Proceedings of International Conference
on Artificial Intelligence

A New Hopfield Neural Network-Based Stereo Matching

Fang-Chih Tien[1]
C. Alec Chang[1]
ChaoTon Su[2]

1. Department of Industrial Engineering
University of Missouri - Columbia
113 Engineering Building West,
Columbia, MO 65211, USA

2. Department of Industrial Engineering and Management,
National Chiao Tung University, Taiwan
TEL: (011-886-35) 722-101

Abstract

A Hopfield neural network-based stereo matching algorithm is presented in this paper. We formulate an effective energy function which is combined with corelation, uniqueness, epipolar line, disparity and 0-1 integer properties. This energy function is minimized by a two-dimensional asynchronous Hopfield neural network. This proposed method is implemented and compared with Nasrabadi's approach. It is found that the proposed method is superior in computation speed and feature matching performances.

1. Introduction

Artificial neural networks are popular methods adopted in many different areas of research such as pattern recognition, optimization, process control and image processing. Among those networks, Hopfield neural networks are known as effective heuristic approaches for solving optimization problems [5][15]. These applications include weighted point matching [2], path determination [1], analog-to-digital (A/D) conversion [16], pattern recognition [13] and the traveling salesman problem [17]. By minimizing the cost function (or energy function), a global solution (or approximation) can be obtained by systematically searching in a multi-variable space [3][4]. Recently, it has been used in solving the correspondence problem.

Mousavi and Schalkoff [11] develop a neural network method to solve both the feature extraction and stereo matching problems. In their method, edges are detected by a multi-layered network that extracts the combination of spatial intensity gradient (first derivative) and zero-crossing (second derivative) of smoothed images. Similarity and epipolar constraints are formulated to match the extracted primitives. A Hopfield network is used in minimizing an energy function subject to these constraints. They improve the formulation by using intra-scanline and inter-scanline constraints to enforce the energy function [10]. Parvin and Medioni [14] use a Hopfield neural network for a multi-scale strategy in matching extracted features. In their network, the three constraints that need to be satisfied are *local*, *adjacency* and *global* constraints. Hu and

Siy [6] propose a novel class of optimization problems called the "Picking Stone Problem" (PSP). It is solved by using a Hopfield neural network to satisfy uniqueness and ordering constraints. Nasrabadi and Choo[12][13] use a two-dimensional binary Hopfield neural network for object recognition and stereo features matching. In their method, features of interest are extracted by the Moravec operator for matching in both images. Then, an energy function is built and minimized subject to compatibility and uniqueness constraints. Lee *et. al.* [7] apply a Hopfield neural network to the stereo correspondence for extracting the 3D structure of a scene. Similarity, smoothness and unique constraints are transformed into the form of an energy function.

In general, all of these algorithms reviewed suffer from excessive computation due to the use of the interconnected relationships between neurons. Computation and storage increases considerably when the number of matching points is large. Also, in many methods, the epipolar line and disparity constraints are not considered in the energy function. For this reason, their proposed algorithms result in more multiple matches or mismatches, and depend heavily upon a false target removing process. Therefore, a new formulation of the stereo matching problem is presented in this paper.

2. Matching by Asynchronous Hopfield Neural Network

The first stage of the proposed algorithm is to find the points of interest, such as corners or edge points, in both images. To find these salient points, many techniques have been developed [7][9]. A modified Sobel operator is adopted in order to reduce the computation. Also, instead of using the minimum variance in the sized window, the sum of the four directional gradient magnitudes is used as an extraction base to produce more consistent effects in both images. Also, by selecting the maximum gradient magnitude within a sized window, a filtering process is adopted to screen out insignificant feature points [12].

When feature points in the left and right images are extracted, a two-dimensional Hopfield network is used to match the conjugate points. Thus, an energy function is constructed as

$$E = \underbrace{A\cdot\sum_{i=1}^{m}\sum_{j=1}^{n}\xi_{ij}n_{ij}}_{(a)} + \underbrace{B\cdot\sum_{i=1}^{m}(1-\sum_{j=1}^{n}n_{ij})^2}_{(b)} + \underbrace{C\cdot\sum_{j=1}^{n}(1-\sum_{i=1}^{m}n_{ij})^2}_{(c)}$$

$$+ \underbrace{D\cdot\sum_{i=1}^{m}\sum_{j=1}^{n}(e-a_{ij}n_{ij})^2}_{(d)} + \underbrace{F\cdot\sum_{i=1}^{m}\sum_{j=1}^{n}(d-b_{ij}n_{ij})^2}_{(e)}, \quad (1)$$

where the coefficients, A, B, C, D, and F are positive weights of component factors. This energy function combines the properties of correlation (a), uniqueness (b and c), epipolar line (d), and disparity (e) between each pair of feature points. The values of these parameters are determined by trial-and-error. Since all components are positive, the energy function has a lower bound.

