Motion Detection
– A neural model and its implementation

Robert Pallbo

Lund University Cognitive Science,
Kungshuset, Lundagård,
S–223 50 Lund,
Sweden.
E-mail: robert.pallbo@fil.lu.se

Abstract

A model for motion detection is presented. In this approach, motion is viewed as a stable pattern propagating over the image – a technique that makes the model unusually insensitive to noisy input sequences. The model is based on a neural network and requires spontaneous activation of the nodes in order to be operative. The model has been tested in computer simulations with over 300 000 nodes and more than 6 million connections. A compilation of results from these simulations is also presented.

1 Introduction

There are two reasons for creating a motion detecting algorithm; either the result is intended for technical applications, or it is intended as a biological model. Depending on which of these two goals one has in mind, the outcome will be different. For a technical application, speed and reliability are crucial aspects. For biological models, the focus is not so much on speed as on biological realism. The model presented in this paper belongs to the latter category – plausible biological models. This means that speed considerations have not been taken into account. In fact, when run on a sequential computer, the simulation is extremely slow.

I will not discuss biological plausibility in this paper. Any reader interested in these considerations is directed to the previous papers on this model (Pallbo, 1992; Pallbo, 1993). Here, I will instead first present the technical equipment used and how the images were prepared for usage in the model. Further, the architecture of the network will be outlined, and, finally, results from simulations run under different conditions will be presented.

2 Technical equipment

All the simulations were done with software programs written in C. To obtain suitable input to the system, however, some hardware was needed. I have used a cheap video camera and a frame grabber to record video sequences of an ordinary office environment. The technical equipment used in the experiments was not specifically selected for this task. Instead, the equipment was compiled on a take-what-you-have basis. In short, the following equipment was used:

LUCS Minor 1, 1994. ISSN 1104–1609.

Camera :	Pulnix TM 526
Ocular :	Ernitec GCD–14, 1:1.4/12mm.
Frame grabber :	VideoPix on a SUN SPARCstation 2
Size of frame :	180×143
Sampling speed :	approx. 5.5 frames/sec.

The frame size 180×143 was obtained by the `vfc_preview_quarter_sq` function which reduces the full PAL image of 768×575 pixels. The captured frame sequences were then stored in files to be used as input.

3 Pre-processings of the input

The images obtained from the frame grabber program used 127 gray levels. The motion detecting algorithm, however, was constructed for black and white pictures without large homogeneous areas. Therefore, a filter was constructed to achieve this property. The filter is a square version of the Mexican hat filter (Figure 1).

+	+	+
+	+/-	+
+	+	+

Figure 1: The filtering is accomplished by a "Mexican hat"-style filter. The nine pixels are compared to the one in the center. If the value of the pixel in the middle exceeds the value of the nine pixels divided by nine, the pixel on the filtered image is set on.

The gray level (ranging from 0–126) of nine pixels arranged in a 3×3 matrix is summarized. The sum is normalized by dividing it by nine. It is then compared to the gray level of the center pixel of the matrix. If the value of this pixel is larger than the normalized value of the total matrix, the corresponding pixel in the filtered picture is set `on` (black), otherwise it is set `off` (white). This was repeated for each pixel on every frame of a recording. The resulting frame sequence (see Figure 2) was used as input to the system. In the sequence, my colleague walks from the right end of the image leftward to the left end. About 100 frames are used for this short walk.

4 Network architecture

In the network architecture, one node corresponds to one pixel. Furthermore, the system extracts eight different directions of the motion. Each direction is represented by one 180×143 matrix of nodes. The system also extracts four different orientations of static lines which adds another four matrices to the system. In all, we have twelve matrices with $180 \cdot 143 = 25\ 740$ pixels which give a total of $12 \cdot 25\ 740 = 308\ 880$ nodes in the entire network (input nodes uncounted).

If we count the connections, we have 15 excitatory and 7 inhibitory connections to each motion selective node, and 10 excitatory and 3 inhibitory connections to each line extracting node. In addition, we also have one input connection to each node. In all, we have $8 \cdot 25\ 740 \cdot (15 + 7 + 1) + 4 \cdot 25\ 740 \cdot (10 + 3 + 1) = 6\ 177\ 600$ connections in the entire network.

The input to each node is primarily of three kinds:

Figure 2: The image as it appears before and after filtering. The filter used is a simple rectangular version of the Mexican hat function.

Input connection: The excitatory connection from the input nodes. Each motion selective node receives input from one single input node.

Lateral excitation: The excitatory connection between nodes selective to the same direction. Each node receives input from 15 other nodes as outlined in Figure 3.

Lateral inhibition: The inhibitory connection between nodes selective to different directions. Each node receives input from one node in each of the other directions as shown in Figure 3.

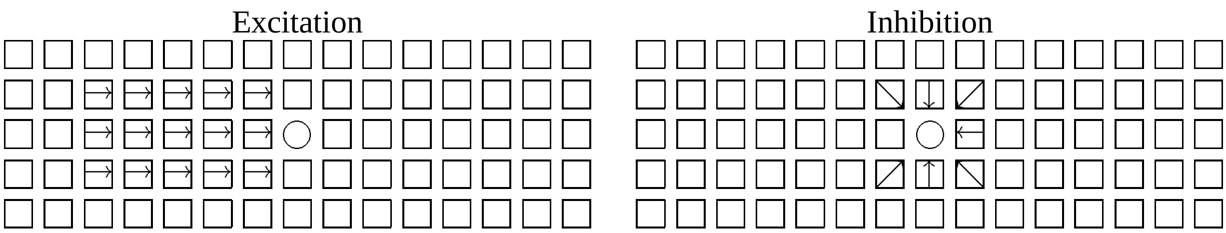


Figure 3: The network architecture for detecting rightward motions. The target cell is circular and the surrounding cells are square. Connected cells are shown with their direction specificity. Left: Lateral excitation from nodes of the same specificity. Right: Lateral inhibition from nodes of different specificity.

