Effect of Noise on the Performance of the Temporally-Sequenced Intelligent Block-Matching and Motion-Segmentation Algorithm

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

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

***Abstract*— Most algorithms for motion-based segmentation depend on the system's ability to estimate optic flow from successive image frames. Block-matching is often used for this, but it faces the problems of noise-sensitivity and texture-insufficiency. Recently, we proposed a two-pathway approach based on locally coupled neural networks to address this issue. The system uses a *pixel-level (P) pathway* to perform robust block-matching in regions with sufficient texture, and a *region-level (R) pathway* to estimate motion from feature matching in low-texture regions. The fused optic-flow from the P and R pathways is then segmented by a pulse-coupled neural network (PCNN). The algorithm has produced very good results on synthetic and natural images. In this paper, we show that its performance shows significant robustness to additive noise in the images.**

I. Introduction

Motion-based segmentation in image sequences typically uses block matching across image frames separated by short time intervals. However, noise in the images often creates spurious matches. Another problem is created by homogeneous regions that do not have enough texture for reliable block matching. The first problem can be alleviated partially by spatial smoothing of the optical flow field [1]. However, optic flow smoothing can blur the flow image at object boundaries. The second problem can be addressed to some degree by algorithms that combine motion segmentation and intensity segmentation [4].

Many researchers have proposed synchronization as a mechanism for feature linking [8], [9], [11], and for image segmentation. Two notable approaches in the latter area are the *pulse-coupled neural network (PCNN)* model [5], and the *locally excitatory, globally inhibitory oscillatory network (LEGION)* model [12]. Approaches based on the PCNN and LEGION models have been successfully applied to image segmentation based on intensity [10], [5], [13] and motion [3].

Recently, we presented a neural method for translational motion segmentation using block-matching in combination with pulse-coupled neural networks [14]. The system uses a *pixel-level (P) pathway* — implemented as a multi-layer neural network — to perform robust block-matching in regions with sufficient texture, and a *region-level (R) pathway* to estimate motion from feature matching in low-texture regions. The fused optic-flow from the P and R pathways is then segmented by a pulse-coupled neural network (PCNN). The system is inspired directly by the LEGION approach [3], which also uses two parallel pathways that process motion and brightness, respectively. Unlike their system ours uses discrete-time binary neurons (as in many PCNN models).

While the algorithm produced very good results on synthetic and natural images, it was not clear whether its performance depended on fragile mechanisms that might not withstand the presence of noise in the images. This is especially important since motion-based segmentation involves matching across two images subject to independent noise. In this paper, we show that the method exhibits significant robustness to additive noise.

II. System Description

Given an image sequence, $I(i,j,t)$, we consider two distinct but temporally proximate frames, $I_A(i,j) = I(i,j,t)$ and $I_B(i,j) = I(i,j,t+k)$, $k > 0$. For uniformity, we normalize intensities in all images to the 0-to-1 range. The proposed approach, termed *temporally-sequenced intelligent block-matching and motion-segmentation (TIBM)*, is shown in Figure 1. The TIBM system comprises three stages:

Stage I: Block Matching and Certainty Determination: In this stage, local block matching is applied to the image frames, $I_A(i,j)$ and $I_B(i,j)$, to obtain the *best block-matching motion estimate (BBME)*, $r^*(i,j)$, for each pixel. In addition, a *block-matching certainty measure(BCM)*, $g(i,j)$, is also computed for each pixel. Typically, highly textured regions generate higher certainty values. These two quantities are then used as input for the next stage.

