Massey University Library
Thesis Copyright Form

Title of thesis:

(1) (a) I give permission for my thesis to be made available to readers in Massey University Library under conditions determined by the Librarian.

(b) I do not wish my thesis to be made available to readers without my written consent for ... months.

(2) (a) I agree that my thesis, or a copy, may be sent to another institution under conditions determined by the Librarian.

(b) I do not wish my thesis, or a copy, to be sent to another institution without my written consent for ... months.

(3) (a) I agree that my thesis may be copied for Library use.

(b) I do not wish my thesis to be copied for Library use for ... months.

Signed SR Long

Date 17/12/92

The copyright of this thesis belongs to the author. Readers must sign their name in the space below to show that they recognise this. They are asked to add their permanent address.

NAME AND ADDRESS DATE

Samantha Robyn Long 17/12/92
71

TRACKING VERTEBRAE IN CINEMATIC FLUOROSCOPIC X-RAYS

A thesis presented in fulfilment of the requirements
for the degree of Master of Technology
at Massey University

Samantha Robyn Long
October 1992

Abstract

This thesis concerns the evaluation of an image subtraction statistic that is used by a prototype of a chiropractic image processing package to track spinal movement. The image subtraction statistic is calculated by summing the absolute differences in pixel intensity of two images. Also included in the thesis is a brief discussion of different methods of tracking and a literature search of alternative statistics that may be appropriate for the image type (low contrast and noisy).

In summary the experimental work concluded that inter frame rotation does not have a significant effect on the performance of the image subtraction statistic when tracking inter-frame but when tracking from a particular frame to one which is significantly later in the sequence rotation must be included in the algorithm. It was also found that discretisation of the image had a detrimental effect on performance. This can be compensated for by adding a sub-pixel location calculation into the algorithm. In the original prototype a median filter (rank 5) was used to smooth the noise in the image to be searched. This was found to have marginal affect on the performance of the statistic.

Many of the algorithms presently defined in the literature were found to be unsuitable for this application as they tracked clearly defined lines or searched for a two-dimensional shape that matched a predefined three-dimensional model.

An algorithm that may prove to be a suitable alternative compared the rate of change in intensity across a window so is based on locating a change of intensity pattern rather than a pixel to pixel comparison.

There are some features that could be included in the tracking procedure to make the algorithm more efficient (the two-dimensional logarithmic search) and provide checks

to safeguard against points incrementally deviating from the correct location as tracking progresses (referencing a moused frame, using the vertebra rigid body property). The benefit of incorporating the safeguard features would have to be weighed against the cost of extra computational time.

In conclusion, the image subtraction technique can be improved from, in some cases, total tracking loss to accuracy within two pixels of the correct location. This is achieved by tracking inter frame, that is from one frame to the next in the video sequence, and including a sub-pixel location calculation.

Acknowledgements

I would like to take the opportunity to thank the following people for their assistance with my research:

- Professor William Monteith for his supervision and motivation.
- Wyatt Page for his patience and comradeship in deciphering and reorganising early programming code.
- Dr Roger Browne, Wyatt Page, and Elliott Blennerhasset for the foundations they laid with their original ideas and work.
- Dr Les Whitehead of "Whitehead Chiropractic" for making available the patient data from his cine X-ray equipment.
- DSIR for their funding and support meetings.
- My husband, Philip, for his patience, understanding, support and encouragement.
- Our Creator who makes all things possible.

Index

Chapter 1
Introduction

In pursuit of progress chiropractors are turning to scientific diagnostic aids. One such innovative chiropractic package was designed and a prototype produced by a team of Palmerston North technologists in association with a local chiropractic clinic.

The package allowed an X-ray video of spinal movement to be captured to computer hard disk, points of interest to be marked on an initial frame using a mouse and these points to be automatically located on each subsequent frame of the video. From here analysis software allowed calculation of vertebra rotation, planar displacement and axial displacement and their rates of change. Graphs of this data allow the chiropractor to monitor how vertebrae interact - a new dimension in diagnosis after years of only static x-ray film and measurements.

1.1 Dynamic versus static measurements of the spine

The use of static X-ray images as a diagnostic tool for medical practitioners means that dynamic features of the human body movement are not shown. For instance, a static X-ray image of the spine shows the position of the vertebrae at the instant of exposure, but the relative positions of the vertebrae through the full range of motions of the body are not available. Characteristic data of static measurements include the relative angular position, feature separation, planar translation and structure definition. Cinematic X-ray motion sequences of the human spine were first produced by Reynolds [Reynolds, 1938] in 1921 but accumulated X-ray exposure for even a short sequence was large and consequently this method was not widely used because of the danger to patient and operator health. The use of modern highly sensitive image intensifiers has dramatically reduced the X-ray dose. This coupled with a highly sensitive video camera, enabling the display of motion X-ray video images on a conventional VDU, makes dynamic measurements of spine movement possible. Some characteristics of the dynamic data are feature angular velocity, feature angular acceleration, feature separation velocity, feature separation acceleration, feature translation velocity, feature translation acceleration and rate of change of acceleration (jerk). The equations used to calculate these dynamic measurements are derived in chapter 3. These dynamic characteristics may prove useful in the diagnosis of spinal disorders.

In order for the dynamic characteristics to be calculated static measurements have to be made for each frame. The process used to define the points of interest and then locate them in each frame is called 'tracking'.

1.2 The Tracking Procedure

The procedure used by the prototype to track spinal movement is for the clinician to display an initial frame on a computer screen and to define the set of points of interest (usually corners of vertebrae or the edge of the skull) using a mouse. The purpose of the tracking software is to map these points through a sequence of frames. When tracking is complete the sequence can be reviewed with the tracked points in place. The sequence of tracked points may also be used to calculate quantitative features of the movement of the spine which may be viewed in tabular or graphical form to aid diagnosis of the spinal condition. The tracking software compares the window surrounding the known point with windows of the same size within a precalculated area of probable point location in the next frame. An algorithm is used to measure the similarity of the windows. In the prototype of the tracking package the algorithm is based on the sum of absolute intensity differences [Schalkoff, 1989]. The absolute difference in intensity between each corresponding pixel within the window is calculated and summed. This algorithm is explored more fully in chapter 4.

1.3 The prototype algorithm

The prototype of the spinal tracking package initially gave a very variable performance. Because of the limited knowledge of the algorithm behaviour for the particular images being used it was unclear what was causing the variations in performance. This report concerns the exploration of the affects of some of the more obvious possible factors. There were some surprising and exciting finds.

The first factor, the quality of the original point selected, was improved by user training and implementing a test on a selected point for tracking quality. The user was requested to select points on definite features in higher contrast areas. The moused point was compared to those in the surrounding area as a point that is not unique from the neighbouring points has questionable tracking ability. For example, a point selected along an edge of a feature will not be clearly distinguishable from other point along the same edge. When a point was deemed as an unsatisfactory choice the user was given the option of selecting an alternative point to track.

The effects of rotation and discretisation were explored. Neither were taken into account in the original algorithm. The main reason for this was that for unknown benefit there would be severe increases in calculational overhead.

From a series of tests it was found that rotation could be disregarded if tracking was strictly interframe. Interframe tracking, or tracking from one frame to the immediate next in the sequence, has itself a cost as points that deviate from their correct position will not recover.

Discretisation was found to cause incorrect locations to be selected. There are two straight forward ways around this problem. The first is to always track from the reference moused frame. This will mean that discretisation will not cause most sensibly selected points to deviate more than a pixel and the tracking error over a sequence will not be cumulative. The cost of always using the reference frame is that rotation and intensity scaling has to be included in the calculations. The second method of overcoming the effects of discretisation is to calculate sub-pixel location. This means that although the location of a point is written to the screen as an integer pixel position when the point is tracked to the next frame the previous location may be a fraction of the way across the pixel.

1.4 Algorithms available from literature

A number of the algorithms documented in literature can not be used directly on the vertebra tracking application but the ideas can be modified to improve the performance of the simple sum of the difference window statistic. For example it is not possible to have a predefined three-dimensional model of the vertebra being tracked but because the z rotation is kept to a minimum a two-dimensional outline of the vertebra can be used to aid tracking if the majority of the points remain accurate.

Other algorithms track lines or curves - edge detection may give enough outline of the vertebra to enable tracking of that line. The success of this method would be very dependant on the stability and clarity of the vertebra outline.

Many algorithms that predict the location of missing points depend on the smoothness of motion criteria in order to interpolate the movement of the point over the image series. Patients spinal movement is often jolty in the problematic cases that would be videoed for analysis.

Two alternative statistics that could give good results for this application are the moment of inertia statistic and the rate of intensity change statistic. The benefit of calculating the moment of inertia of pixels within a set radius of the specified point is that the result is independent of rotation. The rate of change of intensity across a window is also promising as it is looking for change of intensity patterns and is independent of shifts in intensity from frame to frame.

Chapter 2
Capture of cinematic X-ray

2.1 The patient view

The use of cinematic X-rays as a clinical tool for the recording of the relative movement of skeletal features is very straight forward from the patient point of view, involving a brief time in front of the X-ray equipment. During this time the patient is asked to bend the head down towards the chest and then back. A sequence, typically consisting of ten seconds, is recorded. The clinician records the sequence of images on a standard video tape recorder, to be used subsequently for analysis.

2.2 The image capture chain

The image capture chain involves the X ray camera, image intensifier, video camera, video recorder, and frame grabber. The components of this digital image processor are shown in Figure 2-1. The signal going to the video recorder from the highly sensitive CCD (charged couple device) camera, which is directly coupled with the image enhancer, is a composite signal. The frame grabber discretises the analogue video signal to produce an image of 256 by 256 pixels [Pratt, 1991]. The digitised data is stored and then converted back to an analogue format for display on an external video monitor.

2.3 Digitisation

Digitisation is the process of converting the analogue video signal into a series of digital values (pixels). This is performed by sampling the analogue signal at discrete time intervals and converting each individual sample to a digital value in the range 0 to 255 (0 is black and 255 is white). Intensity changes occur over large interframe distances, and these changes may not be easily predictable.

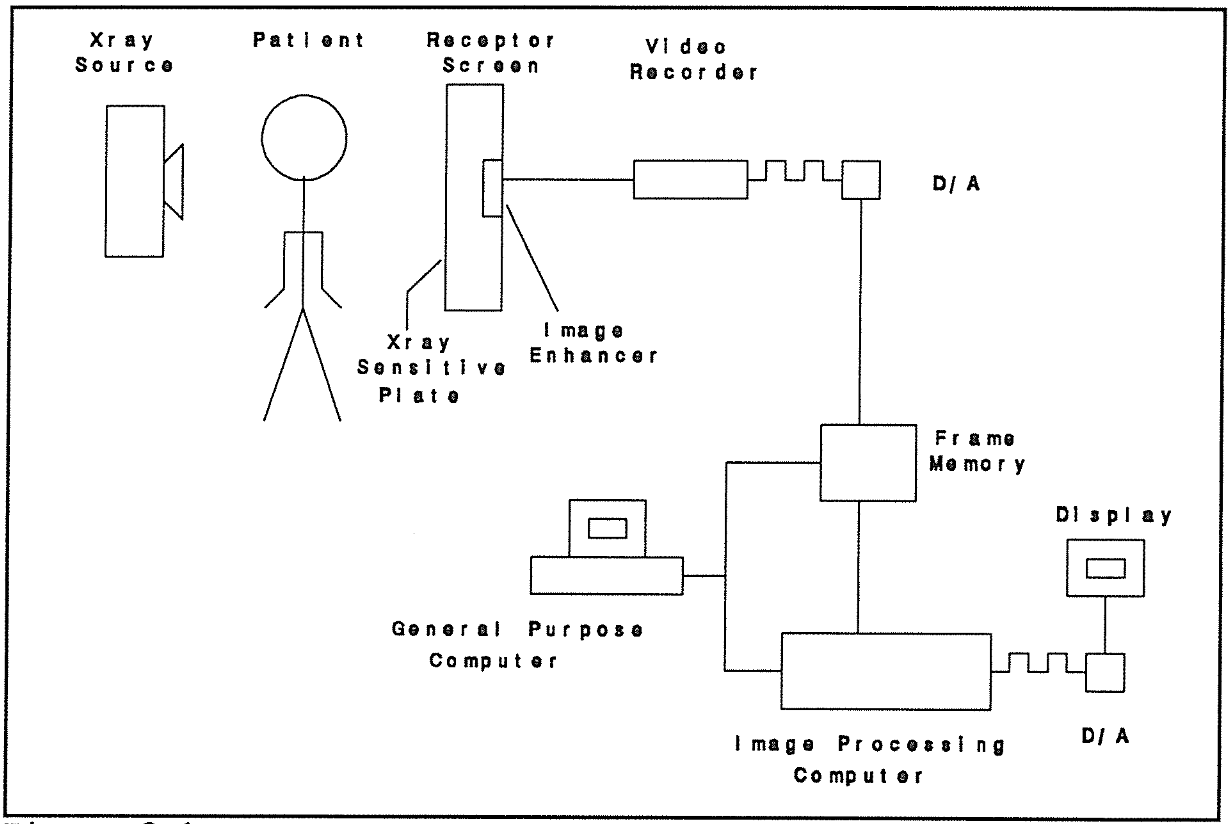


Figure 2-1 Components used to capture cinematic X-ray

2.4 Changes in intensity

The intensity changes are likely to depend on the specific alignment of the camera, the image intensifier, and the X-ray unit, as well as the cleanliness of the video recorder heads and the state of the video tape used. When window matching is attempted from a frame to several later in a video sequence intensity change must be measured and compensated for as it will have a detrimental affect on the measurement of window likeness if a direct pixel to pixel comparison is used. If the similarity of two image windows is measured by an indirect method such as comparing the relative intensity changes across each window then global intensity changes will have a limited affect on the measurement value. The existing chiropractic package uses a direct interframe measurement and as interframe intensity change is usually very gradual no allowance for it is made.

2.5 An example frame

The study reported here will concentrate on the lateral view of the cervical region of the spine. A single frame from a sequence with typical points of interest marked is shown in Figure 2-2. A feature of the images is that some edges are poorly defined. For the cervical region the points that are tracked will typically consist of four points on each vertebrae, plus a few on the skull. It is desirable to pick points in areas of high contrast as near to a corner or vertebra feature as possible. Points marked on a straight edge will not be located easily as the window surrounding the point will not be distinctive from the windows of other points along that same edge. In practical terms when this point is automatically tracked through a sequence and then reviewed the tracked point will randomly slide up and down the edge of the vertebra or skull.

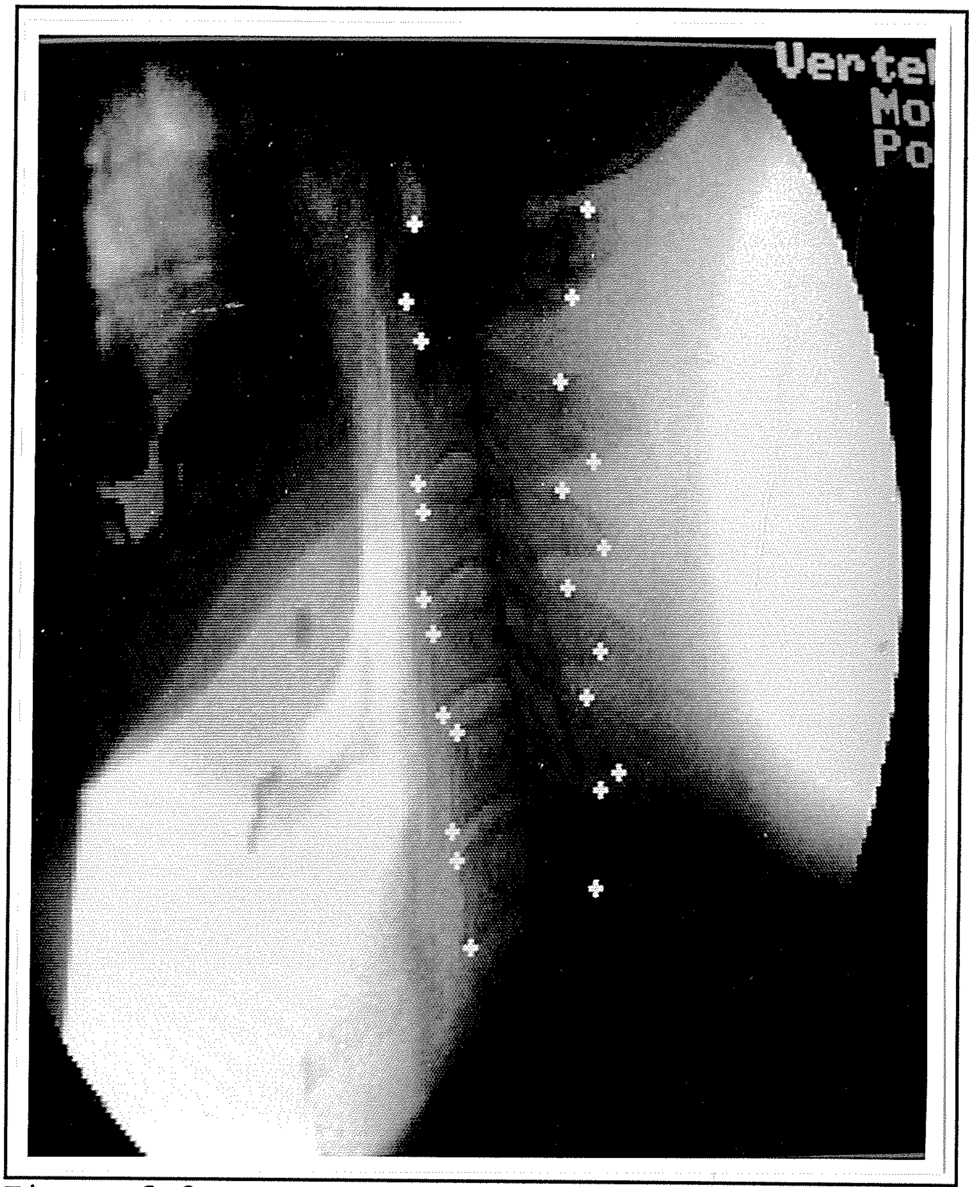


Figure 2-2 Moused points on the lateral view of the cervical region of the Spine

Chapter 3
Measurements of the spine movement

Page [Page, 1991] defined three basic measures of vertebrae behaviour which can be derived from the tracking data. They are rotational, planar and axial displacement. The equations for each of these measurements are given below and fully derived in appendix A. Each of these measurements can be further processed to calculate their first and second derivatives, velocity and acceleration respectively. These three measures were chosen to reflect the various types of vertebrae motion critical to the normal operation of the human spine. Inter-vertebrae rotation provides the primary stimulus for the exchange of nutrients to, and waste products from discs. This function is vital to shock absorbing discs as they have no direct blood supply. Excessive axial translation is undesirable as it stresses the spinal cord by stretching it, while excessive planar translation produces a shearing action that can irritate and potentially damage the outer sheath of the spinal cord. For a healthy patient with normal spinal motion the motion for each vertebra should be smooth throughout the full range of a sequence. For a problematic patient the motion proceeds with several step changes as vertebrae jam then release. Before and after studies of cases identified as problematic who are then treated by a chiropractic specialist should show improvement in their vertebral motion signatures. In the future the motion signatures should offer potential for expert system diagnostics of spinal disfunction and an unbiased view of the impact of spinal treatment.

3.1 Rotational displacement

Rotational displacement is defined as the angle of the principle axis of each vertebra relative to the horizontal. One method of calculating the rotational displacement is to find the angle of the line through the average coordinates of the left hand side marked vertebra points and the average coordinates of the right hand side marked vertebra points. Figure 3-1 is a schematic of two vertebra, a the top one, and b the lower. The normal four corner points marked on each of the two vertebra have a coordinate in order to locate them in the frame, and in the diagram are subscripted with an a or b (to specify the vertebra) and 1, 2, 3 or 4 for top left, top right, bottom right and bottom left corner respectively. The rotational displacement, RD, for vertebra a is given by :

$$RD = \arctan\frac{(y_{a2}+y_{a3}-y_{a1}-y_{a4})}{(x_{a2}+x_{a3}-x_{a1}-x_{a4})}$$

and similarly for vertebra b :

$$RD = \arctan\frac{(y_{b2}+y_{b3}-y_{b1}-y_{b4})}{(x_{b2}+x_{b3}-x_{b1}-x_{b4})}$$

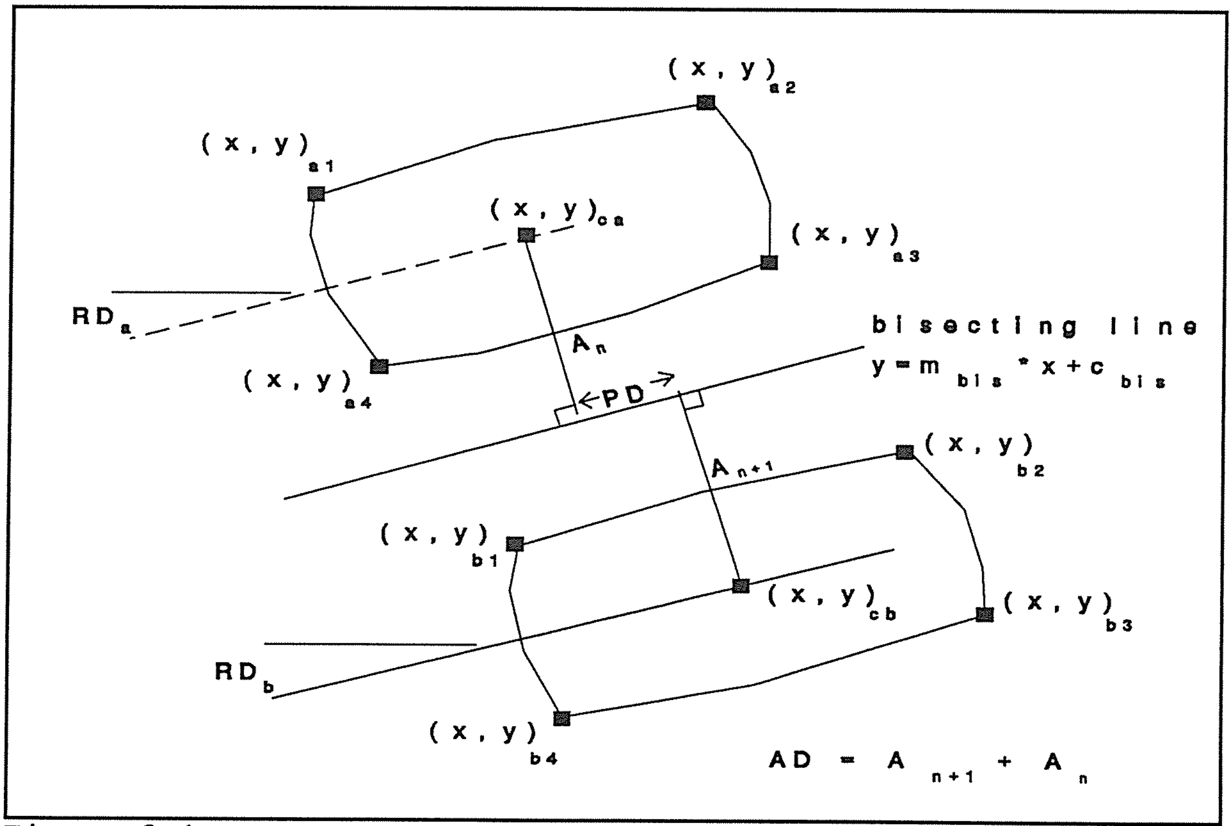


Figure 3-1 Vertebrae geometry

3.2 Axial displacement

Axial displacement, AD, measures the average vertebra separation. Both axial displacement and planar displacement are calculated using perpendicular measures to the line which bisects the space between the two vertebrae in question. This line is calculated as the line through the points given by the average coordinates of the four left hand points on the two vertebra and the average of the four points on the right side of the vertebra. In Figure 3-1 the left point is found from points a1, a4, b1 and b4 and the right hand point from a2, a3, b2 and b3. The AD is then given by the sum of the two distances from the centroid of each vertebrae to the coordinates of the perpendicular intercept of the line from the centroid to the bisection line. The coordinates of the centroid of a vertebrae are found from the average co-ordinates of the four points marked on the vertebrae and will be denoted by $(x,y)_{ca}$ for one vertebrae and $(x,y)_{cb}$ for a second vertebrae. The equation of the line bisecting the space between the two vertebrae is denoted by $y = m_{bis}x + c_{bis}$ where:

$$m_{bis} = \frac{(y_{cb}-y_{ca})}{(x_{cb}-x_{ca})}$$

$$c_{bis} = y_{ca}-m_{bis}*x_{ca}$$

The coordinates of the intersection of this bisecting line and a perpendicular line to the centroid of a vertebra will be denoted as $(x,y)_{ia}$ for the first vertebrae and $(x,y)_{ib}$ for the second. The equations for finding the coordinate $(x,y)_{ia}$ are :

$$x_{ia} = \frac{m_{bis}*(y_{ca}-c_{bis})+x_{ca}}{(1+m_{bis}^2)}$$

$$y_{ia} = m_{bis}*x_{ia}+c_{bis}$$

and similarly $(x,y)_{ib}$:

$$x_{ib} = \frac{m_{bis}*(y_{cb}-c_{bis})+x_{cb}}{(1+m_{bis}^2)}$$

$$y_{ib} = m_{bis}*x_{ib}+c_{bis}$$

Using these definitions the axial displacement, AD, is defined as :

$$AD = \sqrt{(x_{ia}-x_{ca})^2+(y_{ia}-y_{ca})^2} + \sqrt{(x_{ib}-x_{cb})^2+(y_{ib}-y_{cb})^2}$$

3.3 Planar displacement

Planar displacement, PD, is computed as the distance between the two points $(x,y)_{ia}$ and $(x,y)_{ib}$ and is given by :

$$PD = \sqrt{(x_{ia}-x_{ib})^2+(y_{ia}-y_{ib})^2}$$

3.4 Dynamic measurements

The above equations give the static measurements of the vertebrae positions in a single frame and these can be made from a static X-ray. Dynamic measurements use these static measurements from each frame in a sequence to calculate the movement and interactions of vertebrae. The rotational displacement (RD) can be used to calculate the rotational velocity (RV) and acceleration (RA) of a vertebra :

$$RV_{(t+1)} = RD_{(t+1)} - RD_{(t)}$$
$$RA_{(t+1)} = RV_{(t+1)} - RV_{(t)}$$

Similarly axial displacement (AD) is used to calculate the axial separation velocity (AV) and acceleration (AA) of two vertebra:

$$AV_{(t+1)} = AD_{(t+1)} - AD_{(t)}$$
$$AA_{(t+1)} = AV_{(t+1)} - AV_{(t)}$$

and from planar displacement (PD) the planar separation velocity (PV) and acceleration (PA) of two vertebra can be determined:

$$PV_{(t+1)} = PD_{(t+1)} - PD_{(t)}$$
$$PA_{(t+1)} = PV_{(t+1)} - PV_{(t)}$$

In order to implement these dynamic measurements it is necessary to track points through successive frames. In the prototype of the chiropractic spinal tracking package the sequence of frames is recorded from the video onto the computer hard disk. In order to track points through the sequence a computer algorithm uses a statistic to locate a given point in the next frame. The current algorithm sums the absolute results of an image subtraction.

Chapter 4
The sum of absolute difference statistic

4.1 Theory of window matching

The aim of image window matching is to match a window from one image with the closest to identical window in a second image. The objective of window matching as used in tracking is to determine the relative movement of an object within an image as accurately as possible. Let f(i,j) represent a measure of the pixel intensity value at the location (i,j) in the image function f. This may be the actual intensity or some form of normalized intensity, the most useful of which is the intensity divided by the local area brightness (the mean pixel intensity within the window). In order to determine the best match window a measure of similarity must be used. The chiropractic package prototype uses a window size of eleven by eleven pixels, for reasons discussed in Chapter 7. The image window centred on the known or estimated location of the point being tracked is searched for in the image where the point is to be located. The algorithm is effectively locating an image window not a point but as the point is known to be the centre of the first window it is reasonable to assume the location in the searched image will be the centre of the best match window. The sum of absolute differences from an image [Schalkoff, 1989] (in this case window) intensity subtraction was used to compare windows.

4.2 Using the sum of absolute differences to compare windows

The sum of absolute differences (SAD) statistic uses image intensity subtraction which although elementary has been found to be a quite powerful technique of showing changes in a dynamic scene [Jain, 1989]. The absolute differences of the two windows are summed giving a positive integer value which is a measure for how closely the two windows matched. Obviously for a perfect window match the value obtained will be zero. Let the two frames be denoted by p and q, and let f_p' represent f_p translated and rotated. Then the tracking procedure for a single point consists of minimizing S_1 where :

$$S_1 = \sum_{i,j}^{window} | f_p'(i,j) - f_q(i,j) |$$

The simplest approach is to adjust the location and rotation of image p until S_1 is minimized.

4.3 Initial performance of the SAD statistic

The SAD algorithm performed with limited success when implemented for interframe vertebra tracking. Four points, one at each corner, were marked on vertebra C1 to C6 on frame 16 of a 32 frame sequence. These points were tracked from frame 16 through to frame 31 and then from frame 16 through to frame 0. The radial error of each point was calculated as the distance from the correct location to the tracked location of the point. The sum of the four radial errors of each vertebra are shown for each frame in Table 4-1. The 'smoothing' refers to a median filter (rank 5) that was used to smooth the noise in the image being searched. The data from Table 4-1 is graphed in Figures 4-1 to 4-2.