The Hopfield neural network is a single layer feedback network, and its topology is shown in Figure 2. Following the postulates of Hopfield, it consists of mxn neurons having threshold T_{ij}. Denoting w_{ijkl} as the weight value connecting the output of the klth neuron with the input of the ijth neuron. The feedback input to the ijth neuron is equal to the weighted sum, which can be represented as

$$net_{ij} = \sum_{k=1}^{m}\sum_{l=1}^{n} w_{ijkl} \cdot n_{ij} - T_{ij} ,$$

$$\text{for } i=1, 2, \ldots, m; j=1, 2, \ldots, n \qquad (2)$$

A stochastic activation rule is necessary when the Hopfield neural network is searching for the minimum. A change state of neuron n_{ij}, denoted by Δn_{ij}, causes a change of energy, denoted by ΔE_{ij}. If a randomly selected neuron is changed from zero to one and the value of net_{ij} calculated is negative, then a change state is invoked (else the neuron stays unchanged). On the other hand, if a neuron is changed from one to zero and net_{ij} value is positive, then a change state is also invoked. The change of energy depends on the product of net_{ij} and Δn_{ij} as

$$\Delta E = \Delta n_{ij} \cdot net_{ij} . \qquad (3)$$

3. Operational Procedure of Asynchronous Hopfield Neural Network

An operational procedure for solving the stereo matching problem is summarized as follows:

(1) *Construction of the Energy function.* The energy function is constructed as shown in Equation (1). Let n_{ij} be the selected neuron that will change state to n'_{ij}, and then the net_{ij} is calculated as

$$net_{ij} = \Psi_{ij} - T_{ij} \qquad (4)$$

where

$$\Psi_{ij} = B \cdot \sum_{l=1}^{n} (n'_{il} + n_{il}) + C \cdot \sum_{k=1}^{m} (n'_{kj} + n_{kj}) \qquad (5)$$

and

$$T_{ij} = \{A \cdot \xi_{ij} - 2B \cdot -2C + D \cdot [a_{ij}^2 - 2e \cdot a_{ij}] + F \cdot [b_{ij}^2 - 2d \cdot b_{ij}]\} \cdot \qquad (6)$$

(2) *Initialization.* Let n_{ij} denote the ijth neuron in a vector s. Randomly assign the n_{ij} by 0 or 1. The algorithm is initialized by setting

$$s(0) = (n_{11}, n_{12}, \ldots, n_{ij}, \ldots, n_{mn}) \qquad (7)$$

(3) *Iteration and Convergence.* Update the elements of the state vector asynchronously. The stochastic updating rule is summarized as below

if $n_{ij} \longrightarrow 1$ *and* $net_{ij} > 0$,

then change state, $n'_{ij} = 1$,

if $n_{ij} \longrightarrow 0$ *and* $net_{ij} < 0$,

then change state, $n'_{ij} = 0$,

else *no change.* (8)

Repeat the updating and recursive iteration as

$$s(t+1) <= s'(t) = (n_{11}, n_{12}, \ldots, n'_{ij}, \ldots, n_{mn}) \qquad (9)$$

where s' is the state vector after updating, until the state vector s remains unchanged.

(4) *Outputting.* Let s_{fixed} denote the stable state computed at the end of step 3. The resulting output vector of the network is

$$s_{fixed} = (n_{11}, n_{12}, \ldots, n_{ij}, \ldots, n_{mn}) \qquad (10)$$

After the correspondence is made, there will be some incorrect matches created. False target removing is a process to remove the multiple-matched and mismatched pairs. By analyzing the properties of disparity in both the vertical axis and the horizontal axis, we can detect and remove the imperfect matches caused by system noise.

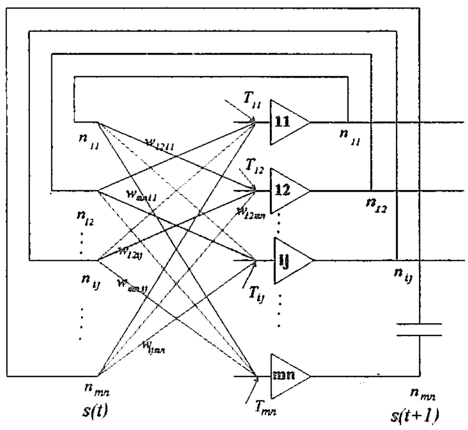
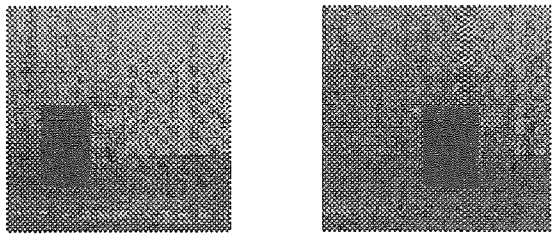