Stage II: Refined Optic Flow Estimation: Stage II processing comprises two pathways: 1) The *pixel-level (P) pathway*; and 2) The *region-level (R) pathway*. Typical images have both *textured regions* and *homogeneous regions*, which can be identified by their BCM values. Block matching works well for the former, but is subject to noise. The P pathway uses a multi-layer locally coupled network to perform robust

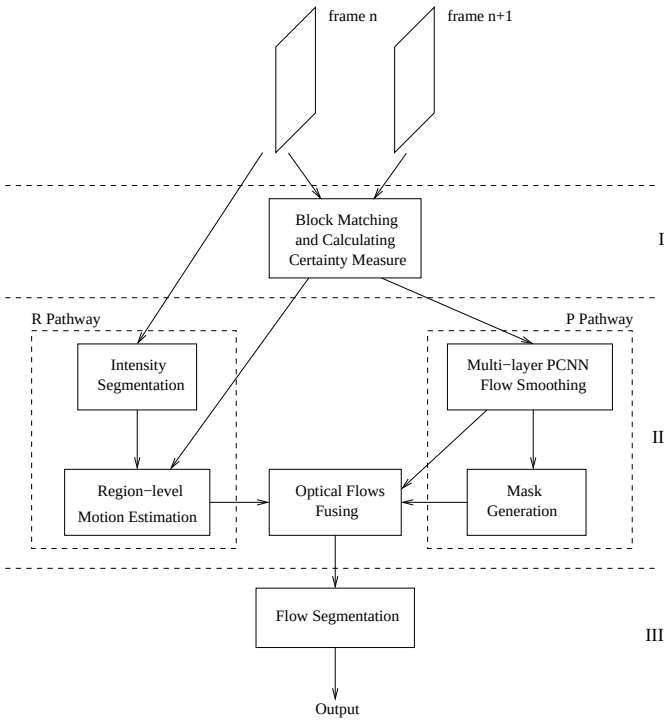

Fig. 1. Flow diagram of the system

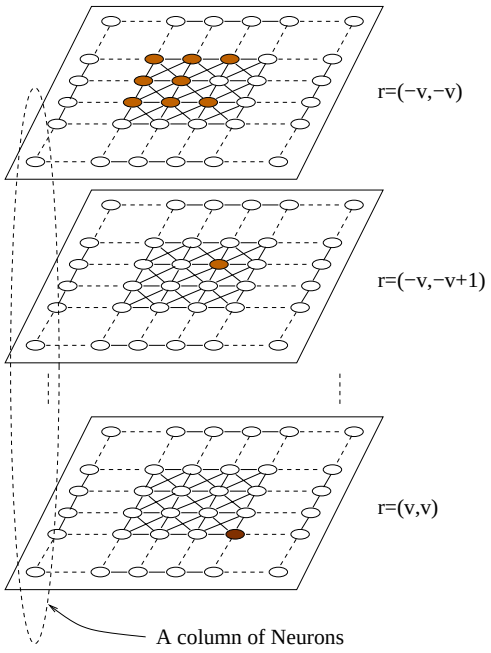

Fig. 2. Architecture of the multi-layer locally coupled network

block matching in textured regions by applying adaptive local smoothing, while the R pathway uses a standard PCNN to detect homogeneous regions and heuristically assigns uniform optic flows to them. Stage II processing produces two optic flow images, $V_P(i,j)$ from the P pathway, and $V_R(i,j)$ from the R pathway, and a *mask*, $M(i,j)$, which indicates which estimate should be used for each pixel. The optic flow images are then combined through the mask to produce the final optic flow image, $V(i,j)$.

Stage III: Segmentation: In this stage the optic flow image, $V(i,j)$, produced by Stage II is segmented using a standard 1-layer PCNN, following the approach used for intensity-based segmentation in [6], [7]. We now describe each stage in detail.

III. The Approach

A. Block Matching and Certainty Determination

The block matching procedure estimates the motion for a pixel by seeking the minimum sum of absolute difference (SAD) between a $b \times b$ block around this pixel in the first image frame, I_A, and all $b \times b$ blocks in a $u \times u$ *search neighborhood* in the successor frame, I_B, where $u > b$. We use $r^*(i,j)$ to denote the BBME for pixel (i,j). The SAD values for (i,j) over all displacements can be seen as forming a $(u-b+1) \times (u-b+1)$ surface, $S(i,j,r) = D_r(i,j)$, whose curvature provides information about the certainty of the match: If the SAD surface around its minimum is flat, the match is uncertain, whereas a deep trough indicates greater certainty [2]. We use this to define a simple, efficiently computed block-matching certainty measure (BCM), $g(i,j)$, for each pixel, (i,j) (see [14] for details).