There were a number of possible causes for the unsatisfactory performance such as:

- the nature of the data (poor definition and low contrast),
- no allowance for the effects of discretisation of the image,
- no allowance for interframe rotation,
- no recovery or self check procedure during tracking.

The affects of each of these causes are examined in the following chapters.

Table 4-1 Sum of radial errors (pixels) for each vertebra for each frame

Frame No	RADIAL ERROR FOR EACH VERTEBRA WHEN TRACKED USING SAD STATISTIC (WHOLE-PIXEL) AND SMOOTHING						RADIAL ERROR FOR EACH VERTEBRA WHEN TRACKED USING SAD STATISTIC (WHOLE-PIXEL) AND NO SMOOTHING					
	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6
0	6.48	9.36	10.80	10.36	9.64	10.47	5.83	10.60	14.56	13.08	8.26	125.52
1	8.72	10.52	11.72	10.26	10.08	11.04	7.24	10.18	13.69	12.91	10.63	104.24
2	8.12	11.73	10.23	9.40	5.65	9.54	7.65	10.52	12.47	11.50	4.65	81.63
3	6.89	11.77	9.30	9.40	8.24	10.87	6.65	10.77	11.06	10.91	8.43	66.61
4	8.12	8.24	9.30	9.05	5.06	9.94	5.83	7.81	13.29	10.91	3.65	46.36
5	7.24	9.16	7.65	7.71	5.06	15.18	4.65	8.74	12.76	8.94	4.65	35.11
6	4.65	8.41	7.89	5.47	9.08	8.00	2.41	7.89	13.26	8.12	8.06	19.54
7	5.65	11.06	6.89	6.47	5.66	6.66	4.83	10.30	12.43	8.72	5.24	14.73
8	6.06	8.41	7.06	6.06	6.06	7.89	3.41	6.65	9.71	7.48	3.83	14.60
9	5.65	9.06	6.06	5.24	6.06	8.80	4.00	8.06	8.48	5.24	6.06	11.90
10	5.24	7.24	5.41	4.41	5.65	6.61	3.00	5.65	5.83	4.83	5.65	7.24
11	3.41	8.65	5.00	4.41	3.41	7.41	4.83	7.65	5.83	4.83	3.41	8.41
12	3.24	7.24	5.00	3.41	2.00	4.65	4.24	6.24	4.41	3.83	2.00	6.24
13	4.83	4.00	4.24	3.00	5.00	1.41	4.41	4.00	2.83	3.00	4.24	1.00
14	1.41	3.41	4.83	2.00	2.00	2.00	2.24	3.41	4.41	2.00	2.00	3.00
15	0.00	1.41	3.00	0.00	3.00	3.00	0.00	1.41	3.00	0.00	3.00	1.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	1.00	1.41	0.00	0.00	0.00	6.39	4.00	1.41	0.00	0.00	0.00	1.00
18	4.83	3.00	2.00	2.41	2.00	8.00	5.66	3.00	2.00	3.00	2.00	2.00
19	2.83	3.83	3.41	3.41	3.00	8.06	5.41	3.41	3.41	3.00	1.00	7.33
20	5.58	6.89	4.41	5.24	4.41	9.00	3.83	6.06	3.41	4.83	5.65	8.71
21	7.24	4.65	7.06	5.24	3.00	10.48	6.24	4.65	5.65	4.41	2.41	14.46
22	5.41	3.24	6.65	6.06	3.24	9.41	5.83	5.65	7.71	3.83	2.00	13.41
23	6.24	6.16	4.83	5.65	3.24	6.39	8.24	8.41	6.66	7.89	2.00	12.00
24	8.65	6.48	6.06	5.06	4.24	7.33	7.24	6.47	5.24	6.47	5.06	16.87
25	8.75	4.65	7.48	6.47	2.83	6.80	8.40	5.41	7.89	7.89	2.41	16.34
26	11.02	7.06	7.89	7.89	3.00	7.33	8.61	7.65	7.26	7.06	5.66	12.63
27	10.78	8.12	8.30	7.40	3.83	10.54	10.48	9.30	8.43	7.81	4.65	13.05
28	10.14	9.49	9.23	8.63	3.00	9.54	7.61	11.60	9.12	8.06	4.24	12.05
29	10.49	9.05	10.68	7.24	4.83	10.48	9.58	10.70	10.89	5.41	6.66	15.45
30	6.77	8.30	12.61	7.24	6.24	12.71	8.84	11.08	8.89	7.24	4.24	17.27
31	6.77	7.48	12.61	10.24	3.41	10.63	8.84	11.26	8.89	8.72	2.00	15.45

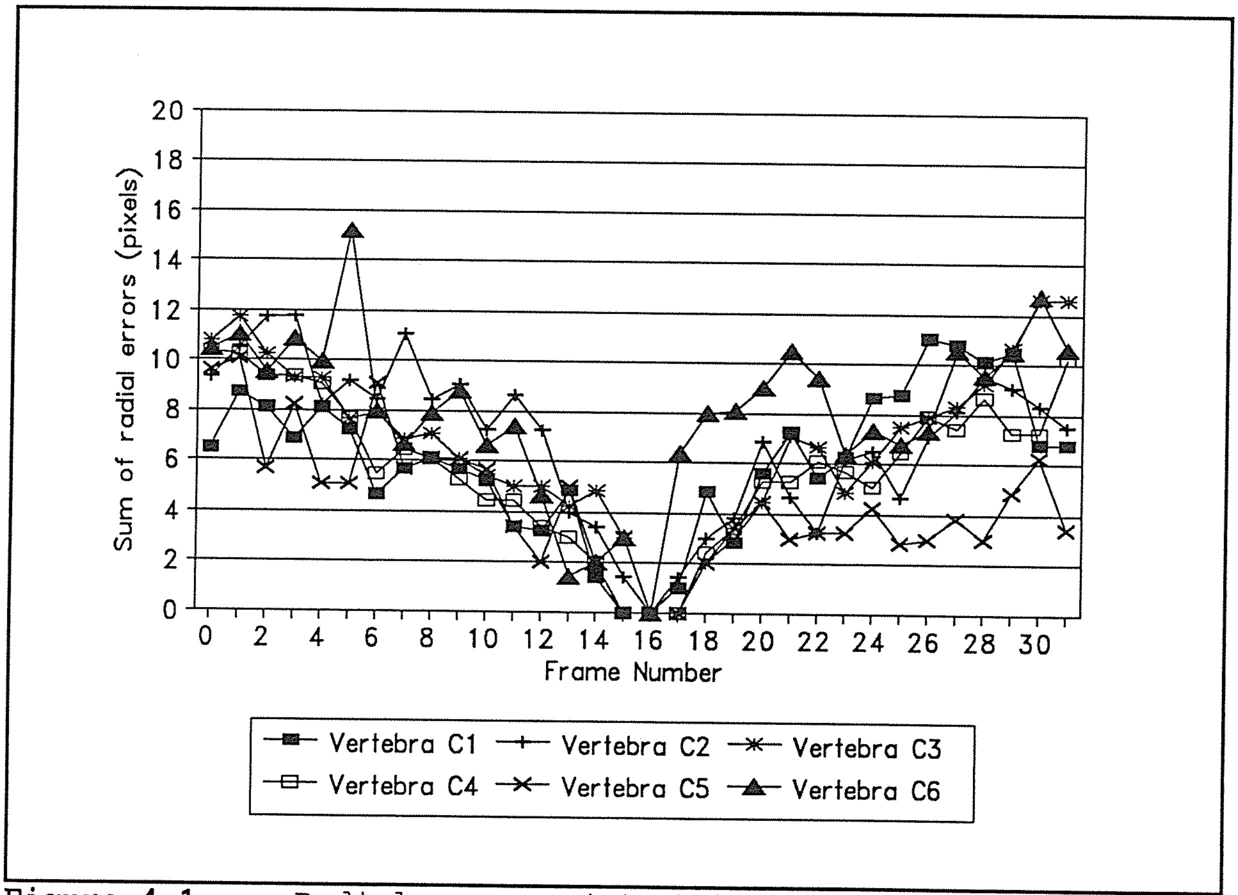


Figure 4-1 Radial error (pixels) of each vertebra when tracked through a smoothed image sequence

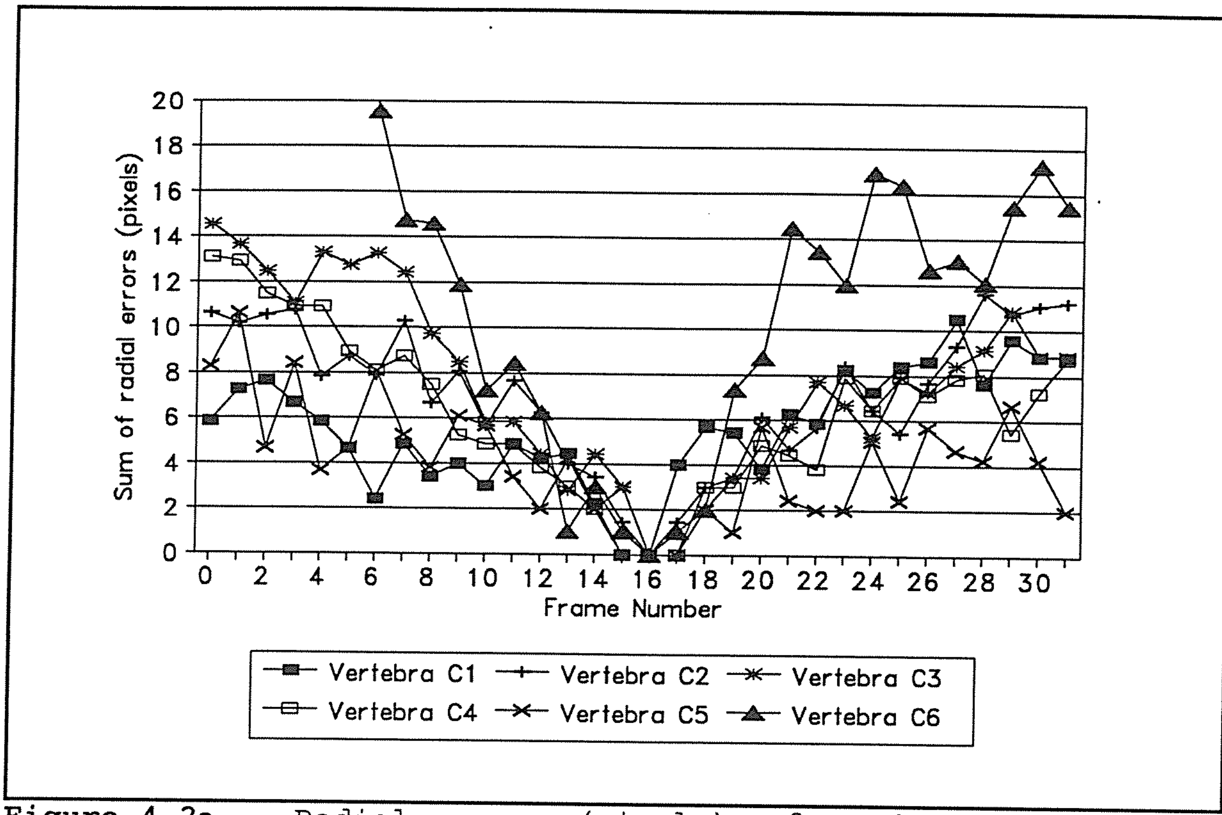


Figure 4-2a Radial error (pixels) of each vertebra when tracked through an unsmoothed image sequence

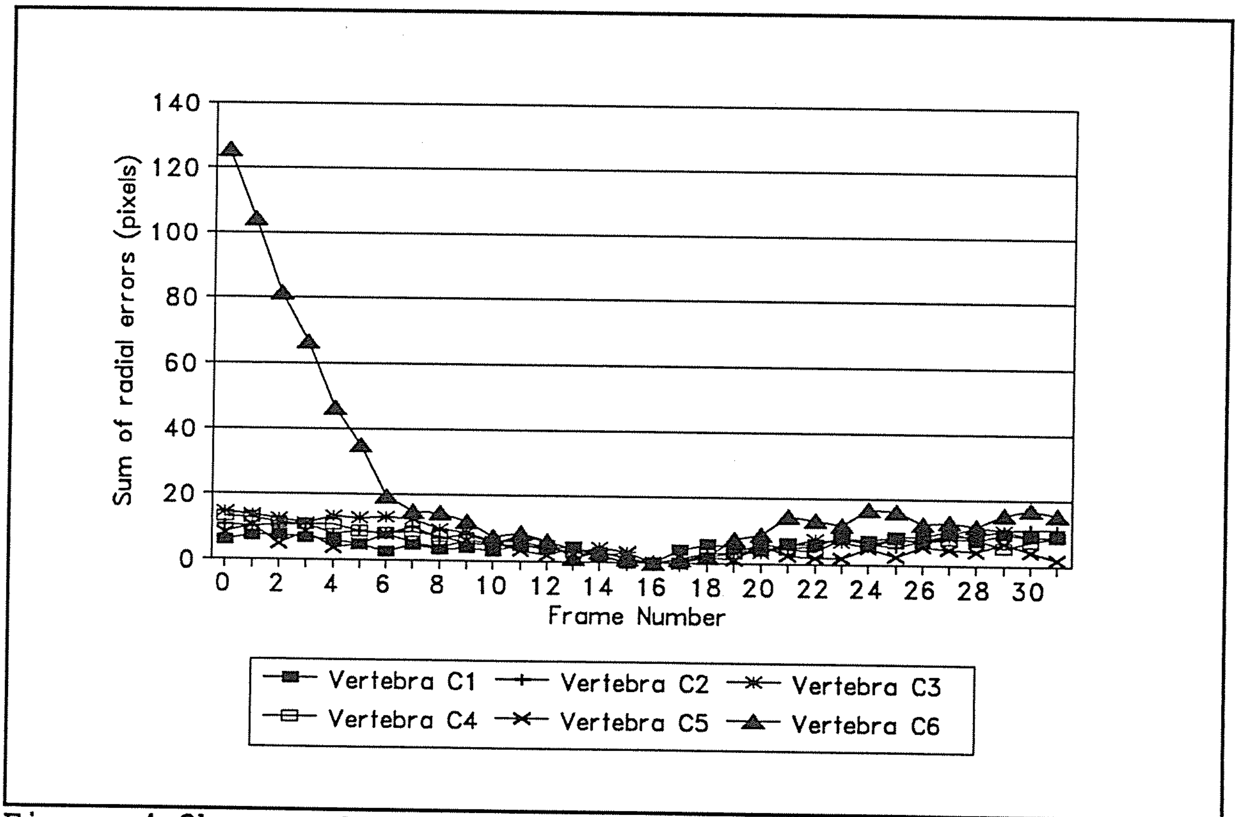


Figure 4-2b Radial error (pixels) of each vertebra when tracking through a smoothed image sequence (extended y axis)

Chapter 5

Effect on sum of absolute difference statistic of neglecting rotation between consecutive frames

5.1 Nature of rotation in an image sequence

Before investigating rotation the nature of the image sequence must be further detailed. In trying to describe the 'normal' it must be stated that because these X-rays will usually be of patients with quite serious neck injuries or abnormalities there will be variance. During the sequence of between thirty and fifty frames the head is bent towards the chest by approximately 35 degrees and then back to roughly the initial position. This means a vertebra will be rotated, around its centre, a maximum of 35 degrees in 15 to 25 frames, that is up to 2.3 degrees per frame if the movement is smooth. Vertebrae below the central pivot vertebra rotate less while the skull and vertebrae above the pivot will be rotating and translating because their centre of rotation is roughly the pivot vertebra.

5.2 Increase in the statistic of the correct location due to rotation

To investigate the effect of neglecting rotation when tracking between consecutive frames, image rotation was simulated. The computer program (listed in Appendix B) used to rotate the image applied area based weighting of the superimposed rotated pixel grid on the original intensity values to calculate the new pixel intensities.

The SAD Statistics for the point being tracked are shown for the various window sizes, and the various amounts of rotation in Table 5-1, and graphed in Figure 5-1. The statistic was calculated for the corner of a vertebra. The centre of rotation was the point being tracked because if the centre of rotation is not the point of interest rotation will result in lateral translation. The statistics increase from zero (at zero degrees) as the rotation increases which means that the pixel intensity values within the window in the original unrotated image and the window in the rotated image are increasingly different as expected.

Table 5-1 The SAD statistic for the origin of rotation

	WINDOW SIZE						
ROTATION	3	5	7	9	11	13	15
0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.20	0.175	0.599	2.153	5.378	10.163	16.872	25.738
0.40	0.350	1.197	4.283	10.705	20.227	33.569	51.213
0.60	0.525	1.794	6.389	15.980	30.192	50.093	76.430
0.80	0.699	2.389	8.472	21.204	40.059	66.444	101.388
1.00	0.873	2.983	10.532	26.376	49.828	82.622	126.089
1.20	1.047	3.576	12.568	31.497	59.499	98.628	150.532
1.40	1.220	4.168	14.581	36.567	69.072	114.462	174.719
1.60	1.394	4.758	16.571	41.586	78.549	130.125	198.650
1.80	1.566	5.347	18.538	46.555	87.928	145.617	222.325
2.00	1.739	5.935	20.481	51.480	97.217	160.945	245.753
3.00	2.598	8.856	29.922	75.441	142.313	235.146	359.196
4.00	3.451	11.748	39.063	98.469	185.336	305.554	466.768
5.00	4.299	14.613	48.010	120.754	226.487	372.777	569.435
6.00	5.141	17.452	56.665	142.700	266.360	438.159	668.687
7.00	5.979	20.266	65.033	164.415	305.483	501.667	767.218
8.00	6.813	23.058	73.104	185.574	343.390	564.472	863.892
9.00	7.644	25.828	80.885	206.189	379.928	624.837	943.908
10.00	8.472	28.578	88.377	226.270	415.099	674.843	1012.676
15.00	12.590	42.234	127.243	319.278	568.093	878.252	1349.484
20.00	16.726	56.044	155.608	391.903	694.362	1076.950	1633.156
25.00	20.956	68.892	167.389	453.937	801.690	1235.314	1853.813
30.00	25.373	78.763	187.688	506.280	917.718	1391.884	2059.032

The point being tracked, which is the origin of rotation, was a corner of a vertebra.

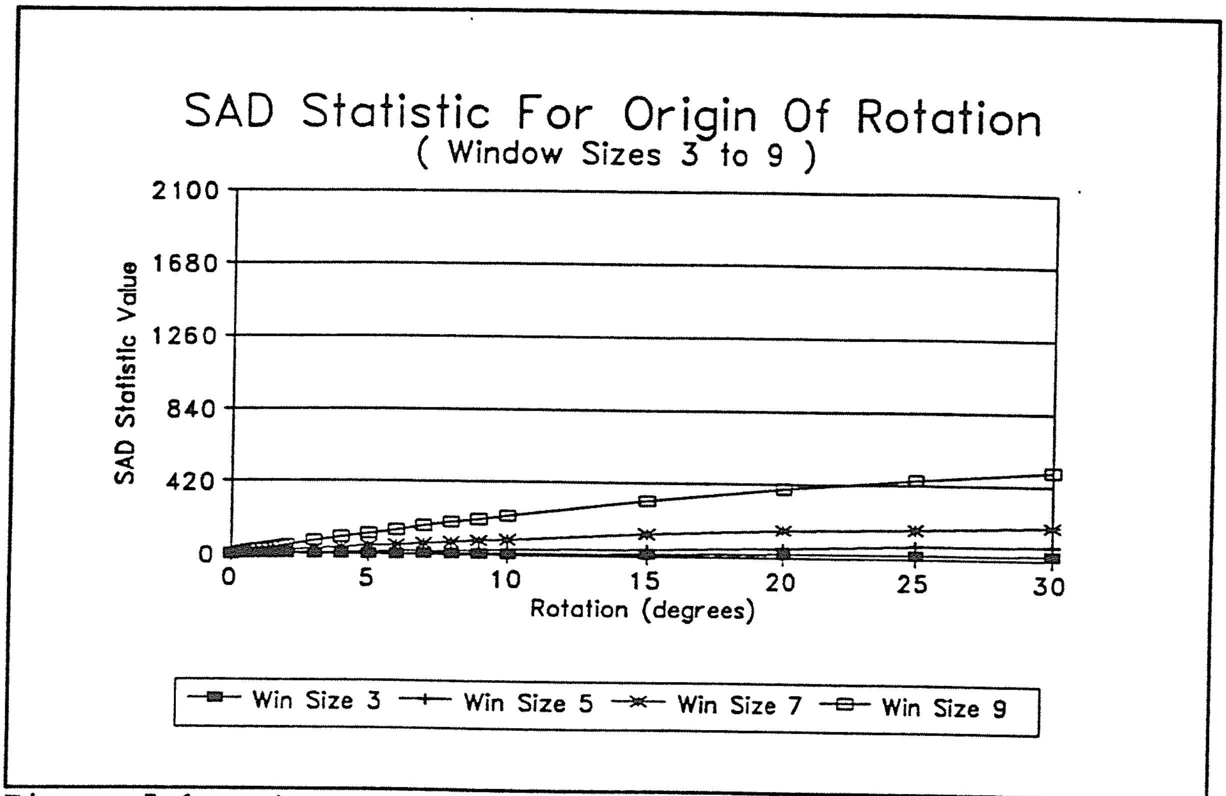


Figure 5-1a The SAD statistic for the origin of rotation

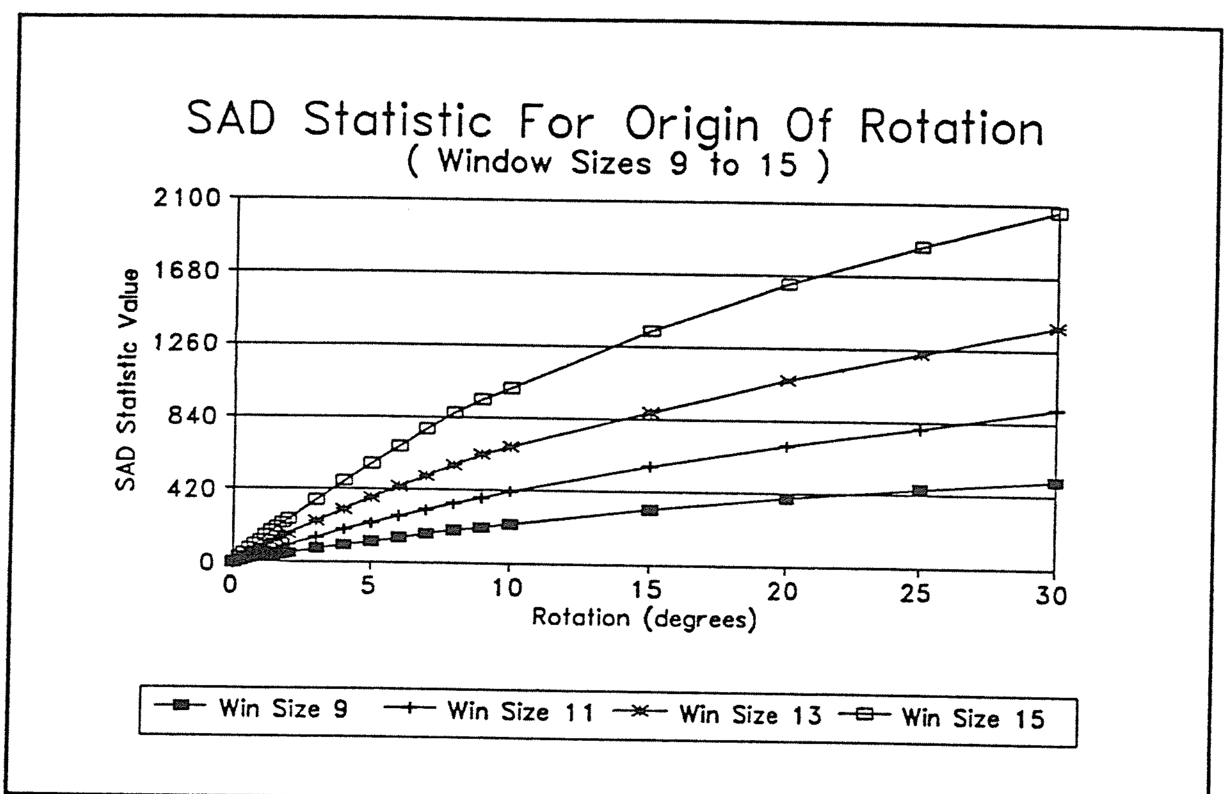


Figure 5-1b The SAD statistic for the origin of rotation

5.3 Difference between correct location statistic and next minimum after rotation

Table 5-2 gives the difference between the statistic for the specified point (28,22), which is also the origin of rotation, and the next lowest statistic found in the search area. The frames being compared were $frame_0$ and $frame_0$ rotated by amounts between 0° and 30°. Where the difference is negative there was a statistic less than the specified point statistic. This means a different (incorrect) location would have been selected by the SAD algorithm for the location of the tracked point in that rotated frame. From Table 5-2 and the graphs in Figure 5-2 it can be seen that the difference in the statistic decreases as rotation increases and levels out after 15°, this effect being more marked as the window size increases. When rotation is larger than 15° the difference fluctuates between 35 and -35 for all window sizes. In Figure 5-3 the normalised line for a window size of three does not conform to the shape of the lines for the other window sizes. This may be explained by looking at Figure 5-2a and noting that the line for the window size of three is almost horizontal and this data is then scaled enormously to compare it with the data for the larger window sizes which show a greater negative gradient in Figure 5-2b.

The difference between the minimum and next minimum statistic is a measure of the confidence with which one position can be selected as the correct location over another, often adjacent, position. To give a notion of the usual levels of confidence the difference between the two minimum statistics when the same point is tracked from $frame_0$ to $frame_1$ (the next image in the sequence) is shown in Table 5-3.

As 3 degrees is safely more than the estimated 2.3 degrees maximum rotation per frame of a vertebra then it can be concluded that inter frame rotation can be neglected when tracking with the SAD algorithm between consecutive frames for window sizes 3 to 15. It can further be concluded that when tracking from a frame to frames not next in sequence it would be beneficial and usually necessary to take rotation into account.