In the previous papers, the network architecture has been illustrated in a more abstract manner. The figures in Figure 4 illustrate mainly the symmetry of the architecture, and not so much the number of connections. The lateral connections are illustrated from a top view in these figures, and the input connection from a lateral view. The positive and negative signs denote excitatory and inhibitory connections respectively.

A specific strength is associated with each connection in the network. Connections of the same kind (i.e., input connections, lateral excitation, or lateral inhibition) have the same strength. At each interval, the sum of the strengths of all active connections will be summed at each destination node (i.e., all nodes except the input nodes). If the sum exceeds a threshold value (the same for all nodes) the node becomes activated in the next iteration.

In addition to node activations caused by the activity mediated by the connections, a node can also be *spontaneously* activated. Each node is forced into activation, independent of the converging inputs, with a probability determined by an `arousal` parameter. This probability

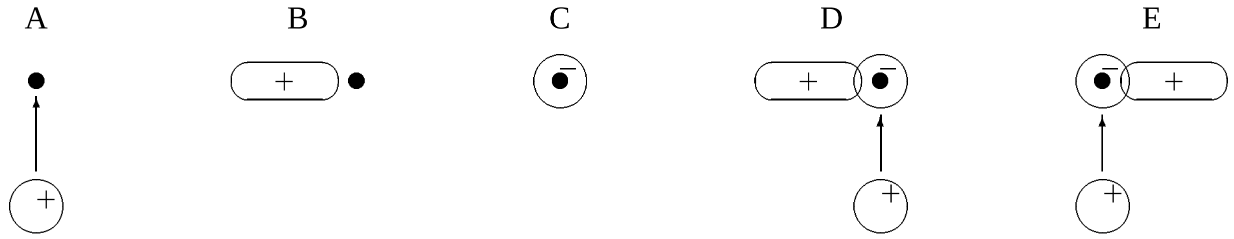


Figure 4: Schematic outline of the network architecture. The target node is illustrated by a filled circle. The other fields illustrate sets of nodes. (A) The input connection. (B) The lateral excitatory connections. (C) Lateral inhibition. (D) All connections to a node selective for right directed motion. (E) Same as D but for left directed motion.

was set to 20‰ in most of the trials described here. For the images presented in this paper, spontaneous activation has been removed. Hence, the noise that appears in these images is not purposively put there.

To produce the spontaneous activity, the SUN C function `random()` was employed rather than `rand()`. The reason for this is that the latter function produced a repetitive pattern across the matrix. The code used for each pixel in each matrix to generate the spontaneous activity was thus the following:

```
if(abs(random())%1000 <= arousal)
   setpixel(ON);
```

Throughout this paper, the parameter settings will be expressed in tables analogous to Table 1. This specific table represents the parameters that gave the best result.

Input connection		Lateral excitation		Lateral inhibition		Node characteristics	
No. of nodes	1	No. of nodes	15	No. of nodes	7	Threshold	1000
Strength	800	Strength	110	Strength	200	Arousal	2.0%

Table 1: Standard parameter settings for the motion detecting network.

4.1 Line orientations

In addition to the motion direction selective nodes, the model also includes nodes selective to line orientations. Four different orientations are extracted. The architecture of this subnetwork is similar to the one selecting motion. The main difference is the lateral excitation. In the line-orientation nodes, this excitation comes from two 1×5 matrices positioned on two opposite sides of the target node (Figure 5). In addition to this architectural difference, the nodes used for this purpose use temporal summation. Half of the internal activity of the nodes (i.e., the sum achieved by integrating the connections) are kept between two iterations.

The parameter settings that gave the best result are shown in Table 2.

4.2 Mutual inhibition

The nodes selective for line orientations and the nodes selective for motion directions inhibit each other. The reason for this is that the nodes selective for line orientations are potential bidirectional motion detectors and the motion selective nodes are potential line orientation detectors. This can

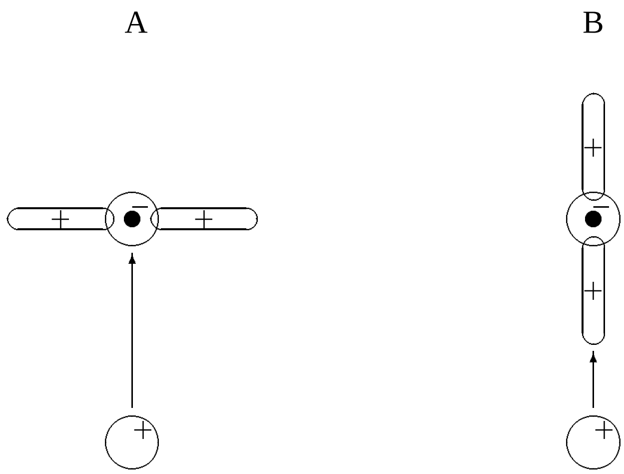


Figure 5: Schematic outline of the network architecture. The target node is illustrated by a filled circle. The other fields illustrate sets of nodes. (A) The target node is selective to horizontal lines. (B) The target node is selective to vertical lines.

