Detecting Corresponding Segments across Images Using Synchronizable Pulse-Coupled Neural Networks

Xiaofu Zhang
Complex Adaptive Systems Laboratory
ECECS Department
University of Cincinnati
Cincinnati, OH 45221

Ali A. Minai
Complex Adaptive Systems Laboratory
ECECS Department
University of Cincinnati
Cincinnati, OH 45221

Abstract

Computational models of locally connected networks of synchronizable neural oscillators — notably pulse-coupled neural networks (PCNN) and locally excitatory globally inhibitory oscillator networks (LEGION) — have been applied to image segmentation. In this paper, we report on research that explores a simple 2-layer PCNN-like network for determining corresponding segments in two images. If the two images are frames in a video sequence this can be used for motion detection and, thus, for motion-based segmentation. More generally, it can be used for finding specific objects in fresh views of a previously imaged scene. The proposed algorithm is called *bidirectional gated block-matching (BGBM)*.

1 Introduction

Computational models of locally connected networks of synchronizable neural oscillators are of interest from both the neuro-engineering and computational neuroscience perspectives. Synchronization was proposed by von der Malsburg [9] and Eckhorn et al [3] as a possible mechanism for feature linking, experimental evidence for which was later reported [5, 6]. This has led to the development of the *pulse-coupled neural network (PCNN)* model by Eckhorn and colleagues [3, 4], and of the *locally excitatory, globally inhibitory oscillatory network (LEGION)* model by Wang [10]. Both models have been applied to several image analysis tasks, but most notably to segmentation [7, 8, 11, 1].

Most applications of the PCNN approach have focused on intensity segmentation — using recruitment based on similarity of intensity and spatial proximity. In this paper, we report on research that explores a simple PCNN-based network for determining corresponding segments in two images, using recruitment based on neighborhood correspondence across two images. If the two images are frames in a video sequence this can be used for motion detection and, thus, for motion-based segmentation. More generally, it can be used for finding specific objects in fresh views of a previously imaged scene.

The literature on image analysis includes several approaches to solving the correspondence problem and determining optic flow, many of which are very effective. However, methods based on synchronizable neurons are of special interest for two reasons: 1) The approach is inherently amenable to parallel implementation; and 2) It may help elucidate possible mechanisms in the early visual system. We present our method as a particularly simple approach to a difficult problem — one that can, with further enhancement, address more complex tasks.

2 Approach

A PCNN is a 2-dimensional (typically $N \times N$ grid of neurons, each coupled to its 4 or 8 neighbours. The firing of a neurons excites the neighbors it is directly connected to via a pulse — representing the action potential in real neurons. In application to segmentation, the neurons are in 1-to-1 correspondence with image pixels, and the intensity of its pixel represents external input to each neuron. The neurons charge up based on external and lateral input. When a neuron's activation reaches a threshold, it fires a pulse (which excites its neighbors) and is discharged. Thus, starting off with similar initial activation, neurons with the most external excitation (highest intensity pixels) fire first, in turn triggering those neighboring neurons which have similar, though slightly lower external input. The wave of activity thus spreads until it encounters a significantly lower intensity boundary on all sides. The total region covered by the wave can then be seem as a segment. Several refinements can be used to improve the process, including the use of modifiable lateral weights and global inhibition for selecting between candidate segments.

Essentially, the mechanism described above provides a neurally implemented algorithm for determining which groups of pixels in an image satisfy two conditions: 1) They are of similar intensity; and 2) They form a compact collection. Condition (1) alone can be determined by simple thresholding, but PCNNs, due to their local connectivity, impose the neighborhood condition as well. The strength of PCNN's lie in the fact that the conditional thresholding is done in a parallel distributed way that is amenable to adaptation and hardware implementation. In extending PCNNs to detecting matching segments in two images, our goal is to generalize the conditional thresholding mechanism from one based on pixel intensity to one that considers similarity of two-dimensional blocks, as used in optic flow algorithms [2].