Table 5-2 The differences between the correct position SAD statistic and the next minimum when tracking from $Frame_0$ to $Frame_0$ rotated

Statistics For Point : (28, 22)

ROTATION	WINDOW SIZE 3	5	7	9	11	13	15
0.00	28.000	88.000	196.000	332.000	513.000	727.000	939.000
0.20	27.813	87.249	192.856	324.025	499.559	701.707	905.325
0.40	27.639	86.498	189.748	316.125	486.303	676.666	872.075
0.60	27.477	85.747	186.677	308.298	471.548	651.877	839.246
0.80	27.416	84.996	183.643	300.545	456.042	627.339	806.837
1.00	27.781	84.245	180.646	292.866	440.692	603.051	775.234
1.20	27.737	83.494	177.685	285.508	425.745	579.292	744.679
1.40	27.526	82.743	174.760	278.356	411.084	556.396	714.725
1.60	27.314	81.992	171.872	271.278	396.579	533.903	682.855
1.80	27.102	81.242	169.021	264.468	382.422	511.856	651.001
2.00	26.890	80.491	166.205	257.956	368.645	491.075	621.298
3.00	25.825	76.736	152.602	227.234	308.646	401.026	495.512
4.00	24.752	72.978	140.177	199.851	256.834	322.274	390.113
5.00	23.961	69.214	127.741	175.638	209.978	251.639	305.027
6.00	22.915	65.441	115.534	153.671	169.157	189.891	232.141
7.00	19.502	61.657	104.008	132.564	134.071	137.433	169.631
8.00	16.160	57.860	94.636	114.772	105.745	94.033	115.355
9.00	13.146	54.046	86.605	100.694	84.942	61.695	88.875
10.00	13.378	50.624	79.054	87.453	64.352	40.757	71.443
15.00	15.087	34.732	36.372	33.476	10.797	-0.405	5.706
20.00	11.061	18.457	8.242	20.098	-13.646	-31.796	-35.058
25.00	4.127	3.362	13.334	25.077	8.116	7.309	-5.294
30.00	3.202	-6.663	7.009	2.792	18.606	25.701	-17.628

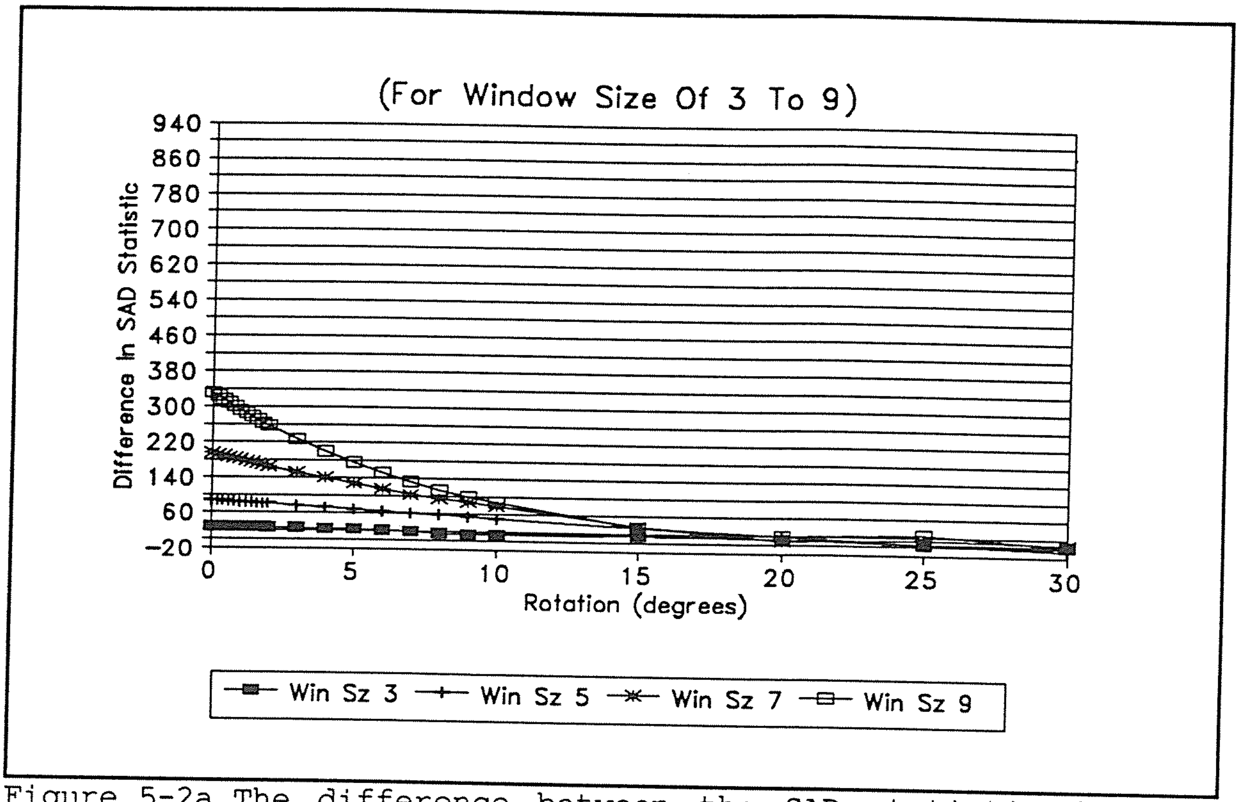


Figure 5-2a The difference between the SAD statistic for the specified point and the next minimum SAD statistic

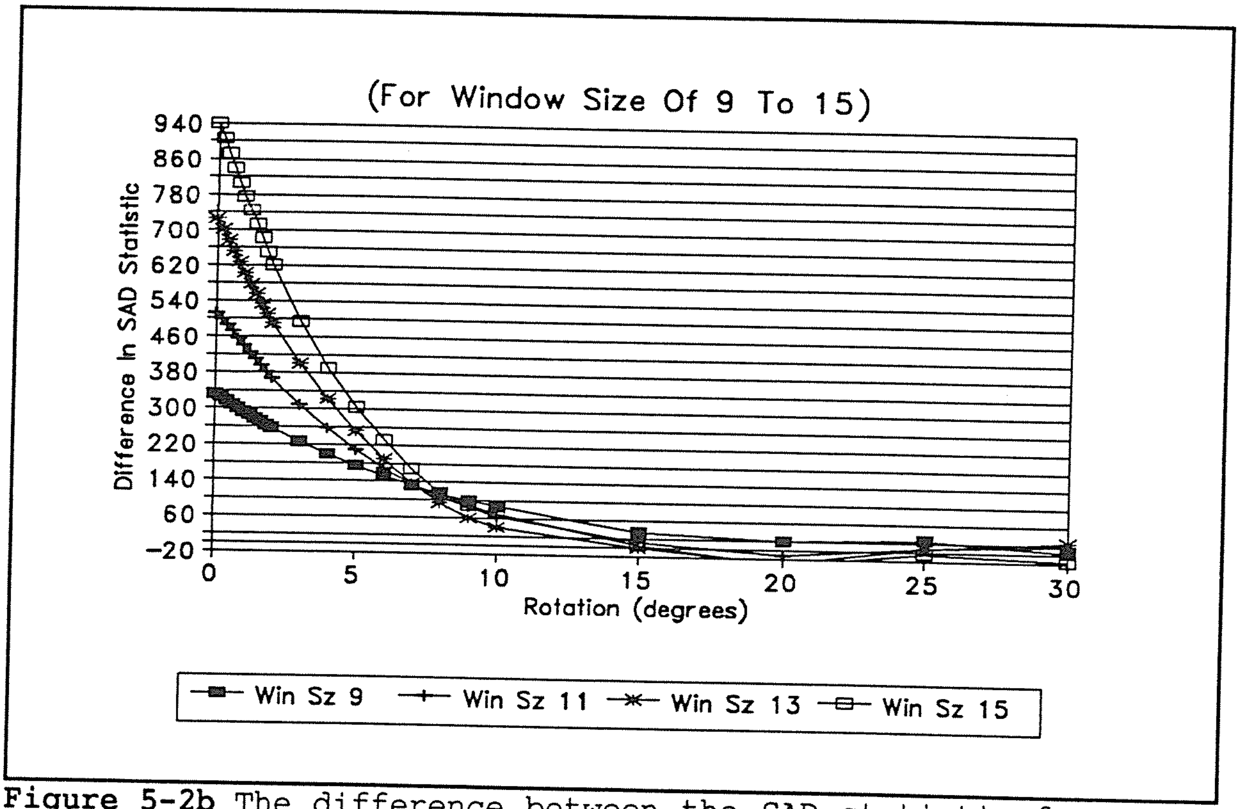


Figure 5-2b The difference between the SAD statistic for the specified point and the next minimum SAD statistic

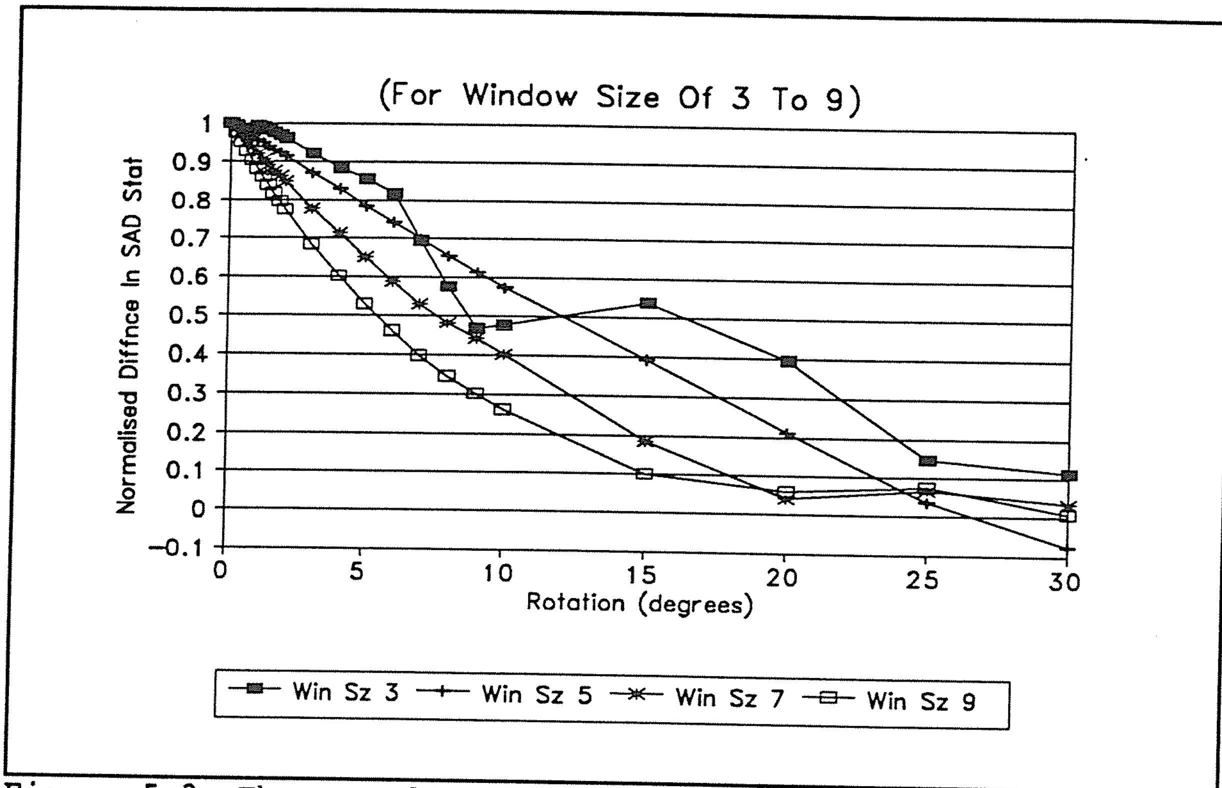


Figure 5-3a The normalised difference between the SAD statistic for the origin of rotation and the next minimum

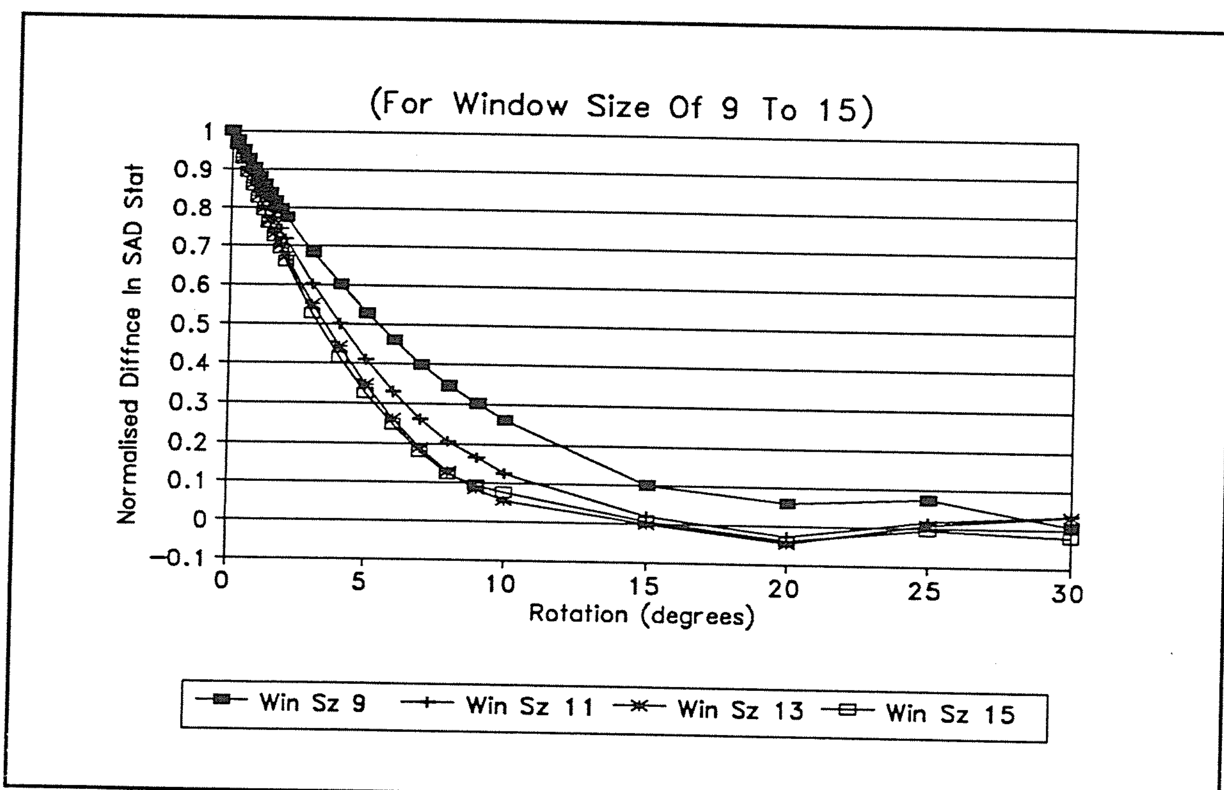


Figure 5.3b The normalised difference between the SAD statistic for the origin of rotation and the next minimum

Table 5-3 The minimum SAD statistics when tracking point (28,22) from $Frame_0$ to $Frame_1$ (the next in the video sequence)

The search width was 5 and no rotation or sub-pixel calculation was performed.

Window Size	Avg Stat	Minimum stats	Position
3	96.124	25.000	(28, 23)
		30.000	(28, 22)
		34.000	(26, 19)
		34.000	(33, 23)
		35.000	(26, 17)

Window Size	Avg Stat	Minimum stats	Position
5	259.967	59.000	(28, 23)
		78.000	(28, 22)
		109.000	(28, 24)
		111.000	(26, 19)
		113.000	(29, 23)

Window Size	Avg Stat	Minimum stats	Position
7	529.215	149.000	(28, 22)
		174.000	(28, 23)
		219.000	(29, 23)
		245.000	(28, 21)
		247.000	(29, 22)

Window size	Avg Stat	Minimum stats	Position
9	903.264	291.000	(28, 23)
		295.000	(28, 22)
		430.000	(29, 23)
		437.000	(28, 21)
		442.000	(27, 22)

Window Size	Avg Stat	Minimum stats	Position
11	1342.909	432.000	(28, 23)
		455.000	(28, 22)
		617.000	(29, 23)
		619.000	(28, 24)
		638.000	(28, 21)

Window Size	Avg Stat	Minimum stats	Position
13	1890.744	620.000	(28, 23)
		686.000	(28, 22)
		894.000	(28, 24)
		896.000	(29, 23)
		907.000	(29, 24)

Window Size	Avg Stat	Minimum stats	Position
15	2503.711	812.000	(28, 23)
		862.000	(28, 22)
		1154.000	(29, 24)
		1157.000	(28, 24)
		1196.000	(29, 23)

5.5 The effect of rotation on a checker board

In order to more fully understand the effects of rotation a repeatable pattern was selected. Checker boards were chosen because of the inherent knowledge we have of them. The size of the checkers was varied from one by one to four by four pixels. This was intended to simulate the effect of the size of the object relative to the size of the pixels when an image undergoes rotation. Vertebra are about twenty by twenty pixels. These checker boards were rotated and the statistic for the origin of rotation was calculated for different window sizes. This data can be seen in Table 5-4 and graphed in Figures 5-4 to 5-7. The normalised data is graphed in Figures 5-8 to 5-11. These results show that the larger the window size the greater the increase in statistic value with rotation. The statistic values plateau at around 10° rotation for the one by one checkers, 20° for the two by two checkers, 30° for the three by three checkers, and above 30° for the four by four checkers. When examining the pictures of the rotated checker boards (found in the appendix B) it can be seen that it is around these levels of rotation that the different size checker boards lose their recognisability. It can be noted that the larger the object compared to the pixel size the greater the rotation before the object loses its recognisability. This can be explained by considering the Nyquist law of sampling [Bose, 1985]. This law states that the sampling frequency must be twice the highest frequency of the signal being sampled. In this case the pixels represent the points of sample and the overlaid checker board pattern is the signal being sampled. Consequently when the checker board of one by one is resampled after rotation two dimensional spatial aliasing is more significant than when the four by four checker board is resampled. The slope of the SAD statistics between 0.2° and 2.0° rotation for the various checker sizes is plotted against the window size and window size squared in Figure 5-12 and 5-13. From the graph it can be seen that the smaller the checker size and the larger the window size the steeper the slope or gain of the SAD statistic . When graphed against the window size squared the plot of slope versus window size approximates a straight line.

Table 5-4a The statistics for the origin of the rotated checker patterns

CHECKER SIZE : 1X1

WIND SIZE / ROTATION	3	5	7	9	11	13	15
0.20	11.978	55.370	151.685	321.838	586.596	966.575	1482.244
0.40	23.878	110.217	301.433	638.463	1161.648	1910.730	2924.858
0.60	35.702	164.542	449.248	949.884	1725.173	2832.494	4327.885
0.80	47.449	218.347	595.134	1256.110	2277.187	3731.892	5691.363
1.00	59.120	271.633	739.097	1557.151	2817.707	4608.949	7015.330
1.20	70.715	324.403	881.139	1853.014	3346.747	5463.690	8299.822
1.40	82.234	376.657	1021.265	2143.708	3864.322	6296.137	9544.871
1.60	93.679	428.397	1159.480	2429.240	4370.445	7106.310	10750.507
1.80	105.049	479.625	1295.786	2709.619	4865.128	7894.230	11916.758
2.00	116.344	530.342	1430.188	2984.852	5348.385	8659.915	13043.649
3.00	171.715	776.314	2073.751	4284.050	7593.640	12155.355	18088.434
4.00	225.276	1009.714	2670.191	5455.498	9554.651	15096.888	22151.669
5.00	277.062	1230.692	3219.842	6499.755	11232.163	17485.289	25233.846
6.00	327.111	1439.376	3722.969	7417.189	12626.493	19320.492	27333.960
7.00	375.455	1635.878	4179.766	8207.976	13737.524	20601.585	28449.496
8.00	422.125	1820.293	4590.359	8872.098	14564.706	21326.799	28862.420
9.00	467.149	1992.696	4954.803	9409.344	15107.050	21647.221	28306.097
10.00	510.554	2153.145	5273.081	9819.306	15371.248	21434.457	27971.979
15.00	704.043	2776.859	6167.699	10299.393	15051.547	21330.650	29002.423
20.00	859.705	3100.015	6248.372	10179.285	15692.546	21825.860	28822.935
25.00	978.373	3295.000	6093.204	10473.239	15572.143	21700.298	28876.397
30.00	1059.449	3198.172	6412.931	10509.518	15420.771	21366.593	28534.509

CHECKER SIZE : 2X2

WIND SIZE / ROTATION	3	5	7	9	11	13	15
0.20	5.995	27.741	76.066	161.540	294.696	486.028	745.994
0.40	11.964	55.332	151.611	321.717	586.417	966.327	1481.917
0.60	17.907	82.774	226.637	480.535	875.171	1440.912	2207.792
0.80	23.824	110.068	301.147	637.999	1160.967	1909.797	2923.639
1.00	29.715	137.215	375.142	794.113	1443.813	2372.995	3629.479
1.20	35.581	164.215	448.625	948.882	1723.718	2830.520	4325.333
1.40	41.422	191.070	521.597	1102.311	2000.690	3282.384	5011.219
1.60	47.237	217.779	594.062	1254.404	2274.736	3728.602	5687.158
1.80	53.028	244.345	666.021	1405.165	2545.864	4169.184	6353.168
2.00	58.794	270.767	737.476	1554.599	2814.083	4604.143	7009.265
3.00	87.258	400.755	1087.268	2282.004	4111.774	6694.986	10141.655
4.00	115.129	527.274	1424.762	2976.822	5337.741	8646.859	13028.572
5.00	142.427	650.412	1750.185	3639.501	6492.734	10460.884	15671.571
6.00	169.174	770.254	2063.748	4270.439	7577.396	12137.981	18071.841
7.00	195.387	886.879	2365.644	4869.990	8592.267	13678.862	20230.205
8.00	221.087	1000.364	2656.048	5438.462	9537.781	15084.038	22147.078
9.00	246.289	1110.778	2935.121	5976.115	10414.271	16353.803	23827.322
10.00	271.012	1218.189	3203.006	6483.167	11221.967	17489.143	25275.060
15.00	388.003	1712.265	4378.547	8564.142	14264.244	21207.813	29025.520
20.00	495.314	2139.133	5286.175	9938.446	15685.826	21923.804	28427.347
25.00	594.632	2502.410	5965.283	10645.299	15707.028	21364.671	28571.855
30.00	687.519	2800.752	6403.018	10718.294	15360.792	21669.640	29160.328

Table 5-4b The statistic for the origin of the rotated checker patterns

CHECKER SIZE : 3X3

WIND SIZE / ROTATION	3	5	7	9	11	13	15
0.20	4.004	16.661	54.361	107.887	187.905	336.958	498..567
0.40	8.005	33.265	108.405	215.253	374.658	670.905	992.895
0.60	12.003	49.813	162.134	322.100	560.265	1001.851	1483.000
0.80	15.998	66.304	215.549	428.431	744.746	1329.806	1968.895
1.00	19.991	82.741	268.653	534.250	928.062	1654.781	2450.596
1.20	23.981	99.122	321.447	639.559	1110.263	1976.783	2928.115
1.40	27.969	115.449	373.933	744.363	1291.340	2295.825	3401.467
1.60	31.954	131.722	426.112	848.663	1471.297	2611.913	3870.666
1.80	35.937	147.941	477.985	952.463	1650.141	2925.059	4335.724
2.00	39.918	164.108	529.555	1055.766	1827.875	3235.270	4796.654
3.00	59.793	244.163	782.906	1564.926	2700.089	4742.615	7039.834
4.00	79.629	322.965	1028.893	2062.089	3545.309	6177.855	9181.641
5.00	99.440	400.572	1267.696	2547.596	4364.104	7541.922	11223.430
6.00	119.239	477.041	1499.483	3021.773	5157.004	8835.640	13166.410
7.00	139.039	552.427	1724.411	3484.929	5924.502	10059.736	15011.644
8.00	158.852	626.782	1942.629	3937.359	6667.054	11214.828	16734.814
9.00	178.693	700.160	2154.276	4379.342	7385.083	12301.428	18315.634
10.00	198.574	772.611	2359.484	4811.147	8078.977	13320.841	19793.399
15.00	299.068	1122.711	3292.903	6726.335	11231.142	17502.556	25618.862
20.00	402.733	1456.952	4079.397	8252.042	13657.956	20356.532	28914.238
25.00	511.649	1780.597	4767.631	9464.119	15299.635	22017.155	29571.751
30.00	628.346	2093.508	5370.374	10336.451	16288.540	22645.333	28933.753

CHECKER SIZE : 4X4

WIND SIZE / ROTATION	3	5	7	9	11	13	15
0.20	1.996	16.633	43.517	72.000	134.301	262.178	398.964
0.40	3.978	33.153	86.836	143.804	267.946	522.212	794.759
0.60	5.946	49.561	129.958	215.413	400.938	780.112	1187.397
0.80	7.900	65.858	172.886	286.829	533.282	1035.884	1576.889
1.00	9.841	82.043	215.621	358.054	664.982	1289.537	1963.248
1.20	11.768	98.117	258.163	429.090	796.041	1541.078	2346.484
1.40	13.682	114.082	300.514	499.940	926.464	1790.516	2726.610
1.60	15.582	129.937	342.675	570.605	1056.253	2037.857	3103.636
1.80	17.469	145.683	384.648	641.086	1185.414	2283.110	3477.574
2.00	19.342	161.321	426.433	711.387	1313.949	2526.280	3848.434
3.00	28.515	237.900	632.594	1060.242	1947.372	3711.156	5656.930
4.00	37.367	311.843	834.253	1404.856	2565.697	4845.013	7389.996
5.00	45.909	383.209	1031.554	1745.462	3169.350	5928.624	9048.778
6.00	54.148	452.053	1224.634	2082.284	3758.737	6962.689	10634.314
7.00	62.094	518.425	1413.624	2415.546	4334.243	7947.836	12147.537
8.00	69.752	582.375	1598.651	2745.464	4896.233	8884.621	13564.037
9.00	77.132	643.946	1779.835	3072.253	5445.056	9773.532	14863.493
10.00	84.240	703.180	1957.294	3396.122	5981.043	10616.264	16096.214
15.00	115.921	965.530	2791.807	4979.047	8535.751	14237.838	21336.816
20.00	141.677	1173.919	3420.782	6548.674	10887.670	16926.231	24839.302
25.00	162.067	1330.168	3903.480	7959.352	12760.890	18988.722	27544.058
30.00	177.519	1437.043	4298.384	9012.191	14161.552	20530.990	28921.915

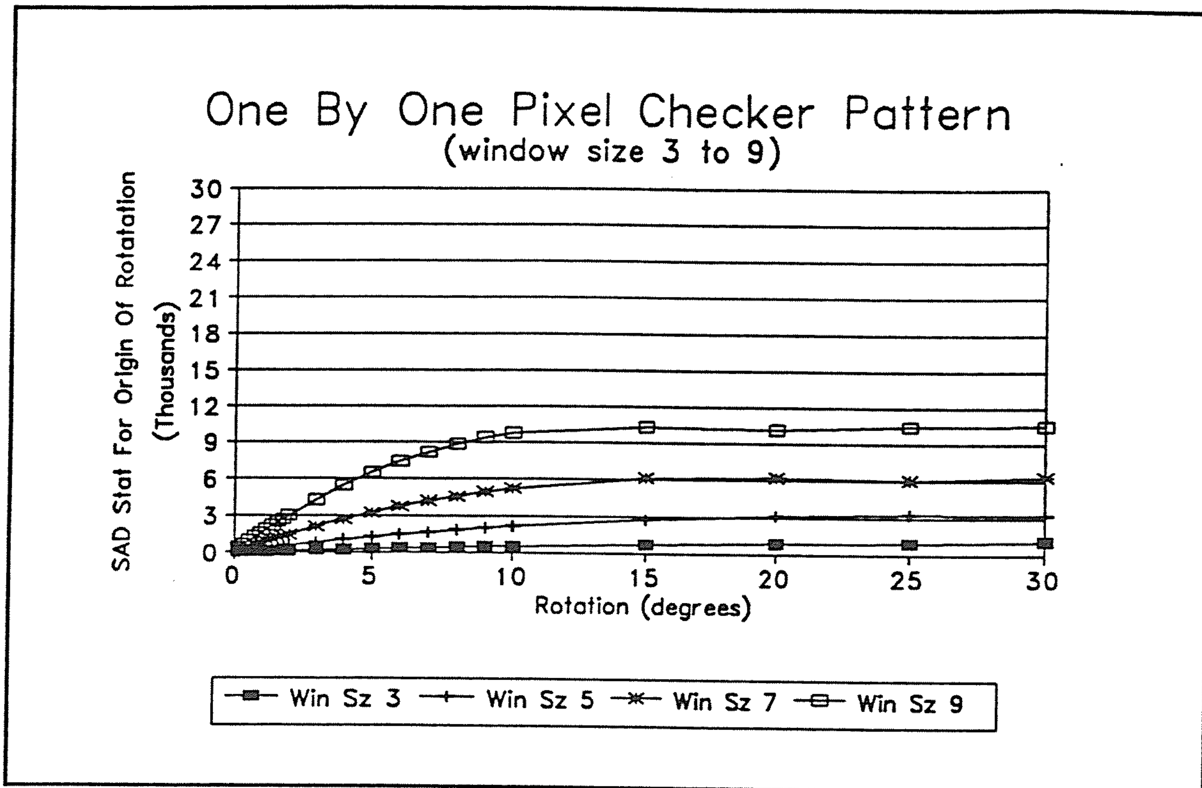


Figure 5-4a The SAD statistics for the origin of a rotated one by one pixel checker pattern

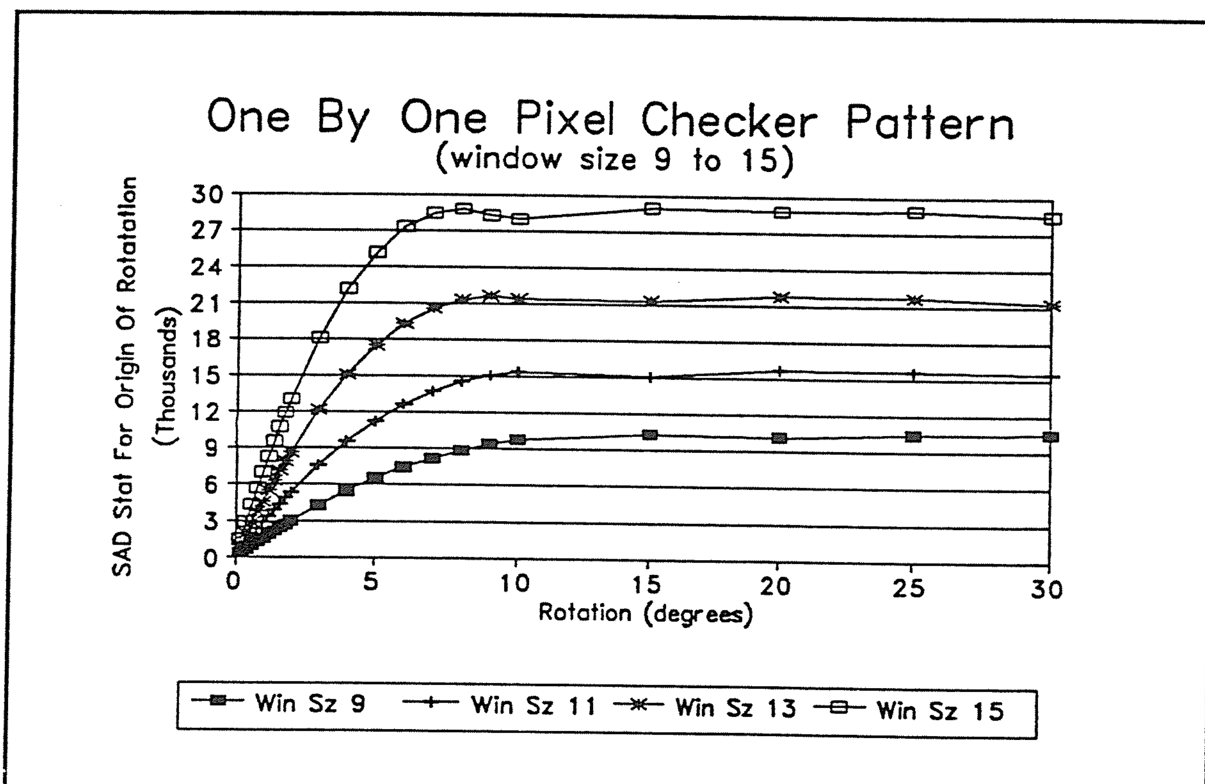


Figure 5-4b The SAD statistics for the origin of a rotated one by one pixel checker pattern