Input connection	
No. of nodes	1
Strength	512

Lateral excitation	
No. of nodes	10
Strength	80

Lateral inhibition	
No. of nodes	3
Strength	100

Node characteristics	
Threshold	1000
Arousal	2.0%

Table 2: Standard parameter settings for the line detecting network.

be seen most clearly when comparing the architecture of these two subsystems. The strengths of the inhibitory connections are shown in Table 3. Refer to section 5.4 for a discussion on why this inhibition is needed.

Mutual inhibition	
line extraction→motion detection	512
motion detection→line extraction	500

Table 3: The inhibitory strength between the two kinds of selectivity.

5 Results

This section offers a compilation of case studies on how different parameter settings will affect the outcome. Some notes on the snapshots illustrated here will be helpful: To allow comparisons, all images illustrated are the 30:th frame in the sequence except for Figure 8 and section 5.5. The figures illustrating all eight motion directions are positioned as illustrated in Figure 6.

For compact illustrations, certain figures show only four frames. These are, in order, the source image used as input, the left motion detected, a combination of motions detected to the left, to the upper left and to the lower left, and, finally, a combination of the vertical and horizontal lines detected. The frames showing mixtures of images were obtained by an `OR`-operation on the pixels, i.e., if a pixel was set `on` in at least one of the images involved, it will also appear in the combination.

↖	↑	↗
←		→
↙	↓	↘

Figure 6: Motion directions.

5.1 STANDARD PARAMETER SETTINGS

A good selectivity is obtained when running the simulations with the parameters set as in section 4, which from here on will be referred to as the standard parameters. As can be seen in Figure 7, the person that walks leftward in the input sequence appears primarily in the matrix of left motions, but also in the matrices of upper left and lower left motions and only minor in the other fields. In summary, there is a strong convergence of the moving pattern to the matrix of left motions.

From the figure it can also be seen that the very same pixel in the input image appears in some cases in more than one of the matrices that indicate motion occurrences. That is, the output tells us that the one and same spot is simultaneously moving in different directions. There is not necessarily a contradiction in this – it all depends on how the output is supposed to be used. In the case of this model, the output is not intended for a human observer, nor for generating the information that a certain spot is moving in a specific direction with a specific speed. Rather, the output is intended as input to other neural networks. These networks interpret the activity in the output from this model as *suggestions* of how a certain spot is moving. Given the original principles under which the network operates, the activity of a single node contributes only to a minor degree to the global function. Indeed, the activity of a single node is negligible – but, of course, the individual activities of all nodes cannot be ignored at the same time.

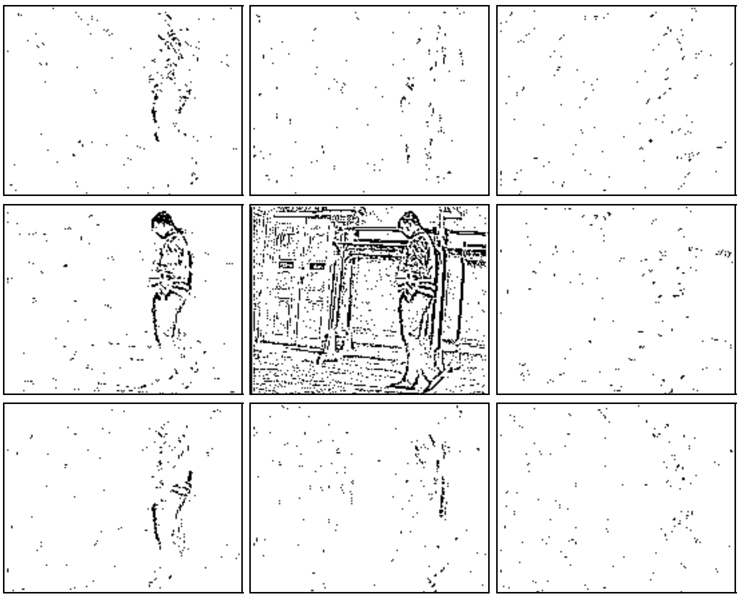


Figure 7: Motion detection obtained with standard parameter settings.

Figure 9 shows the line extracted with the standard parameters. As can be seen in Figure 7, the source image does not contain diagonal lines, but contains several lines that are approximately vertical or horizontal. Because the line orientation detection was supplied only for the reason of inhibiting the motion detecting nodes, this respect of the network has not been given as