B. Pixel-level Motion Estimation (P) Pathway

The P pathway comprises a multi-layer PCNN-like locally coupled network that spatially smoothes the optical flow obtained from block matching. Functionally, the P pathway attempts to assign the motion estimate for a pixel consistent with that for its neighbors if: 1) most of its neighbors have same motion estimates; and 2) the motion estimates for its neighbors are of sufficiently high reliability. Thus, reliable pixels with reliable block-matching estimates are accorded greater weight in motion assignment. Blurring around boundaries is avoided by using a definition of motion reliability that tends to be very low near object boundaries, thus shielding pixels near these boundaries from the smoothing process.

1) P Network Architecture: The architecture of the multi-layer network is shown in Figure 2. The inputs to the network are the two image frames, I_A and I_B. The network consists of $(2v+1)^2$, $v = (u-b)/2$, layers of locally coupled networks, each in 1-to-1 correspondence with the pixels in frame I_A. Each layer in the network corresponds to one motion displacement, $r = (r_x, r_y)$. A neuron in layer r corresponding to pixel (i,j) in the image is denoted by (i,j,r). Clearly, there is a unique SAD value, $S(i,j,r)$, corresponding to each neuron (i,j,r) in the network. A *column* is the set of neurons at position (i,j) in all the layers, representing all possible displacements of pixel (i,j). In each column, the neuron, $(i,j,r^*(i,j))$, with the lowest $S(i,j,r)$ is termed the *dominant neuron*; it corresponds to the BBME for pixel (i,j).

Each neuron, (i,j,r), receives a fixed *external input* inversely proportional to $S(i,j,r)$, and a *certainty input* which is $g(i,j)$ for dominant neurons and zero for others. Neuron (i,j,r) also receives two types of connections from each of its eight lateral neighbors, (k,l,r): 1) A *proximity connection*

of weight $w^p_{ij,kl,r}$; and 2) A *certainty connection* of weight $w^c_{ij,kl,r}$. The proximity connections are all set to unity, while the certainty connections are set as: $w^c_{ij,kl,r} = G^P(k,l,r)$. Thus, certainty connections are directed, while proximity connections are bidirectional and symmetric.

2) P Network Operation: The network has a *fast time-scale* indexed by t and a *slow time-scale* indexed by T. Each increment of the slow time-scale is called a *cycle*, and comprises q increments (steps) of the fast time-scale. The cycle corresponding to t is denoted $T(t) = \lfloor t/q \rfloor$. Each cycle begins with the firing of at most one neuron, which is termed the *seed neuron* for that cycle. The seed neuron's activity spreads by local recruitment on the fast time scale, and all neurons fired in a cycle form a potential segment. We assume that cycles are long enough to complete even the largest segment.

The dynamics of each neuron is described by:

$$L^P(i,j,r,t) = \sum_{(k,l,r)\in N_8(i,j,r)} \lambda_1\, w^p_{ij,kl,r} Z^P(k,l,r,t-1) + \sum_{(k,l,r)\in N_8(i,j,r)} \lambda_2\, w^c_{ij,kl,r} Z^P(k,l,r,t-1) \quad (1)$$

$$U^P(i,j,r,t) = E^P(i,j,r)[1 + L^P(i,j,r,t)] \quad (2)$$

$$Z^P(i,j,r,t) = \begin{cases} 1 & if\ [G^P(i,j,r) > \phi(i,j,T(t))\ and\ H(i,j,r,t) = 1] \\ & or\ U^P(i,j,r,t) > \theta(i,j,t) \\ 0 & else \end{cases} \quad (3)$$