Consider the case with two $n \times n$ intensity images, $I_A(i,j)$ and $I_B(i,j)$. As for intensity-based segmentation, the segment starts with a single seed pixel in image I_A. A corresponding "best match" pixel is found in image I_B. The segment then grows in image I_A by recruiting neighbors of currently active segment pixels *provided the distribution of intensity in their neighborhood matches the distribution around a candidate pixel in image* I_B. As long as such pixels can be found, the segment grows. When no further pixels satisfy the condition, the segment is complete. Essentially, the question being answered by the system is this: *Starting from the seed pixel in* I_A*, find the largest segment which has a corresponding (matching) segment in* I_B. We implement this approach using a 2-layer PCNN-like architecture, and the resulting network is called the *bidirectional gated block-matching (BGBM) network.*

An alternative approach for identifying corresponding segments in two images is motion-based segmentation. This typically follows a two-step process. First, block matching between the two images is used to estimate the optic flow explicitly at each pixel [2]. Then, contiguous regions with similar optic flow are grouped together to form segments. Cesmeli and Wang [1] have used this approach to obtain motion-based segmentation in LEGION networks. While this two-step technique works well in many cases, it does require the explicit pre-calculation of an optic flow image. We are seeking a somewhat more "on-line" or "connectionist" approach, where the optic flow is implicit in the network rather than explicitly calculated as an image. Thus, in keeping with connectionist orthodoxy, we use local block matches to set up inter-layer weights, which are the only repository of processed information. The presentation of the raw images to this network leads to the desired segmentation via network dynamics as described below. Since the inter-layer weights in our system implicitly embody optic flow information, the system can be seen as a distributed, on-line "approximation" to the two-step procedure.

3 System Description

The network consists of two $n \times n$ layers, Layer A and Layer B, of neurons. Each layer represents one of the images, with 1-to-1 correspondence between neurons and pixels. Each neuron is connected laterally to its 8 immediate neighbors with weights of magnitude 1. The 8-neighborhoods of neuron $A(i,j)$ and $B(i,j)$ are denoted $N_8^A(i,j)$ and $N_8^B(i,j)$, respectively. Layer A projects to Layer B and vice-versa. The inter-layer weights are determined by the corresponding images as follows. First, the $\nu \times \nu$ block around $A(i,j)$ is compared with all $\nu \times \nu$ blocks in the $0.25n \times 0.25n$ neighborhood of B centered around $A(i,j)$. This neighborhood is denoted $N_W^B(i,j)$. The difference between pixels $B(i,j)$ and $(B(k,l) \in N_W^B(i,j)$ is given by:

$$\delta_{ij,kl} = \sum_{r_x=-(\nu-1)/2}^{(\nu-1)/2} \sum_{r_y=-(\nu-1)/2}^{(\nu-1)/2} |I_A(k+r_x, l+r_y)$$

$$-I_B(i+r_x, j+r_y)|$$

From this, the neuron $B(k,l) \in N_W^B(i,j)$ with the best match to $A(i,j)$ is identified as the one with the smallest $\delta_{ij,kl}$. This can be done using an efficient local search procedure such as the one suggested by Wang et al. [12]. The weight from $A(i,j)$ to $B(k,l)$ is set to $W_{ij,kl}^{21} = \sqrt{(k-i)^2 + (l-j)^2}$, and all other weights from $A(i,j)$ are set to 0. Weights $W_{ij,kl}^{12}$ from $B(k,l)$ to $A(i,j)$ are set up in the same way, using a neigborhood $N_W^A(k,l)$ in A.

The process begins by the user firing an initial neuron in layer A. The choice of this neuron will depend on the user's requirements, but it will generally lie in the interior of the area that needs to be matched. The network operates iteratively with Layers A and B driving each other. Each iteration, t, of the network operation comprises four steps:

Step 1: Spread excitation in Layer A: In this step, the currently active neurons in Layer A spread their excitation to their neighbors. Those neighbors that have not fired recently then become eligible for firing.