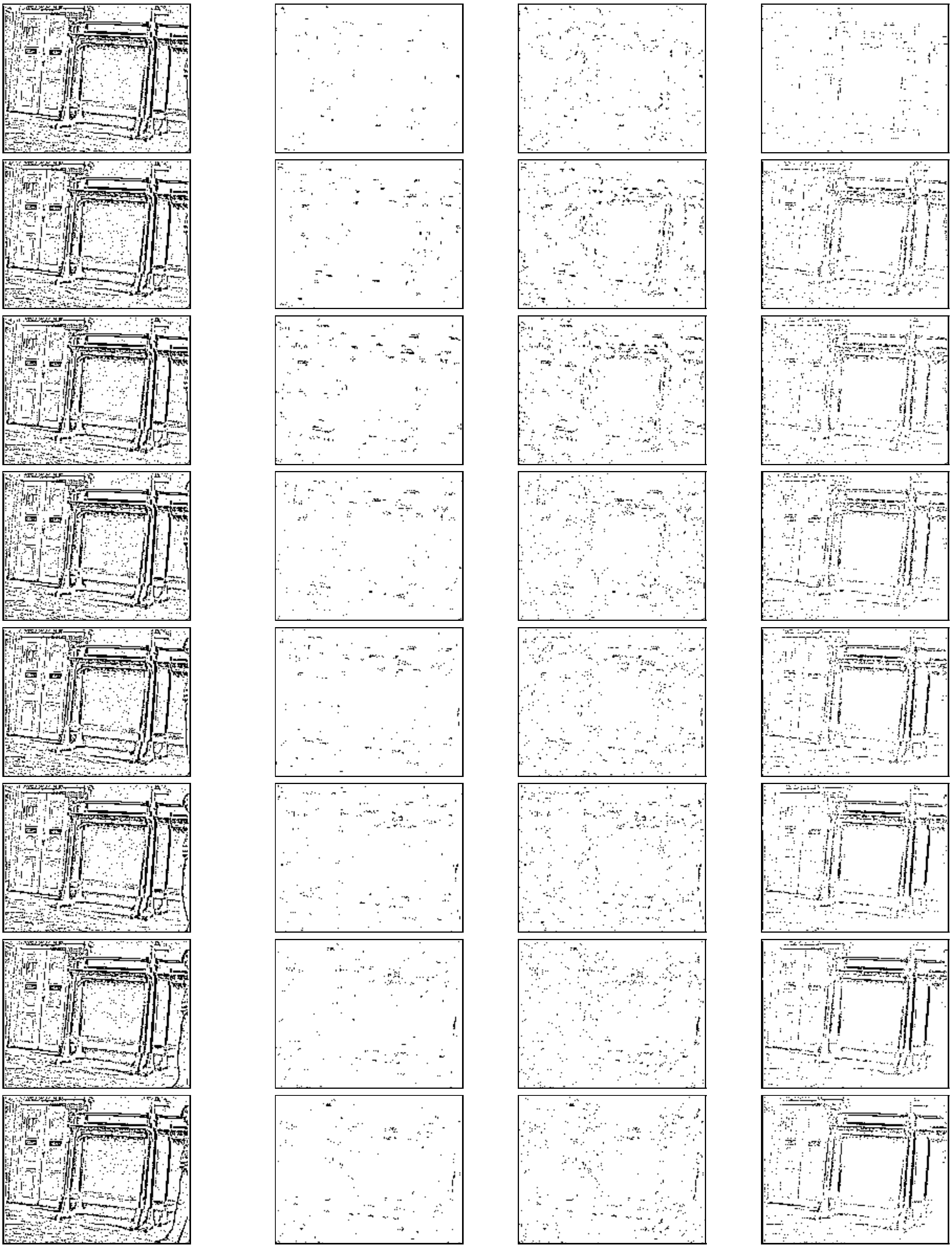


Figure 8: The 16 first frames of source, left motion, left + upper left + lower left motion and vertical + horizontal line orientations.

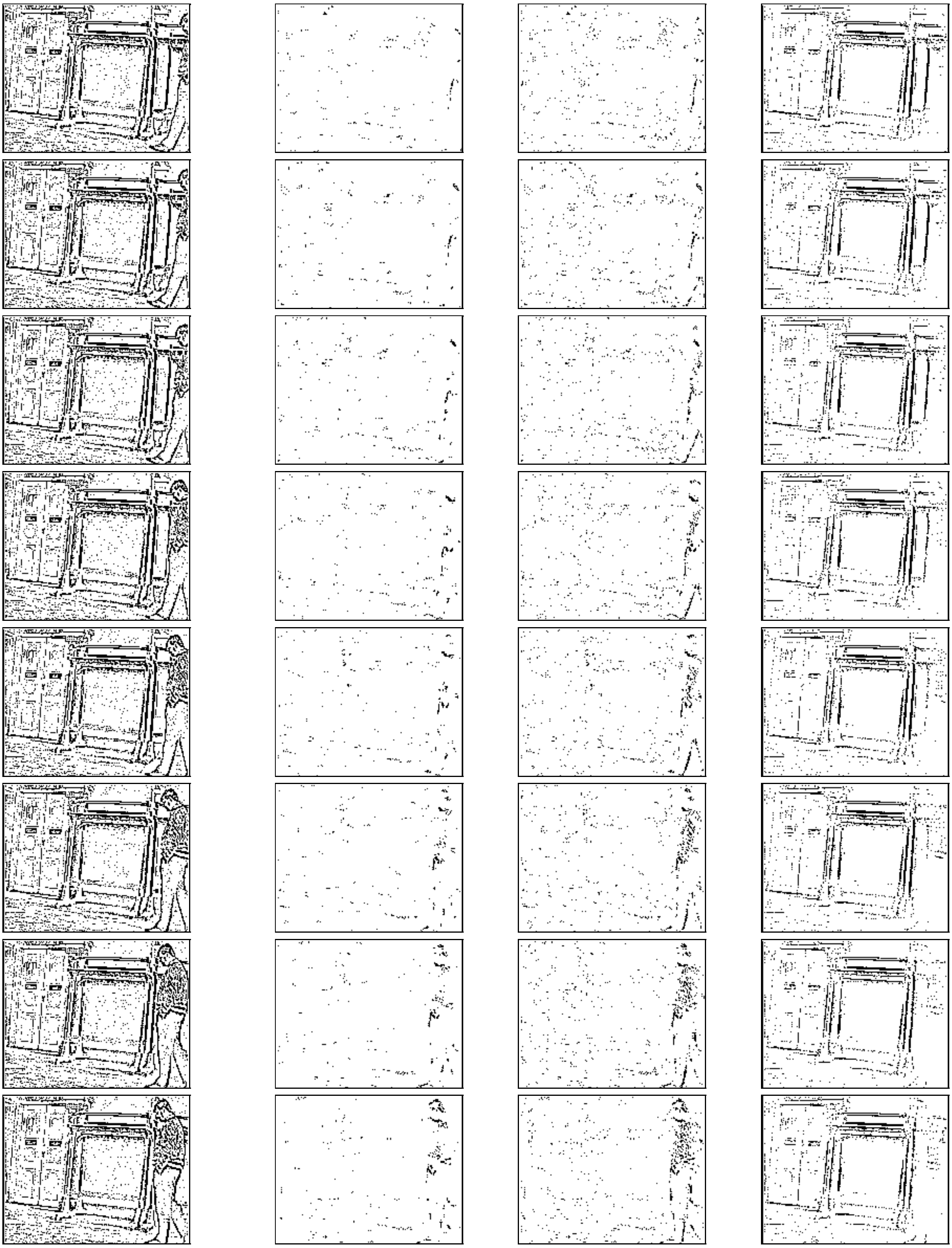
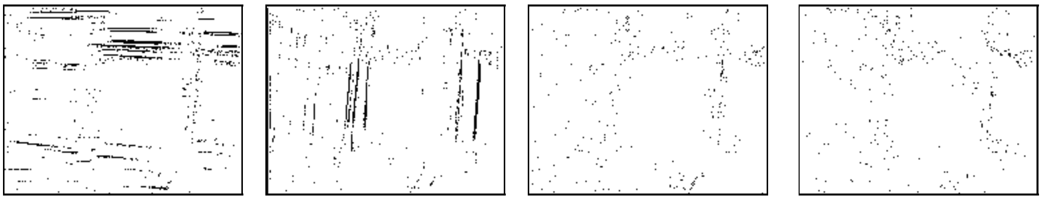


