

# On Disparity Matching in Stereo Vision via a Neural Network Framework

JUNG-HUA WANG\* AND CHIH-PING HSIAO\*\*

\*Department of Electrical Engineering  
National Taiwan Ocean University  
Keelung, Taiwan, R.O.C.

\*\*Aetex Biometric Corp.  
Taipei, Taiwan, R.O.C.

(Received December 15, 1998; Accepted March 29, 1999)

## ABSTRACT

This paper presents a neural framework for dealing with the problem of disparity matching in stereo vision. Two different types of neural networks are used in this framework: one is called the vitality conservation (VC) network for learning clustering, and the other is the back-propagation (BP) network for learning disparity matching. The VC network utilizes a vitality conservation principle to facilitate self-development in network growing. The training process of VC is smooth and incremental; it not only achieves the biologically plausible learning property, but also facilitates systematic derivations for training parameters. Using the [intensity, variation, orientation, x, y] of each pixel (or a block) as the training vector, the VC network dismembers the input image into several clusters, and results can be used by the BP network to achieve accurate matching. Unlike the conventional  $k$ -means and self-organizing feature map (SOFM), VC is a self-creating network; the number of clusters is self-organizing and need not be pre-specified. The BP network, using differential features as input training data, can learn the functional relationship between differential features and the matching degree. After training, the BP network is first used to generate an initial disparity (range) map. With the clustering results and the initial map, a matching algorithm that incorporates the BP network is then applied to recursively refine the map in a cluster-by-cluster manner. In the matching process, useful constraints, such as an epipolar line, ordering, geometry and continuity, are employed to reduce the occurrence of mismatching. The matching process continues until all clusters are matched. Empirical results indicate that the proposed framework is very promising for applications in stereo vision.

**Key Words:** stereo vision, disparity matching, self-creating network, back-propagation network, clustering, neural networks, self-organizing feature map

## 1. Introduction

Stereo imaging or stereo vision (Kanade and Okutomi, 1994) refers to a process which transforms the information of two plane images into a 3-D description of the scene and recovers depth information in terms of the exact distance. With depth information, one can create models of the terrain and other natural environments for use in various applications, such as virtual reality, flight simulation, air navigation, and robotics. Due to its inherent characteristics, stereo vision is a better choice for achieving high-resolution 3-D description of *moving* objects than other methods, such as laser range finding. The basic idea of stereo vision is illustrated in Fig. 1. An arbitrary point in a 3-D scene is projected onto different locations in stereo images. Assume that a point  $p$  in a surface is projected

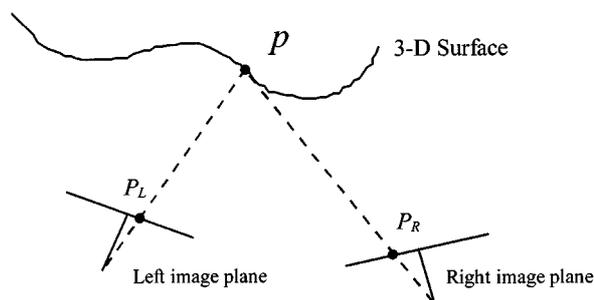


Fig. 1. Basic concept of stereo vision.

onto two cameras image planes,  $P_L$  and  $P_R$ , respectively. When the imaging geometry is known, the disparity between these two locations provides an estimate of the corresponding 3-D position. Specifically, the location of  $p$  can be calculated from the known

information,  $P_L$  and  $P_R$ , and the internal and external parameters of these two cameras, such as the focal lengths and positions of two cameras (Dhond and Aggarwal, 1989; Barnard and Fishler, 1987). Shown in Fig. 2 is a parallel configuration, where one point,  $p(x,y,z)$ , is projected onto the left and right imaging planes at  $P_L(x_l,y_l)$  and  $P_R(x_r,y_r)$ , respectively. The coordinates of  $p$  can be calculated as follows:

$$\begin{aligned} x &= b(x_l+x_r)/[2(x_l-x_r)], \\ y &= b(y_l+y_r)/[2(x_l-x_r)], \\ z &= bf/(x_l-x_r), \end{aligned} \tag{1}$$

where  $(x_l-x_r)$  = the *disparity*, base line  $b$  = the distance between the left and right cameras, and  $f$  = the focal length of the camera. It should be noted that even if the camera is arbitrarily configured, we can always generate a parallel configuration through the process of rectification (Papadimitriou and Dennis, 1996). Stereo vision systems are not only low in cost, but also offer great utility in depth recovery. Depth is a visual cue essential to applications such as surface reconstruction, pattern recognition, and 3D computer vision. Although other sources, such as camera vergence and lens focus, have also been investigated (Nayar and Nakagawa, 1994; Subbarao and Tao, 1995; Capurro *et al.*, 1996; Park *et al.*, 1998; Horng *et al.*, 1998) for their use in depth recovery, much research has emphasized stereo disparity as a source of depth information. This is because among the three most important issues in stereo vision, namely, feature extraction, disparity matching, and surface reconstruction, disparity matching still is the most challenging task due to its complexity (Marr and Poggio, 1976). As can be seen in Figs. 1 and 2, in order to obtain the disparity, one has to first determine if a point  $P_L$  matches  $P_R$  (called a template point or candidate). However, it is by no means a trivial task to accurately match the pixels in left and right images. Thus, as shown in Fig. 2, the goal is to find a correspondence which minimizes a measure of the error between a pair of pixels ( $P_L, P_R$ ). In this parallel configuration, it can be shown that a match pair must be found on  $y_l=y_r$ , which is called the *epipolar* line constraint in the stereo image. This constraint is important because it can be used to reduce the search area from the entire image to one horizontal line across the image.

Traditionally, the two mainstream types of techniques used in solving the problem of disparity matching are area-based (Marapane and Trivedi, 1989; Kanade and Okutomi, 1994) and feature-based (Nasrabadi *et al.*, 1989; Nasrabadi and Choo, 1992) techniques. Area-

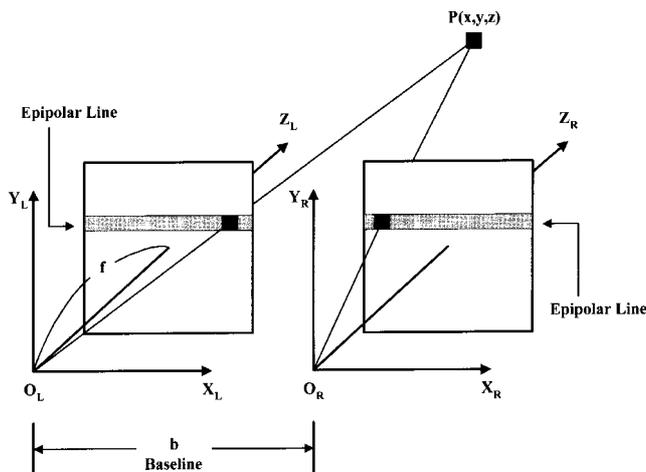


Fig. 2. The parallel configuration in stereo vision. The baseline between the two cameras is  $b$ , and the focal length is  $f$ .

based (aka window-based) techniques utilize correlation between the intensity patterns in the neighborhood of a pixel in the left image and those in the neighborhood of a corresponding pixel at the right image. They are simple and fast. However, the intensity value of each pixel is sensitive to changes in absolute intensity, contrast and illumination. Hence, area-based techniques usually give low accuracy results. The sum of the squared difference (SSD) or auto-correlation has often been used as a criterion to determine the best matching pair. However, mismatches can arise if the pair with the minimum SSD is chosen as the best matching pair because the functional relationship between the matching degree and the correlation in intensity patterns is by no means structural or linear and cannot be fully described using such a simple clear-cut rule. In addition, the size of the window can significantly affect the matching accuracy. Kanade and Okutomi (1994) have presented an adaptive window algorithm with impressive results. Still, it is virtually impossible to correctly match pixels using only information about distortion and variation inside a window area. Feature-based techniques, on the other hand, use symbolic features derived from intensity images rather than image intensities *per se*. Symbolic features, such as edge points and edge segments, are often used in these techniques. For example, Cochran and Medioni (1992) have used the edge extraction technique developed by Nevatia and Babu (1980) to obtain twelve different edge orientations. Nasrabadi and Choo (1992) have used a special operator developed by Moravec (1980) to obtain directional variances. Because feature-based techniques allow simple comparisons between the attributes of features, they are generally faster and more accurate than area-based methods. Still,

finding an appropriate interpolation method for non-featured areas is still a problem, especially in cases where reconstruction of a 3-D surface is desired.