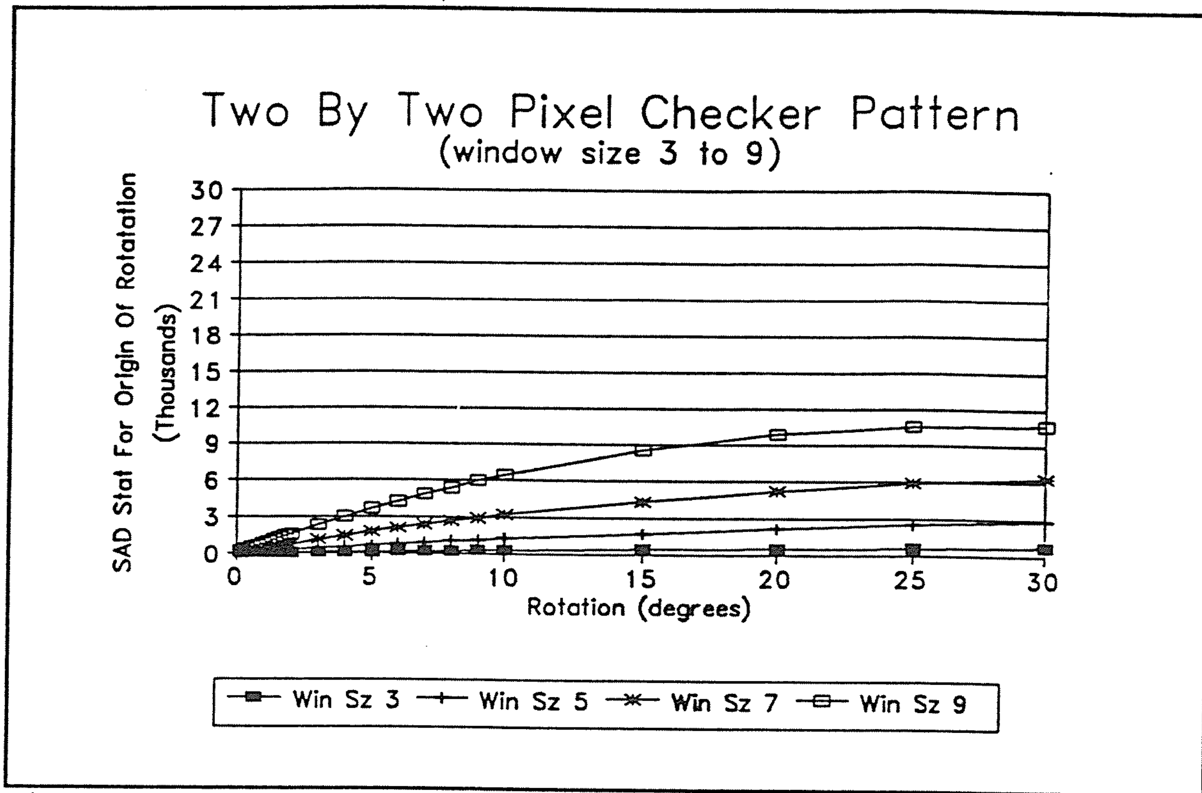


Figure 5-5a The SAD statistics for the origin of a rotated two by two pixel checker pattern

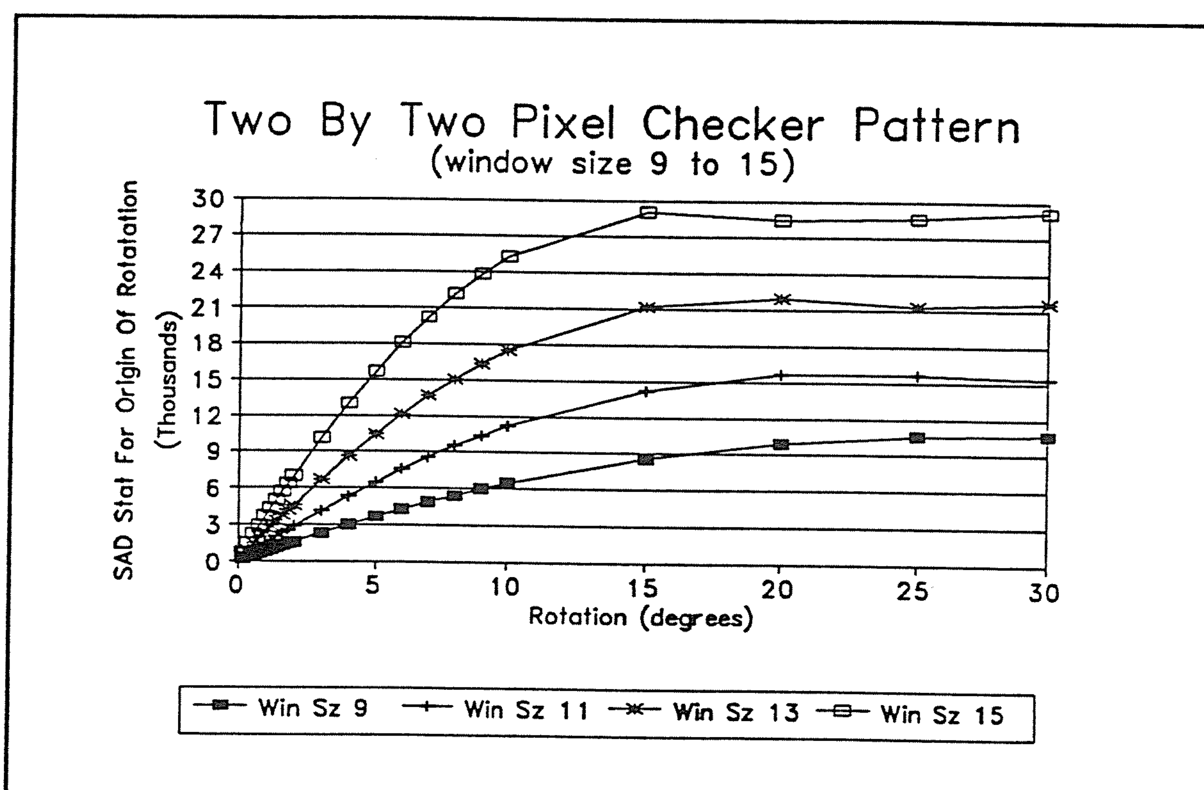


Figure 5-5b The SAD statistics for the origin of a rotated two by two pixel checker pattern

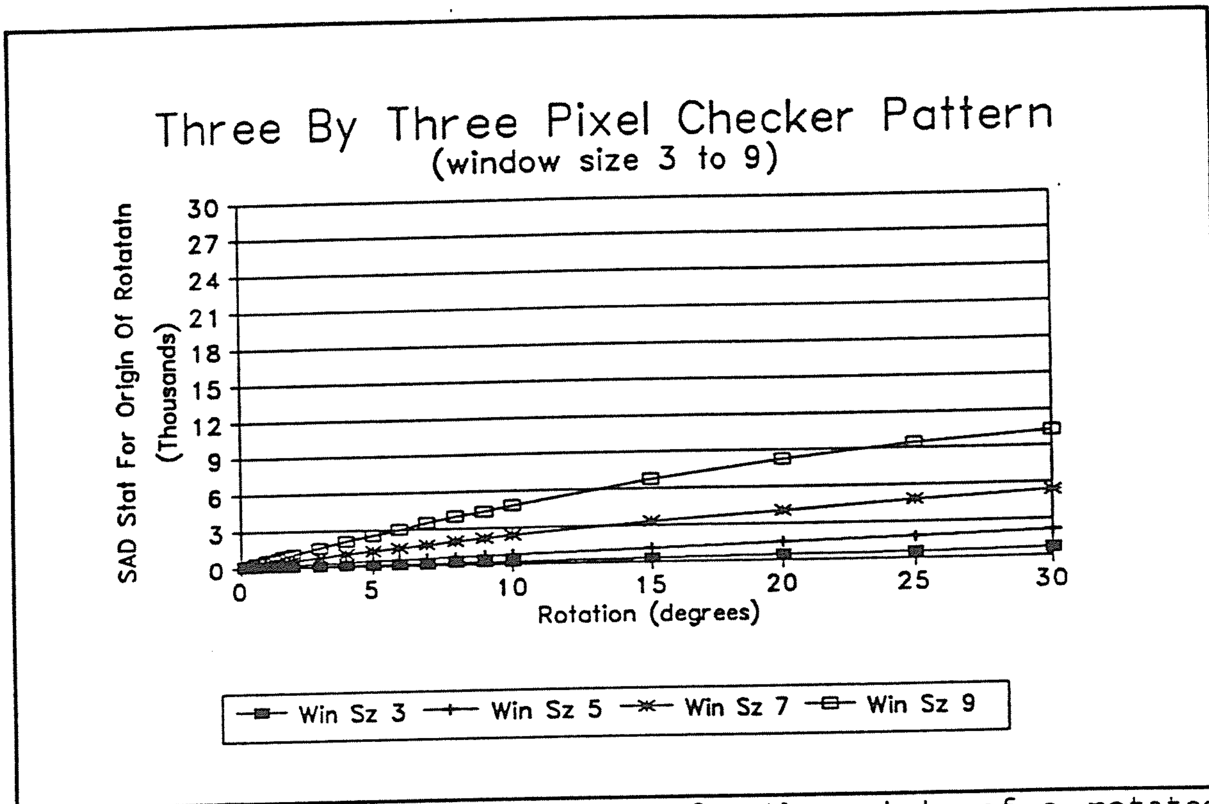


Figure 5-6a The SAD statistics for the origin of a rotated three by three pixel checker pattern

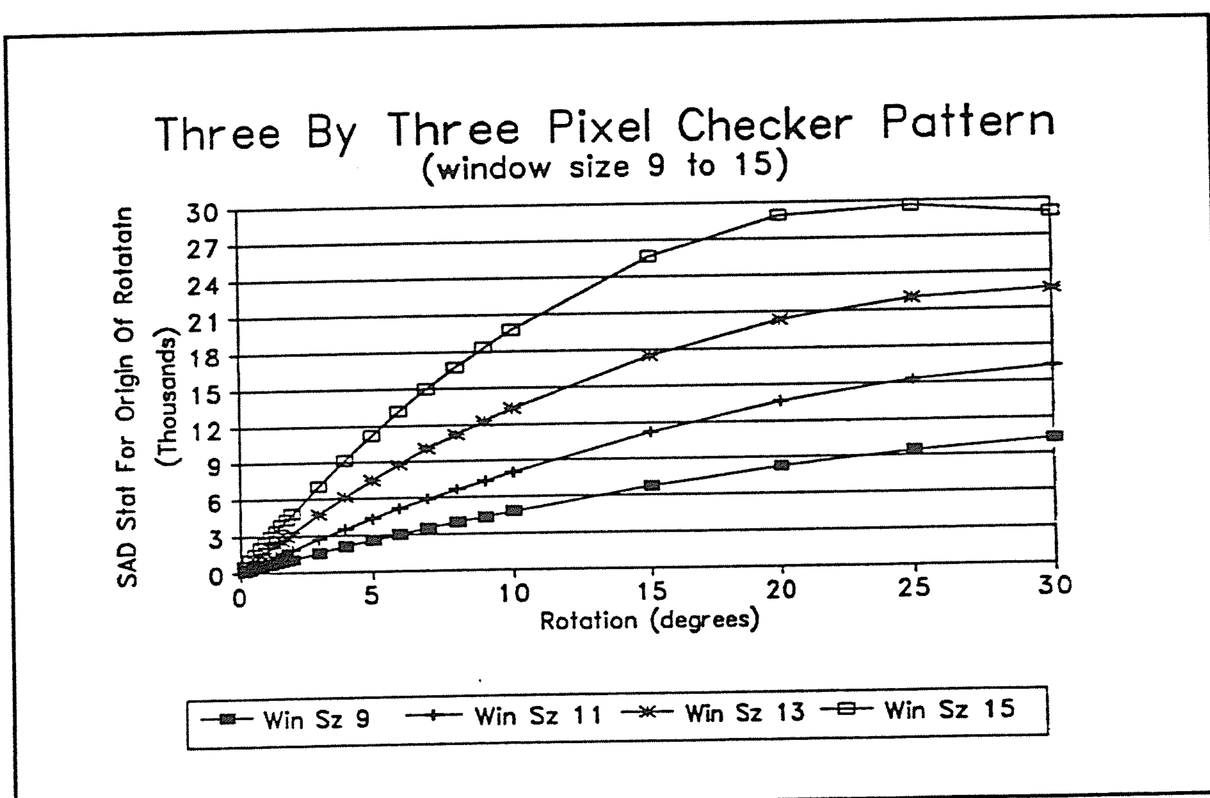


Figure 5-6b The SAD statistic for the origin of a rotated three by three pixel checker pattern

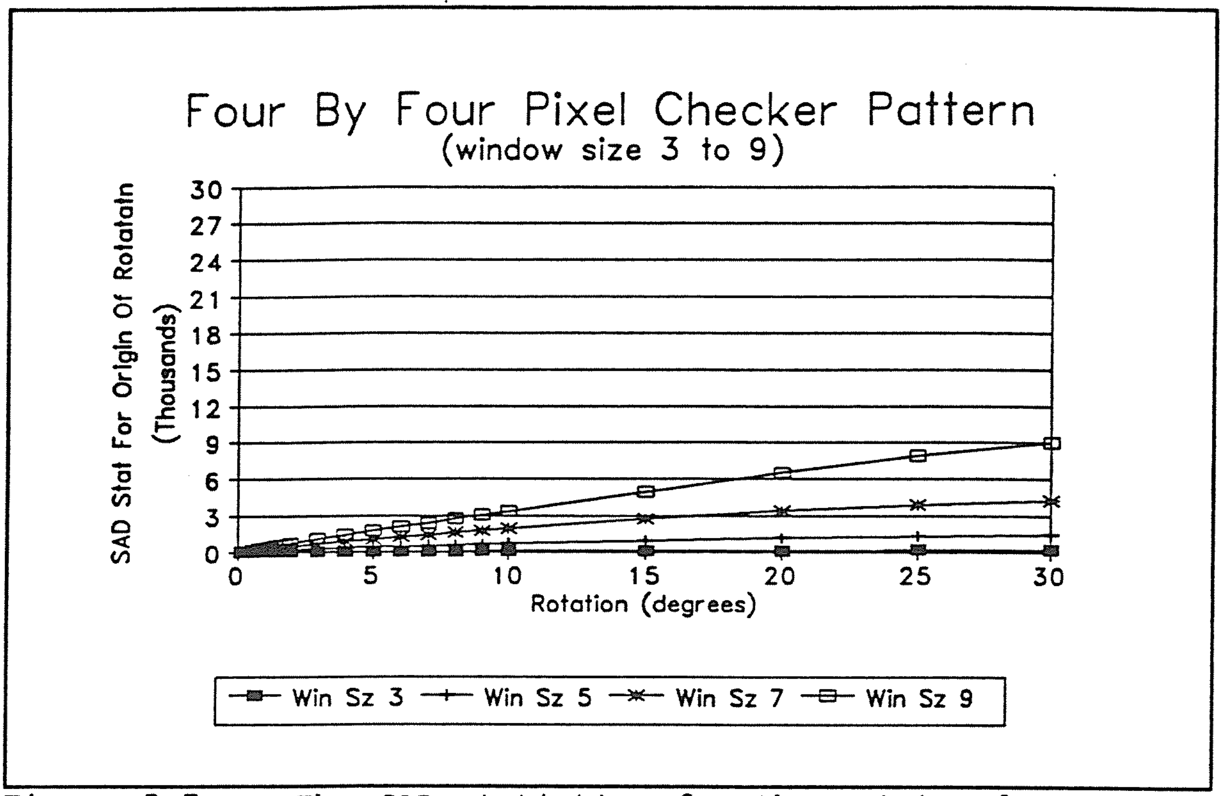


Figure 5-7a The SAD statistics for the origin of a rotated four by four pixel checker pattern

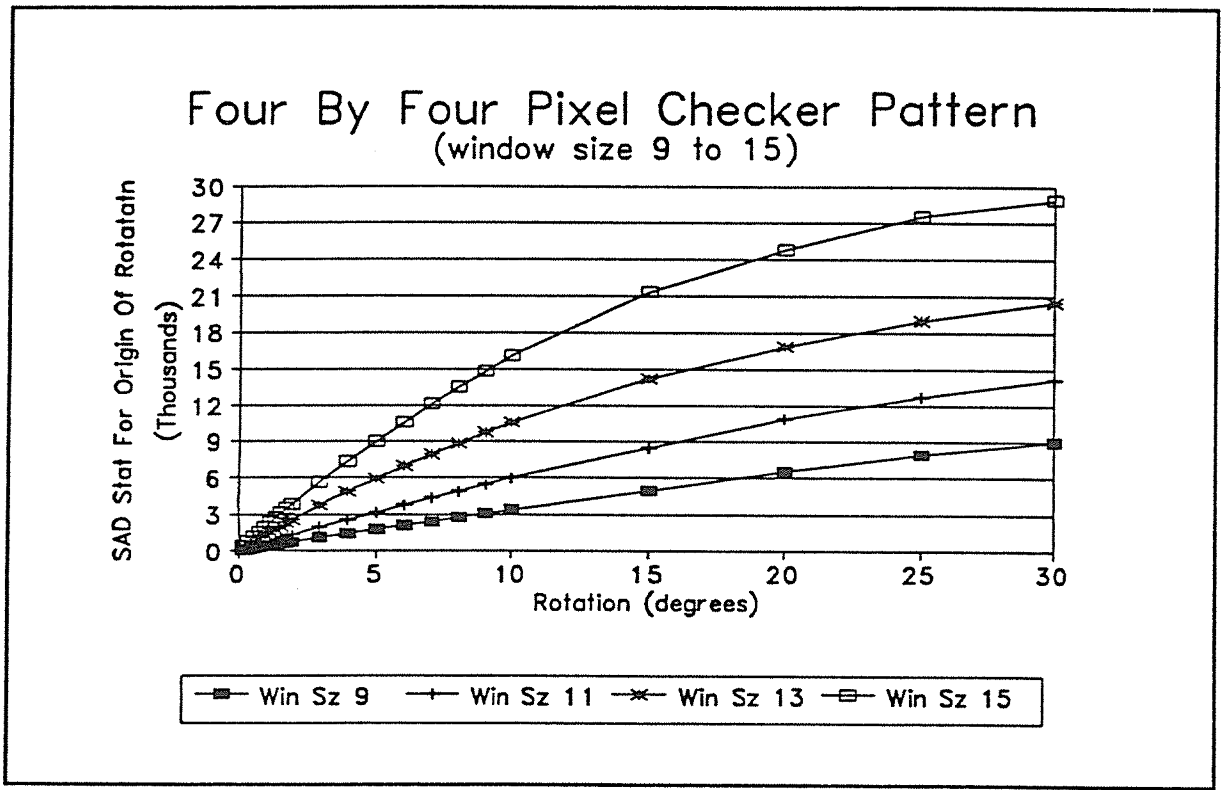


Figure 5-7b The SAD statistics for the origin of a rotated four by four pixel checker pattern

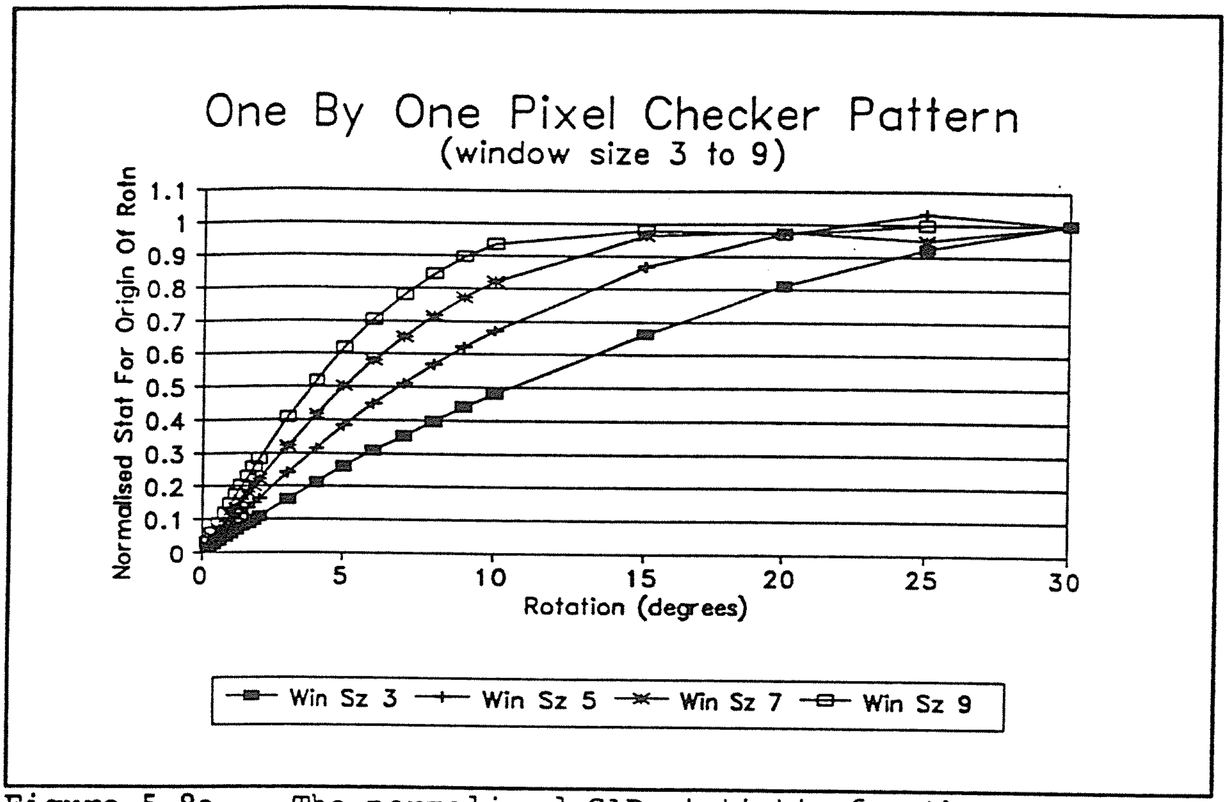


Figure 5-8a The normalised SAD statistic for the origin of a rotated one by one pixel checker pattern

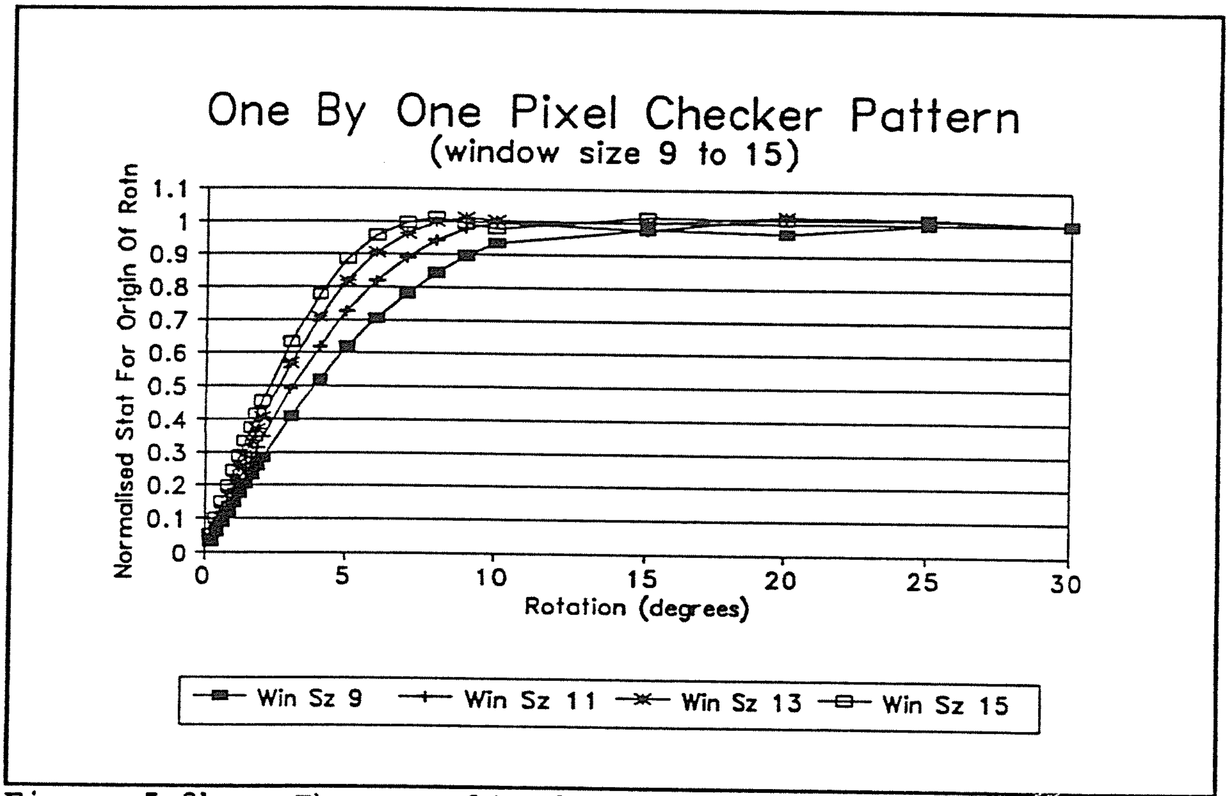


Figure 5-8b The normalised SAD statistic for the origin of a rotated one by one pixel checker pattern

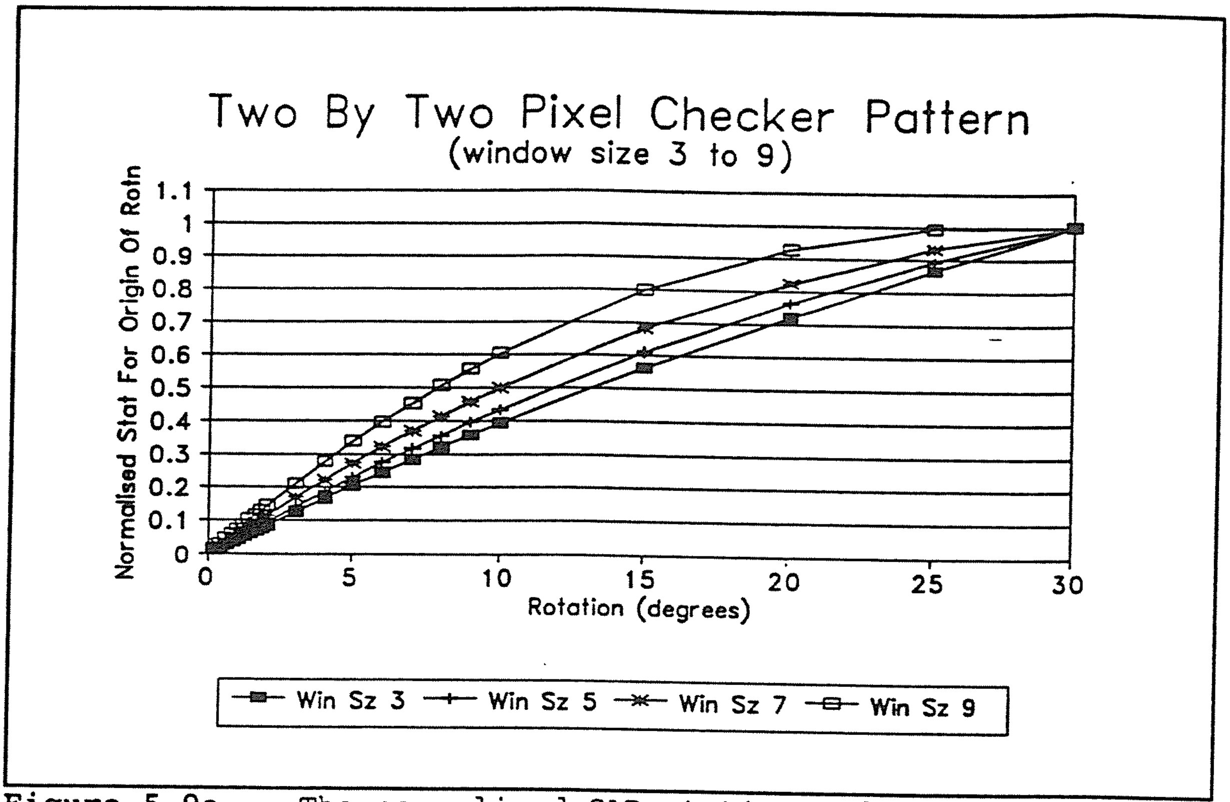


Figure 5-9a The normalised SAD statistic for the origin of a rotated two by two pixel checker pattern