Figure 8: continued.

detailed examination as the motion detection part. The paper is primarily concerned with how the parameter settings affected the motion selectivity.



Horizontal Vertical /–diagonal \–diagonal

Figure 9: Line orientation detection obtained with standard parameter settings

Because the system is stochastic in nature, two successive frames are not sufficient for extracting motion. Figure 8 shows the 16 first frames of a simulation. For comprehension, the output is shown in selection. Note how the detection of the moving figure grows out from a number of ignition points.

5.2 "EPILEPSY"

When the lateral excitatory connections are too strong (see Table 4, 5, and 6), an interesting phenomenon emerges. At sites where motion are detected, the lateral connections will be enough to keep the activity going once it has been ignited. Moreover, the excitation spreads, and large black spots of activity appear (Figure 10). The first interpretation of the phenomena that comes to mind is *epilepsy* – uncontrolled activation.

Note that even in the standard parameter settings, the lateral excitation would be able to support such an activity. The reason that epilepsy does not appear in this case is that the inhibition is more in balance with the excitation. There is, however, still a risk that epilepsy will occur. Under normal conditions, though, it does not. This is one reason that the input must not contain large homogeneous areas as the original source obtained from the camera. The filtering procedure reduces the risk of epilepsy to a minimum. If epilepsy does occur with the standard settings, it will spread to the entire matrix in which it appears, if given enough time. There is no exhaustion mechanism in the model to prevent or interrupt an epileptic seizure once it has been ignited. Such a mechanism could, however, easily be added if desired.

Input connection		Lateral excitation		Lateral inhibition		Node characteristics	
No. of nodes	1	No. of nodes	15	No. of nodes	7	Threshold	1000
Strength	900	Strength	128	Strength	300	Arousal	1.0%

Table 4: The parameters for the motion selective nodes. Note the strong excitatory connections.

Input connection		Lateral excitation		Lateral inhibition		Node characteristics	
No. of nodes	1	No. of nodes	10	No. of nodes	3	Threshold	1000
Strength	512	Strength	128	Strength	256	Arousal	1.0%

Table 5: The parameters for the line orientation selective nodes.

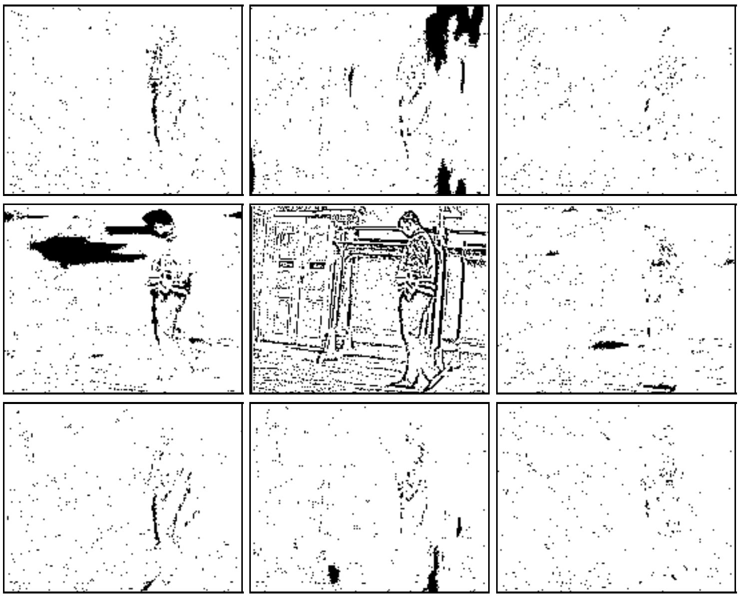


Figure 10: To much excitation causes uncontrolled activation to spread.