$$\phi(i,j,0) = 0 \quad (4)$$

$$\Omega(i,j,t) = \begin{cases} 1 & if\ \phi(i,j,T(t)) < \phi_{min} \\ 0 & else \end{cases} \quad (5)$$

$$H(i,j,r,t) = \begin{cases} \delta(i,j,r) & if\ \Omega(i,j,t-1) = 1 \\ H(i,j,r,t-1)(1 - Z^P(i,j,r,t-1)) & else \end{cases} \quad (6)$$

$$\phi(i,j,T(t)) = \begin{cases} \phi_{init} & if\ \Omega(i,j,t-1) = 1 \\ e^{-\alpha_\phi}\phi(i,j,T(t)-1) & else \end{cases} \quad (7)$$

$$\theta(i,j,t) = \begin{cases} E^P(i,j,r^*) & if\ \Omega(i,j,t-1) = 1 \\ \theta(i,j,t-\tau)[1 - \sum_r Z^P(i,j,r,t-\tau)] & else \\ \quad + \sum_r [U^P(i,j,r,t-\tau) Z^P(i,j,r,t-\tau) & \\ \quad \times Z^P(i,j,r,t-\tau+1)] & \end{cases} \quad (8)$$

where $L^P(i,j,r,t)$ is the time-varying lateral input, $Z^P(i,j,r,t)$ is the binary output of (i,j,r), $N_8(i,j,r)$ denotes the 8-neighborhood of neuron (i,j,r) in layer r, $U^P(i,j,r,t)$ is the internal activity of neuron (i,j,r) at t, $E^P(i,j,r)$ is the external input, $G^P(i,j,r)$ is the certainty input, τ is the duration of neuron's firing activity, q denotes the number of fast time-scale steps in a slow time-scale cycle, and λ_1, λ_2, ϕ_{init}, ϕ_{min}, α_ϕ are user-specified parameters that are explained below. Each neuron has a *seed potential*, $H(i,j,r,t)$, which is non-zero only for dominant neurons, allowing them to fire as seed neurons. All neurons in the same column, (i,j) have two common thresholds: A *certainty threshold*, $\phi(i,j,T)$, that changes at the slow time-scale, and an *activity threshold*, $\theta(i,j,t)$, that tracks the highest activity level seen so far for any neuron in column C_{ij}. Once the certainty threshold falls below a certain value ϕ_{min}, a resetting signal $\Omega(i,j,t)$ is generated and propagated to all neurons in that column.

The certainty thresholds of all columns are set to zero when the network starts operation. Therefore, at time $t = 0$, a resetting signal is generated. Each resetting signal accomplishes three things: 1) Charges the seed potential, $H(i,j,r,t)$, to 1 for dominant neurons; 2) charges the certainty threshold, $\phi(i,j,T)$, to the initial value, ϕ_{init}; and 3) sets the activity threshold, $\theta(i,j,t)$, to, $E^P(i,j,r^*)$, the maximum external input in a column. Once the certainty threshold is charged up, the resetting signal disappears at the next step. The ϕ's of all columns start to decay identically at the slow time-scale. When they fall below ϕ_{min} at the same time, resetting signals are produced again in each column. The duration for ϕ decaying from ϕ_{init} to ϕ_{min} is termed an *epoch*. The network is ready to operate from the same initial status at the beginning of each epoch.

During each epoch, dominant neurons initiate activity in order of their certainty input. This, in turn, produces a segment via spreading activation to other non-dominant but high-certainty neurons nearby in the same layer. This ensures that only pixels with locally consistent estimates are included in the segment. By the end of the epoch, all pixels in sufficiently textured regions have a "best" displacement estimate indicated by the index of the last network layer on which they fired. The degree to which similarity of block-match displacement or certainty is emphasized during segmentation can be controlled by setting λ_1 andλ_2 appropriately.