A neural network derives its computing power through, first, its massively parallel distributed structure and, second, its ability to learn and, therefore, generalize; generalization refers to production by the network of reasonable outputs for inputs not encountered during training. The present paper deals with disparity matching by using a neural framework, in which a self-creating neural network (named VC) is incorporated along with a back-propagation (BP) neural network to achieve high performance disparity matching. The goal is to combine the advantages of the area-based and feature-based methods. The framework employs the capabilities of the BP network in function approximation (Hornik *et al.*, 1989) and generalization to learn the non-structured knowledge required to achieve higher accuracy in disparity matching compared to area-based methods. In addition, the VC network, which utilizes a *conservation principle* (Wang and Hsiao, 1997) to facilitate network development, is introduced. The conservation principle, originally developed for learning vector quantization, is applied in this paper to learning image clustering, the result of which can be incorporated with the BP network and a matching algorithm to implement a matching process in which pixels are matched on a cluster-by-cluster basis. Thus, the main contributions of this paper are an effective matching algorithm, which applies the conservation principle to online learning image clustering, incorporation of the BP and VC networks to implement a high performance disparity matching system, and further validation through simulation studies.

The organization of this paper is as follows. In Section II, the architecture of the neural-based framework is described. The matching process in which pixels are matched on a cluster-by-cluster basis by combining the BP network and a matching algorithm is discussed. Constraints imposed by the epipolar line, ordering, geometry and continuity are shown to be useful in reducing the number of unmatched pixels as well as unnecessary interpolation during surface reconstruction. In Section III, the VC network is introduced and its use in learning image clustering is described. A merging algorithm is developed to agglomerate code vectors into several clusters. Empirical results are presented in Section IV to show the performance in terms of matching of the proposed framework. Finally, concluding remarks are given.

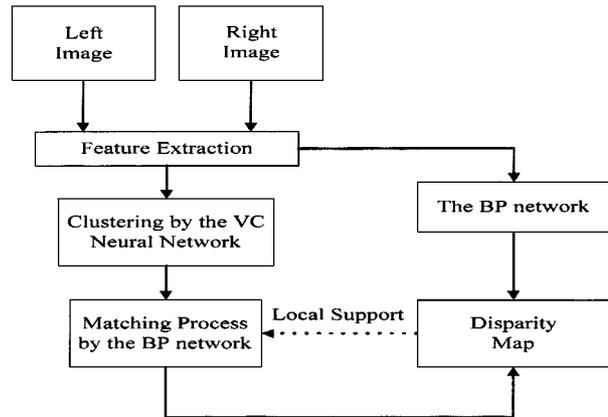


Fig. 3. Block diagram of the proposed disparity matching system.

## II. The Neural-based Stereo Vision System

### 1. System Overview

Figure 3 illustrates the block diagram of our proposed system. To avoid the drawback of area-based methods, our idea is to employ a BP network to compute the matching degree (0.0-1.0) between the two local windows ( $7 \times 7$ ) from the left and right images. The two pixels to be matched correspond to the centers of the two windows. Several constraints are used to help determine the best matching pair. In feature extraction, the Sobel operation is applied to input images to obtain the variation and orientation of each pixel; these data will form the basic input features. To prepare the training data, 200 matched and unmatched pixels were hand picked (by means of visual examination) from both flat regions and highly varying regions in the image pair. For example, it is easy to spot the pixel in the right image which matches a very bright pixel (i.e., an interesting pixel) surrounded by a dark region in the left image. In addition, the continuity constraint can be employed to obtain more matched pairs because pixels near the interesting pixel must have the same disparity. Fifty pairs of matched and unmatched pixels are randomly selected to offline train the BP network. During training, the differences in intensity, variation and orientation between two local windows are fed to the BP network. The BP network after training should have the ability to differentiate the matched pairs from unmatched pairs. The trained BP network is first used to generate an initial or primitive disparity map, which will then be used as a reference map for the subsequent matching process. Using the BP network and the clustering results from the VC network, a matching process further refines the primitive map on a cluster-

by-cluster basis. Because the matching problem can be treated as a many-to-one problem, in order to reduce the search space, the process also involves application of some useful constraints so as to effectively extract out the best matching pixel from among multiple candidates. Due to its critical importance, the matching process will be elaborated on later in Subsection II.4.

## 2. The Differential Input Features

Because the intensity value of each pixel is sensitive to changes in contrast and illumination, and despite the fact that intensity is the information most commonly used in matching process, the variation and orientation of each pixel deserve more detailed investigation with regard to their use as features. Denote the intensity of an arbitrary pixel in location  $(x,y)$  as  $f(x,y)$ . The gradient of  $f(x,y)$  is defined as

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \end{bmatrix}, \quad (2)$$

and its magnitude is defined as the variation of  $f(x,y)$  and is written as

$$|\nabla f| = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2}. \quad (3)$$

The orientation  $\alpha(x,y)$  of the vector  $\nabla f$  at  $(x,y)$  is

$$\alpha(x,y) = \tan^{-1} \left( \frac{G_x}{G_y} \right). \quad (4)$$

In this work, variation and orientation values are all normalized to  $[0,255]$ . The Sobel operator gives good approximations of  $G_x$  and  $G_y$  (Gonzalez and Woods, 1992). Using the Sobel operator in feature extraction has two advantages: (1) it can quickly calculate variation and orientation simultaneously; (2) it gives accurate orientations. Each pixel in the left and right image has three basic features: intensity, variation and orientation. When the BP network is used to compute the matching degree between two pixels from the left and right images, each input vector in fact consists of data from two  $7 \times 7$  windows (one from the left image, the other from the right) in which the centers are the pixels to be matched. In particular, the differences between the three features are calculated for each  $7 \times 7$  window to form a 147-Dimensional input feature vector. That is:

$$\text{Differences of Intensity} = f_{Li} - f_{Ri},$$

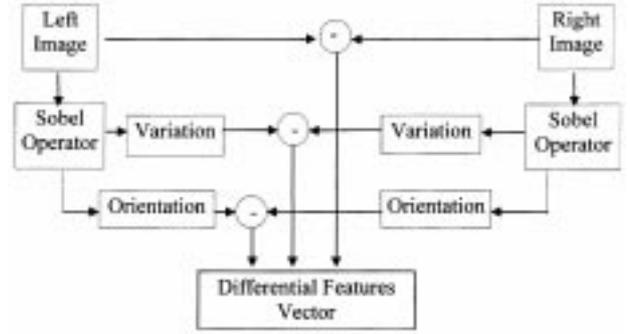


Fig. 4. Feature extraction in the BP network.

$$\text{Differences of Variation} = |\nabla f_{Li}| - |\nabla f_{Ri}|, \quad (5)$$

$$\text{Differences of Orientation} = \alpha_{Li} - \alpha_{Ri},$$

where  $f_{Li}$ ,  $|\nabla f_{Li}|$ ,  $\alpha_{Li}$  and  $f_{Ri}$ ,  $|\nabla f_{Ri}|$ ,  $\alpha_{Ri}$  are the intensity, variation and orientation values of the  $i$ th pixel in the left and right images, respectively. Note that these differential features constitute the actual input vectors both in the training and in matching processes. Figure 4 illustrates how this differential feature vector is generated. As will be shown later, the major advantage of using this differential feature is that it produces the property of image-independence; that is, a BP network trained with one pair of stereo images can be used to match another pair.