Mutual inhibition	
line extraction→motion detection	256
motion detection→line extraction	512

Table 6: The inhibitory strength between the subnetworks.

5.3 Removing the lateral inhibition

In the standard parameter settings, a node selective for a specific direction receives inhibition from nodes located at the same topological position but tuned to different directions. This means that only the nodes with the strongest support for that motion (i.e., the node that receives the most excitatory inputs) will become activated. It should therefore come as no surprise that when the lateral inhibitions between these nodes are removed, more nodes will indicate motion. The result of such a simulation, run with the parameters from Table 7, can be seen in Figure 11.

Input connection	
No. of nodes	1
Strength	800

Lateral excitation	
No. of nodes	15
Strength	110

Lateral inhibition	
No. of nodes	7
Strength	0

Node characteristics	
Threshold	1000
Arousal	2.0%

Table 7: The same parameters as in the standard case was used except for the inhibition strength which was set to zero.

5.4 Removing the mutual inhibition

When the mutual inhibition between the nodes selective for line orientations and the nodes selective for motion directions are removed, a strange phenomenon occurs. As the model works under the principle that motion is propagated in each matrix when support can be found in the source image, the motion selective nodes become selective not only for motions but also for static lines. The reason for this is that if a motion detection is ignited on such a static line, it will propagate along that line with support from the source image.

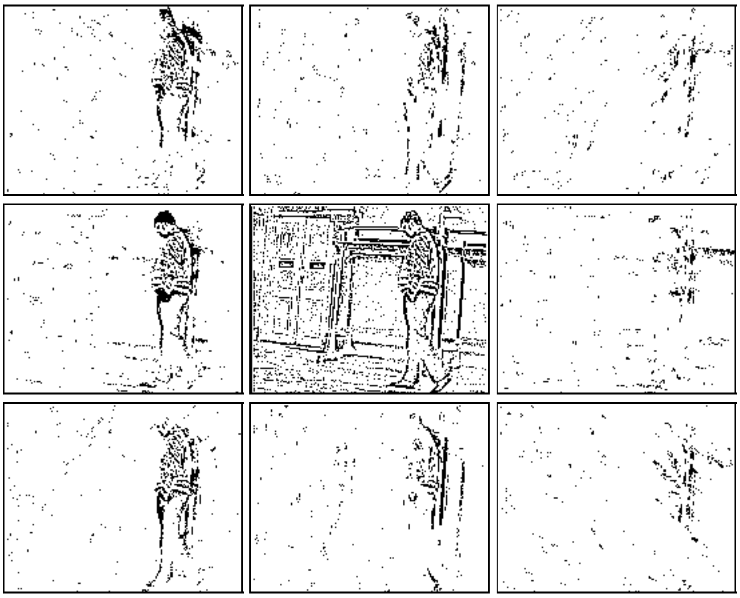


Figure 11: Without the lateral inhibition the selectivity looses its acuity.

The static lines in the source image can be said to decrease the threshold in the motion selective nodes. To compensate for this effect, the static lines can be detected by another mechanism which in turn inhibits the nodes detecting motion, or, in other words, increases the threshold.

Another possibility to overcome the problem would be to engage transitive nodes in the input chain, i.e., nodes that only responds to changes in the image. This was suggested to me by Sivert Lindström and there is no doubt that it would work and indeed be an easy solution. I was, however, interested in whether it would be possible to overcome the problem without engaging transitive nodes. And, as has been shown in this paper, it was.

The manner in which I implemented the static line detectors introduced a new problem. These nodes were potential motion detectors. Therefore, it was necessary to inhibit these nodes by the motion detecting nodes for the same reasons as the opposite connection. The situation thus arisen involved circular dependencies between the sub-networks of motion and line detection which made it hard to tune the parameters in the model.

An interesting point can be made on how this mutual inhibition relates to biology. It has been observed that the inhibition to a motion selective cell is contra-intuitively coming from nodes with the same selective characteristics as the inhibited cell itself (Sivert Lindström, personal communication). A leftward motion selective cell appears to be inhibited by leftward motion selective cells. The mutual inhibition involved in this model offers a possible explanation of this strange phenomenon. A node selective for leftward motion needs, in this model, primarily to be inhibited by nodes selective for horizontal lines. This horizontal line selective node in turn operates potentially as a motion detector for leftward and rightward motions. Under the special circumstances of clinical experiments, the nodes selective for lines inhibiting the motion selective node, and this node itself might appear to have the same selective characteristics.

Input connection	
No. of nodes	1
Strength	900

Lateral excitation	
No. of nodes	15
Strength	50

Lateral inhibition	
No. of nodes	7
Strength	200

Node characteristics	
Threshold	1000
Arousal	2.0%

Table 8: The parameters used when studying the effects of removing the mutual inhibition.