Since the multi-layer P network is completely dynamic, with segments appearing and disappearing in each epoch, these must be buffered to produce the output, $V_P(i,j)$ and the mask, $M(i,j)$, for purposes of further processing. Both M and V_P are initialized to 0 at $T = 0$. If the average certainty of the segment is larger than a threshold γ, the optic flow of all neurons in the segment is set to the displacement corresponding to the layer of the segment, and the mask is set to 1.

C. Region-Level Motion Estimation (R) Pathway

In this pathway, first, a PCNN network [6], [7] detects regions of homogeneous intensity in the first intensity image frame I_A. The optical flow output V_R in this pathway is initially set to the BBME: $V_R(i,j,0) = r^*(i,j)$. For each intensity segment, the BBME displacement with the maximum total certainty is assigned as the unique estimate vector for the whole homogeneous region.

D. Fusing the Output Optical Flow Estimates

Fusing the output optical flow estimates of the P and R pathways is accomplished using the mask matrix M. Regions where $M(i,j) = 1$ have high motion estimation certainty, and optical flow smoothing functions well through the strong certainty connections between neighboring neurons in the

multi-layer network architecture. Therefore, we use the pixel-level pathway's output as the final output where $M(i,j) = 1$. In contrast, optical flow smoothing performs poorly where $M(i,j) = 0$ since the certainty connections are too weak. However, since low motion estimation certainty usually means regions with little texture, we naturally turn to the region-level pathway to find out the optical flow. The final optical flow, $V(i,j) = (V_x(i,j), V_y(i,j))$, is computed by:

$$V(i,j) = M(i,j)V_P(i,j) + [1 - M(i,j)]V_R(i,j) \quad (9)$$

E. Optical Flow Image Segmentation

Once the final optical flow image, $V(i,j)$, is obtained, it can be treated just like an intensity image and is segmented using a standard 1-layer PCNN [6], [7]. The external input signal to neuron (i,j) in the network is set to: $S^V(i,j) = \|V(i,j)\|$, where $\|.\|$ is the Euclidean norm. The connection weight between neuron (i,j) and neuron (k,l) in its 8-neighbors is

$$w^V_{ij,kl} = exp\,[-\xi\|V(i,j) - V(k,l)\|\,] \quad (10)$$

where ξ is a user-specified parameter. Thus, neighboring neurons corresponding to pixels with similar motion assignments have stronger connections, and are likely to be included in the same segment. The ξ parameter controls the discrimination of the weights along the similarity dimension.

IV. Results

The multi-layer TIBM approach described above has been evaluated using both synthetic and natural image sequences. The primary focus is on demonstrating the benefit of TIBM compared to simple block-matching. We use the *same* parameter values for all images to demonstrate that the system's performance does not depend sensitively on parameter tuning.

The size of the matching block, b, is set at 5 and 9 for synthetic images and real images, respectively. For the multi-layer network in the P pathway, $\lambda_1 = 0.02$, $\lambda_2 = 0.3$, ϕ_{init}=1.1, $\phi_{min} = 0.01$, and $\alpha_\phi = 10^{-5}$. We use the average block matching certainty of the whole image as the certainty threshold, γ. For the intensity-based PCNN in the R pathway, and the 1-layer PCNN network that segments the optical flow images, parameters are set as suggested in [6], [7]. The value of ξ depends on the velocity differences between objects in the image.

A. Synthetic Images

In the first image sequence shown in Figure 3(a), a 48×48 object, indicated (for illustration only) by the white outline, moves on the 80×80 background at a speed of 4 pixels leftward per frame. Both object and background have Gaussian distributed intensity with 0.5 mean and 0.1 standard deviation (STD). We add zero mean Gaussian noise to both frames, with STD=0.04, 0.08, and 0.12 respectively. The corresponding segmentation results are shown in Figure 4(d)-(i). When STD=0.08, the power level of noise is actually greater than the strength of texture (since we add noise to both frames). As