Figure 1. The topology of Hopfield networks

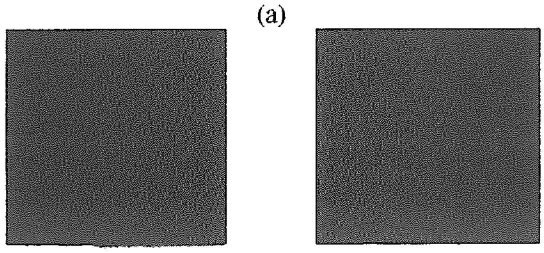
4. Implementation and Discussion

The proposed algorithm is implemented on several scanned images, and one of examples are shown here for further discussion. A test part is shown in Figure 2. Paired images are acquired by two cameras that are mounted on a precise X-Y table and the base line distance is fixed at 1 inch.

At the first stage, the feature points in the left and right images are extracted by the adopted feature extraction method. The Figure 2(a) shows the original scanned images. The number of feature points found in the left and right images are N_l= 30 and N_r=29, respectively. Before running the network, the coefficients required by trial-and-error are first determined and are summarized in Table 1, which is listed in Appendix. The energy function is then constructed by following Equation (1). In the initialization stage, the state vector $s(0)$ is randomly assigned by 0 or 1. Then, a neuron n_{ij} is randomly selected, and its Δn_{ij} and net_{ij} are computed. Following the activation rule, we can determine the change state of the neuron and derive a new state vector $s(t+1)$. Recursively, the new state vector $s(t+1)$ is fed back to the system until the network reaches a stable state. After the Hopfield neural network is stable, the outputting vector s_{fixed} and disparity for each conjugate pair are obtained. Obviously from the Figure 2(b), there are unmatchable feature points that exist due to the missing part and imperfect illumination. The false target removing algorithm is required at this stage. The final matching results are shown in Figure 3, where the points with no mark are correctly matched.



(a)



(b)

Figure 2. (a) Left and right images, (b) Features extracted with a 9x9 screening window

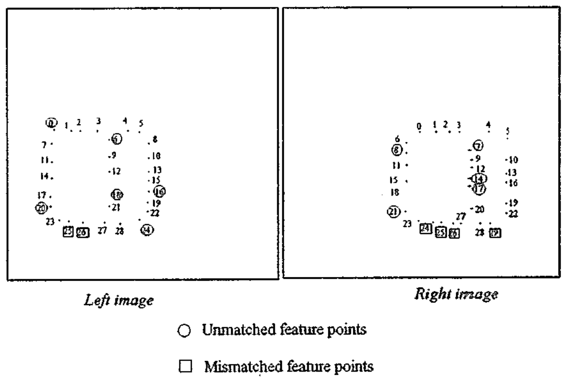


Figure 3. Image matching of test part

A comparison between the proposed method and Nasrabadi's method [12] is made in this paper. Both of the algorithms are implemented in *C* language and run on an IBM 6000, model 580 minicomputer. The same example presented previously is used for bench marking. The comparison is based on two performance measures: *computation efficiency*, and *matching performance.* The matching performance evaluates the matching results by means of the number of mismatches, missed matches, multiple matches, and correct matches. The matching speed and storage are considered as the matching efficiency. As shown in Table 2 in Appendix, the proposed algorithm presents superior performance in both measures. Also, it consistently performs better in speed and matching than Nasrabadi's method in other 14 experiments. When the number of points is large, the computation speed is considerably faster. It is obvious that in Nasrabadi's method the interconnection relationship, which is expressed as $\sum_{i}^{m}\sum_{j}^{n}\sum_{k}^{m}\sum_{l}^{n}T_{ijkl}n_{ij}n_{kl}$, requires the computation order $(mxn)^2$. In contrast, the epipolar line and disparity constraints used in the proposed method have a computation order (mxn). Additionally, the computer memory used to store the weight matrix w_{ij} is mxn times less than T_{ijkl} based on the same reason. Theoretically and empirically, the proposed algorithm not only has a fast matching speed, but it also offers superior matching results.