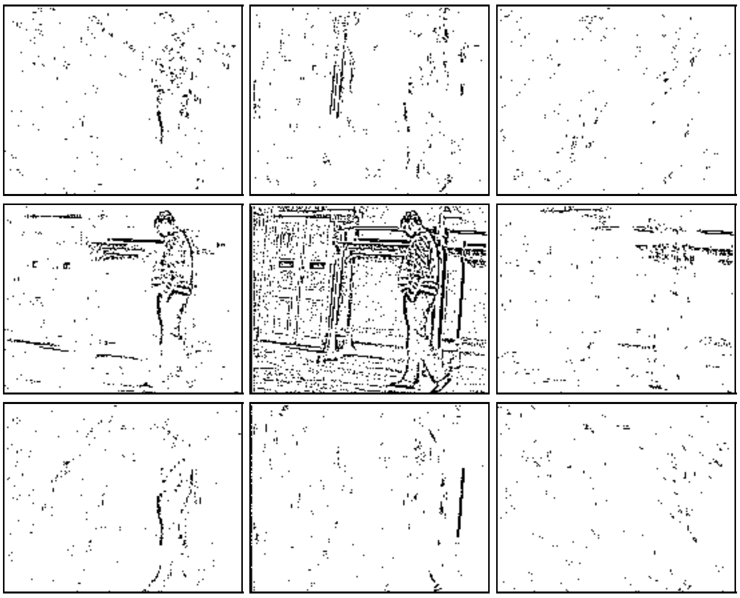


Figure 12: Without the inhibition from static lines, “ghost” motions will appear.

5.5 The camera mounted on a mobile robot

To test the model under more realistic conditions, the camera was mounted on a mobile robot (Figure 13). It was attached directly on the aluminum chassis of the robot without the use of any shock absorber.

The robot was maneuvered by radio control and driven on the floor of the department. It was run forward, backward and turning around. The only useful data for the model turned out to be when the robot was driving straight ahead. While the robot was turning, the camera image became obscure. This problem is not inherent in the model, but would be solved by using a faster camera.

Figure 14 shows the motion extracted when the robot has been driving forward for about ten frames. The result shows that the visual flow that appears during such motions indeed are detected. The flow to the right and left sides as well as the diagonal flows show the best results. The upward and downward motions are, however, more obscure.

One problem exists, however. The line orientation extraction turned out to be too slow for this application (see Figure 15), therefore it does not inhibit the motion detectors from detecting static patterns. While in motion, though, there are few such static patterns appearing in the image. The result probably would have been better if the sample speed had been increased. This was not possible with the equipment at hand, however.

5.6 Noise resistance

Some readers might object that the signal to noise ratio is bad in this model. They are correct in that there is noise in the output, but the output is not intended for human observers. Rather, it is intended for networks operating with the same technique used in this model. By showing that this model is unusually insensitive for noise in its input, I hope that the reader will agree that the noise appearing in the output of this model is not a problem.

As was mentioned earlier, noise is actually added to the network in the form of spontaneous activity. The amount of this noise is modest, however. In a sequence of trials I have added noise to the source image as well. I have tried with 10, 25, and 50% of noise. That is, the 50% noise

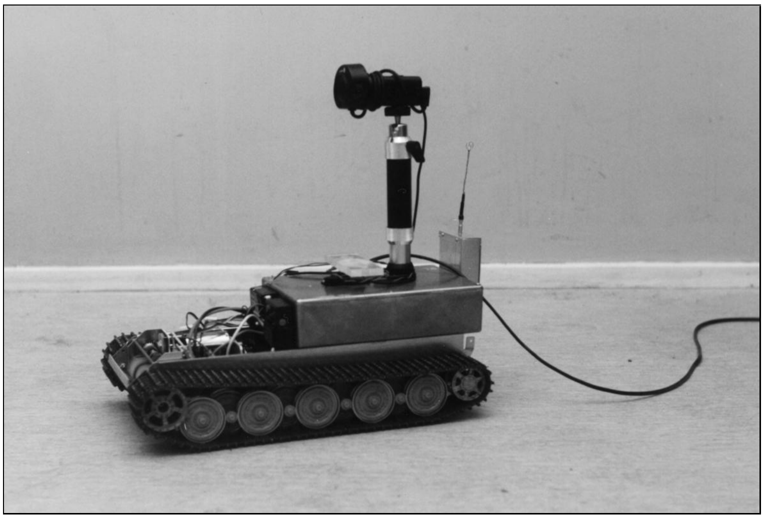


Figure 13: The robot used for the recordings described in this section

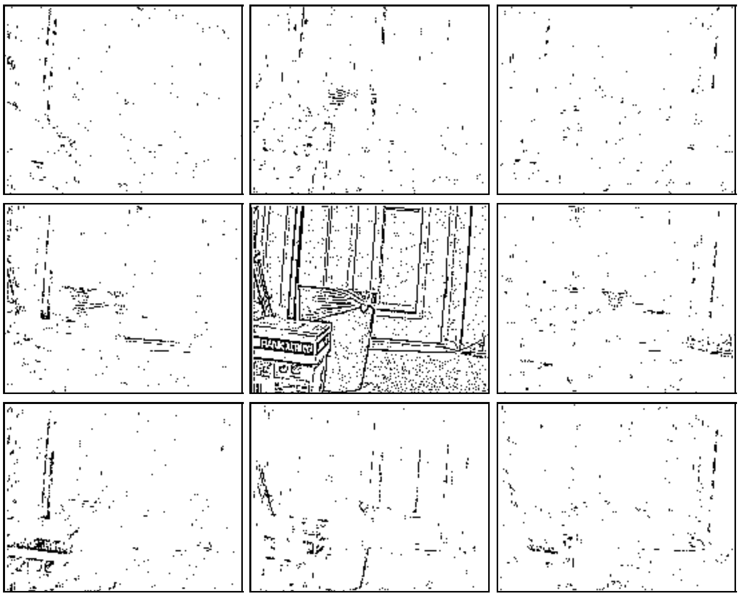
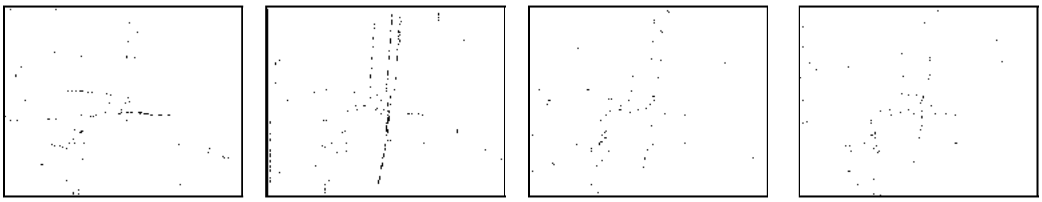


Figure 14: The motion extracted when the camera was mounted on a moving robot.