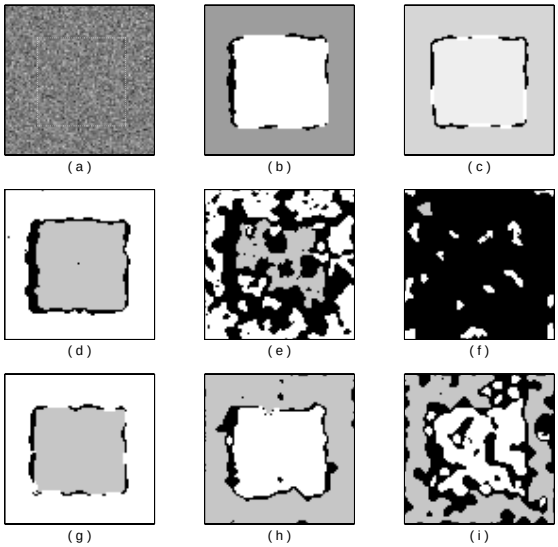

Fig. 3. Synthetic image: (a) First image frame; (b) Segmentation result based on block matching; (c) Segmentation result based on TIBM; (d)(e)(f) Segmentation result based on block matching, both frames with additive noise of zero mean, STD=0.04, 0.08, and 0.12 respectively; (g)(h)(i) Segmentation result based on TIBM, both frames with additive noise of zero mean, STD=0.04, 0.08, and 0.12 respectively.

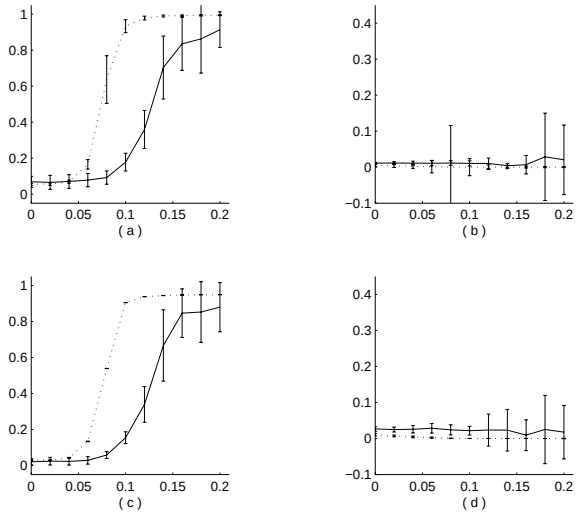

Fig. 4. Dependence on additive noise STD: TIBM (solid line), block matching (dotted line). (a) Fraction of object pixels not classified as object; (b) Fraction of non-object pixels classified as object; (c) Fraction of background pixels not classified as background; (d) Fraction of non-background pixels classified as background.

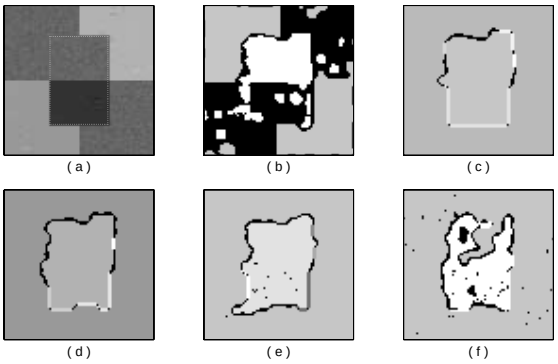

Fig. 5. Synthetic image with homogeneous regions: (a) First image frame; (b) Segmentation based on block matching, both frames with additive noise of zero mean, 0.005 STD; (c)-(f) Segmentation based on TIBM, with noise STD 0.005, 0.01, 0.015 and 0.02, respectively

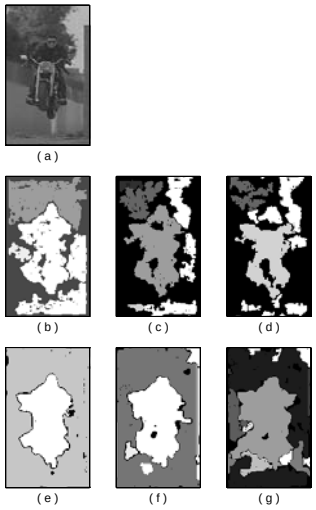

Fig. 6. "Motorcycle rider" image sequence: (a) First image frame; (b)(c)(d) Segmentation based on block matching, noise STD=0 (b), $2M_{std}$ (c), and $4M_{std}$ (d); (e)(f)(g) Segmentation based on TIBM, noise STD=0 (e), $2M_{std}$ (f), and $4M_{std}$ (g).