## 3. The BP Network

In traditional matching methods, one often needs to determine an *appropriate* window size. It must be large enough to cover intensity variations while small enough to avoid the effects of projective distortions (Kanade and Okutomi, 1994). Considering the fact that the functional relationship between the matching degree and the correlation in intensity patterns cannot be easily obtained, this paper presents use of the function approximation capability of the BP network to replace the traditional auto-correlation or SSD. The first clear insight into the versatility of neural networks for use in function approximation came with the discovery of Kolmogorov's theorem (Hecht-Nielsen, 1987), which essentially states that the BP network can implement any function of practical interest to any desired degree of accuracy. More formally, we rewrite the theorem as follows:

**Theorem of Function Approximation.** Given any  $\epsilon > 0$  and any  $L_2$  function  $f: [0,1]^n \rightarrow R^m$ , there exists a three-layer back-propagation neural network that can approximate  $f$  to within  $\epsilon$  mean squared error accuracy.

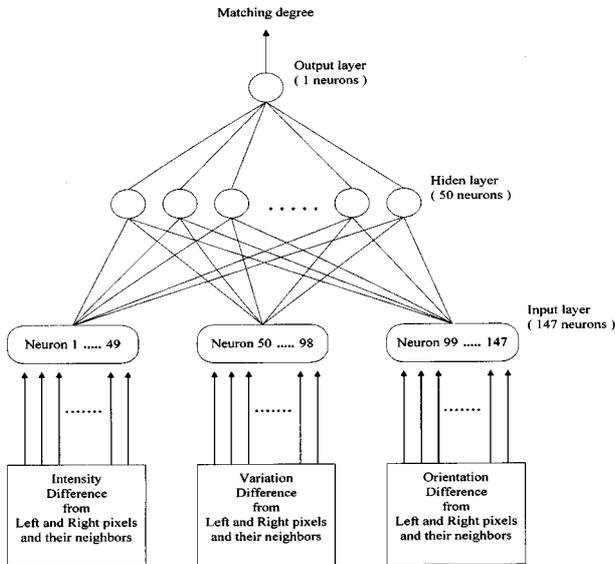


Fig. 5. The BP neural network.

The proof of this theorem, along with the related derivations of the back-propagation neural network, can be found in Hornik *et al.* (1989) and is thus omitted here.

Thus, unlike traditional area-based methods, using the supervised steepest descent algorithm (Haykin, 1998), the BP network can learn to *approximate* any functional mapping between well-defined input vectors and outputs. Figure 5 shows the architecture of the three-layer BP network that we used in this paper. The input layer has 147 neurons ( $=7 \times 7 \times 3$ , i.e., two  $7 \times 7$  windows from the left and right images); the hidden layer contains 50 perceptron-type (Haykin, 1998) neurons; a perceptron output neuron gives a real number, i.e.,  $0.0 \leq \text{match\_degree} \leq 1.0$ . A few feature vectors of matched and unmatched pixel-pairs are sampled as training data. During training, whenever a match pair appears in the input, the network is taught to output a target value of 1.0 by propagating the error  $= (1.0 - \text{match\_degree})$  back to the hidden layer; otherwise, error  $= (\text{match\_degree} - 0.0)$  when a mismatch occurs.

Initially, a primitive disparity map is generated by the BP network itself; then, the map is refined by a matching process which contains the same BP network. As in the training process, we use differential features to generate the primitive disparity map. We first note that, under the geometric constraint, the larger the image size is, the greater the value of maximum disparity can be. For example, for a  $128 \times 128$  image, the corresponding maximum disparity (denoted as  $D_{max}$ ) normally will not exceed 10 whereas for a  $256 \times 256$  image, the corresponding  $D_{max}$  normally is less than 20. Starting from the upper left pixel in the left image, the

best matching pixel in the right image is determined pixel by pixel. To be specific, for an arbitrary matching pixel in the left image  $P_L(x_1, y_1)$ , we compute the matching degrees for the next  $D_{max}$  pixels located to the left of the  $P_R(x_r, y_r)$  in the right image, where  $x_r = x_1$  and  $y_r = y_1$ . Among the candidates  $P_R(x_r, y_r)$ ,  $P_R(x_r - 1, y_r)$ ,  $P_R(x_r - 2, y_r)$ ,  $\dots$ ,  $P_R(x_r - D_{max}, y_r)$ , the one that produces the largest output match\_degree is chosen, say,  $P_R(x_r - 3, y_r)$ . Hence, the disparity for the point  $(x_1, y_1)$  is 3. Continuing in this manner until all the pixels are matched and the corresponding disparities are obtained, the primitive disparity map is drawn by assigning to the point that has the largest disparity the highest intensity level (say 255), and to the one with null disparity an intensity level of zero.

#### 4. The Matching Algorithm

After the initial disparity map is obtained, the BP network is incorporated along with a VC network and a matching algorithm to work on this map to achieve more accurate disparity. In doing so, we note that a pixel in the left image might have several pixels with high matching degrees in the right image. In order to effectively reduce the number of unmatched pixels and reduce the distortion caused by unnecessary interpolation during surface reconstruction, our idea is to first classify the pixels in both input images into  $T_C$  (e.g. 10) clusters using the VC network. Pixels belonging to the same cluster in the feature space should have similar characteristics; hence, pixels in the  $j$ th cluster in left image tend to have matching pixels in the same cluster in the right image. Performing clustering before matching should greatly reduce the search space needed to find correspondence pixels and increase the likelihood of correct matches.

Bearing with this in mind, Fig. 6 shows the matching process on a cluster-by-cluster basis. The BP network computes the matching degrees between an arbitrary pixel in the  $i$ th cluster in the left image and pixels in the same cluster in the right image. We denote those pixels having output match\_degree  $> 0.9$  as the candidate set  $C$ . Then, the best match (in the right image) for the said pixel will be determined by applying various constraints to each pixel in  $C$  so as to exclude the incorrect ones. Some useful constraints are as follows.

- (1) **Epipolar Line Constraint:** As shown in Fig. 2, a matching pair of pixels should always be located on the same epipolar line in parallel stereo images. This limits the search space for a given pixel in the left image to the corresponding scan line in the right image. However, if the axes of the cameras are not parallel, the epipolar lines in the

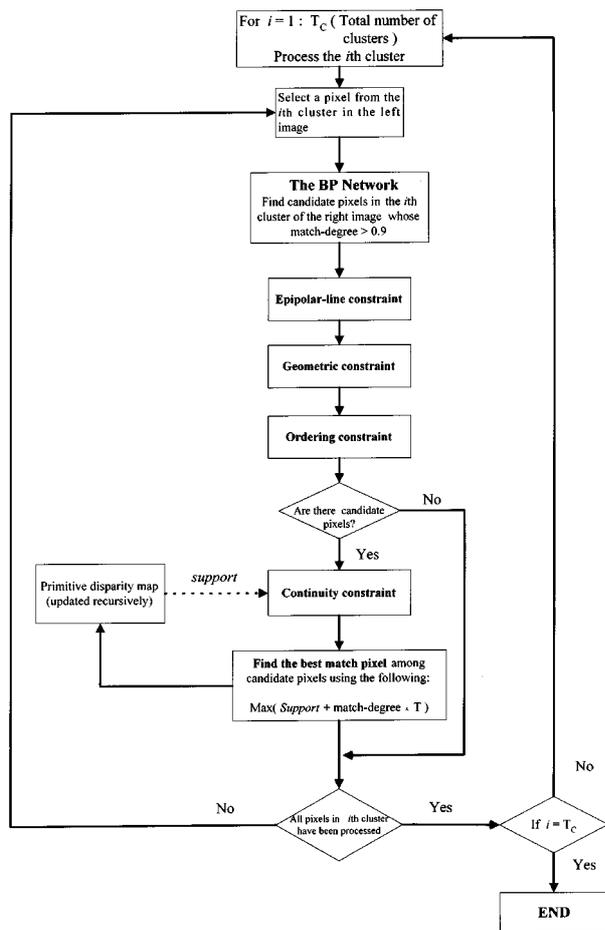


Fig. 6. Illustration of the matching process shown in Fig. 3.

images will appear to be inclined to the horizontal (Dhond and Aggarwal, 1989).