Horizontal Vertical /–diagonal \–diagonal

Figure 15: Line orientation detection obtained using the moving robot.

level means that half of the nodes in the source image are randomly given the value `on` or `off`.

The results of these trials are shown in Figure 16–19. Up to 25% of noise, the result is acceptable. At the level of 50% noise, the result is not as good, but the motion selectivity is still evident. The nodes selective for line orientations show a better result for all trials. The reason for this is, in essence, that these nodes use temporal integration of its input, and is thus even more stable against temporal fluctuation (noise).

The reason for the comparatively strong noise resistance of this model is that the model processes motion as a *stable* phenomenon. The common approach to motion detection is otherwise to detect what has *changed* in two successive images. The problem that such models confront is that *noise* is also something that changes between successive images. The model presented here, does not have to deal with the complications that arise in earlier models.

In models that detect what has changed between two successive images, noise will create such an effect with the probability $2p(1-p)$, or $2p - 2p^2$. For noise to be a problem in the model presented here, the noise must correlate across several successive images. Two pixels correlate in their noise activity between two successive images with the probability p^2, where p is the noise level. The details of how the noise affects the activity in the output matrices is, however, more complex than that. The more detailed picture depends on the number of nodes connected and the value of the threshold among other. But it should be clear that the problem with noise grows slower than linear to the noise level in this model.

ACKNOWLEDGEMENTS

I thank Lars Kopp for kindly lending me his video camera, Christian Balkenius for help with the construction of the software used in the simulations and Christer Johansson for valuable comments. Further, I am grateful to all those people who have contributed with comments throughout this work.

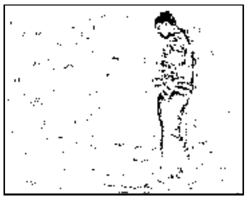
This work has been supported by the Swedish Council for Research in the Humanities and Social Sciences.

REFERENCES

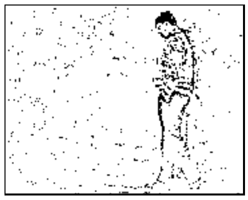
Pallbo, R. (1992). Neuronal selectivity without intermediate cells. *Lund University Cognitive Studies*, 13.

Pallbo, R. (1993). Visual motion detection based on a cooperative neural network architecture. In Sandewall, E. & Jansson, C. G., editors, *Scandinavian conference on artificial intelligence – '93*, pages 193–201. IOS Press.

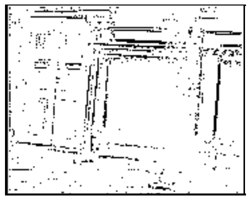
Source



Left motion



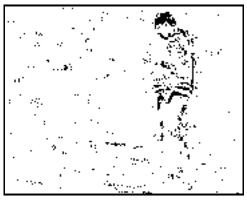
Left, upper left and lower left motion



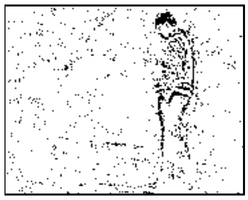
Vertical and horizontal lines

Figure 16: 0 % added noise.

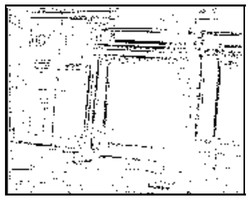
Source



Left motion



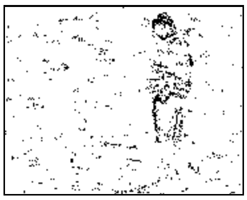
Left, upper left and lower left motion



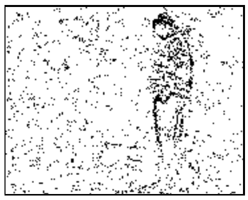
Vertical and horizontal lines

Figure 17: 10 % added noise.

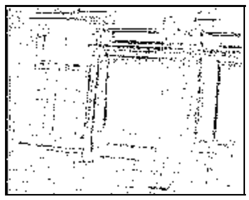
Source



Left motion



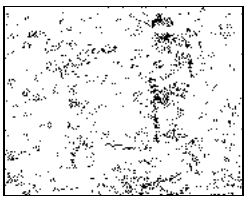
Left, upper left and lower left motion



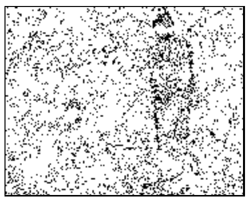
Vertical and horizontal lines

Figure 18: 25 % added noise.

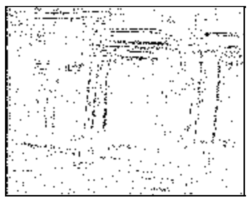
Source



Left motion



Left, upper left and lower left motion



Vertical and horizontal lines

Figure 19: 50 % added noise.