The excitation for each neuron, $A(i,j)$ is updated as follows:

$$Y_{ij}^A(t) = \begin{cases} 1 & if \sum_{A(k,l)\in N_8^A(i,j)} Z_{kl}^A(t-1) > 0 \\ & \quad or \sum_{A(k,l)} Z_{kl}^A = 0 \\ 0 & else \end{cases}$$

$Y_{ij}^A = 1$ indicates whether $A(i,j)$ has a neighbor that fired in iteration $t-1$, or if no Layer A neurons fired (implying that a new segment can be started). In effect, $Y_{ij}^A(t)$ indicates eligibility to be recruited as part of the current segment.

Neurons with $Y_{ij}^A = 1$ at the end of Step 1 represent the potential population of A neurons that could be recruited into the segment if they find a satisfactory match in Layer B.

Step 2: Firing of Layer B neurons: In this step, the eligible Layer A neurons fire Layer B neurons. To fire, a Layer B neuron must satisfy two conditions: 1) It receives a strong input from Layer A, indicating that its neighborhood matches that of an eligible Layer A neuron; and 2) It is adjacent to at least one Layer B neuron that fired in the previous iteration. The second condition is waived if no Layer B neurons fired in the previous step, indicating that a new segment is beginning. The equations for this step are:

$$Y_{ij}^B(t) = \begin{cases} 1 & if \sum_{B(k,l)\in N_8^B(i,j)} Z_{kl}^B(t-1) > 0 \\ & \quad or \sum_{B(k,l)} Z_{kl}^B = 0 \\ 0 & else \end{cases}$$

$$\bar{E}_{N_8^B(i,j)}(t-1) = \frac{\sum_{B(k,l)\in N_8^B(i,j)} E_{kl}^B(t-1) Z_{kl}^B(t-1)}{\sum_{B(k,l)\in N_8^B(i,j)} Z_{kl}^B(t-1)}$$

$$E_{ij}^B(t) = Y_{ij}^B \sum_{A(k,l)} W_{ij,kl}^{21} Y_{kl}^A(t)$$

$$Z_{ij}^B(t) = \begin{cases} 1 & if\ e^{-|E_{ij}^B(t) - \bar{E}_{N_8^B(i,j)}(t-1)|} > \phi^B \\ 0 & else \end{cases}$$

Here, ϕ^B is a fixed threshold, Y_{ij}^B is an eligibility variable, indicating whether $B(i,j)$ has a neighbor that fired in iteration $t-1$ or if no Layer B neurons fired, E_{ij}^B is the net excitation to $B(i,j)$, and Z_{ij}^B is its output. Note that Y_{ij}^B can be calculated easily using a neural formulation with neighborhood connectivity and an interneuron summing total Layer B activity.

Neurons with $Z_{ij}^B = 1$ at the end of Step 2 are those that match the intensity neighborhood of an eligible Layer A neuron, and are eligible for recruitment to the Layer B segment because they are adjacent to the edge of the segment obtained so far.

Step 3: Firing of Layer A neurons: In this step, those Layer A neurons that became eligible for firing in Step 1, and whose inclusion in the segment is confirmed by their receiving a significant input from active Layer B neurons are actually fired. The equations for this are:

$$\bar{E}_{N_8^A(i,j)}(t-1) = \frac{\sum_{A(k,l)\in N_8^A(i,j)} E_{kl}^A(t-1) Z_{kl}^A(t-1)}{\sum_{A(k,l)\in N_8^A(i,j)} Z_{kl}^A(t-1)}$$

$$E_{ij}^A(t) = Y_{ij}^A \sum_{B(k,l)} W_{ij,kl}^{12} Z_{kl}^B(t)$$

$$Z_{ij}^A(t) = \begin{cases} 1 & if\ e^{-|E_{ij}^A(t) - \bar{E}_{N_8^A(i,j)}(t-1)|} > \phi^A \\ 0 & else \end{cases}$$

The A-to-B-to-A loop can be seen as a gated recurrent self-connection for Layer A neurons. Essentially, each eligible Layer A neuron ends up firing itself if it satisfies match and neighborhood conditions in Layer B. Neurons that fire in Step 3 are included in the I_A segment.