- (2) **Geometric Constraint:** For parallel stereo images, objects located at infinity distance will cause zero disparity. If there is a disparity, the position of a pixel in the right image should always be on the left side of its matching pixel in the left image.
- (3) **Ordering Constraint:** A left-right ordering relation between two pixels in the left image should have the same ordering for the correspondence pixels in the right image.
- (4) **Continuity Constraint:** This, in its most primitive form, refers to the depth constancy theory developed by Marr and Poggio (1976). Later, Grimson (1985) observed that the 1-D continuity constraint using Marr-Poggio theory caused difficulty in propagation of disparity at occluding boundaries between objects and along thin elongated surfaces. His implementation instead imposes a more complicated (but more accurate) regional continuity check on disparity in order

to validate matches. In our implementation, for a matching pixel  $P_L(x_l, y_l)$ , a candidate pixel  $P_R(x_r, y_r)$  in the right image looks for local support within a support window in the space of the *disparity map*. The range of the window is defined by

$$\{x_l - \omega \leq x' \leq x_l + \omega, y_l - \varepsilon \leq y' \leq y_l + \varepsilon\}, \quad (6)$$

where  $(\omega, \varepsilon)$  denote the width and the height of the region, respectively. The local *support* for  $P_R(x_r, y_r)$  is defined as the number of pixels  $P(x', y')$  inside the window (centered at  $(x_l, y_l)$ ) that have a disparity of  $(x_l - x_r)$ .

Note that the epipolar and geometric constraints were also employed by the BP network to generate the initial disparity map. In Fig. 6, the best match is chosen as the one that has the maximum value of  $(support + match\_degree \times T)$ , where  $T$  is a control factor. Increasing or decreasing  $T$  has the effect of suppressing or enhancing the *local support* inferred from the continuity constraint. Here, an interesting point is that the underlying physical meaning of *match\_degree* is analogous to the window size effect in the median filtering. For pixels in image details and edges (i.e., high frequency components), their support values are much smaller than those of pixels located in smooth areas. Thus, the matching process tends to preserve more high frequency components, provided that the value of *support* is kept relatively smaller than that of *match\_degree*. After the best match is determined, the new disparity value is used to replace the *old* disparity in the disparity map. In this recursive manner, the new disparity as well as its preceding replacements should provide more accurate local support so that subsequent pixels can be matched.

### III. VC for Learning Clustering

As implied in the previous section, the reason way we perform clustering before matching is that we can enjoy the advantage of feature-based methods, i.e., fast and accurate matching results. As a result, the proposed neural framework combines the advantages of feature-based and area-based methods. This section discusses how we apply the vitality conservation principle (VC) (Wang and Hsiao, 1997) to learning image clustering. Unlike the  $k$ -means method (Gose *et al.*, 1996), VC is a self-creating network, the number of clusters is self-organizing and does not need to be pre-specified.

#### 1. The Self-creating Network: VC

In the following, we will introduce the conserva-

tion principle and demonstrate how it can be incorporated into the competitive learning algorithm (Kohonen, 1989) to deal with the most important issue in developing a self-creating network, namely, the mechanism that determines *when* and *where* to generate a node during the training process. The basic idea behind vitality is to estimate the winning frequency of each individual node (neuron) so that nodes which are excessively or rarely (*in a relative sense*) accessed can be determined. The winner is the only node that has the right to update its weight vector. The total vitality in the proposed network at any time is a constant, hence, the name conservation. Combined with a procedure that redistributes the learning rate variables after generation, the conservation principle not only provides a novel approach to the problem of harmonizing equi-error and equi-probable criteria (Matsuyama, 1996), but also facilitates systematic derivations of various training parameters. The VC network will be shown to be (1) fast in terms of computation time, (2) smooth and incremental so that it can overcome the dead-node problem, stability-plasticity dilemma, and the deficiency of the local minimum, and (3) flexible enough for learning both vector quantization and clustering.

Define *vitality*  $\theta_k(t)$  as the measure of the winning frequency of the  $k$ th node after the  $t$ th input presentation. The  $k$ th node is selected as the winner if it is the least distance  $d_k$  from the present input vector. Not only is its weight vector updated, but its vitality is also increased by an amount of  $\Delta\theta^+(t)$ . On the other hand, an amount of  $\Delta\theta^-(t)$  will be subtracted from the vitality of the non-winning nodes. In this sense, vitality represents the *a priori* probability that a node will win at the time the input vector is presented. Considering that integrating an arbitrary probability density function is 1, it follows that *vitality conservation* can be stated as

$$\sum_{k=1}^{N(t)} \theta_k(t) = 1, \quad t \geq 0, \quad (7)$$

where  $M(t)$  is the total number of nodes at time  $t$ . Given an initial vitality  $\theta_k(0)$ , the vitality  $\theta_k(t)$  at time  $t$  can be formulated as

$$\theta_k(t) = \theta_k(0) + \sum_{i=0}^t [W_k(i)\Delta\theta^+(i) - (1 - W_k(i))\Delta\theta^-(i)], \quad (8)$$

where

$$W_k(i) = \begin{cases} 1, & \text{if } k\text{th node wins at time } i \\ 0, & \text{otherwise.} \end{cases}$$

After each input presentation, the vitality of the

winner is checked; if it is larger than the threshold, then that winner will generate a son node. The mother/son pair will then equally share the original vitality of the mother node. Thus, the following parameters are vital in training a VC network:

$\Delta\theta^+$ =increment in vitality for a winning node;

$\Delta\theta^-$ =decrement in vitality for a non-winning node;

$\theta_{born}(t)$ =dynamic threshold for node-generation;

$\theta_{ini}$ =initial vitality.

We start by letting  $\theta_k(0) = \theta_{ini}$  = the initial vitality of the  $k$ th node,  $k=1, 2, \dots, M(0)$ . Considering that all nodes have not received any input initially, it makes sense to assume that all nodes have the identical initial winning probability. Thus,  $\theta_k(0) = \frac{1}{M(0)}$ . For other training parameters, readers can refer to (Wang and Hsiao, 1997). The following exponential function with a decaying rate  $\lambda$  can be used to obtain the solution for  $\Delta\theta^-(t)$ :

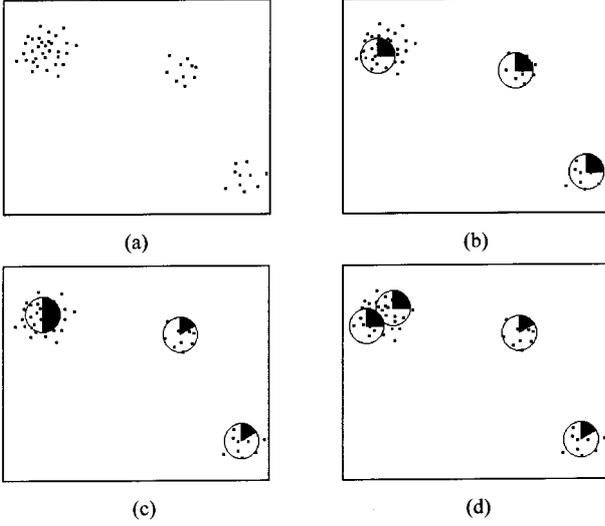
$$\Delta\theta^-(t) = \frac{1}{M(0)} \lambda e^{-\lambda t}, \quad (9)$$

and it follows that

$$\Delta\theta^+(t) = \frac{M(t) - 1}{M(0)} \lambda e^{-\lambda t}. \quad (10)$$

After each input presentation, the vitality values of all the nodes are updated. Considering the fact that equal partitioning of the probability space yields maximum entropy, it follows that half of the original vitality associated with the mother node should be given to the son node after it is generated. As to the threshold  $\theta_{born}(t)$ , it is easily seen that  $\frac{1}{M(t)} < \theta_{born}(t) < 1$  must hold for the generation process to work properly. This can be understood by recalling that the initial vitality =  $1/M(0)$ . Naturally, a VC network with a smaller  $\theta_{born}(t)$  grows faster than one with a larger  $\theta_{born}(t)$ .

