 128 pixels. In the simulation we used 4 such pattern pairs to obtain a reasonably good initial weight.

![Image](image.png)

**Figure 2:** Input patterns. (a) image, (b) model.

**Features.** Features are computed as 1D Gabor wavelet responses. There are \( n = 5 \) different scales, each differs by 0.5 octave. Because we use 4 pattern pairs, the feature set at each point is a vector of size \( 20 \times 1 \).

**Connection weights.** We uniformly sample \( N = 17 \) nodes from image and model each, and compute the initial connection weights according to equation 1. The initial weights are shown in figure 3(a). Weights at different time steps are shown in figure 3(b)-(e). One can see from the figure that as the weights evolve, one scale wins out rapidly, and a correct continuous mapping is formed after about 50 iterations. The final weights, after 1000 iterations, are shown in figure 3(e), which indicates that the mapping we get is an attractor. The step size in the Euler method for solving equation 2 is 0.3.

**Parameters:**

- \( N = 17 \) number of sampling points.
- \( P = 5 \) number of relative scales, corresponding to the size ratios of model to image \( \lambda = 1/2, 1/\sqrt{2}, 1, \sqrt{2}, 2 \).
- \( C \) matrices are as in the last section, size \( 9 \times 9 \), \( \sigma_1 = 2, \sigma_2 = 1 \).
- \( \alpha = 0.01, \beta = 0.25 \)

Simulations with various input patterns show that even though the scale is sampled discretely, the system is very robust to scale change, and can deal with continuous scale variation. The accurate relative scale of image and model can be computed from the slope of the connection weights.

5 **Discussion**

Our new system of coupling synapses directly takes much less iteration steps than the original DLM model [11], which takes several thousands of steps. The time consuming part of DLM is in the sequential computation of signal correlations, which needs some 200 iteration steps to allow the running blob move over the layers, before link dynamics can be updated. In our new model, mapping is formed after about 50 iterations. These two models, however, is not completely comparable because in [11] the patterns are two dimensional, and there are multiple models, while in the present simulation patterns are 1D and we only consider mapping creation but not recognition, there being only 1 model. On the other hand, the original DLM did not deal with variations in scale (orientation). More work is needed to evaluate the speedup of this current system.

As to the neurobiological basis of direct communicating synapses, we propose that the communication is mediated by astrocytes, a type of glial cells, which connect directly to the synapses of neurons. This is biologically more plausible than using a third party to provide control signals.

Ongoing work includes simulating 2D patterns and extending this system to deal with other variations such as rotation in depth.
Figure 3: Connection weights.
Acknowledgments

The authors would like to thank the developers of the FLAVOR software environment, which served as the platform for this work. This work was supported by ARO, Grant Number DAAG55-98-1-0293 and by ONR, Grant Number N00014-98-1-0242.

References