Step 4: Decide whether to terminate the segment: In this step, a decision may be made to terminate the segment if the number of newly fired A neurons are too small a fraction of those that were eligible. This is simply a noise-control measure to prevent a one or two accidentally matching pixels from causing a leakage of the segment into the surrounding region. The decision to terminate is made as follows:

$$q(t) = \frac{\sum_{A(i,j)} Z_{ij}^A(t)}{\sum_{A(i,j)} Y_{ij}^A(t)}$$

If $q(t) < \theta$, the process is terminated; else $t = t+1$ and the algorithm is repeated from step 1.

In order to evaluate whether the network model adequately captures the logic of the standard two step process for motion-based segmentation (obtain velocity image and then segment that image), we also obtained data from another algorithm — termed the *1-layer network* — which comprised the following steps:

1. Use local block matching and best-match (as used for setting up inter-layer weights in our model) to estimate the velocity, $v(i,j) = [v_x(i,j) v_y(ij)]$ for each pixel.
2. Set up a 1-layer PCNN, C, with the weight to each neuron $C(i,j)$ from each of its 8-neighbours, $C(k,l)\ Wij,kl = exp(-\zeta|\Delta_{ij,kl}|)$, where $|\Delta_{ij,kl}| = sqrt(v_x(i,j) - v_x(k,l))^2 + (v_y(i,j) - v_y(k,l))^2$ is the Euclidean difference of the velocities for the two pixels, and ζ is a parameter.

3. Starting with an activated PCNN neuron in the object, let the PCNN complete the segment via recruitment as in the standard PCNN algorithm.

Basically, this algorithm uses a PCNN to segment a velocity image much as it would be used to segment an intensity image [7, 8]. This approach, using LEGION instead of PCNN, has also been used by Cesmeli and Wang [1] as part of their more sophisticated motion-based segmentation method.

The 1-layer network represents a common, albeit simple, off-line approach to the motion-based segmentation problem. We use its results comparatively to determine whether our on-line approach can at least achieve the same performance as the off-line one.

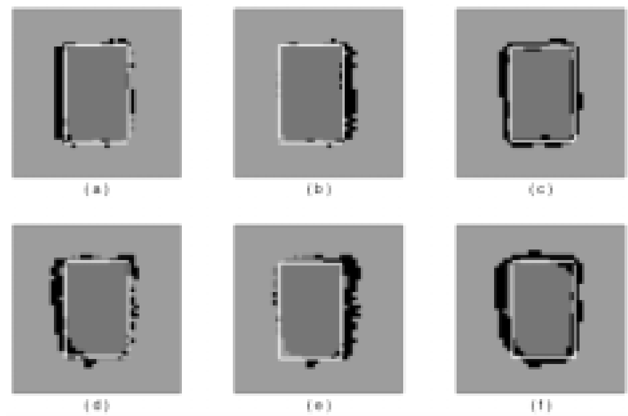

Figure 3: Segmentation results with additive noise: (a)(b)(c) Noise variance = 0.05; (d)(e)(f) Noise variance = 0.1

4 Results

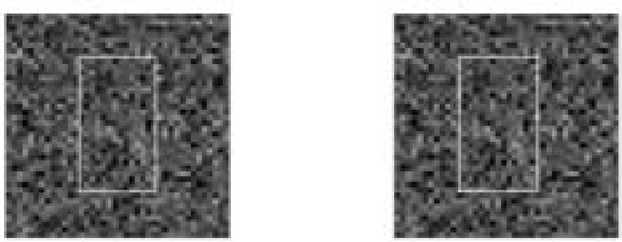

Figure 1: Image frames with one moving object.