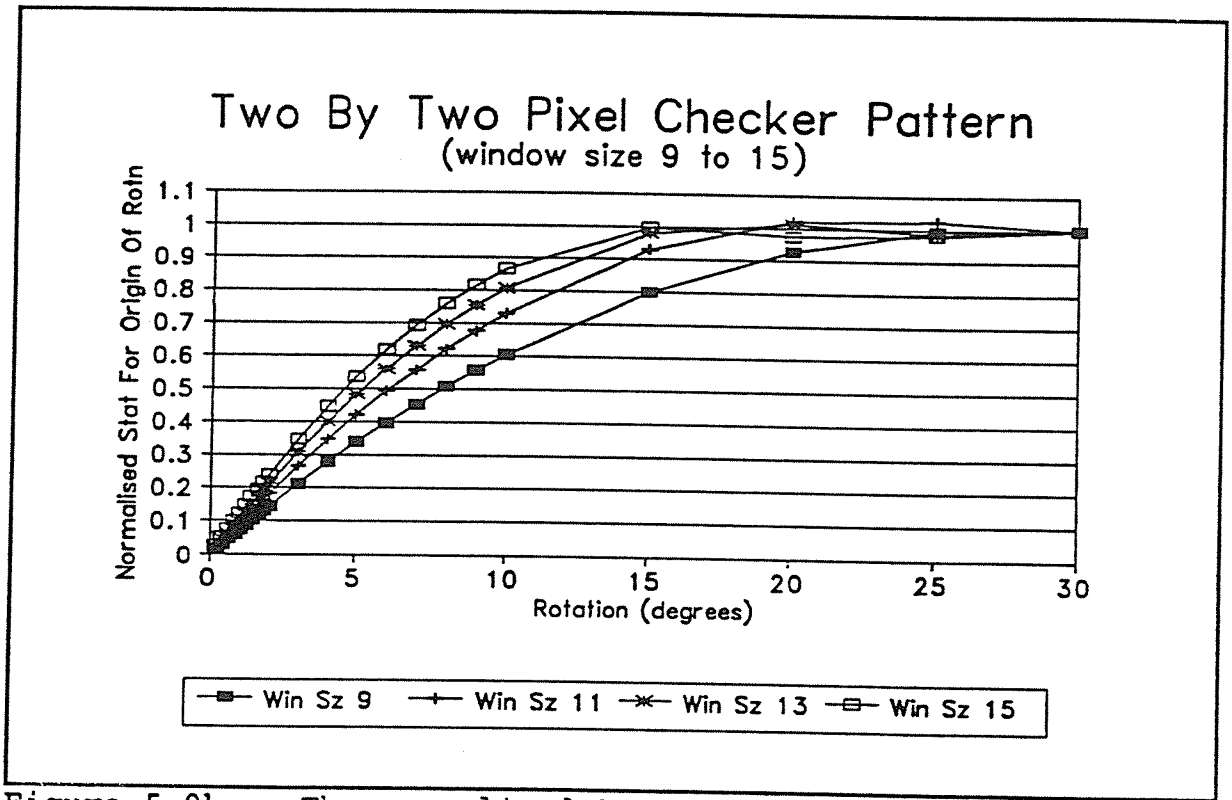


Figure 5-9b The normalised SAD statistic for the origin of a rotated two by two pixel checker pattern

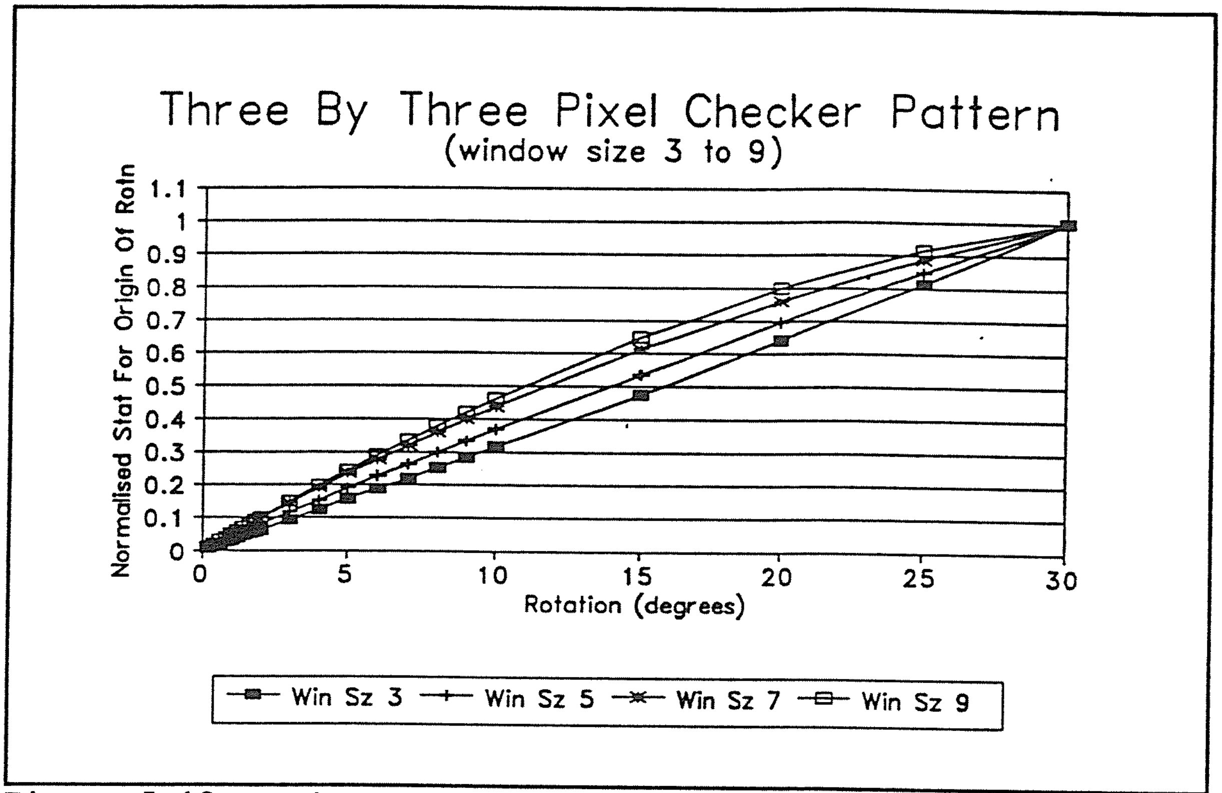


Figure 5-10a The normalised SAD statistic for the origin of a rotated three by three pixel checker pattern

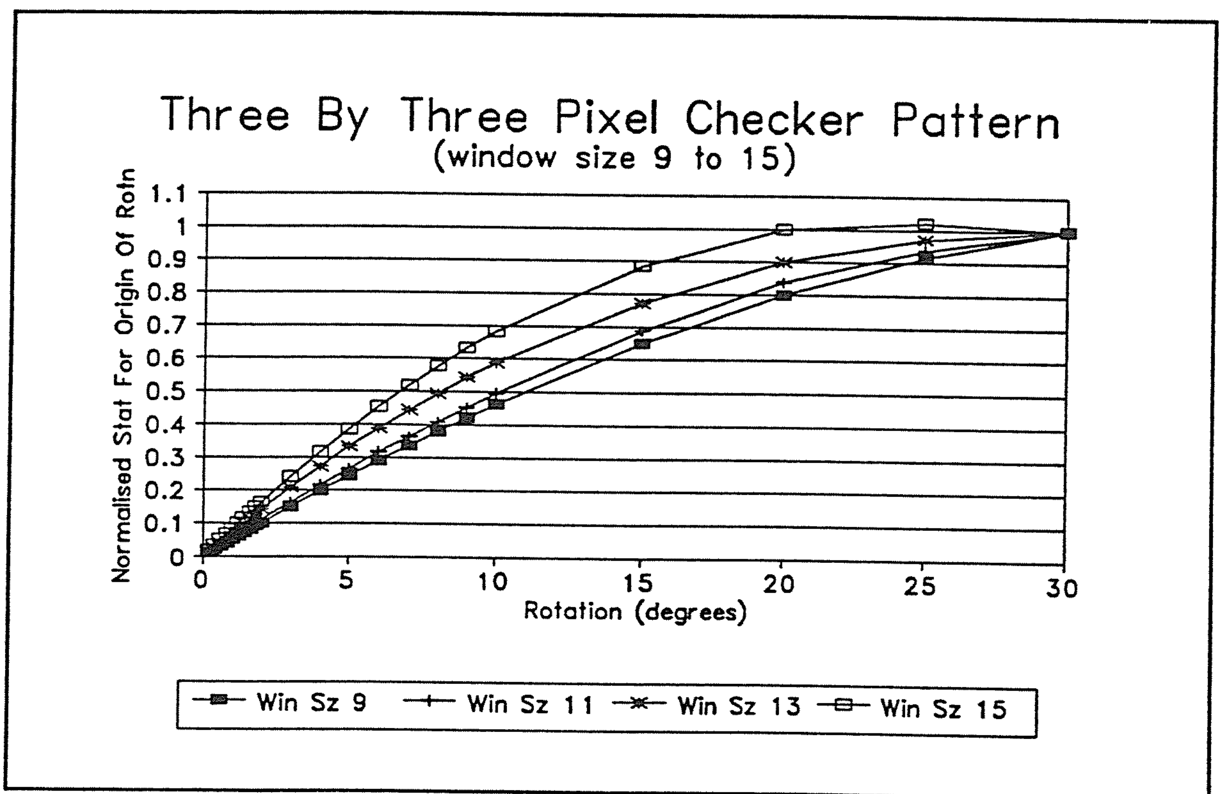


Figure 5-10b The normalised statistic for the origin of a rotated three by three pixel checker pattern

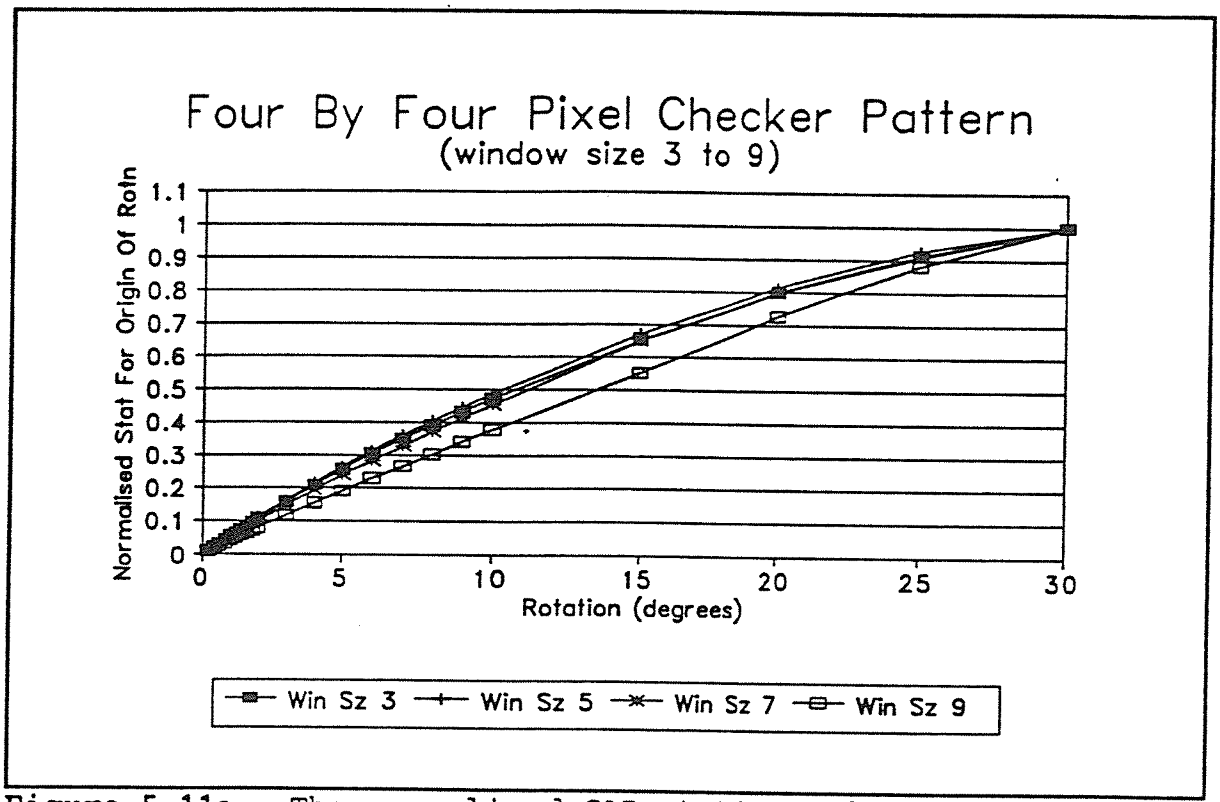


Figure 5-11a The normalised SAD statistic for the origin of a rotated four by four pixel checker pattern

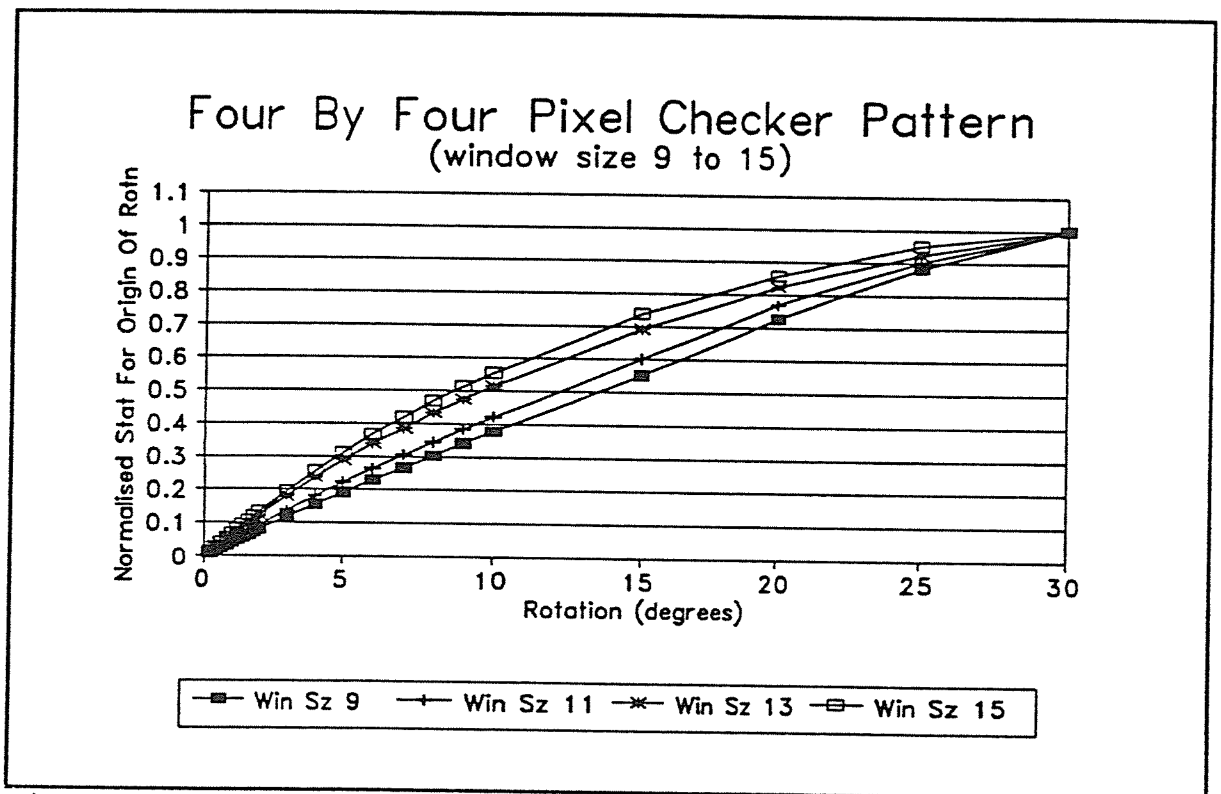


Figure 5-11b The normalised SAD statistic for the origin of a rotated four by four pixel checker pattern

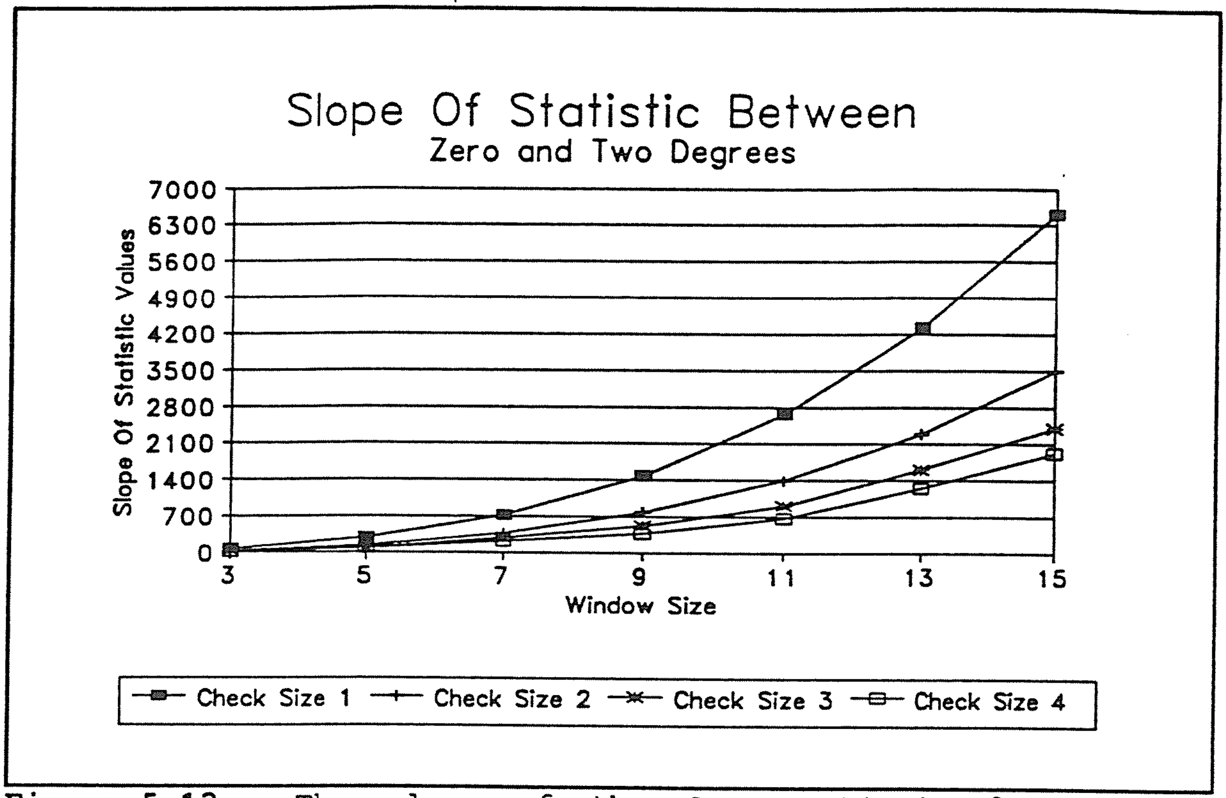


Figure 5-12 The slope of the SAD statistic for checker patterns rotated from zero to two degrees

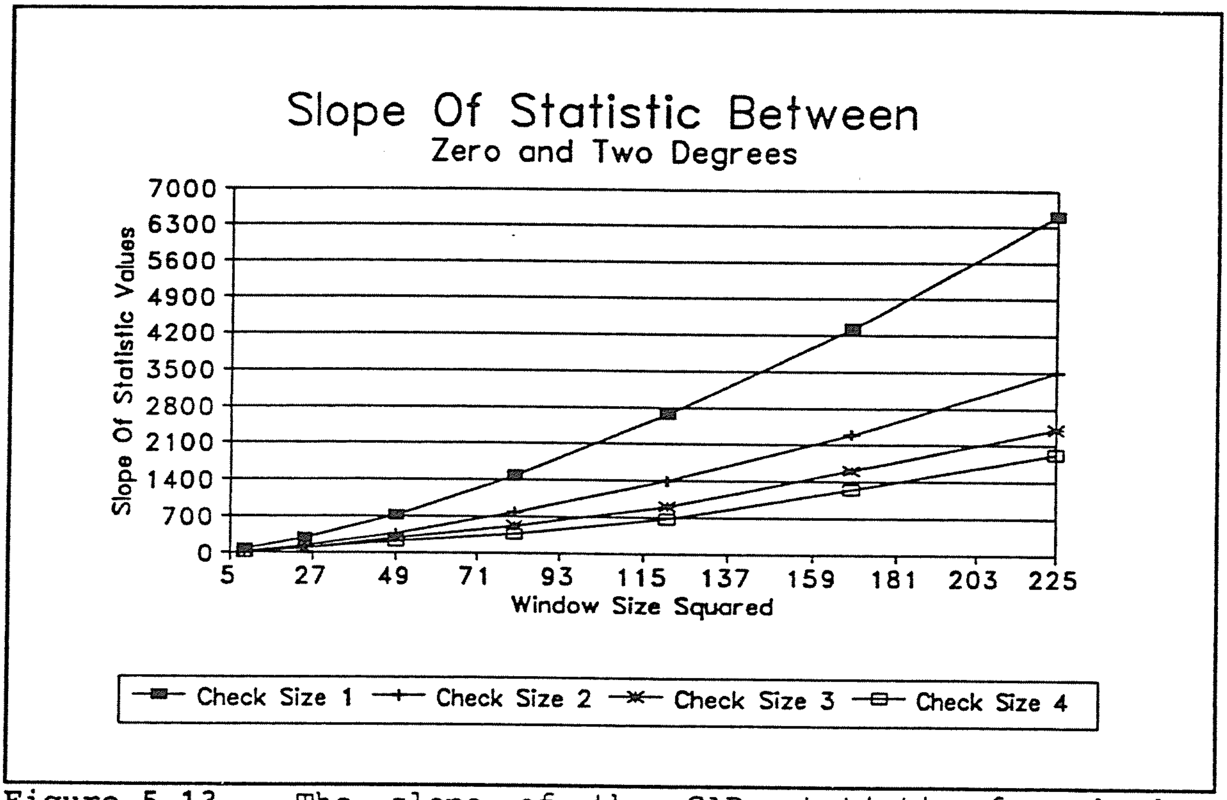


Figure 5-13 The slope of the SAD statistic for checker patterns rotated from zero to two degrees versus the window size squared

5.6 The affect of contrast level on the sum of absolute difference statistic of a rotated checker board

To understand the effect of contrast on the SAD statistic a series of one by one pixel checker boards with various levels of contrast were rotated. The contrast measure and intensities used are shown in Figure 5-14. The SAD statistics for the various contrast level checker boards are shown in Table 5-5 and the data is graphed for the various window sizes in Figures 5-15 to 5-21. The SAD statistic increases and then plateaus for all window sizes and contrast levels. The greater the contrast the greater the initial rate of increase in the statistic, with increasing rotation, and the greater the final plateau value. The larger the window size the smaller the rotation when the statistic plateaus. The graphs suggest that the higher the contrast of the image the greater the detrimental effect of rotation will be on the statistics performance when the object that is being tracked undergoes rotation.

In conclusion, for the vertebra tracking problem if in the future images become very high contrast then the effects of rotation will be more detrimental than the original results presented in section 5-2 and rotation may have to be included in the tracking algorithm.

CONTRAST LEVELS

C	I1	I2
1.0	0	255
0.8	46	230
0.6	82	204
0.4	107	179
0.2	122	153
0.0	128	128

Where:

$$C = \frac{\Delta I}{MAX\ I} = \frac{|I1 - I2|}{MAX(I1, I2)}$$

Figure 5-14 Contrast measure and intensities used

Table 5-5a The SAD statistics for a rotated one by one pixel checker pattern of varying contrast levels

Contrast value is 0.00, I1 = 128, I2 = 128

WIND_SIZE ROTATION	3	5	7	9	11	13	15
0.20	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.40	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.60	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.80	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.20	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.40	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.60	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.80	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
9.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
10.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
15.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
20.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
30.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Contrast value is 0.20, I1 = 122, I2 = 153

WIND_SIZE ROTATION	3	5	7	9	11	13	15
0.20	1.456	6.731	18.440	39.125	71.312	117.505	180.194
0.40	2.903	13.399	36.645	77.617	141.220	232.285	355.571
0.60	4.340	20.003	54.614	115.476	209.727	344.342	526.135
0.80	5.768	26.544	72.350	152.704	276.835	453.681	691.891
1.00	7.187	33.022	89.851	189.301	342.545	560.304	852.844
1.20	8.597	39.437	107.119	225.268	406.859	664.213	1008.998
1.40	9.997	45.790	124.154	260.608	469.780	765.413	1160.357
1.60	11.388	52.080	140.956	295.319	531.309	863.904	1306.924
1.80	12.771	58.307	157.527	329.405	591.447	959.691	1448.704
2.00	14.144	64.473	173.866	362.864	650.196	1052.774	1585.699
3.00	20.875	94.375	252.103	520.806	923.148	1477.710	2198.986
4.00	27.386	122.750	324.611	663.217	1161.546	1835.308	2692.948
5.00	33.682	149.613	391.432	790.166	1365.479	2125.663	3067.644
6.00	39.766	174.983	452.596	901.697	1534.985	2348.766	3322.952
7.00	45.644	198.871	508.128	997.832	1670.052	2504.506	3458.566
8.00	51.317	221.291	558.044	1078.569	1770.611	2592.670	3508.765
9.00	56.791	242.249	602.349	1143.881	1836.543	2631.623	3441.133
10.00	62.067	261.755	641.041	1193.720	1868.662	2605.758	3400.515
15.00	85.590	337.579	749.799	1252.083	1829.796	2593.138	3525.785
20.00	104.513	376.865	759.606	1237.482	1907.721	2653.340	3503.965
25.00	118.939	400.569	740.742	1273.217	1893.084	2638.075	3510.464
30.00	128.796	388.797	779.611	1277.628	1874.682	2597.507	3468.901

Table 5-5b The SAD statistics for a rotated one by one pixel checker pattern of varying contrast levels

Contrast value is 0.40, I1 = 107, I2 = 179

WIND_SIZE ROTATION	3	5	7	9	11	13	15
0.20	3.382	15.634	42.829	90.872	165.627	272.915	418.516
0.40	6.742	31.120	85.110	180.272	327.995	539.500	825.842
0.60	10.081	46.459	126.846	268.203	487.108	799.763	1221.991
0.80	13.397	61.651	168.038	354.666	642.971	1053.711	1606.973
1.00	16.693	76.696	208.686	439.666	795.588	1301.350	1980.799
1.20	19.967	91.596	248.792	523.204	944.964	1542.689	2343.479
1.40	23.219	106.350	288.357	605.282	1091.103	1777.733	2695.022
1.60	26.451	120.959	327.383	685.903	1234.008	2006.488	3035.437
1.80	29.661	135.424	365.869	765.069	1373.683	2228.959	3364.732
2.00	32.850	149.744	403.818	842.782	1510.132	2445.152	3682.913
3.00	48.484	219.195	585.530	1209.614	2144.087	3432.100	5107.323
4.00	63.607	285.096	753.936	1540.376	2697.784	4262.651	6254.589
5.00	78.229	347.489	909.132	1835.225	3171.434	4937.023	7124.851
6.00	92.361	406.412	1051.191	2094.265	3565.127	5455.198	7717.824
7.00	106.011	461.895	1180.169	2317.546	3878.830	5816.918	8032.799
8.00	119.188	513.965	1296.101	2505.063	4112.388	6021.685	8149.389
9.00	131.901	562.644	1399.003	2656.756	4265.520	6112.157	7992.310
10.00	144.156	607.947	1488.870	2772.510	4340.117	6052.082	7897.971
15.00	198.789	784.054	1741.468	2908.064	4249.849	6022.772	8188.920
20.00	242.740	875.298	1764.246	2874.151	4430.836	6162.596	8138.240
25.00	276.246	930.353	1720.434	2957.150	4396.841	6127.143	8153.336
30.00	299.138	903.013	1810.710	2967.393	4354.100	6032.920	8056.803

Contrast value is 0.60, I1 = 82, I2 = 204

WIND_SIZE ROTATION	3	5	7	9	11	13	15
0.20	5.731	26.491	72.571	153.977	280.646	462.440	709.152
0.40	11.424	52.731	144.215	305.461	555.769	914.153	1399.344
0.60	17.081	78.722	214.934	454.454	825.377	1355.154	2070.596
0.80	22.701	104.464	284.731	600.963	1089.478	1785.454	2722.927
1.00	28.285	129.958	353.607	744.990	1348.080	2205.066	3356.354
1.20	33.832	155.204	421.564	886.540	1601.189	2614.001	3970.895
1.40	39.344	180.204	488.605	1025.617	1848.813	3012.269	4566.566
1.60	44.819	204.959	554.732	1162.225	2090.958	3399.882	5143.380
1.80	50.259	229.468	619.945	1296.367	2327.630	3776.847	5701.351
2.00	55.663	253.732	684.247	1428.047	2558.835	4143.175	6240.491
3.00	82.154	371.413	992.147	2049.624	3633.036	5815.503	8654.074
4.00	107.779	483.079	1277.503	2610.081	4571.245	7222.825	10598.053
5.00	132.555	588.801	1540.473	3109.687	5373.819	8365.511	12072.664
6.00	156.500	688.643	1781.185	3548.616	6040.910	9243.530	13077.424
7.00	179.629	782.656	1999.731	3926.953	6572.462	9856.445	13611.131
8.00	201.958	870.885	2196.172	4244.690	6968.212	10203.410	13808.687
9.00	223.499	953.368	2370.533	4501.726	7227.687	10356.710	13542.525
10.00	244.265	1030.132	2522.807	4697.864	7354.087	10254.917	13382.672
15.00	336.836	1328.537	2950.821	4927.553	7201.132	10205.252	13875.669
20.00	411.310	1483.145	2989.417	4870.089	7507.806	10442.176	13789.796
25.00	468.084	1576.431	2915.180	5010.726	7450.202	10382.103	13815.374
30.00	506.873	1530.106	3068.147	5028.083	7377.781	10222.449	13651.805

Table 5-5c The SAD Statistics for a rotated one by one pixel checker pattern of various contrast levels