After a node-generation, the coordinates of the present input vector are used as the insertion place for the new node. This simple insertion strategy has the advantages of saving computation time and avoiding the likelihood of producing dead nodes. Figure 7 illustrates this generation process. Figure 7(a) shows a 2-D structured input data with 4 nodes. Initially, each node (marked as a circle) has a vitality value of  $1/4$  (the bold area), as shown in Fig. 7(b). As the training proceeds, Fig. 7(c) shows that nodes located in the denser local density area accumulate more vitality



**Fig. 7.** Illustration of the generation process in VC. (a) 2-D input data; (b) 3 initial nodes, each with 1/3 vitality initially; (c) after certain input presentations; (d) a node with vitality exceeding the threshold generates a new son node and shares vitality with its son node.

accumulation, and vice versa. In Fig. 7(d), the node with vitality greater than  $\theta_{born}$  will generate a new node and share its vitality with the new node. Most competitive learning networks use a global rate  $\eta_k(t)$ , whose initial value  $\eta_{init}$  has a great effect on the final training results. Usually  $0 < \eta_{init} < 1$ , and  $\eta_{init}$  decays monotonically by a constant  $\gamma$ .

However, too large a value of  $\gamma$  will result in premature training, and too small a value of  $\gamma$  will result in convergence that is too slow. To solve this dilemma, we let

$$\gamma = \frac{\text{card}(x(t)) - 1}{\text{card}(x(t))}, \quad (11)$$

where  $\text{card}(x(t))$  is the cardinal number of  $x(t)$ . Denote  $\eta_k(t)^s$  as the learning rate of the  $k$ th node at the beginning of the  $s$ th iteration. Assume that the  $k$ th node never wins during a training iteration; as  $\text{card}(x(t)) \gg 1$ , then  $\eta_k(t)^s$  converges to

$$\begin{aligned} \eta_k(t)^s &= \eta_{init} \gamma^{\text{card}(x(t))}, \quad s=1 \\ &\approx \eta_{init} e^{-1}, \\ \eta_k(t)^s &= \eta_k(t)^{s-1} \gamma^{\text{card}(x(t))}, \quad s>1 \\ &\approx \eta_k(t)^{s-1} e^{-1}. \end{aligned} \quad (12)$$

From Eq. (12),  $\eta_k(t)$  is irrelevant to the number of input vectors. In VC, the learning rates of the new node and

its mother node are re-initialized to  $\eta_{init}$ . In addition, if the neighboring nodes around the mother/son can also react to the new situation more quickly incurred by the node-generation, then the quantization error can be decreased. Without loss of generality, we consider a 2-D uniform input stretching from  $(0,0)$  to  $(n,m)$ . Assuming that twelve codewords (i.e., nodes) are used, it can be shown that the mean Voronoi space (Haykin, 1998) of each codeword is  $(n \times m)/12$  with diameter  $\rho(0) \approx ((n \times m)/12)^{1/2}$ . For simplicity, when a new node is generated, the learning rates of its neighboring nodes are increased by an amount  $\Delta\eta_k$ , given by

$$\Delta\eta_k \propto e^{-\frac{dk}{\rho(t)}}. \quad (13)$$

As training proceeds, the dynamic equation of  $\rho(t)$  for  $N$ -dimension input data with length  $s$  in each dimension is approximated by

$$\rho(t) \propto \frac{s}{\sqrt{N}M(t)}. \quad (14)$$

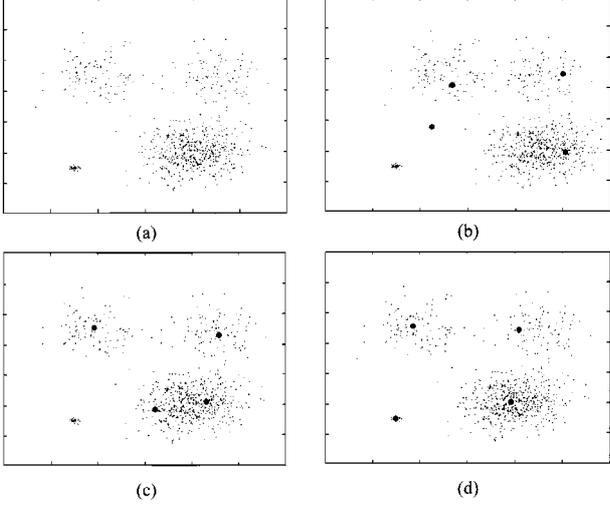
Thus, letting  $\eta_{min}$  be the stop learning rate, the local dynamic learning rate  $\eta_k(t)$  in VC can be stated as follows:

$$\eta_k(t) = \begin{cases} \eta_{init} & \left\{ \begin{array}{l} \text{when } t=0; \text{ or} \\ k\text{th node is a new born node; or} \\ k\text{th node is a mother node at time } t. \end{array} \right. \\ \text{Max}\{\eta_{min}, \text{Min}\{\eta_{init}, \gamma\eta_k(t-1) + I(t)\Delta\eta_k\}\}, & \text{otherwise,} \end{cases} \quad (15)$$

where

$$\text{the indicator function } I(t) = \begin{cases} 0 & \text{if } M(t) = M(t-1) \\ 1 & \text{if } M(t) > M(t-1). \end{cases}$$

As can be seen,  $\eta_k(t)$  is bounded by  $[\eta_{min}, \eta_{init}]$ . Usually  $\eta_{min} = 0.001$  will work well. Unlike the monotonically decreasing learning rate in SOFM (Kohonen, 1989), Eq. (15) provides dynamic perturbations in  $\eta_k(t)$  during the training process. While the decay factor  $\gamma$  guarantees a long-term decreasing trend for  $\eta_k(t)$ , the intermittent perturbations caused by node-generation can help avoid trapping in a local minimum. The redistribution of learning rate variables and vitality-sharing together create an important property: harmony



**Fig. 8.** (a) Input data. The clustering result obtained using (b) SOFM, (c)  $k$ -means, and (d) VC.

between *learning quality* and *growth rate*. This is because too many nodes generated in the earlier stage of training will force the network to use more computation time to complete an iteration. With Eq. (15) and vitality-sharing after a node-generation, VC attempts to keep the growth rate and learning quality in balance as training proceeds, in the sense that a new node will not be generated until enough input presentations have been presented. Note that implementation of Eq. (15) is computationally efficient because the input distortion  $d_k$  has been obtained earlier in selecting the winner node. Finally, once  $M(t)$  grows to the pre-specified  $M_f$ , the learning rate will *globally* decay, and the network will then enter the stage of fine-tuning and eventually converge in less than  $\ln(\eta_{min}-\eta_{init})/\ln(\gamma)$  input presentation.

## 2. From VQ to Clustering

In applying VC to clustering, we first add the  $x$  and  $y$  coordinates of the pixel into the original feature vector to form the input training vector [intensity, variation, orientation,  $x$ ,  $y$ ]. The left image is used to train the VC network; after training, the network can then be used to cluster both images. After clustering, every pixel (in the left and right images) will be labeled with a cluster number.

We used a 2-D input distribution to visually examine the performance of the VC in learning clustering. The input data contained four clusters. Figure 8 compares the clustering results obtained using SOFM,  $k$ -means (Gose *et al.*, 1996), and the VC network. Clearly, VC has the least number of misclassifications.