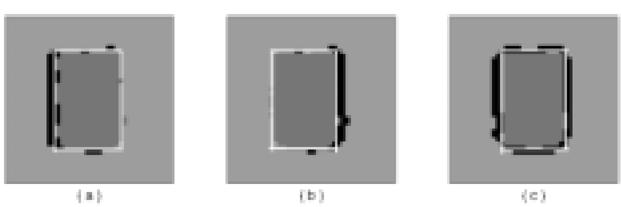

Figure 2: Segmentation results for one moving object: (a) BGBM Layer 1; (b) BGBM Layer 2; (c) 1-Layer Network.

We test our method using synthetic images with one or two moving objects, as well as natural images. In the synthetic images, we focus on the case where both object and background have significant and irregular intensity variation, since this is where block-matching is most applicable (see below). In particular, we consider the case where background and object have *identical* intensity distributions, so that motion is the only cue available for segmentation.

Figure 1 shows two 40×40 frames which include a 24×16 moving object indicated by its outline. Note that the actual image used for segmentation does not include the outline; it is provided only for clarity. Both object and background have a Gaussian intensity distribution with mean 0.5 and variance 0.2. Thus, the object is effectively invisible against the background. For both BGBM and the 1-layer networks, the matching block size, ν, was set at 5. Figures 2 (a) and (b) show the object as segmented in the two layers of the BGBM network, and Figure 2 (c) shows the results obtained by the 1-layer network. In each case, dark gray indicates the matched object region, light gray the matched background, and black the unmatched region (due to motion). As can be seen the BGBM network was able to segment the object corresctly on the basis of inter-frame correspondence.

Figure 3 shows the segmentation results when independent time-varying zero mean Gaussian noise is added to both frames. Figure 3 (a,b,c) are for noise variance of 0.05, and Figure 3 (d,e,f) for noise variance of 0.1. In each case, the rightmost picture is for the 1-layer network and the others for the BGBM network. The segmentation quality does deteriorate, but even at noise variance 50% of the image intensity variance, are reasonable. They could be improved further by using larger ν, but eventually, this begins to reduce the accuracy in determining the object boundary. Figure 4 shows the dependence of segmentation error on the variance of additive noise. The solid lines are for BGBM and the dotted lines for the 1-layer network.

Figures 5 and 6 show the performance of BGBM and the 1-layer network as the object and background variances, respectively, are changed while keeping the other fixed at 0.2. The mean intensity is always 0.5.

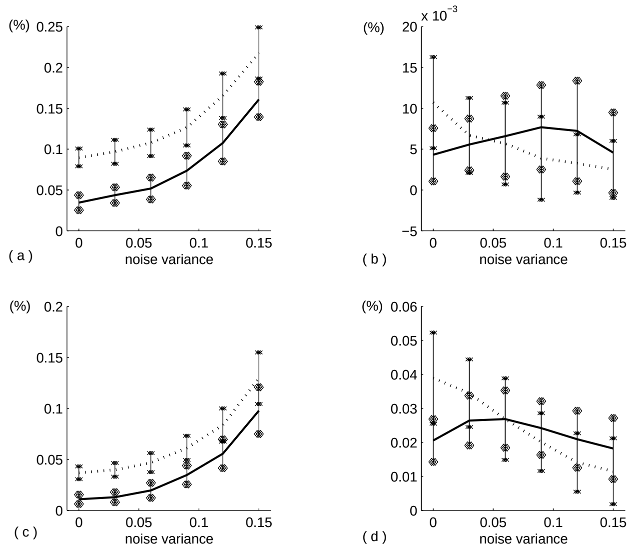

Figure 4: Dependence on additive noise variance: BGBM (solid line), 1-layer network (dotted line). (a) Fraction of object pixels not classified as object; (b) Fraction of non-object pixels classified as object; (c) Fraction of background pixels not classified as background; (d) Fraction of non-background pixels classified as background.