Contrast value is 0.80, I1 = 46, I2 = 230

WIND_SIZE ROTATION	3	5	7	9	11	13	15
0.20	8.643	39.953	109.451	232.228	423.269	697.450	1069.541
0.40	17.230	79.529	217.504	460.695	838.208	1378.723	2110.486
0.60	25.761	118.728	324.163	685.406	1244.831	2043.839	3122.866
0.80	34.238	157.552	429.430	906.370	1643.147	2692.816	4106.709
1.00	42.659	196.002	533.309	1123.591	2033.169	3325.673	5062.042
1.20	51.026	234.079	635.802	1337.077	2414.908	3942.427	5988.891
1.40	59.338	271.784	736.913	1546.832	2788.374	4543.095	6887.279
1.60	67.596	309.118	836.644	1752.864	3153.576	5127.691	7757.229
1.80	75.800	346.083	934.999	1955.176	3510.524	5696.229	8598.759
2.00	83.950	382.679	1031.979	2153.775	3859.226	6248.723	9411.888
3.00	123.905	560.164	1496.354	3091.236	5479.333	8770.923	13052.046
4.00	162.552	728.578	1926.726	3936.516	6894.336	10893.441	15983.950
5.00	199.919	888.028	2323.337	4690.019	8104.777	12616.836	18207.952
6.00	236.033	1038.608	2686.378	5352.011	9110.881	13941.061	19723.328
7.00	270.916	1180.399	3015.988	5922.618	9912.566	14865.457	20528.264
8.00	304.592	1313.466	3312.259	6401.828	10509.435	15388.749	20826.217
9.00	337.080	1437.867	3575.230	6789.488	10900.773	15619.956	20424.791
10.00	368.400	1553.642	3804.890	7085.303	11091.410	15466.432	20183.702
15.00	508.015	2003.695	4450.418	7431.719	10860.724	15391.527	20927.239
20.00	620.336	2236.874	4508.629	7345.052	11323.249	15748.856	20797.726
25.00	705.963	2377.569	4396.665	7557.161	11236.370	15658.254	20836.302
30.00	764.465	2307.701	4627.370	7583.338	11127.145	15417.463	20589.607

Contrast value is 1.00, I1 = 0, I2 = 255

WIND_SIZE ROTATION	3	5	7	9	11	13	15
0.20	11.978	55.370	151.685	321.838	586.596	966.575	1482.244
0.40	23.878	110.217	301.433	638.463	1161.648	1910.730	2924.858
0.60	35.702	164.542	449.248	949.884	1725.173	2832.494	4327.885
0.80	47.449	218.347	595.134	1256.110	2277.187	3731.892	5691.363
1.00	59.120	271.633	739.097	1557.151	2817.707	4608.949	7015.330
1.20	70.715	324.403	881.139	1853.014	3346.747	5463.690	8299.822
1.40	82.234	376.657	1021.265	2143.708	3864.322	6296.137	9544.871
1.60	93.679	428.397	1159.480	2429.240	4370.445	7106.310	10750.507
1.80	105.049	479.625	1295.786	2709.619	4865.128	7894.230	11916.758
2.00	116.344	530.342	1430.188	2984.852	5348.385	8659.915	13043.649
3.00	171.715	776.314	2073.751	4284.050	7593.640	12155.355	18088.434
4.00	225.276	1009.714	2670.191	5455.498	9554.651	15096.888	22151.669
5.00	277.062	1230.692	3219.842	6499.755	11232.163	17485.289	25233.846
6.00	327.111	1439.376	3722.969	7417.189	12626.493	19320.492	27333.960
7.00	375.455	1635.878	4179.766	8207.976	13737.524	20601.585	28449.496
8.00	422.125	1820.293	4590.359	8872.098	14564.706	21326.799	28862.420
9.00	467.149	1992.696	4954.803	9409.344	15107.050	21647.221	28306.097
10.00	510.554	2153.145	5273.081	9819.306	15371.248	21434.457	27971.979
15.00	704.043	2776.859	6167.699	10299.393	15051.547	21330.650	29002.423
20.00	859.705	3100.015	6248.372	10179.285	15692.546	21825.860	28822.935
25.00	978.373	3295.000	6093.204	10473.239	15572.143	21700.298	28876.397
30.00	1059.449	3198.172	6412.931	10509.518	15420.771	21366.593	28534.509

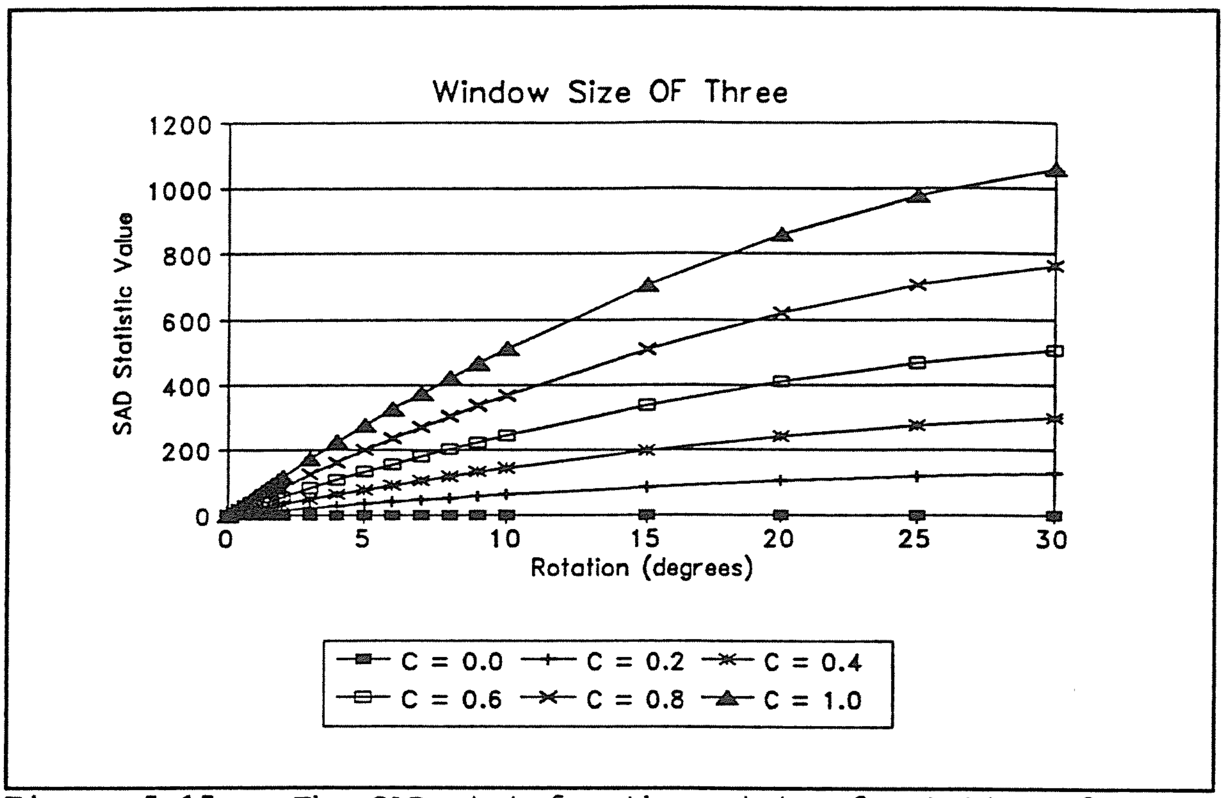


Figure 5-15 The SAD stat for the origin of rotation of a one by one pixel checker pattern of various contrasts

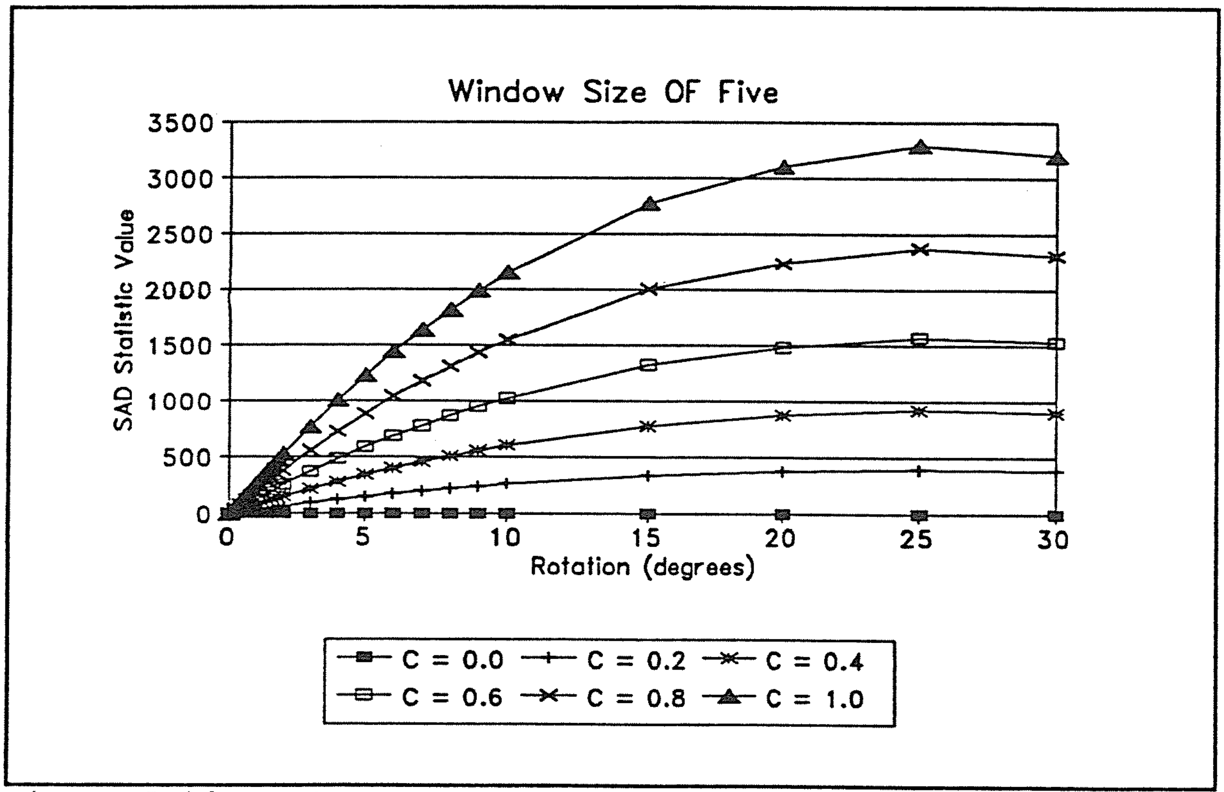


Figure 5-16 The SAD stat for the origin of a rotated one by one pixel checker pattern of various contrasts

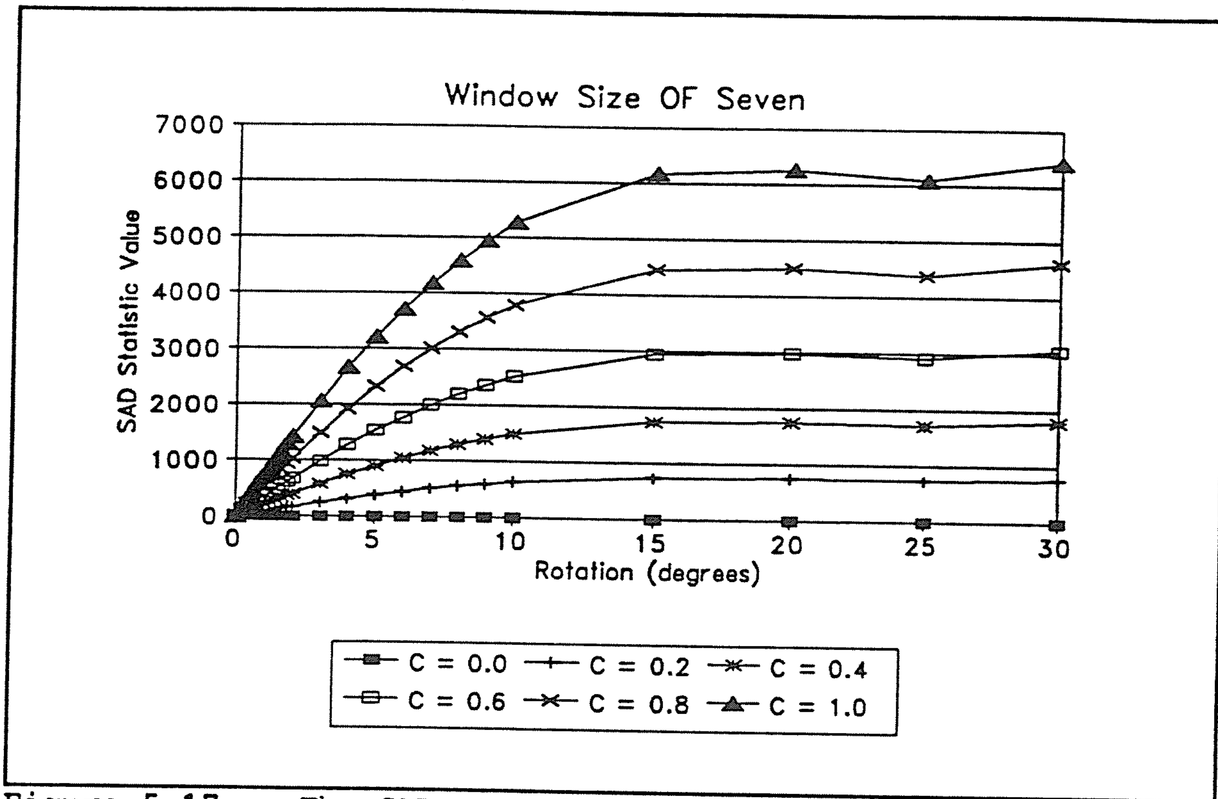


Figure 5-17 The SAD stat for the origin of a rotated one by one pixel checker pattern of various contrasts

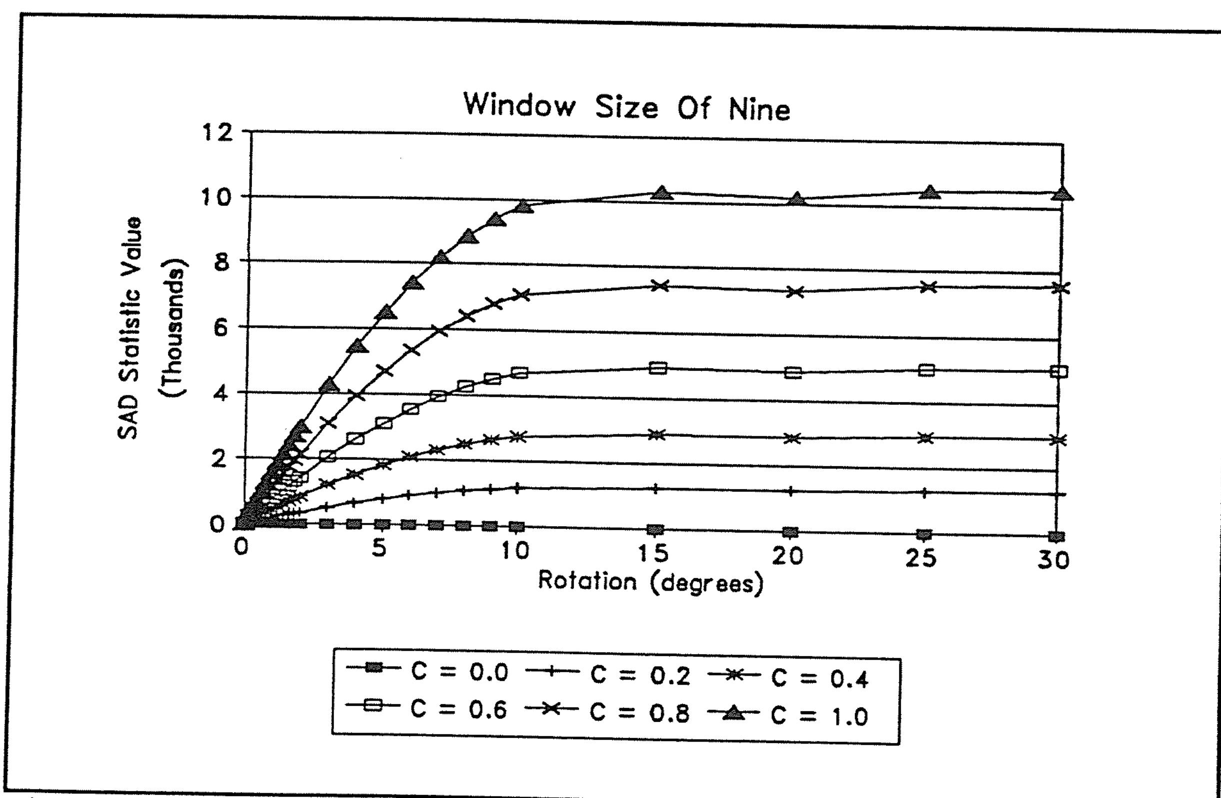


Figure 5-18 The SAD stat for the origin of a rotated one by one pixel checker pattern of various contrasts

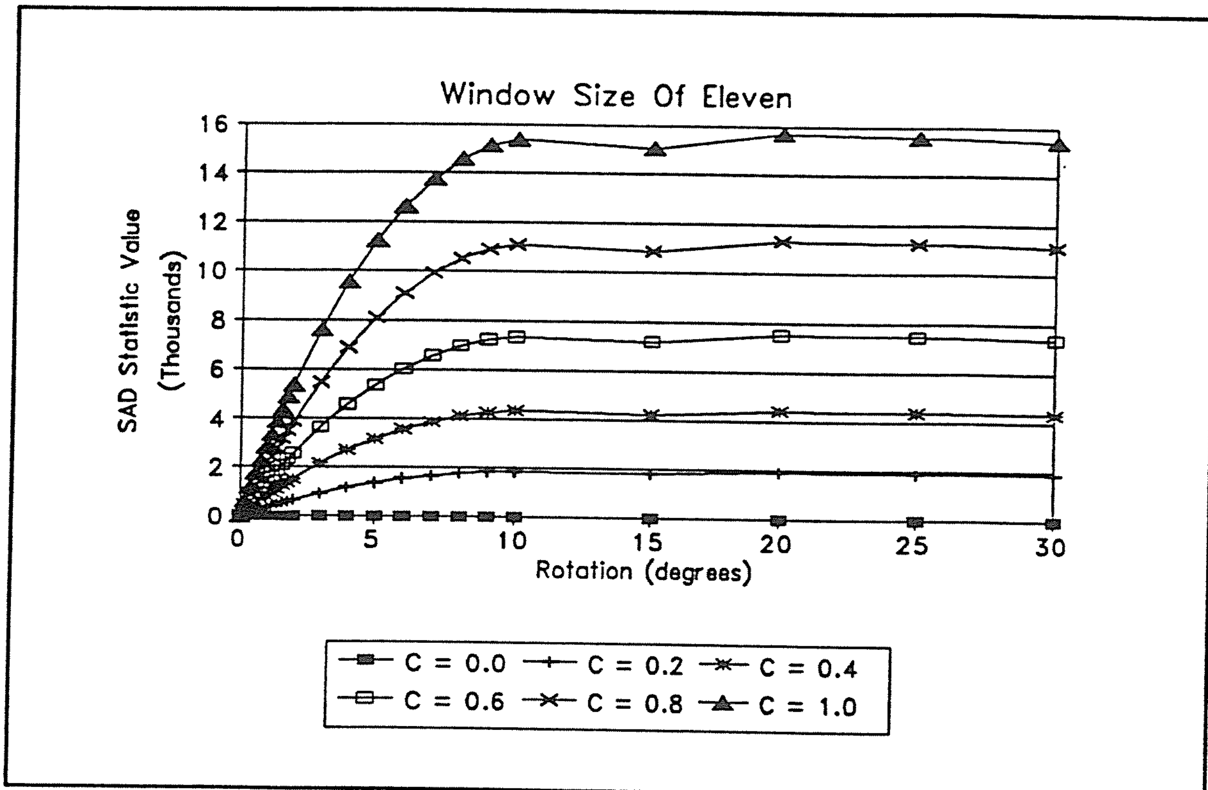


Figure 5-19 The SAD stat for the origin of a rotated one by one pixel checker pattern of various contrasts

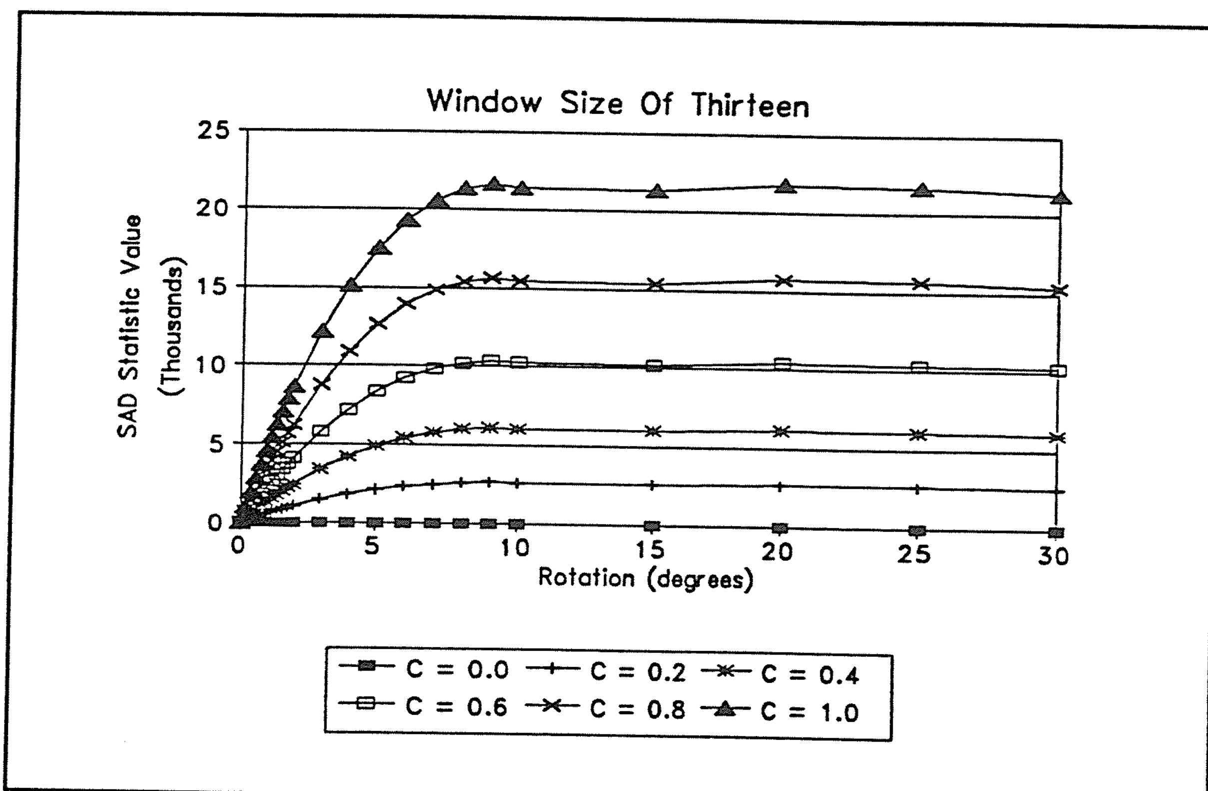


Figure 5-20 The SAD stat for the origin of a rotated one by one pixel checker pattern of various contrasts

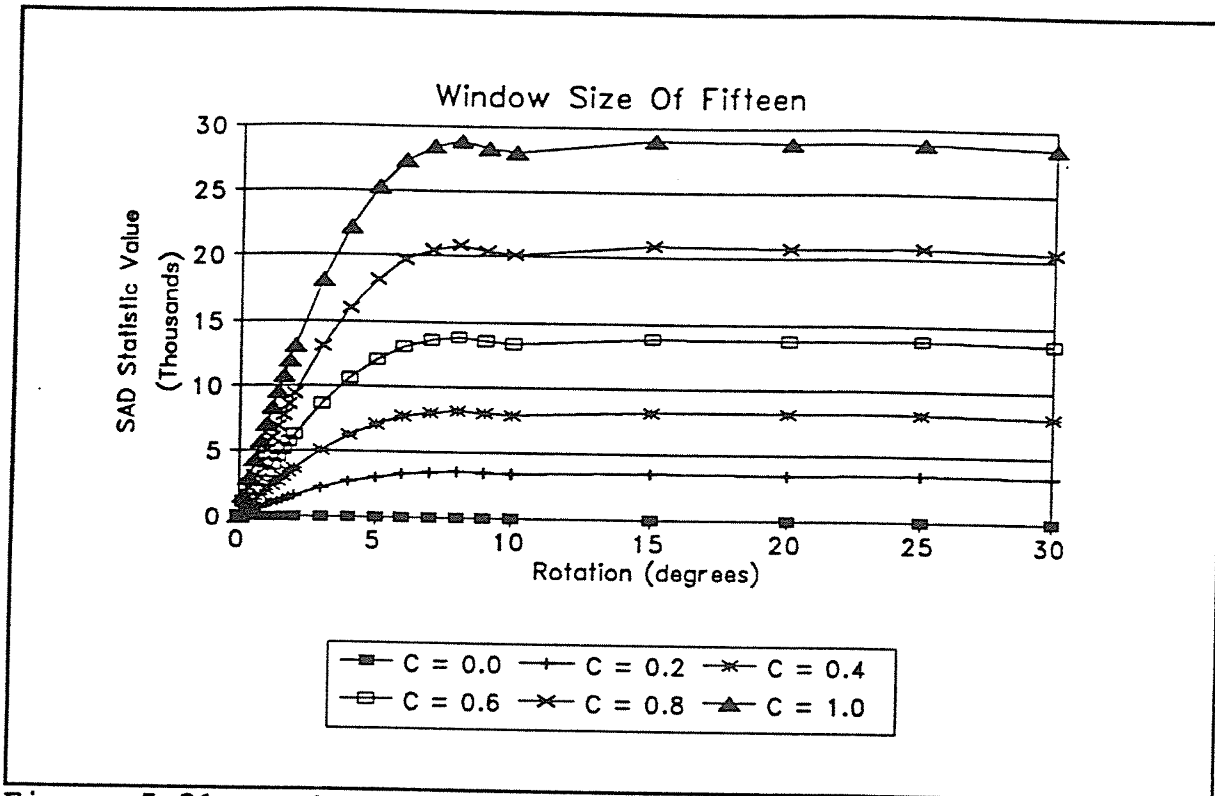


Figure 5-21 The SAD stat of the origin of a rotated one by one pixel checker pattern of various contrasts

Chapter 6

Effect of sub-pixel translation between consecutive frames

6.1 Mathematical simulation of sub-pixel effect

The effect on the sum of absolute difference statistic of translation between zero and one pixel can be easily simulated mathematically. For translation in the x direction between -1 and +1 pixels, where a positive translation is a movement to the right, $f_p(i,j)$ is the original intensity of the pixel at position (i,j) in frame p, and $f_p'(i,j)$ is the new intensity after the sub-pixel translation. The intensity of a pixel after translation is proportional to the intensity of the pixels before translation and the area of each pixel that falls within the new pixel boundaries. If x denotes the amount of translation, and w is the width of the pixel then :

$$f_p'(i,j) = f_p(i-\frac{x}{|x|},\ j) * \frac{|x| * w}{w^2} + f_p(i,j) * \frac{(w-|x|) * w}{w^2}$$

$$\Delta\ f_p(i,j) = f_p(i,j) - f_p'(i,j)$$

$$\begin{aligned}\Delta\ f_p(i,j) &= f_p(i,j) - (f_p(i-\frac{x}{|x|},\ j) * \frac{|x| * w}{w^2} + f_p(i,j) * w*(w-\frac{|x|}{w^2}) \\ &= f_p(i,j) * (1 - \frac{w-|x|}{w}) - f_p(i-\frac{x}{|x|},\ j) * \frac{|x|}{w} \\ &= \frac{|x|}{w} * (f_p(i,j) - f_p(i - \frac{x}{|x|},\ j)\end{aligned}$$

These equations are fully derived in appendix C.

The SAD Statistic, for an x direction translation between -1 and +1 pixels, where the window radius is denoted by n, is given by

$$S_1 = \frac{|x|}{W} * \sum_{i,j}^{n} | f_p(i,j) - f_p(i - \frac{x}{|x|}, j) |$$

Similarly, for a y translation between -1 and +1 pixels, the SAD Statistic is given by

$$S_1 = \frac{|y|}{W} * \sum_{i,j}^{n} | f_p(i,j) - f_p(i, j - \frac{y}{|y|}) |$$

Using these formulae the effect of sub-pixel translation on the SAD Statistic can be seen for the specified point on the corner of the C1 vertebra, where C1 is the vertebra immediately beneath the skull, for different window sizes and translation in the x and y direction in Table 6-1 and graphed in Figure 6-1. The amount the statistic is effected is given by the translation (between -1 and +1 pixels) multiplied by the values in the Table 6-1 corresponding to the correct window size and direction.

From this data it can be seen that negative x and positive x translation have nearly the same effect on the statistic, similarly negative y and positive y have the same effect. Therefore the predominant effect on the statistic is whether the translation is in the x or y direction (not the sign), what window size is being used, and the modulus of the translation undergone. The statistic increases with an increase in modulus of translation and an increase in window size. However Table 6-1 only holds for an exclusive x or y translation not a combined translation.

Table 6-1 Mathematical simulation of the SAD statistic for point (28, 22) in $frame_0$ when tracked to $frame_0$ translated between -1 and +1 pixels

For the specified location (28,22) in $frame_0$.
Translation to the right is positive , to the left is negative.

For X Translation the SAD Statistic is given by the values below multiplied by the translation undergone.

Window Size	Negative Direction	Positive Direction
3	62	57
5	120	120
7	222	284
9	509	467
11	714	741
13	1003	1035
15	1339	1389

For Y Translation the SAD Statistic is given by the values below multiplied by the translation undergone.