To obtain the clustering result in Fig. 8(d), our approach first employed VC to quantize the input data into  $M_f=32$  code vectors. Then, a merging algorithm was used, in a self-organizing manner, to aggregate the code vectors (for which the coordinates were represented by nodes in the VC network) into four clusters. Figure 9 shows the merging algorithm. Therefore, after the merging process was completed, a cluster label was assigned to each and every input data point. To provide better insight into how the algorithm works, Figure 10 shows the step-by-step results after applying the merging

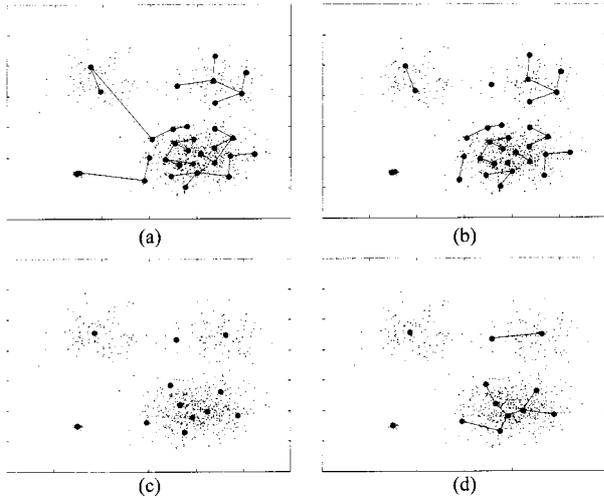
```

STEP_1:
  Search the associated codeword  $node_j$  for each input vector  $x$ , and assign  $x$  to  $node_j$ 
  For  $j = 1 : M_f$ 
    For  $k = 1 : M_f$ 
      If  $node_j$  is mother of  $node_k$ 
         $L(node_j, node_k) = 1$ 
      Else
         $L(node_j, node_k) = 0$ 
      End
    End
  End
STEP_2:
  For  $i = 1 : F$ 
     $LMAX_i = \max(\|L(node_j, node_k)\| \mid \text{if } L(node_j, node_k) = 1 \ \& \ node_j, node_k \in Family_i)$ 
    If  $\|L(node_j, node_k)\| = LMAX_i$ 
       $L(node_j, node_k) = 0$  //Remove the link
    End
  End
   $\bar{L} = \text{avg}(LMAX_i \mid j = 1 \dots F)$ 
STEP_3:
  For  $i = 1 : F$  // Some links may have been removed, so  $F$  is a variable
     $CF_i = \text{centroid}(node_j \mid j = 1 \dots M_f, node_j \in Family_i)$ 
    merge  $node_j \in Family_i$  into a node  $Q_i$ , the location of  $Q_i$  is  $CF_i$ 
    If label of  $x_k = node_j$  and  $node_j \in Family_i$ , label( $x_k, Q_i \mid k = 1 \dots V$ )
  End
STEP_4:
  For  $i = 1 : F$ 
    For  $j = 1 : F$ 
      If  $i \neq j$  &  $\|(CF_i, CF_j)\| < \bar{L}$ 
         $L(CF_i, CF_j) = 1$ 
      End
    End
  End
STEP_5:
  For  $i = 1 : F$ 
     $CF_i = \text{centroid}(Q_j \mid j = 1 \dots \text{number of } Q_j, Q_j \in Family_i)$ 
    merge  $Q_j \in Family_i$  into a node  $Q_i'$ , the location of  $Q_i'$  is  $CF_i$ 
    If label of  $x_k = Q_j$  and  $Q_j \in Family_i$ , label( $x_k, Q_i' \mid k = 1 \dots V$ )
  End

```

**Notes:**  $V$ : total number of input data.  $x_i$ : an arbitrary input data,  $i=1, 2, \dots, V$ .  $M_f$ : a pre-specified number of nodes after training the VC network.  $L(y,z)$ : create a logical link, where  $y, z$  can represent a node or a centroid (e.g., if  $y$  and  $z$  has a mother-son relation,  $L(y,z)=1$ , else  $L(y,z)=0$ ).  $Family_i$ : A family refers to a set in which for any arbitrary node or centroid, there exists a connected path (comprising a few links) to another node in the set.  $F$ : number of families, where  $F$  is a variable.  $CF_i$ : centroid of  $Family_i$ .  $LMAX_i$ : Max path (Euclidean distance) in  $Family_i$ .  $\bar{L}$ : average of  $LMAX_i$ . label( $x, var_j$ ): label  $x$  as  $var_j$ .

**Fig. 9.** The merging algorithm.



**Fig. 10.** Results after each step in the merging algorithm. The result after (a) STEP\_1, (b) STEP\_2, (c) STEP\_3, and (d) STEP\_4. The result after STEP\_5 is shown in Fig. 8(d).

algorithm to the quantization obtained using the VC network.

## IV. Experimental Results

In the following simulations, the BP network we used had  $7 \times 7 \times 3$  input neurons, 50 hidden neurons and 1 output neuron. Fifty matched pixel pairs and 50 unmatched pixel pairs constituted the training data set. It should be noted that, due to the generalization capability of the BP network, the training vectors can use a different stereo image pair; that is, a trained BP network is image-independent.

### 1. Clustering Result

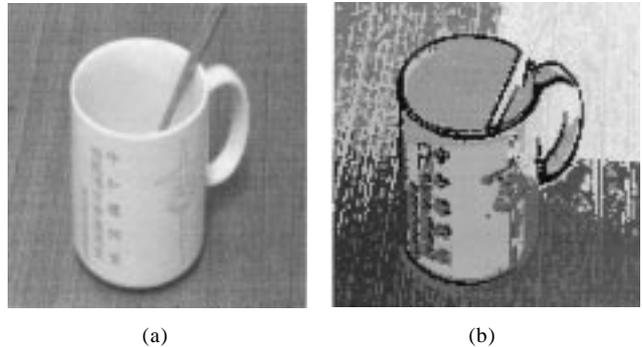
The test image was an indoor picture, shown in Fig. 11(a). To save computation time, the image was first divided into 16384 equal-size blocks, each with  $2 \times 2$  pixels. In this case, the feature vector for each block was the mean of four vectors, namely,  $[f(x_i, y_i), |\nabla f(x_i, y_i)|, \alpha(x_i, y_i), x_i, y_i], i=1 \dots 4$ . Then, 1000 5-D feature vectors were randomly selected to train the VC network that was initialized with 3 nodes and allowed to grow to 10 nodes at most. The network converged in less than 1 second on a Pentium-133 PC, which is a very small amount of total processing time. In comparison, the BP matcher needed about 40 minutes to finish execution. However, it should be noted that the efficiency of the BP matcher can be significantly improved if it is implemented with parallel hardware (e.g., neural chips). After training, all of the 16384 feature vectors were input into the VC network for

clustering. The result is shown in Fig. 11(b), where each cluster is plotted with a different grayscale level.

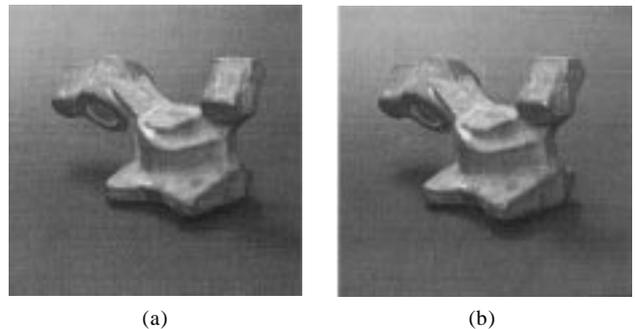
Testing on the Renault images shown in Fig. 12 led to similarly good results, as can be seen in Fig. 13. Note that the VC network was trained using only the left image, and that the trained network was used to cluster both the left and right images. Also, it is interesting to note that the “image segmentation” effect as seen in the tables in Fig. 11(b) and Fig. 13(b) is mainly due to the  $x$  and  $y$  input features. This is good because of the smaller searching space it offers. This also implies that matching can still proceed even if the epipolar line constraint is not applicable wherever the parallel image configuration is not available.

### 2. Disparity Matching

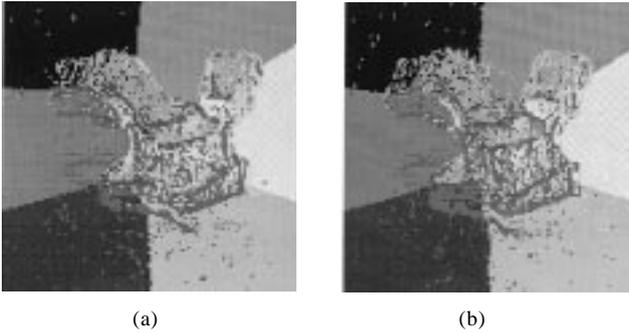
First, we will present the performance of the BP matcher obtained without using the matching algorithm on any constraints. The result will be compared with that obtained using SSD. The Fruit images shown in Fig. 14 were tested, and the results are shown in Fig. 15. In order to visualize the disparity map, each disparity value is represented by a gray level intensity.