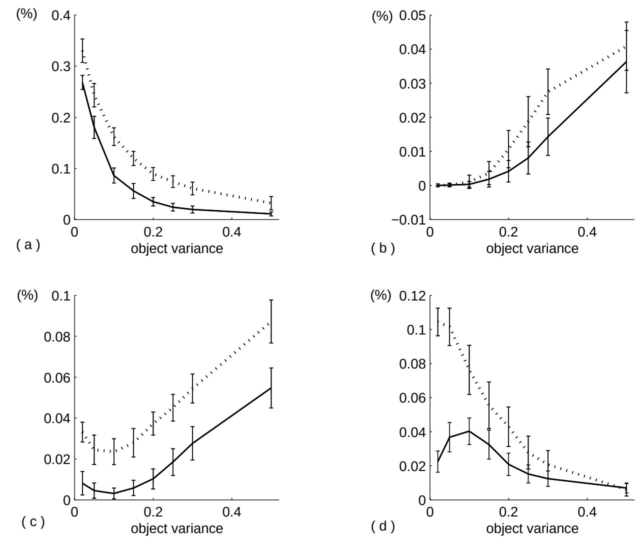

Figure 5: Dependence on object variance: BGBM (solid line), 1-layer network (dotted line). (a) Fraction of object pixels not classified as object; (b) Fraction of non-object pixels classified as object; (c) Fraction of background pixels not classified as background; (d) Fraction of non-background pixels classified as background.

Figure 7 shows the performance of the networks on an image with two objects, one of which occludes the other. As shown, BGBM was able to match the portions of the two objects that were visible in both images.

Finally, Figure 8 shows performance on a natural image. It is noticeable that the algorithm successfully segmented textured regions, but had some difficulty with uniform ones (see below).

5 Discussion and Conclusion

The BGBM method, which is based on block-matching, is best suited to objects/segments with significant and irregular intensity variation. It is less likely to work with homogeneous or regularly textured objects. We note, however, that intensity-based, edge-based, or texture-based segmentation is usually sufficient to segment homogeneous or regularly textured objects. It is precisely when none of these work that our approach is most applicable. We envisage, therefore, that BGBM will be used in parallel with other correspondence or segmentation algorithms, and applied only to those parts of images that are suited to it.

Acknowledgement: This research was supported by a grant from the Ohio Board of Regents. The authors gratefully acknowledge useful discussions with Prof. DeLiang Wang.

References

[1] E. Cesmeli and D.L. Wang. Motion segmentation based on motion/brightness integration and oscillatory correlation. *IEEE Trans. Neural Networks*, 11:935–947,2000.

[2] H. Bülthoff, J. Little and T. Poggio. A parallel algorithm for real-time computation of optic flow. *Nature*, 337:549–553,1989.

[3] R. Eckhorn, R. Bauer, W. Jordan, M. Brosch, W. Kruse, M. Munk and H.J. Reitboeck. Coherent oscillations: A mechanism of feature linking in the visual cortex? *Biol. Cybern*, 60:121–130,1988.

[4] R. Eckhorn, H.J. Reitboeck, M. Arndt and P.W. Dicke. Feature-linking via synchronization among distributed assemblies: Simulation of results from cat cortex. *Neural Computation*, 2:293–307,1990.

[5] C.M. Gray, P. König, A.K. Engel and W. Singer. Oscillatory responses in cat visual cortex exhibit intercolumnar synchronization which reflects global stimulus properties. *Nature*, 338:334–337,1989.