shown in Figure 4(e), simple block matching fails to generate correct segments at this time. However, TIBM still produces a reasonable result clearly identifying the object based on coherent motion (Figure 4(h)). Even when the noise STD is 0.12, so that noise is far stronger than the texture information, TIBM can still pick out the moving object to some degree, while direct block-matching fails completely.

To determine whether the results shown in Figure 3(a) are typical, we repeated the experiment 50 times and obtained the average dependence of segmentation error on the noise STD. Figure 4 shows the fraction of pixels in the object and background that were classified incorrectly at thr different noise-levels. The solid lines are for TIBM and the dotted lines for block matching. Clearly, there is a wide range of noise where TIBM is doing acceptably while block-matching is not. It should be noted that since there is no homogeneous region in this image, Stage II processing is really accomplished by the P pathway alone (though, of course, this is determined automatically by the algorithm). Thus, the results demonstrate the improvement provided by just the P pathway, even without the use of the R pathway.

To evaluate the performance of TIBM on images with both textured and homogeneous regions, we used the synthetic image shown in Figure 5(a). A 48×32 object moves across the 80×80 background at the speed of 4 pixels leftward per frame. The upper left and lower right quarters of the background, and the upper half of the object are textured regions with Gaussian distributed intensity of mean 0.5 and STD 0.02. The rest are homogeneous regions with distinct intensities. Several 3×3 patches with zero mean and 0.04 variance Gaussian intensity distribution are added to these homogeneous blocks at random locations serving as the feature points. Zero mean Gaussian noise, with STD=0.005, 0.01, 0.015, and 0.02, is added to both frames. The segmentation results are shown in Figure 5(b)-(f). Segmentation based on blocking matching is unable to deal with homogeneous regions even when STD=0.005, while TIBM still works even when noise level is slightly greater than the strength of texture (STD=0.015). These results provide strong evidence that TIBM can tolerate much more noise than block matching.

B. Natural Images

To evaluate the noise immunity of TIBM on natural images, we need to ensure that the amount of noise added is reasonable with respect to the intensity distribution in the image. We use a heuristic approach to obtain a rough estimate of the intensity variation in the image and add noise that is a multiple of this value. We divide the image into blocks of the same size as the matching block, calculate the standard deviation of intensity in each of these blocks, and then take the mean of these values to obtain M_{std}, the block-mean STD of intensity. Since the STD for some blocks — especially at region boundaries — may be extremely high, we exclude the blocks in the top and bottom 5 % of the block STD range. Although M_{std} is not a rigorous measure of intensity variation, it provides a reasonable and easily calculated estimate, and serves as a guideline for choosing the strength of noise.

We evaluate the performance of TIBM by adding zero mean Gaussian noise to both frames in a natural image sequence. In Figure 6(a), a motorcycle rider jumps across a dry canal while the camera is tracking him. The rider and the motorcycle fall downward with slight rightward movement, while the background has upward motion. When no noise is added, segmentation based on block-matching can detect the rider approximately (Figure 6(b)), while the segmentation result of TIBM is almost perfect (Figure 6(e)). We then add zero mean Gaussian noise with STD equal to $2M_{std}$ and $4M_{std}$ — i.e., twice and four times the mean intensity variation in the image. The results, shown in Figure 6(d)-(g) demonstrate that segmentation produced through simple block-matching degrades much more than that obtained with TIBM.