Window Size	Negative Direction	Positive Direction
3	45	38
5	88	104
7	202	196
9	332	345
11	519	513
13	727	736
15	977	939

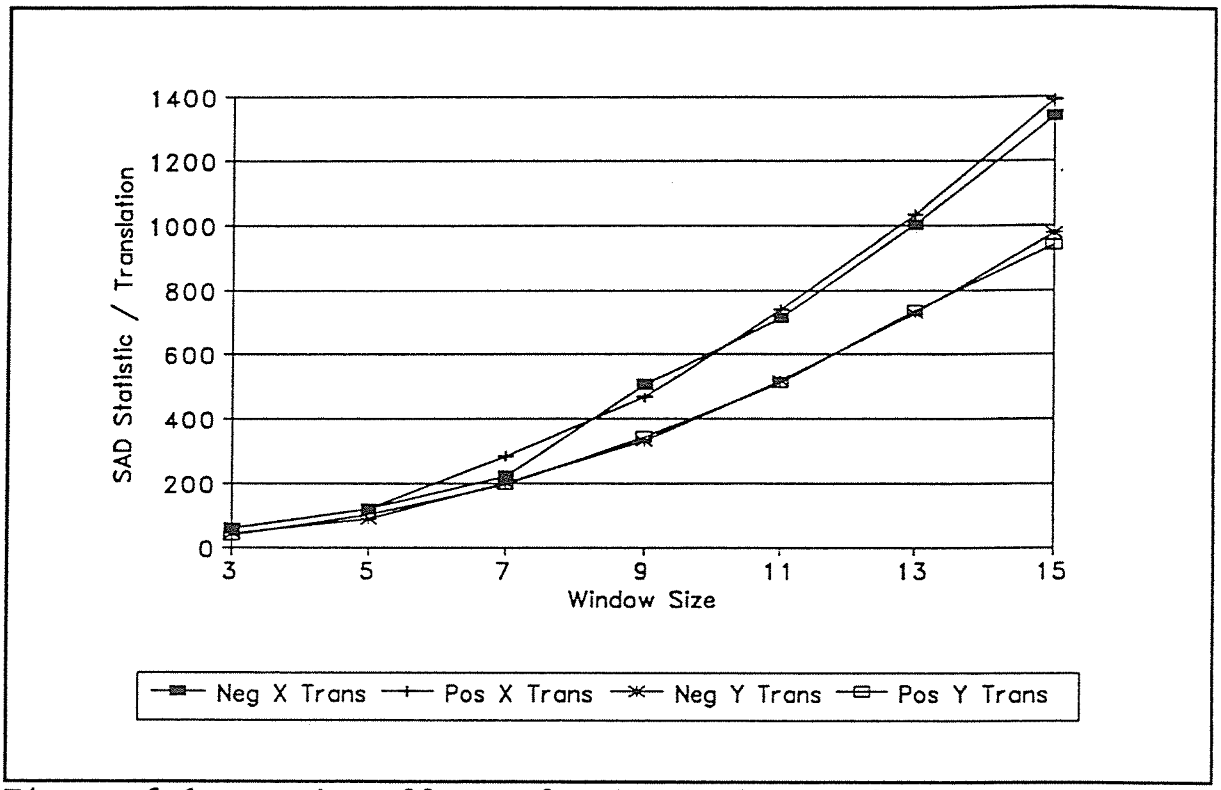


Figure 6-1 The effect of sub pixel translation on the SAD statistic

6.2 The difference between the actual sum of absolute difference statistic and the next minimum after a sub-pixel translation

The difference between the SAD statistic for the correct location, and the next lowest minimum statistic for various window sizes when tracking from $frame_0$ to $frame_0$ translated by a sub-pixel amount between negative two and two pixels in either the x direction or y direction are shown in Table 6-2 and 6-3. These Tables are graphed in Figures 6-2 and 6-3. A negative value in the Tables, or graphs, corresponds to a condition where the minimum statistic was given by a location other than the correct one which means an incorrect location would have been recorded during tracking. Tables 6-4 to 6-10 shows the difference between the correct position statistics and next lowest when the image was being translated in both the x and y direction from -0.7 to +0.7 for the various window sizes. This data is mesh plotted in Figures 6-4 to 6-10. These figures include mesh plots of the negative values only in each table.

On the mesh plots the x-axis is the translation in the x direction, the y-axis is the translation in the y direction and the z-axis is the difference between the SAD statistic for the correct position and the next lowest statistic.

For a window size of three by three there were seventy-six wrong positions chosen, for a window size of five, seven and eleven there were eight incorrect positions, for a window size of nine there were twelve wrong positions, for a thirteen by thirteen window there were four, and no incorrect positions were chosen by the size fifteen by fifteen window. When the magnitude of the x or y translation was 0.5 there are two equally correct positions for the algorithm to select - these are adjacent positions.

Table 6-2 Difference between correct position statistic and next minimum SAD statistic for X translation between -2.0 and +2.0 pixels

Translation	Window Size						
X	3	5	7	9	11	13	15
-2.0	28.000	88.000	196.000	332.000	513.000	727.000	939.000
-1.9	19.200	77.100	158.000	283.100	415.900	606.200	760.000
-1.8	14.400	67.000	120.800	222.400	324.000	483.600	597.800
-1.7	8.900	48.000	70.200	165.900	240.900	369.800	451.600
-1.6	1.200	24.000	19.600	113.400	132.000	187.800	247.800
-1.5	-3.500	0.000	31.000	21.000	13.500	16.000	25.000
-1.4	1.200	24.000	81.600	76.600	159.000	219.800	297.800
-1.3	8.400	47.800	119.300	145.700	234.800	346.400	482.300
-1.2	16.800	61.200	144.600	198.200	315.000	451.000	619.200
-1.1	21.800	74.600	171.300	258.900	410.800	582.200	789.900
-1.0	28.000	88.000	196.000	332.000	513.000	727.000	939.000
-0.9	19.200	77.100	158.000	283.100	415.900	606.200	760.000
-0.8	14.400	67.000	120.800	222.400	324.000	483.600	597.800
-0.7	8.900	48.000	70.200	165.900	240.900	369.800	451.600
-0.6	1.200	24.000	19.600	113.400	132.000	187.800	247.800
-0.5	-3.500	0.000	31.000	21.000	13.500	16.000	25.000
-0.45	-2.400	12.000	56.300	27.800	86.250	117.900	161.400
-0.40	1.200	24.000	81.600	76.600	159.000	219.800	297.800
-0.35	4.800	36.000	106.850	121.750	199.300	301.500	424.150
-0.30	8.400	47.800	119.300	145.700	234.800	346.400	482.300
-0.25	12.500	54.500	131.750	170.750	272.000	394.000	544.750
-0.20	16.800	61.200	144.600	198.200	315.000	451.000	619.200
-0.15	19.200	67.900	157.450	225.950	360.300	513.800	701.350
-0.10	21.800	74.600	171.300	258.900	410.800	582.200	789.900
-0.05	24.700	81.300	184.250	295.250	464.700	654.400	875.700
0.00	28.000	88.000	196.000	332.000	513.000	727.000	939.000
0.05	23.300	82.550	177.000	307.900	463.450	669.850	847.400
0.10	19.200	77.100	158.000	283.100	415.900	606.200	760.000
0.15	16.800	71.750	139.100	252.250	369.150	544.100	677.000
0.20	14.400	67.000	120.800	222.400	324.000	483.600	597.800
0.25	12.000	60.000	95.500	193.750	280.750	425.000	521.500
0.30	8.900	48.000	70.200	165.900	240.900	369.800	451.600
0.35	4.050	36.000	44.900	138.150	204.750	289.700	384.200
0.40	1.200	24.000	19.600	113.400	132.000	187.800	247.800
0.45	-1.150	12.000	-5.700	69.800	59.250	85.900	111.400
0.50	-3.500	0.000	31.000	21.000	13.500	16.000	25.000
0.6	1.200	24.000	81.600	76.600	159.000	219.800	297.800
0.7	8.400	47.800	119.300	145.700	234.800	346.400	482.300
0.8	16.800	61.200	144.600	198.200	315.000	451.000	619.200
0.9	21.800	74.600	171.300	258.900	410.800	582.200	789.900
1.0	28.000	88.000	196.000	332.000	513.000	727.000	939.000
1.1	19.200	77.100	158.000	283.100	415.900	606.200	760.000
1.2	14.400	67.000	120.800	222.400	324.000	483.600	597.800
1.3	8.900	48.000	70.200	165.900	240.900	369.800	451.600
1.4	1.200	24.000	19.600	113.400	132.000	187.800	247.800
1.5	-3.500	0.000	31.000	21.000	13.500	16.000	25.000
1.6	1.200	24.000	81.600	76.600	159.000	219.800	297.800
1.7	8.400	47.800	119.300	145.700	234.800	346.400	482.300
1.8	16.800	61.200	144.600	198.200	315.000	451.000	619.200
1.9	21.800	74.600	171.300	258.900	410.800	582.200	789.900
2.0	29.000	88.000	196.000	332.000	513.000	727.000	939.000

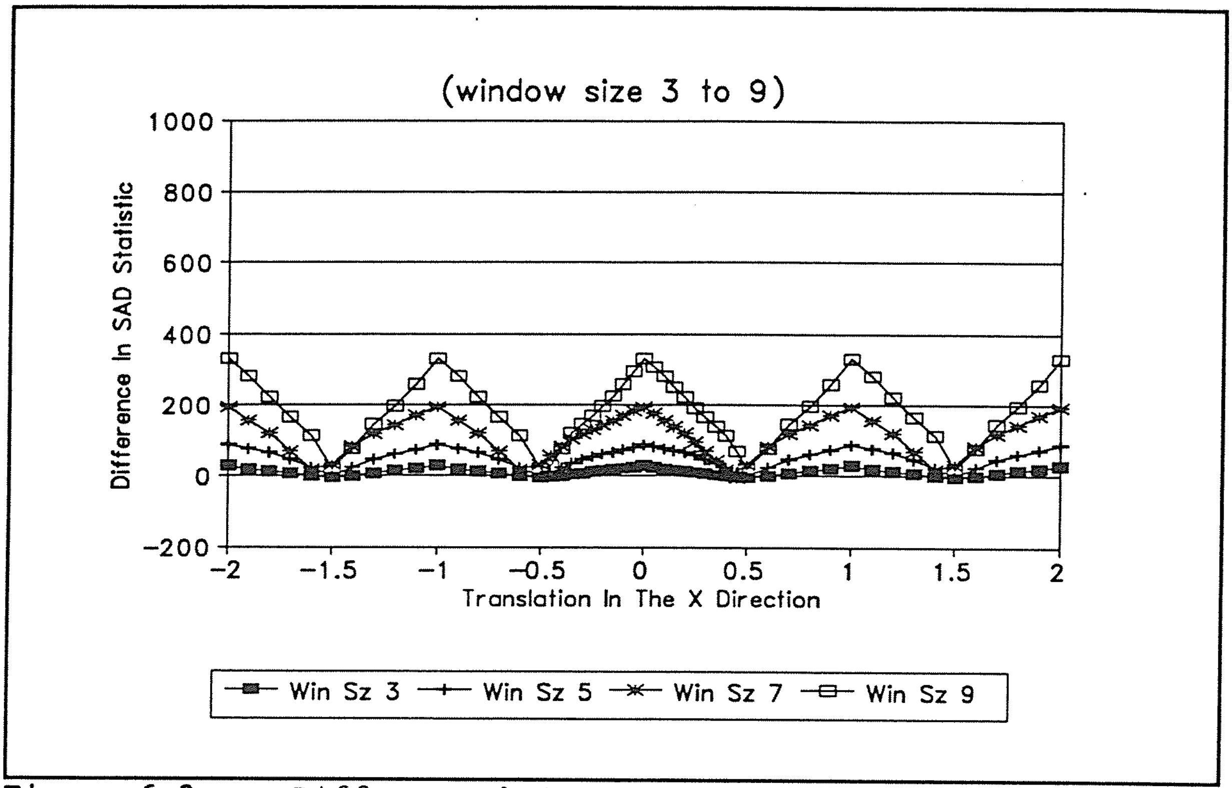


Figure 6-2a Difference between correct position statistic and next minimum SAD statistic for X translations between -2 and +2 pixels

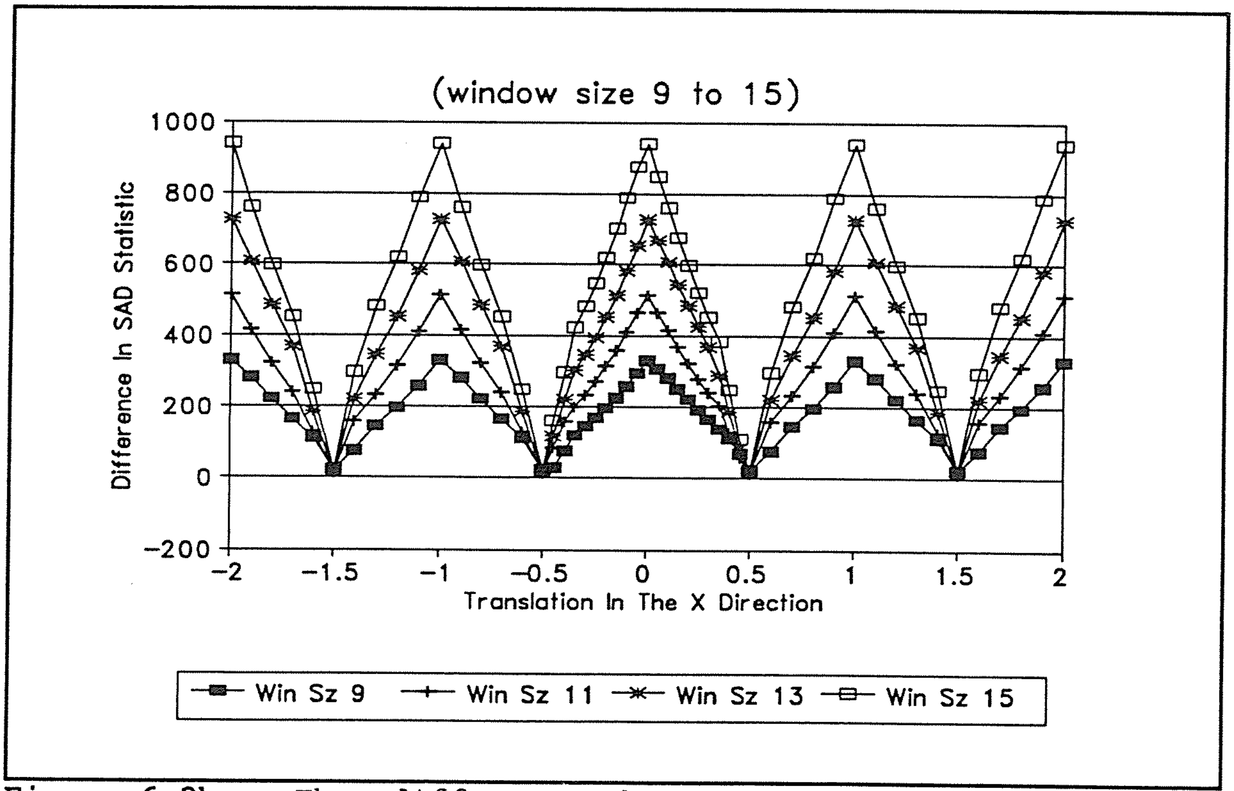


Figure 6-2b The difference between the correct position statistic and next minimum statistic for X translations between -2 and +2 pixels

Table 6-3 Difference between the correct position statistic and the next minimum SAD statistic for Y translation from -2 to +2 pixels

Translation Y	WINDOW SIZE 3	5	7	9	11	13	15
-2.0	17.000	88.000	196.000	332.000	513.000	727.000	939.000
-1.9	14.100	68.800	162.200	264.300	415.800	580.700	785.400
-1.8	11.200	49.600	122.400	196.600	312.600	434.400	593.800
-1.7	8.300	30.400	82.600	128.900	209.400	288.100	402.200
-1.6	5.400	11.200	42.800	61.200	106.200	141.800	210.600
-1.5	2.500	8.000	3.000	6.500	3.000	4.500	19.000
-1.4	4.400	27.200	36.800	74.200	100.200	150.800	172.600
-1.3	8.000	46.400	76.600	141.900	203.400	297.100	364.200
-1.2	11.000	65.600	116.400	209.600	306.600	443.400	555.800
-1.1	14.000	84.800	156.200	277.300	409.800	589.700	747.400
-1.0	17.000	88.000	196.000	332.000	513.000	727.000	939.000
-0.9	14.100	68.800	162.200	264.300	415.800	580.700	785.400
-0.8	11.200	49.600	122.400	196.600	312.600	434.400	593.800
-0.7	8.300	30.400	82.600	128.900	209.400	288.100	402.200
-0.6	5.400	11.200	42.800	61.200	106.200	141.800	210.600
-0.5	2.500	8.000	3.000	6.500	3.000	4.500	19.000
-0.45	0.650	17.600	16.900	40.350	48.600	77.650	76.800
-0.40	4.400	27.200	36.800	74.200	100.200	150.800	172.600
-0.35	7.100	36.800	56.700	108.050	151.800	223.950	268.400
-0.30	9.800	46.400	76.600	141.900	203.400	297.100	364.200
-0.25	12.500	56.000	96.500	175.750	255.000	370.250	460.000
-0.20	15.600	65.600	116.400	209.600	306.600	443.400	555.800
-0.15	18.700	75.200	136.300	243.450	358.200	516.550	651.600
-0.10	21.800	84.800	156.200	277.300	409.800	589.700	747.400
-0.05	24.900	86.600	176.100	311.150	461.400	662.850	843.200
0.00	28.000	88.000	196.000	332.000	513.000	727.000	939.000
0.05	25.500	78.400	182.100	298.150	467.400	653.850	881.200
0.10	22.200	68.800	162.200	264.300	415.800	580.700	785.400
0.15	19.800	59.200	142.300	230.450	364.200	507.550	689.600
0.20	17.800	49.600	122.400	196.600	312.600	434.400	593.800
0.25	16.250	40.000	102.500	162.750	261.000	361.250	498.000
0.30	13.100	30.400	82.600	128.900	209.400	288.100	402.200
0.35	11.250	20.800	62.700	95.050	157.800	214.950	306.400
0.40	11.000	11.200	42.800	61.200	106.200	141.800	210.600
0.45	7.650	1.600	22.900	27.350	54.600	68.650	114.800
0.50	3.500	8.000	3.000	6.500	3.000	4.500	19.000
0.6	4.800	27.200	36.800	74.200	100.200	150.800	172.600
0.7	12.000	46.400	76.600	141.900	203.400	297.100	364.200
0.8	17.000	65.600	116.400	209.600	306.600	443.400	555.800
0.9	23.000	84.800	156.200	277.300	409.800	589.700	747.400
1.0	29.000	88.000	196.000	332.000	513.000	727.000	939.000
1.1	22.200	68.800	162.200	264.300	415.800	580.700	785.400
1.2	17.800	49.600	122.400	196.600	312.600	434.400	593.800
1.3	13.100	30.400	82.600	128.900	209.400	288.100	402.200
1.4	11.000	11.200	42.800	61.200	106.200	141.800	210.600
1.5	3.500	8.000	3.000	6.500	3.000	4.500	19.000
1.6	4.800	27.200	36.800	74.200	100.200	150.800	172.600
1.7	12.000	46.400	76.600	141.900	203.400	297.100	364.200
1.8	17.000	65.600	116.400	209.600	306.600	443.400	555.800
1.9	23.000	84.800	156.200	277.300	409.800	589.700	747.400
2.0	29.000	88.000	196.000	332.000	513.000	727.000	939.000

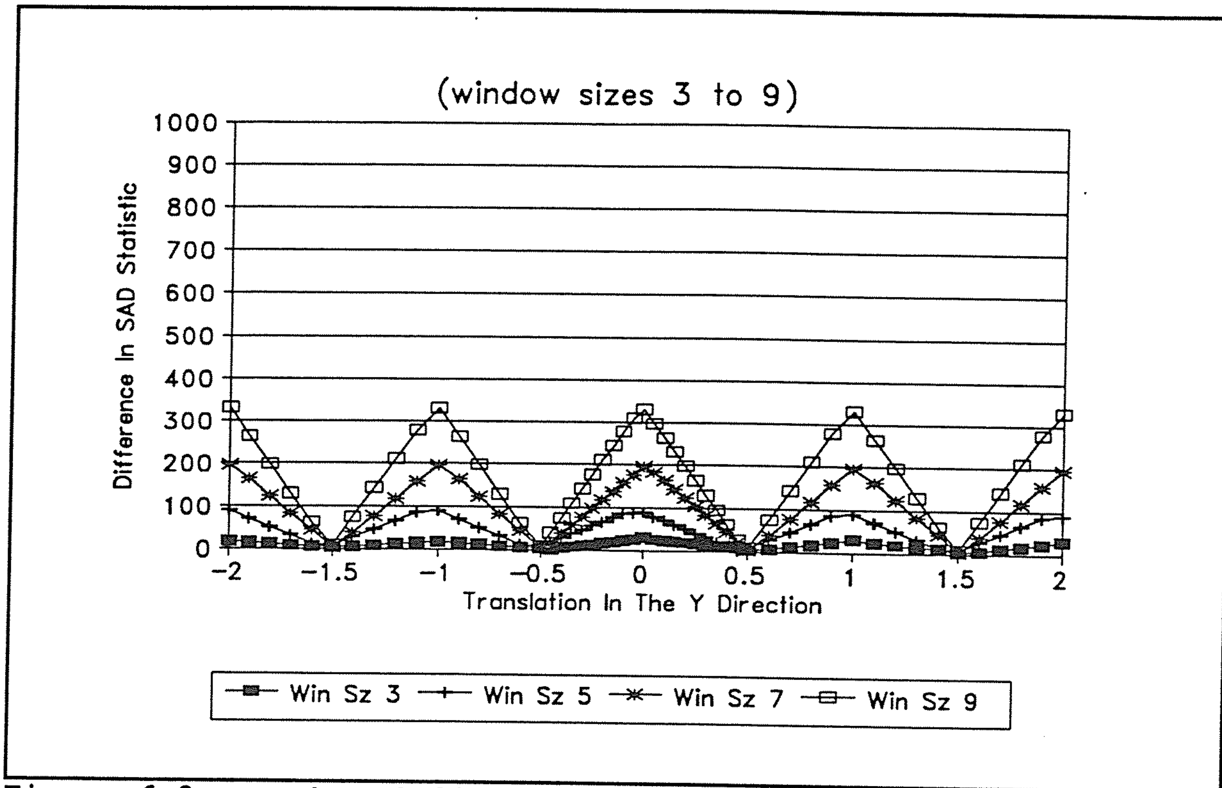


Figure 6-3a The difference between the correct position statistic and the next minimum SAD statistic for Y translations between -2 and +2 pixels

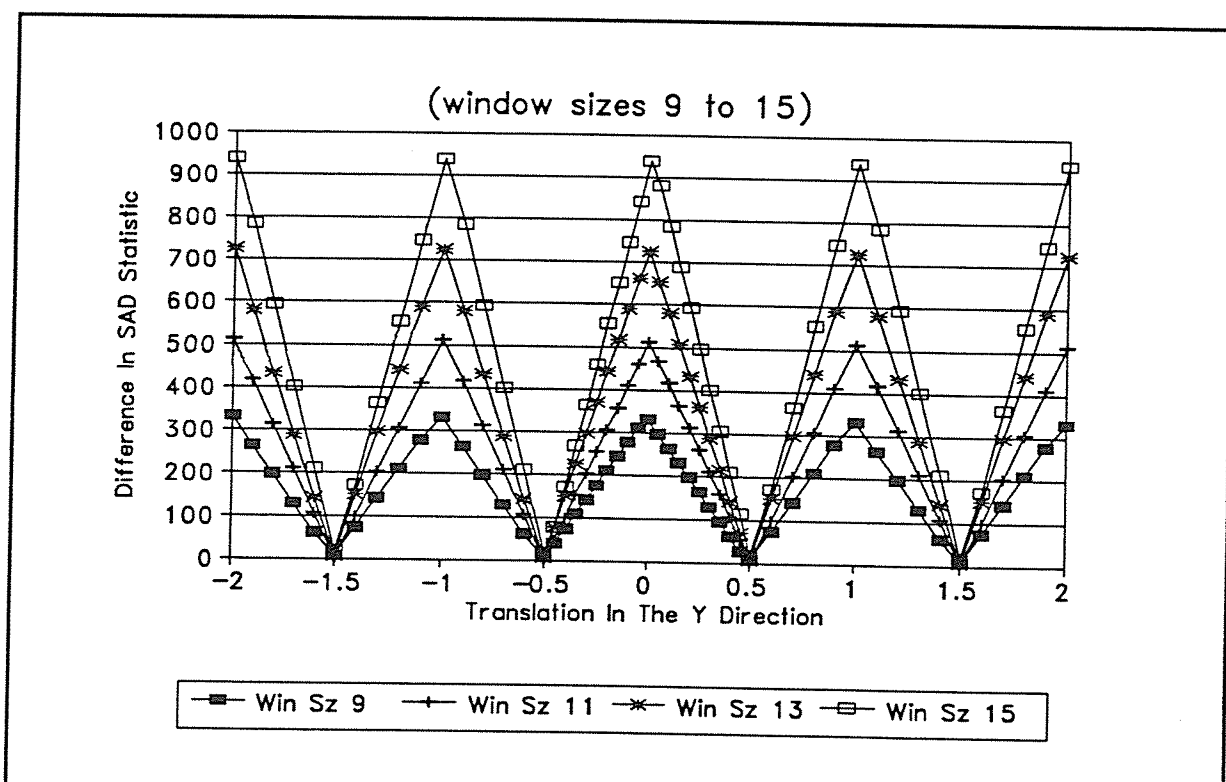


Figure 6-3b The difference between the correct position statistic and the next minimum SAD statistic for Y translations between -2 and +2 pixels

Table 6-4 Difference between SAD statistic for the correct location & next minimum statistic for a window size of three

Window Size : 3
Maximum Difference : 28.000
Minimum Difference : -15.360

Translation In The Y Direction

Tran X	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
-0.7	0.820	-0.500	-0.800	-7.980	-2.910	3.960	6.390	8.300	7.940	3.980	0.820	-0.500	-0.800	-7.980	-2.910
-0.6	1.300	-0.440	-1.100	-7.760	-5.420	0.680	3.720	5.400	5.580	3.440	1.300	-0.440	-1.100	-7.760	-5.420
-0.5	1.300	-0.100	0.000	0.800	1.250	1.300	0.000	2.500	3.000	2.500	1.300	-0.100	0.000	0.800	1.250
-0.4	-12.900	-15.360	-5.200	1.240	3.920	4.920	4.760	4.400	-2.340	-9.400	-12.900	-15.360	-5.200	1.240	3.920
-0.3	-8.140	-11.220	-4.650	2.120	5.800	9.080	9.520	9.800	3.100	-2.560	-8.140	-11.220	-4.650	2.120	5.800
-0.2	-3.260	-7.080	-5.100	1.680	8.720	12.280	14.720	15.600	8.320	3.120	-3.260	-7.080	-5.100	1.680	8.720
-0.1	2.540	-2.940	-5.550	1.440	8.950	14.920	19.420	21.800	13.800	8.760	2.540	-2.940	-5.550	1.440	8.950
0.0	8.900	1.200	-3.500	1.200	8.400	16.800	21.800	28.000	19.200	14.400	8.900	1.200	-3.500	1.200	8.400
0.1	10.600	3.060	-1.650	-2.460	5.130	13.360	18.870	22.200	19.280	14.580	10.600	3.060	-1.650	-2.460	5.130
0.2	11.240	5.320	-0.200	-6.280	1.080	8.920	15.300	17.800	18.380	15.160	11.240	5.320	-0.200	-6.280	1.080
0.3	11.860	7.580	0.700	-7.980	-2.910	3.960	10.950	13.100	14.300	13.640	11.860	7.580	0.700	-7.980	-2.910
0.4	11.380	8.800	2.500	-7.760	-5.420	0.680	6.720	11.000	10.940	12.320	11.380	8.800	2.500	-7.760	-5.420
0.5	9.800	9.400	4.750	0.800	1.250	1.300	0.000	3.500	10.100	12.000	9.800	9.400	4.750	0.800	1.250
0.6	-12.900	-15.360	-5.200	1.240	3.920	4.920	4.760	4.800	-2.340	-9.400	-12.900	-15.360	-5.200	1.240	3.920
0.7	-8.140	-11.220	-4.650	2.120	5.800	9.080	9.520	12.000	5.420	-2.260	-8.140	-11.220	-4.650	2.120	5.800

Table 6-5 Difference between SAD statistic for correct location & next minimum statistic for a window size of five