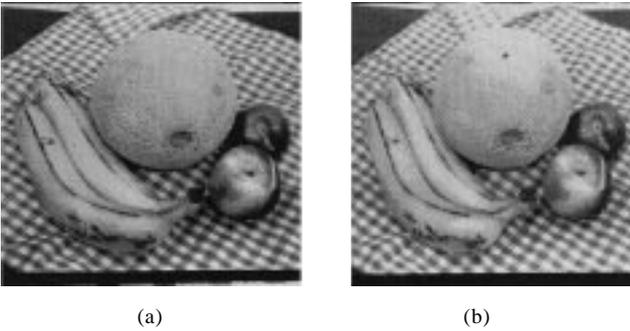
**Fig. 11.** (a) An indoor image (256x256): a cup with a spoon; (b) the clustering result.  $T_c=10$ .



**Fig. 12.** Original stereo images of Renault (256x256), (a) left view, (b) right view.



**Fig. 13.** Clustering results of the Renault images; (a) left image, (b) right image.  $T_c=10$ .



**Fig. 14.** Original stereo images of fruits (256×256); (a) left view, (b) right view.

Although both results show ambiguous regions in the smooth area, the disparity map obtained using the BP matcher evidently shows higher accuracy than does that obtained using SSD. This simulation result clearly indicates that the BP matcher can generate an acceptable disparity matching result.

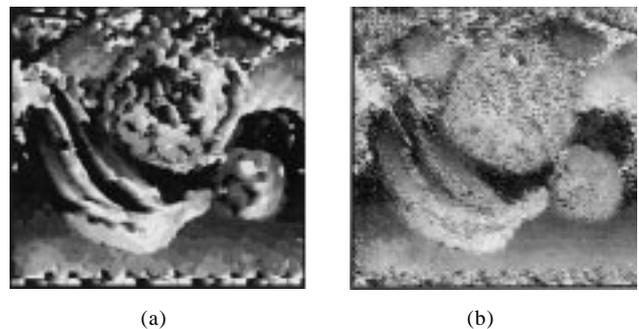
In the following, we will demonstrate how the proposed neural framework can be applied to achieve accurate disparity matching. The implementation involves a matching process incorporated along with the BP network and VC. Recall that in the matching process, a matching pixel  $P_L$  in the left image is checked against multiple candidates in the right image that have the same cluster label as  $P_L$ . In examining Fig. 6, one may wonder if the ordering sequence of the cluster to be processed can affect matching accuracy. Indeed, determining which cluster is to be processed first and which one next is a nontrivial task. That problem will be addressed in our future work.

Since this is a subject that is beyond the scope of this paper, we here will only give a practical example to show the feasibility of our proposed neural framework. In determining the sequence of clusters to be matched, some *a priori* knowledge can be used. For example, the human visual system is sensitive to intensity changes;

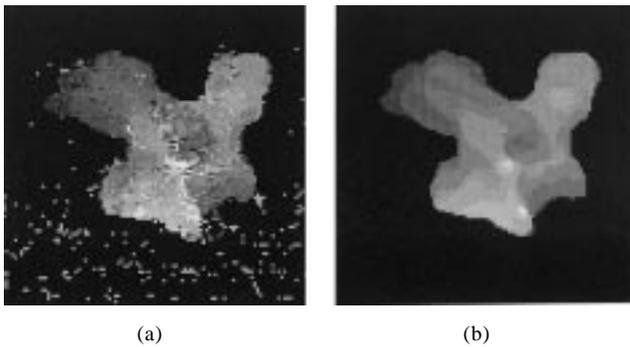
hence, the input feature of variation can be used as an index for sequencing the input clusters. At the extreme, one may cluster input images simply according to their levels of variation. In this case, the matching process shown in Fig. 3 is implemented using the following procedure:

- (1) Classify all pixels into several clusters based on their levels of variation. Starting from clusters with the highest average variation value and continuing to the lowest one, perform steps (2)-(4) for each cluster. Use the BP network to choose candidate pixels.
- (2) Unlike in step (1) where only the pixel with the largest matching degree can be chosen and used in generating the primitive disparity map, in this stage, multiple potential matching pixels can be chosen by the BP network and processed in step (3).
- (3) Apply the four previously described constraints to the candidate pixels in order to determine the best matching pixel. Calculate the disparity of the best matching pixel.
- (4) Update the disparity map with the newly obtained disparity.
- (5) Iterate steps (2)-(4) until all the clusters are processed, except for the one with the lowest variation level cluster.
- (6) Use interpolation techniques to process the area of the cluster with the lowest variation level and unmatched pixels.

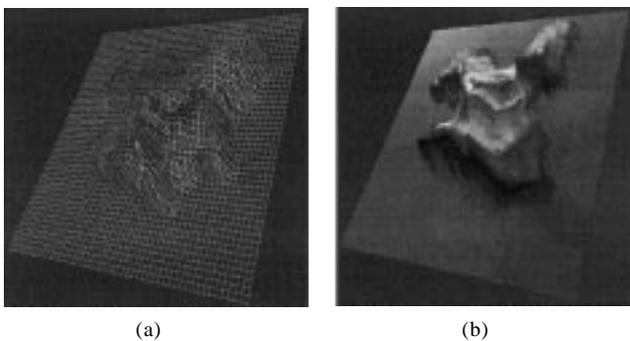
The rationale behind step (6) is that a very low variation area indicates a flat surface. If we continue to process the lowest variation cluster using the matching algorithm, we could severely blur image details due to the lack of significant features needed to obtain correct matches. The test image was the Renault stereo image, yet it is interesting to note that here we used the fruit images shown in Fig. 14 instead to train the BP network; afterwards, the Renault images were used



**Fig. 15.** Disparity map without any constraints or clustering results; (a) SSD, (b) the BP network with  $D_{\max}=15$ .



**Fig. 16.** (a) Disparity map of the stereo image, (b) after applying interpolation and median filtering to (a).



**Fig. 17.** (a) The 3-D mesh model reconstructed from the disparity map, (b) the reconstructed 3-D model with the left image used as its texture.

to generate the primitive disparity map. As can be seen in the final disparity map shown in Fig. 15, our matching algorithm can still work well, sufficiently verifying that the BP network indeed has the capabilities of generalization and function approximation. Note that the “salt and pepper” area in Fig. 16(a) can be easily removed by means of median filtering. The result is shown in Fig. 16(b). A 3D mesh reconstructed from the disparity map is shown in Fig. 17(a). Figure 17 (b) is the 3D model with the the left image used as its texture.

## V. Concluding Remarks

We have developed a stereo vision system capable of capturing depth information from stereo images. The system uses two neural networks, a VC network for learning clustering and a BP network for learning disparity matching. The BP network has been trained to act as an area-based disparity matcher. A matching algorithm incorporating the clustering results by means of VC is then applied to refine the primitive disparity map, which is generated by the BP network itself.

Constraints, such as epipolar lines, ordering, geometry and continuity, are used to effectively reduce the search space. Refining the disparity map is performed in a cluster-by-cluster manner. The matching process continues until all the clusters are matched. As such, our proposed system is more like a combination of the area-based and feature-based methods. We have also shown that the BP network trained using fruit images can be used to match the Renault images. This verifies the generalization capability of the BP network. It also implies that the knowledge extracted by the BP network is image-independent and useful for disparity matching.