Figure 7 shows the results of adding zero mean Gaussian

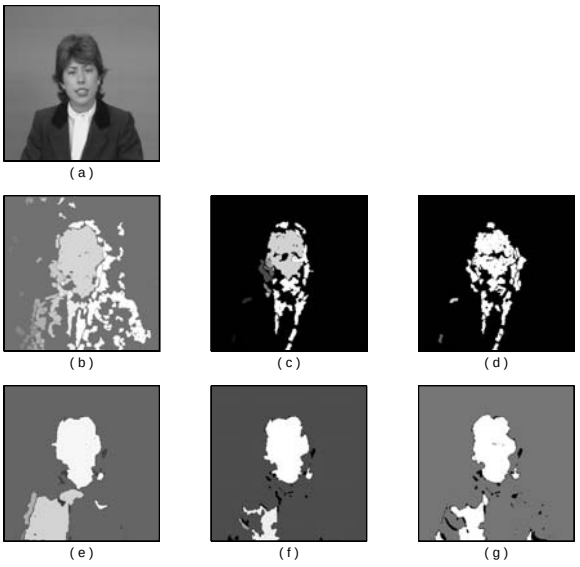

Fig. 7. "Claire" image sequence: (a) First image frame; (b)(c)(d) Segmentation based on block matching, noise STD=0, $2M_{std}$, and $3M_{std}$;(e)(f)(g) Segmentation based on TIBM, noise STD=0, $2M_{std}$, and $3M_{std}$.

noise to the "Claire" sequence. In this scene, her head is bowing down thus causing slight deformation in the head region, and the right side of her body is slightly rising up. The additive noise has standard deviation 0, $2M_{std}$, and $3M_{std}$ from left to right. Once again, TIBM has more robust output than block matching in the presence of noise.

V. Conclusion

In the approach described in this paper, we have studied the degree to which the temporally-sequenced intelligent block-matching based motion-estimation (TIBM) is robust against the presence of additive noise in the images. Previous results [14] have shown that TIBM performs significantly better than simple block-matching when additive noise is not present. The results in the present paper show that the advantage provided by TIBM extends to the case with noise, and that TIBM can provide reasonable motion estimates at much higher levels of noise. This is significant because it shows that the superior performance of TIBM in the noise-free case is no obtained by the use of fragile methods that might fall apart in the presence of a little noise. Indeed, TIBM appears to work fairly well even when the signal-to-noise ration is extremely poor. However, further study is needed to understand the limitations and strengths of TIBM on various types of images.

Acknowledgment

This research was supported in part by a grant from the Ohio Board of Regents to Ali Minai and DeLiang Wang. The authors would like to thank DeLiang Wang for motivating this research, and for collaborating in many of its formative aspects.

References

[1] L. Alparone, M. Barni, F. Bartolini, and V. Cappellini. Adaptively weighted vector-median filters for motion-fields smoothing. In *IEEE ICASSP-96*, volume 4, pages 2267–2270, 1996.

[2] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *Int. J. Computer Vision*, 2:283–310, 1989.

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

[4] F. Dufaux, F. Moscheni, and A. Lippman. Spatio-temporal segmentation based on motion and static segmentation. In *Proc. IEEE Conf. on Image Processing*, volume 1, pages 306–309, 1995.

[5] 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.

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

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

[8] P. M. Milner. A model for visual shape recognition. *Psychol. Review*, 81:521535, 1974.

[9] H. J. Reitboeck. A multi-electrode matrix for studies of temporal signal correlations within neural assemblies. *In: H. Haken, E. Basar, H. Flohr, A. J. Mandell (Eds), Synergetics of the Brain. Berlin, Springer*, pages 174–182, 1983.

[10] M. Stoecker, H. J. Reitboeck, and R. Eckhorn. A neural network for scene segmentation by temporal coding. *Neurocomputing*, 11:123–134, 1996.

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

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

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

[14] X. F. Zhang and A. A. Minai. Temporally-sequenced intelligent block-matching and motion-segmentation using locally coupled neural networks. *IEEE Trans. Neural Networks (Special Issue on Temporal Coding for Neural Information Processing)*, (to appear).