[6] C.M. Gray and W. Singer. Stimulus-specific neu-

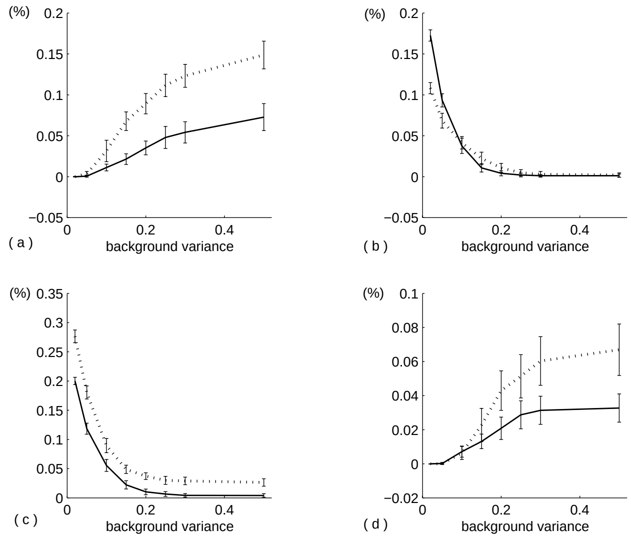

Figure 6: Dependence on background variance: BGBM (solid line), 1-layer network (dotted line). (a) Fraction of object pixels not classified as object; (b) Fraction of non-object pixels classified as object; (c) Fraction of background pixels not classified as background; (d) Fraction of non-background pixels classified as background.

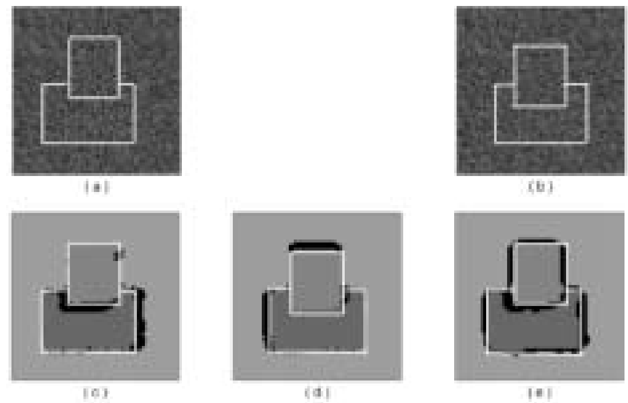

Figure 7: Images and results for occluding object case.

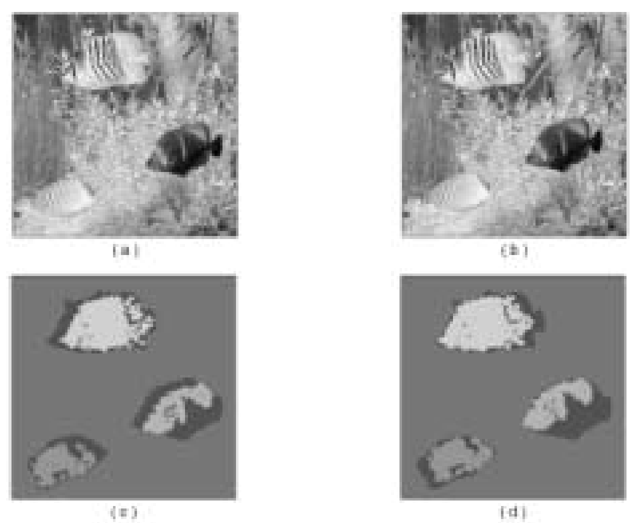

Figure 8: Images and results for natural image

ronal oscillations in the orientation columns of cat visual cortex. *Proc. Nat. Acad. Sci. USA*, 86:1689–1702,1989.

[7] J.L. Johnson and M.L. Padgett. PCNN models and applications. *IEEE Trans. Neural Networks*, 10:480–498,1999.

[8] G. Kuntimad and H.S. Ranganath. Perfect image segmentation using pulse coupled neural networks. *IEEE Trans. Neural Networks*, 10:591–598,1999.

[9] C. von der Malsburg and W. Schneider. A neural cocktail party processor. *Biol. Cybern.*, 54:29–40,1986.

[10] D.L. Wang and D. Terman. Locally excitatory globally inhibitory oscillatory networks. *IEEE Trans. Neural Networks*, 6:283–286,1995.

[11] D.L. Wang and D. Terman. Image segmentation based on oscillatory correlation. *Neural Computation*, 9:805–836,1997.

[12] H.-S. Wang and R.M. Mersereau. Fast algorithms for estimation of motion vectors. *IEEE Trans. Image Proc.*, 8:435–438,1999.