We have successfully applied the self-creating VC network to learning image clustering. Analogous to the feature-based methods, the VC network can provide fast but good clustering performance by using fairly simple features, i.e., [intensity, variation, orientation,  $x, y$ ] for a single pixel, or [mean intensity, mean variation, mean orientation,  $\bar{x}, \bar{y}$ ] for an image block. While we believe that two pixels belonging to the same cluster have a good chance of being a matching pair, there still exist other unknown factors or outliers that can cause mismatches. One way to decrease the mismatch rate is to divide the image into smaller clusters (cells), and to group these cells based on specific objects. This in a sense may be viewed as segmentation of the image. An image can be viewed as a combination of several segments (objects), and each segment may consist of many small clusters. Such a course-to-fine approach or tree-structured representation of an image, we believe, can effectively reduce the occurrence of mismatch.

In the future works, we note that a more general *active* vergence stereo vision system (Ahuja and Abbott, 1993; Blake *et al.*, 1993; Grosso and Tistarelli, 1995) can be used to overcome the disadvantages of ordinary parallel stereo vision systems. Several cues (Ahuja and Abbott, 1993), such as the vergence angle and focus, can be used to obtain a more accurate range map of a scene. The internal parameters of the cameras from the stereo images also deserve investigation for use in improving the accuracy of the range map.

## Acknowledgment

This research was supported by the National Science Council of the Republic of China under Grant No. NSC 87-2213-E-019-007.

## Reference

- Ahuja, N. and A. L. Abbott (1993) Active stereo: integrating disparity, vergence, focus, aperture, and calibration for surface estimation. *IEEE Trans. Pattern Anal. Machine Intell.*, **15**, 1007-1029.
- Barnard, S. T. and M. A. Fischler (1987) Stereo Vision. *Encyclopedia of Artificial Intelligence*, pp. 1083-1090. John Wiley, New

## A Neural Framework for Disparity Matching

- York, NY, U.S.A.
- Blake, A., D. McCowen, H. R. Lo, and P. J. Lindsey (1993) Trinocular active range-sensing. *IEEE Trans. Pattern Anal. Machine Intell.*, **15**, 477-483.
- Capurro, C., F. Panerai, and G. Sandini (1996) Dynamic vergence. *IEEE Conf. Intelligent Robots and Systems*, **3**, 1241-1248.
- Cochran, S. D. and G. Medioni (1992) 3-D surface description from binocular stereo. *IEEE Trans. Pattern Anal. Machine Intell.*, **14**, 981-994.
- Dhond, U. R. and J. K. Aggarwal (1989) Structure from stereo—a review. *IEEE Trans. Syst., Man, Cybern.*, **3**, 1489-1510.
- Gonzalez, R. C. and R. E. Woods (1992) *Digital Image Processing*. Addison-Wesley, Reading, MA, U.S.A.
- Gose, E., R. Johnsonbaugh, and S. Jost (1996) *Pattern Recognition and Image Analysis*. Prentice Hall, Upper Saddle River, NJ, U.S.A.
- Grimson, W. E. L. (1985) Computational experiments with a feature-based stereo algorithm. *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-7**, 17-34.
- Grosso, E. and M. Tistarelli (1995) Active/dynamic stereo vision. *IEEE Trans. Pattern Anal. Machine Intell.*, **17**, 868-879.
- Haykin, S. (1998) *Neural Networks*, 2nd Ed. Prentice-Hall, Upper Saddle River, NJ, U.S.A.
- Hecht-Nielsen, R. (1987) Counter propagation networks. *Proc. of the Int'l. Conf. on Neural Networks*, **2**, pp. 19-32. IEEE Press, New York, NY, U.S.A.
- Hornig, J. L., J. L. Semmlow, G. K. Hung, and K. J. Ciuffreda (1998) Initial component control in disparity vergence: a model-based study. *IEEE Trans. Biomed. Eng.*, **45**, 249-257.
- Hornik, K., M. Stinchcombe, and H. White (1989) Multilayer feed forward networks are universal approximators. *Neural Networks*, **2**(5), 359-366.
- Kanade, T. and M. Okutomi (1994) A stereo matching algorithm with an adaptive window: theory and experiment. *IEEE Trans. Pattern Anal. Machine Intell.*, **16**, 920-932.
- Kohonen, T. (1989) *Self-Organization and Associative Memory*, 3rd Ed. Springer-Verlag, New York, NY, U.S.A.
- Marapane, S. B. and M. M. Trivedi (1989) Region-based stereo analysis for robotic applications. *IEEE Trans. Syst., Man, Cybern.*, **19**, 1447-1464.
- Marr, D. and T. Poggio (1976) Cooperative computation of stereo disparity. *Science*, **194**, 283-287.
- Matsuyama, Y. (1996) Harmonic competition: a self-organizing multiple criteria optimization. *IEEE Trans. Neural Networks*, **7**, 652-668.
- Moravec, H. (1980) *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. Ph.D. Dissertation. Stanford Artificial Intell. Lab., AIM-340, Stanford Univ., Stanford, CA, U.S.A.
- Nasrabadi, N. M. and C. Y. Choo (1992) Hopfield network for stereo vision correspondence. *IEEE Trans. Neural Networks*, **3**, 5-13.
- Nasrabadi, N. M., W. Li, B. G. Epranian, and C. A. Butkus (1989) Use of Hopfield network for stereo vision correspondence. *IEEE ICSMC*, **2**, 429-432.
- Nayar, S. K. and Y. Nakagawa (1994) Shape from focus. *IEEE Trans. Pattern Anal. Machine Intell.*, **16**, 824-831.
- Nevatia, R. and K. Babu (1980) Linear feature extraction and detection. *Comput. Graphics Image Processing*, **13**, 257-269.
- Papadimitriou, D. V. and T. J. Dennis (1996) Epipolar line estimation and rectification for stereo image pairs. *IEEE Trans. Image Processing*, **5**, 672-676.
- Park, S. Y., Y. B. Lee, and S. I. Chien (1998) Linear relation for vergence control of parallel stereo camera. *IEE Electronics Letters*, **34**, 255-256.
- Subbarao, M. and C. Tao (1995) Accurate recovery of three-dimensional shape from image focus. *IEEE Trans. Pattern Anal. Machine Intell.*, **17**, 266-274.
- Wang, J. H. and C. P. Hsiao (1997) Representation-burden conservation network applied to learning VQ. *Neural Processing Letters*, **5**(3), 209-217.

# 立體視覺系統之對應點匹配—應用神經網路

王榮華\* 蕭志平\*\*

\*國立臺灣海洋大學電機工程學系

\*\*數通科技股份有限公司

## 摘 要

本文提出一套神經網路架構應用於解決立體視覺對應點匹配的問題。架構中應用了兩種神經網路，(1)活動力守恆(VC)網路—用於學習叢集分類；(2)倒傳遞網路(Back-Propagation, BP) —用於學習對應點的匹配。其中VC網路是由我們所發展完成的一個自我發展型(self-creating)神經網路。藉由VC網路可最大化熵值(Entropy)的特性，而可用來求得一個近似最佳向量量化器。我們亦提出一融合演算法(Merging algorithm)利用前述向量量化的結果以計算出最佳叢集分類。不像 $k$ -means、SOFM，吾人不需預先指定所要分類資料的群數。論文中我們說明了VC網路如何應用於增加對應點匹配準確率。首先，藉由Sobel之運算得到影像像素之特徵，如：變化量和方向性。這些特徵被用來訓練BP網路成為一個匹配器(吾人可使用具代表性的影像來訓練BP網路，以期具有概泛化的能力，如此吾人即可應用此匹配器於所有的影像)。訓練之後的匹配器可用來產生一個初始的深度圖。這些特徵再加上 $x$ 和 $y$ 座標值經由VC網路分類，而BP便可利用分類之後的叢集尋找對應點。其中使用了一些限制條件來縮小搜尋空間，如：幾何限制(geometry constraint)，一致線限制(epipolar line constrain)和順序限制(ordering constraint)；另外再使用連續限制(continuity constraint)和初始的深度圖以找出最佳的匹配點。實驗結果顯示此一結合BP網路和VC網路之BP匹配器確實兼具區塊基礎(Area-based)及特徵基礎(Feature-based)兩種傳統方法之優點。