References

[1] Cavalieri , S., Stefano, A. Di and Mirabella O. (1994) Optimal Path Determination in a Graph by Hopfield Neural Network, *Neural Networks*, Vol. 7, No. 2, 387-404, 1994

[2] Hertz, John, Krogh, Anders and Palmer, Richard G., (1991) Weighted point matching, *Introduction to the theory of neural computation*, Addison-Wesley publishing company, 72-79

[3] Hopfield, J. (1982) Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proc. Nat. Acad. Sci.*, vol. 79, 2554-2558

[4] Hopfield, J. (1982) Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons, *Proc. Nat. Acad. Sci.* (81), 3088-3092

[5] Hopfield, J. and Tank, D. W. (1985) (1985) 'Neural' Computation of Decisions in Optimization Problems, *Biol. Cybern.*, Vol. 52, 141-152

[6] Hu, Joe-E and Siy, Pepe (1993) An Ordering-oriented Hopfield Network and its Application in Stereo Vision, *SPIE* Vol. 1965, Applications of Artificial Neural Networks IV, 556-567, 1993

[7] Huang, M. S. Lew, T. S. and Wong, Kam (1994) Learning and Feature Selection in Stereo Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 9, 869-881

[8] Lee, J. J., Shim, J. C. and Ha, U. H. (1994) Stereo Correspondence Using the Hopfield Neural Network of A New Energy Function, *Pattern Recognition*, Vol. 27, No. 11, pp. 1513-1522

[9] Moravec, H. (1981) *Robot Rover Visual Navigation*, Ann Arbor, MI: U.M.I. Research Press

[10] Mousavi, M. S. and Schalkoff, R. J. (1990) Stereo Vision: a Neural Network Application to Constraint Satisfaction Problem, *SPIE Vol. 1382 Intelligent Robots and Computer Vision IX: Neural, Biological*, and 3-D Methods, 228-239

[11] Mousavi, M. S. and Schalkoff, R. J. (1990) A Parallel Distributed Algorithm for Feature Extraction and Disparity Analysis of Computer Images, *Proceedings of IEEE Computer Society Conference on Computer Vision an Pattern Recognition*, 428-435

[12] Nasrabadi, Nasser M. and Choo, Chang Y. (1992) Hopfield Network for Stereo Vision

Correspondence, *IEEE Transactions on Neural Networks*, Vol. 3, No. 1, 5-13

[13] Nasrabadi, Nasser M. and Li, Wei (1991) Object Recognition by a Hopfield Neural Network, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No 6, Nov./Dec., 1523-1535

[14] Parvin, B. and Medioni, G. (1991) A Layered Network for the Correspondence of 3D Objects, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, 1808-1813

[15] Robert Hecht-Nielsen, *Neurocomputing*, Addison-Wesley Publishing Company, 1990

[16] Tank, David and Hopfield, J. (1986) Simple 'Neural' Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit, *IEEE Transactions on Circuits and Systems*, Vol. CAS-33, No. 5

[17] Wilson, G. V. and Pawley, G. S. (1988) On the Stability of the Traveling Salesman Problem Algorithm of Hopfield and Tank, *Biological Cybernetics*, Vol. 58, 63-70

Appendix

Exp.	*window size*	$N_l \times N_r$	d	e	$A:B:C:D:F$
Coeff	9	30×29	58	2	5:1750:1750:1:100

Table 1. The Coefficients used in examples

Criteria	Nasrabadi's	Proposed method
Energy function	$E = -(\frac{1}{2})\sum_{i=1}^{N_l}\sum_{k=1}^{N_r}\sum_{j=1}^{N_l}\sum_{l=1}^{N_r} T_{ikjl} V_{ik} V_{jl}$ $+\sum_{i=1}^{N_l}(1-\sum_{k=1}^{N_r} V_{ik})^2 + \sum_{k=1}^{N_r}(1-\sum_{k=1}^{N_l} V_{ik})^2$	$E = A \cdot \sum_{i=1}^{m}\sum_{j=1}^{n} \xi_{ij} n_{ij} + B \cdot \sum_{i=1}^{m}(1-\sum_{j=1}^{n} n_{ij})^2 + C \cdot \sum_{j=1}^{n}(1-\sum_{i=1}^{m} n_{ij})^2$ $+ D \cdot \sum_{i=1}^{m}\sum_{j=1}^{n}(e - a_{ij} n_{ij})^2 + F \cdot \sum_{i=1}^{m}\sum_{j=1}^{n}(a - b_{ij} n_{ij})^2$
Complexity	$O[(mxn)^2] \approx O[n^4]$	$O[(mxn)] \approx O[n^2]$
Storage	$O[(mxn)^2] \approx O[n^4]$	$O[(mxn)] \approx O[n^2]$
Example size 30x30	810,000 times per iteration	900 times per iteration
Estimated time	7.5 hrs.	30 sec.
Running time	2.5-3 hrs.	39-49 sec.
Correct matches	18	20
Multiple matches	2	1
Mismatches	3	3
Missed matches	2	0

Table 2. A comparison of matching performance and computation time