Window Size : 5
Maximum Difference : 88.000
Minimum Difference : -8.480

Translation In The Y Direction

Tran X	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
-0.7	33.050	28.160	12.600	2.200	10.330	18.140	25.490	30.400	34.290	34.180	33.050	28.160	12.600	2.200	10.330
-0.6	19.500	20.240	8.200	-8.480	-3.880	2.480	8.200	11.200	16.580	18.840	19.500	20.240	8.200	-8.480	-3.880
-0.5	5.350	7.800	4.000	12.200	17.050	13.100	9.250	8.000	1.350	3.300	5.350	7.800	4.000	12.200	17.050
-0.4	8.560	4.480	2.000	17.440	29.380	28.680	26.740	27.200	19.320	12.640	8.560	4.480	2.000	17.440	29.380
-0.3	20.670	14.360	2.650	22.260	39.160	43.300	44.630	46.400	37.390	28.820	20.670	14.360	2.650	22.260	39.160
-0.2	31.940	19.760	1.700	23.160	44.420	56.720	62.400	65.600	55.220	43.880	31.940	19.760	1.700	23.160	44.420
-0.1	42.790	21.880	0.850	23.580	46.310	63.020	76.200	84.800	72.130	57.460	42.790	21.880	0.850	23.580	46.310
0.0	48.000	24.000	0.000	24.000	47.800	61.200	74.600	88.000	77.100	67.000	48.000	24.000	0.000	24.000	47.800
0.1	51.870	29.620	6.750	16.120	36.030	48.420	59.770	68.800	67.050	59.900	51.870	29.620	6.750	16.120	36.030
0.2	43.200	33.200	12.100	9.760	24.020	33.960	42.740	49.600	51.320	48.800	43.200	33.200	12.100	9.760	24.020
0.3	33.050	28.160	12.600	2.200	10.330	18.140	25.490	30.400	34.290	34.180	33.050	28.160	12.600	2.200	10.330
0.4	19.500	20.240	8.200	-8.480	-3.880	2.480	8.200	11.200	16.580	18.840	19.500	20.240	8.200	-8.480	-3.880
0.5	5.350	7.800	4.000	12.200	17.050	13.100	9.250	8.000	1.350	3.300	5.350	7.800	4.000	12.200	17.050
0.6	8.560	4.480	2.000	17.440	29.380	28.680	26.740	27.200	19.320	12.640	8.560	4.480	2.000	17.440	29.380
0.7	20.670	14.360	2.650	22.260	39.160	43.300	44.630	46.400	37.390	28.820	20.670	14.360	2.650	22.260	39.160

Table 6-6 Difference between SAD statistic for correct location & next minimum statistic for a window size of seven

Window Size : 7
Maximum Difference : 196.000
Minimum Difference : -8.320

Tran X	Translation In The Y Direction -0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
-0.7	72.430	32.500	7.850	30.480	43.280	56.320	69.220	82.600	81.210	78.500	72.430	32.500	7.850	30.480	43.280
-0.6	50.140	38.720	1.200	7.360	14.580	24.360	33.700	42.800	45.780	50.280	50.140	38.720	1.200	7.360	14.580
-0.5	25.350	28.500	4.750	15.800	13.300	7.600	2.100	3.000	10.450	20.100	25.350	28.500	4.750	15.800	13.300
-0.4	-0.480	-8.320	16.200	37.440	40.300	39.480	37.940	36.800	24.920	10.800	-0.480	-8.320	16.200	37.440	40.300
-0.3	22.810	9.720	29.150	58.360	64.820	70.360	74.160	76.600	60.610	41.540	22.810	9.720	29.150	58.360	64.820
-0.2	45.740	14.200	31.000	72.400	88.020	99.720	110.220	116.400	95.540	71.080	45.740	14.200	31.000	72.400	88.020
-0.1	64.940	17.060	31.100	79.180	109.920	126.480	144.120	156.200	129.170	97.660	64.940	17.060	31.100	79.180	109.920
0.0	70.200	19.600	31.000	81.600	119.300	144.600	171.300	196.000	158.000	120.800	70.200	19.600	31.000	81.600	119.300
0.1	73.970	26.340	21.750	69.840	96.900	118.640	139.660	162.200	147.090	121.000	73.970	26.340	21.750	69.840	96.900
0.2	73.020	29.400	14.300	53.520	71.220	88.200	104.460	122.400	116.000	105.280	73.020	29.400	14.300	53.520	71.220
0.3	72.430	32.500	7.850	30.480	43.280	56.320	69.220	82.600	81.210	78.500	72.430	32.500	7.850	30.480	43.280
0.4	50.140	38.720	1.200	7.360	14.580	24.360	33.700	42.800	45.780	50.280	50.140	38.720	1.200	7.360	14.580
0.5	25.350	28.500	4.750	15.800	13.300	7.600	2.100	3.000	10.450	20.100	25.350	28.500	4.750	15.800	13.300
0.6	-0.480	-8.320	16.200	37.440	40.300	39.480	37.940	36.800	24.920	10.800	-0.480	-8.320	16.200	37.440	40.300
0.7	22.810	9.720	29.150	58.360	64.820	70.360	74.160	76.600	60.610	41.540	22.810	9.720	29.150	58.360	64.820

Table 6-7 The difference between the SAD statistic for the correct location & next minimum for a window size of nine

Window Size : 9
Maximum Difference : 332.000
Minimum Difference : -12.500

Tran X	Translation In The Y Direction -0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
-0.7	120.610	111.100	41.600	25.260	40.420	63.660	92.020	128.900	128.030	125.760	120.610	111.100	41.600	25.260	40.420
-0.6	78.920	78.240	42.100	-3.440	2.060	12.760	33.460	61.200	67.040	76.560	78.920	78.240	42.100	-3.440	2.060
-0.5	35.050	42.500	28.000	33.300	36.800	36.300	25.000	6.500	6.150	24.600	35.050	42.500	28.000	33.300	36.800
-0.4	8.220	-7.240	-12.500	58.200	74.980	83.280	83.660	74.200	54.620	28.880	8.220	-7.240	-12.500	58.200	74.980
-0.3	49.410	26.380	2.050	73.440	110.580	128.620	141.960	141.900	115.110	81.800	49.410	26.380	2.050	73.440	110.580
-0.2	91.120	57.440	10.100	78.680	144.340	171.160	197.900	209.600	174.840	132.960	91.120	57.440	10.100	78.680	144.340
-0.1	130.590	87.540	16.650	76.880	146.720	198.540	249.980	277.300	232.490	180.560	130.590	87.540	16.650	76.880	146.720
0.0	165.900	113.400	21.000	76.600	145.700	198.200	258.900	332.000	283.100	222.400	165.900	113.400	21.000	76.600	145.700
0.1	169.000	119.140	31.100	61.820	113.660	159.580	209.000	264.300	242.410	213.720	169.000	119.140	31.100	61.820	113.660
0.2	157.580	123.200	37.000	49.600	78.860	113.680	150.660	196.600	188.180	172.400	157.580	123.200	37.000	49.600	78.860
0.3	120.610	111.100	41.600	25.260	40.420	63.660	92.020	128.900	128.030	125.760	120.610	111.100	41.600	25.260	40.420
0.4	78.920	78.240	42.100	-3.440	2.060	12.760	33.460	61.200	67.040	76.560	78.920	78.240	42.100	-3.440	2.060
0.5	35.050	42.500	28.000	33.300	36.800	36.300	25.000	6.500	6.150	24.600	35.050	42.500	28.000	33.300	36.800
0.6	8.220	-7.240	-12.500	58.200	74.980	83.280	83.660	74.200	54.620	28.880	8.220	-7.240	-12.500	58.200	74.980
0.7	49.410	26.380	2.050	73.440	110.580	128.620	141.960	141.900	115.110	81.800	49.410	26.380	2.050	73.440	110.580

Table 6-8 The difference between the SAD statistic for the correct location & next minimum for a window size of eleven

Window Size : 11
Maximum Difference : 513.000
Minimum Difference : -16.360

Translation In The Y Direction

Tran X	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
-0.7	164.160	143.040	28.750	30.520	59.500	98.160	148.720	209.400	199.360	180.320	164.160	143.040	28.750	30.520	59.500
-0.6	102.340	98.360	16.100	-16.360	-3.000	19.000	57.760	106.200	107.420	107.040	102.340	98.360	16.100	-16.360	-3.000
-0.5	39.200	46.200	8.250	64.000	64.500	57.000	32.800	3.000	14.200	31.000	39.200	46.200	8.250	64.000	64.500
-0.4	22.380	4.200	33.600	109.480	123.680	130.440	123.520	100.200	78.900	47.120	22.380	4.200	33.600	109.480	123.680
-0.3	81.760	51.720	35.950	154.080	178.960	200.640	213.080	203.400	170.800	123.240	81.760	51.720	35.950	154.080	178.960
-0.2	139.380	94.760	27.400	158.240	229.140	266.760	299.520	306.600	260.140	196.720	139.380	94.760	27.400	158.240	229.140
-0.1	193.140	119.320	19.550	158.420	233.590	310.940	378.780	409.800	344.080	264.360	193.140	119.320	19.550	158.420	233.590
0.0	240.900	132.000	13.500	159.000	234.800	315.000	410.800	513.000	415.900	324.000	240.900	132.000	13.500	159.000	234.800
0.1	241.570	142.600	4.750	121.520	180.220	250.560	330.480	415.800	371.400	315.560	241.570	142.600	4.750	121.520	180.220
0.2	220.980	146.960	18.600	77.000	122.280	176.600	239.760	312.600	289.380	251.040	220.980	146.960	18.600	77.000	122.280
0.3	164.160	143.040	28.750	30.520	59.500	98.160	148.720	209.400	199.360	180.320	164.160	143.040	28.750	30.520	59.500
0.4	102.340	98.360	16.100	-16.360	-3.000	19.000	57.760	106.200	107.420	107.040	102.340	98.360	16.100	-16.360	-3.000
0.5	39.200	46.200	8.250	64.000	64.500	57.000	32.800	3.000	14.200	31.000	39.200	46.200	8.250	64.000	64.500
0.6	22.380	4.200	33.600	109.480	123.680	130.440	123.520	100.200	78.900	47.120	22.380	4.200	33.600	109.480	123.680
0.7	81.760	51.720	35.950	154.080	178.960	200.640	213.080	203.400	170.800	123.240	81.760	51.720	35.950	154.080	178.960

Table 6-9 The difference between the SAD statistic for the correct location & the next minimum statistic for a window size of thirteen

Window Size : 13
Maximum Difference : 727.000
Minimum Difference : -16.120

Translation In The Y Direction

Tran X	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
-0.7	234.850	209.540	71.400	54.020	93.520	141.420	209.080	288.100	277.250	257.240	234.850	209.540	71.400	54.020	93.520
-0.6	144.860	140.920	62.000	-16.120	2.880	29.800	79.040	141.800	145.380	147.200	144.860	140.920	62.000	-16.120	2.880
-0.5	49.750	63.100	23.000	88.900	87.400	79.100	50.600	4.500	12.050	35.800	49.750	63.100	23.000	88.900	87.400
-0.4	43.160	12.480	21.200	158.760	176.400	185.680	180.200	150.800	121.160	78.640	43.160	12.480	21.200	158.760	176.400
-0.3	132.450	84.180	28.350	198.660	260.380	289.020	306.520	297.100	252.470	190.920	132.450	84.180	28.350	198.660	260.380
-0.2	217.020	150.120	24.100	205.520	328.080	387.080	429.080	443.400	380.900	297.760	217.020	150.120	24.100	205.520	328.080
-0.1	296.830	173.360	19.950	213.260	338.300	439.280	543.960	589.700	501.890	395.560	296.830	173.360	19.950	213.260	338.300
0.0	369.800	187.800	16.000	219.800	346.400	451.000	582.200	727.000	606.200	483.600	369.800	187.800	16.000	219.800	346.400
0.1	377.890	212.760	19.500	173.880	266.840	355.820	467.240	580.700	524.810	449.240	377.890	212.760	19.500	173.880	266.840
0.2	314.480	229.960	50.000	123.920	182.720	251.120	339.000	434.400	406.720	359.520	314.480	229.960	50.000	123.920	182.720
0.3	234.850	209.540	71.400	54.020	93.520	141.420	209.080	288.100	277.250	257.240	234.850	209.540	71.400	54.020	93.520
0.4	144.860	140.920	62.000	-16.120	2.880	29.800	79.040	141.800	145.380	147.200	144.860	140.920	62.000	-16.120	2.880
0.5	49.750	63.100	23.000	88.900	87.400	79.100	50.600	4.500	12.050	35.800	49.750	63.100	23.000	88.900	87.400
0.6	43.160	12.480	21.200	158.760	176.400	185.680	180.200	150.800	121.160	78.640	43.160	12.480	21.200	158.760	176.400
0.7	132.450	84.180	28.350	198.660	260.380	289.020	306.520	297.100	252.470	190.920	132.450	84.180	28.350	198.660	260.380

Table 6-10 The difference between the SAD statistic for the correct location & the next minimum statistic for a window size of fifteen

Window Size : 15
Maximum Difference : 939.000
Minimum Difference : 3.400

Translation In The Y Direction

Tran X	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
-0.7	321.240	279.480	74.100	98.080	152.630	217.340	304.930	402.200	388.380	355.720	321.240	279.480	74.100	98.080	152.630
-0.6	206.560	192.440	85.500	5.480	33.260	71.600	134.740	210.600	217.920	213.800	206.560	192.440	85.500	5.480	33.260
-0.5	83.800	95.000	34.250	89.000	84.550	72.100	35.150	19.000	44.500	70.200	83.800	95.000	34.250	89.000	84.550
-0.4	36.560	3.400	23.400	181.160	200.240	212.600	205.600	172.600	128.920	77.640	36.560	3.400	23.400	181.160	200.240
-0.3	153.120	96.400	37.950	268.080	311.270	347.900	372.770	364.200	299.900	222.600	153.120	96.400	37.950	268.080	311.270
-0.2	263.200	179.640	34.100	279.800	417.860	478.000	534.580	555.800	467.600	361.160	263.200	179.640	34.100	279.800	417.860
-0.1	362.600	231.460	29.350	290.240	467.510	598.660	685.090	747.400	626.340	487.560	362.600	231.460	29.350	290.240	467.510
0.0	451.600	247.800	25.000	297.800	482.300	619.200	789.900	939.000	760.000	597.800	451.600	247.800	25.000	297.800	482.300
0.1	456.690	276.160	15.300	245.680	379.870	497.380	642.010	785.400	709.980	597.220	456.690	276.160	15.300	245.680	379.870
0.2	422.280	294.360	49.000	191.480	270.560	360.440	475.200	593.800	555.800	488.760	422.280	294.360	49.000	191.480	270.560
0.3	321.240	279.480	74.100	98.080	152.630	217.340	304.930	402.200	388.380	355.720	321.240	279.480	74.100	98.080	152.630
0.4	206.560	192.440	85.500	5.480	33.260	71.600	134.740	210.600	217.920	213.800	206.560	192.440	85.500	5.480	33.260
0.5	83.800	95.000	34.250	89.000	84.550	72.100	35.150	19.000	44.500	70.200	83.800	95.000	34.250	89.000	84.550
0.6	36.560	3.400	23.400	181.160	200.240	212.600	205.600	172.600	128.920	77.640	36.560	3.400	23.400	181.160	200.240
0.7	153.120	96.400	37.950	268.080	311.270	347.900	372.770	364.200	299.900	222.600	153.120	96.400	37.950	268.080	311.270

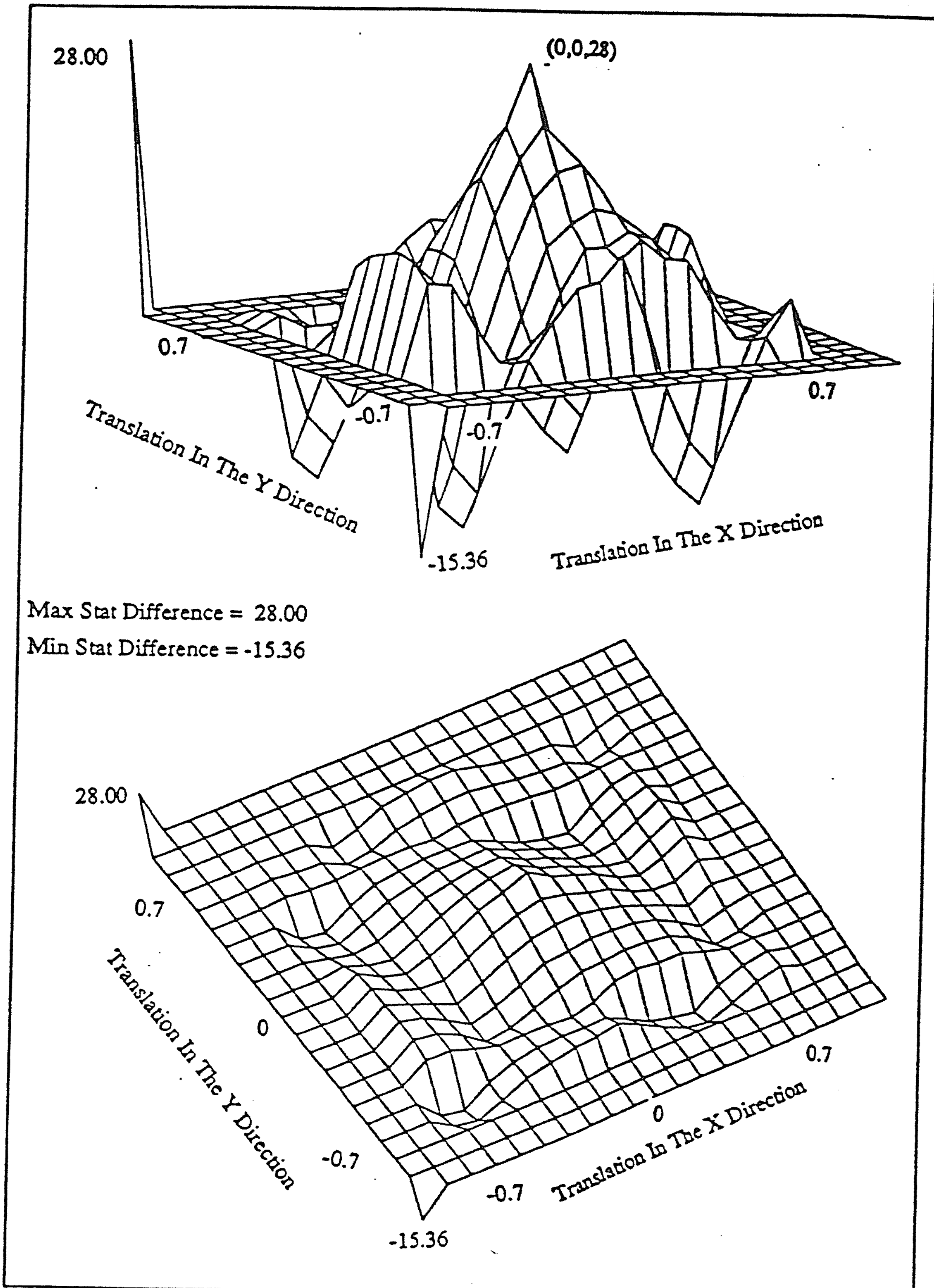


Figure 6-4a Mesh plot of difference between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of three

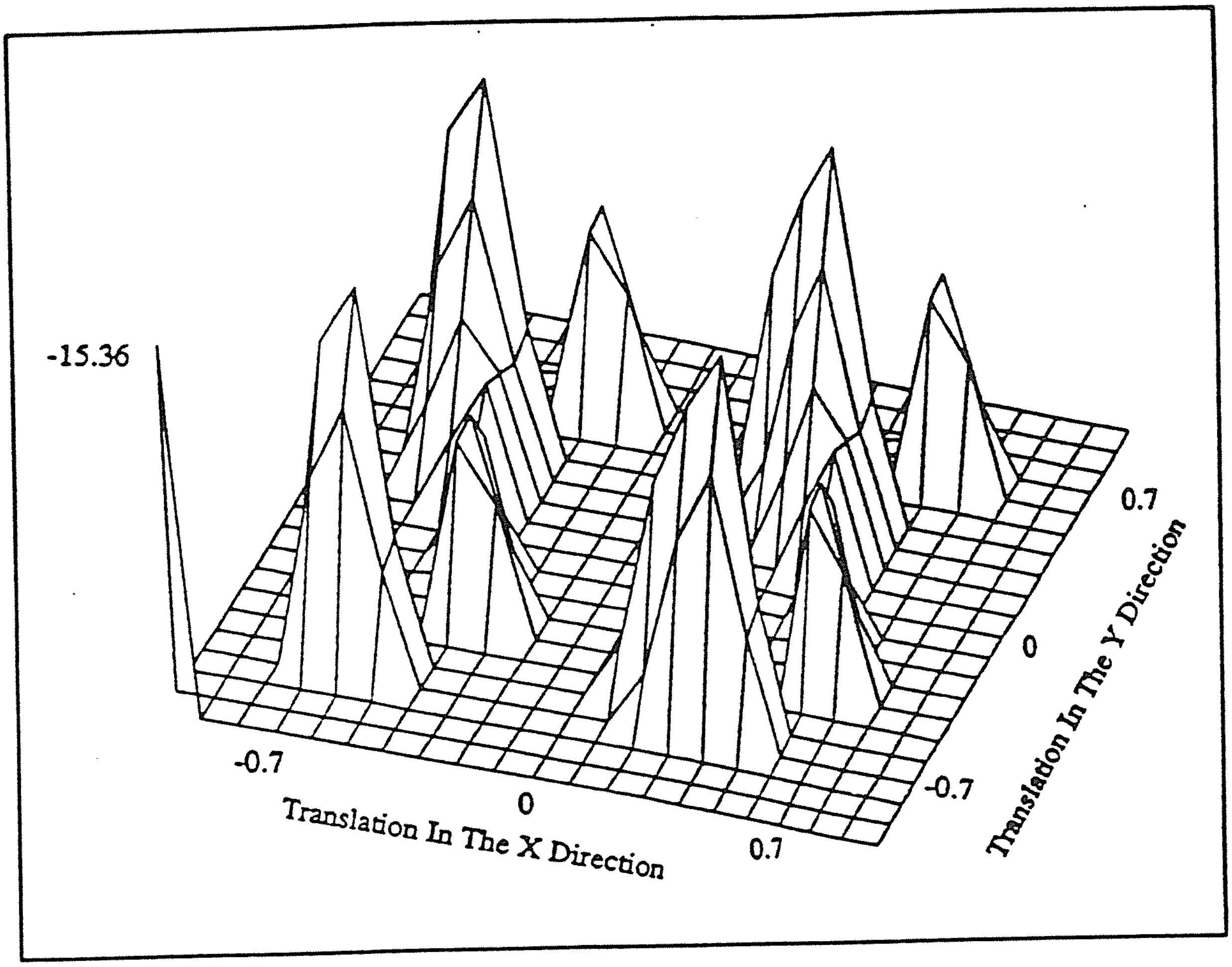


Figure 6-4b   Mesh plot of the negative differences between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of three

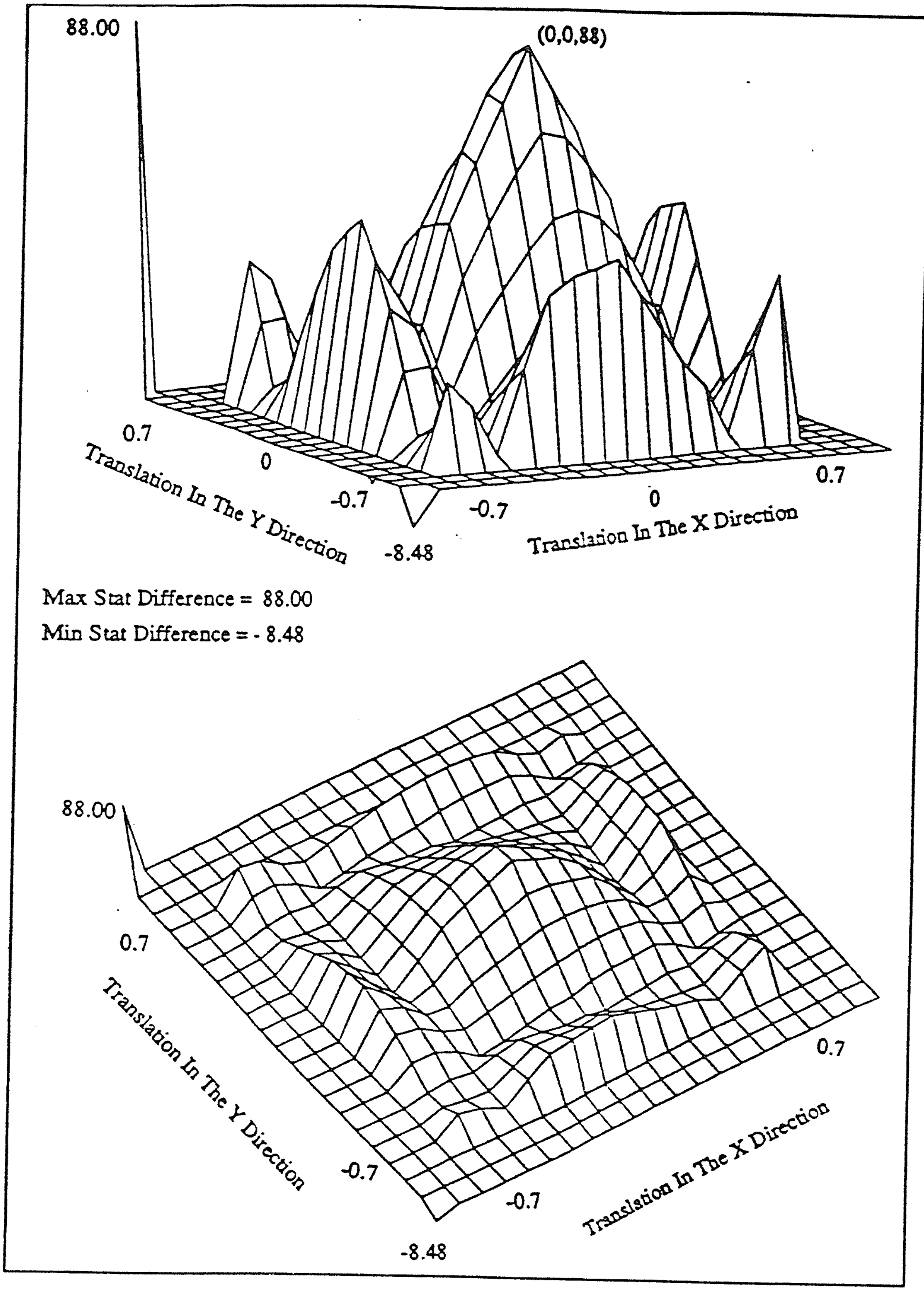


Figure 6-5a Mesh plot of difference between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of five

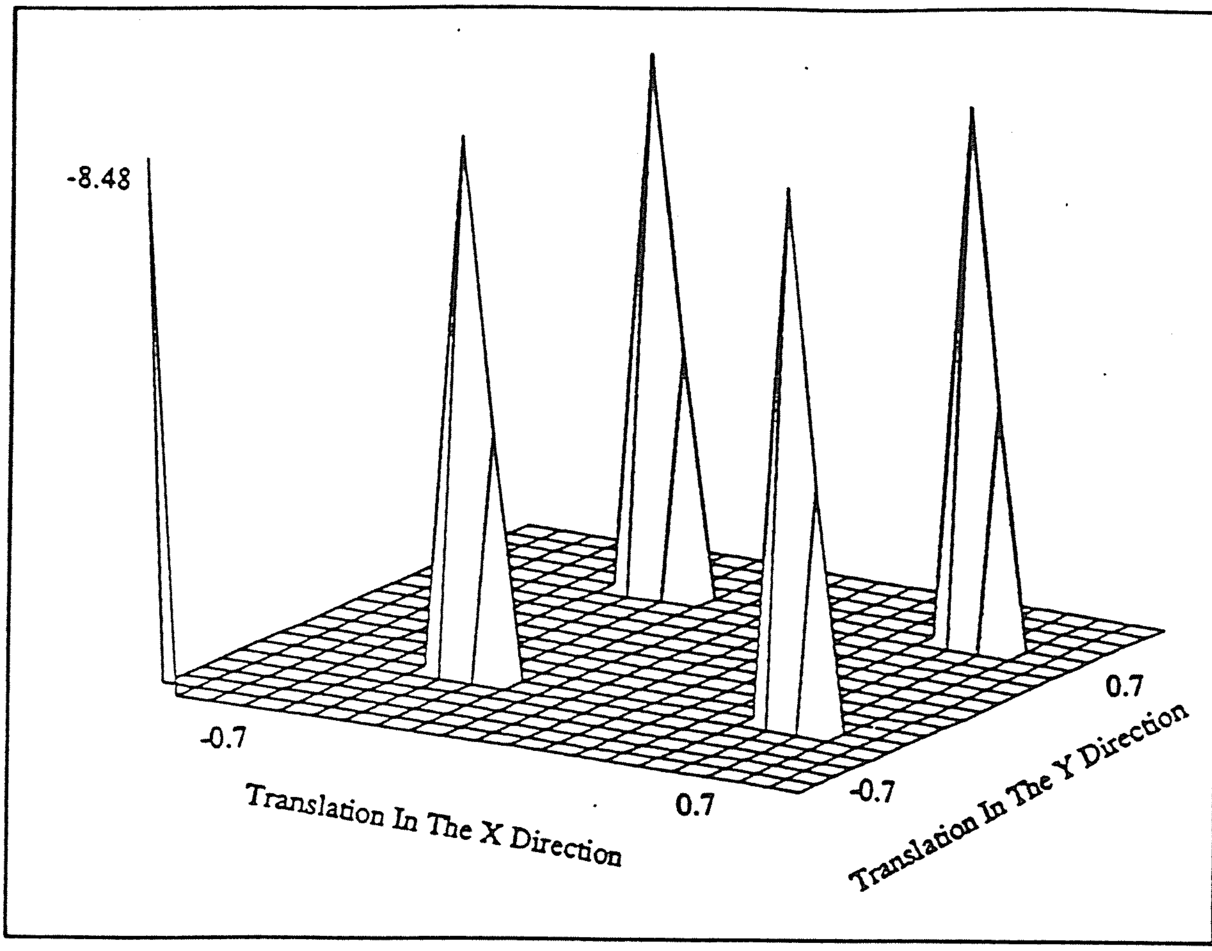


Figure 6-5b Mesh plot of the negative differences between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of five

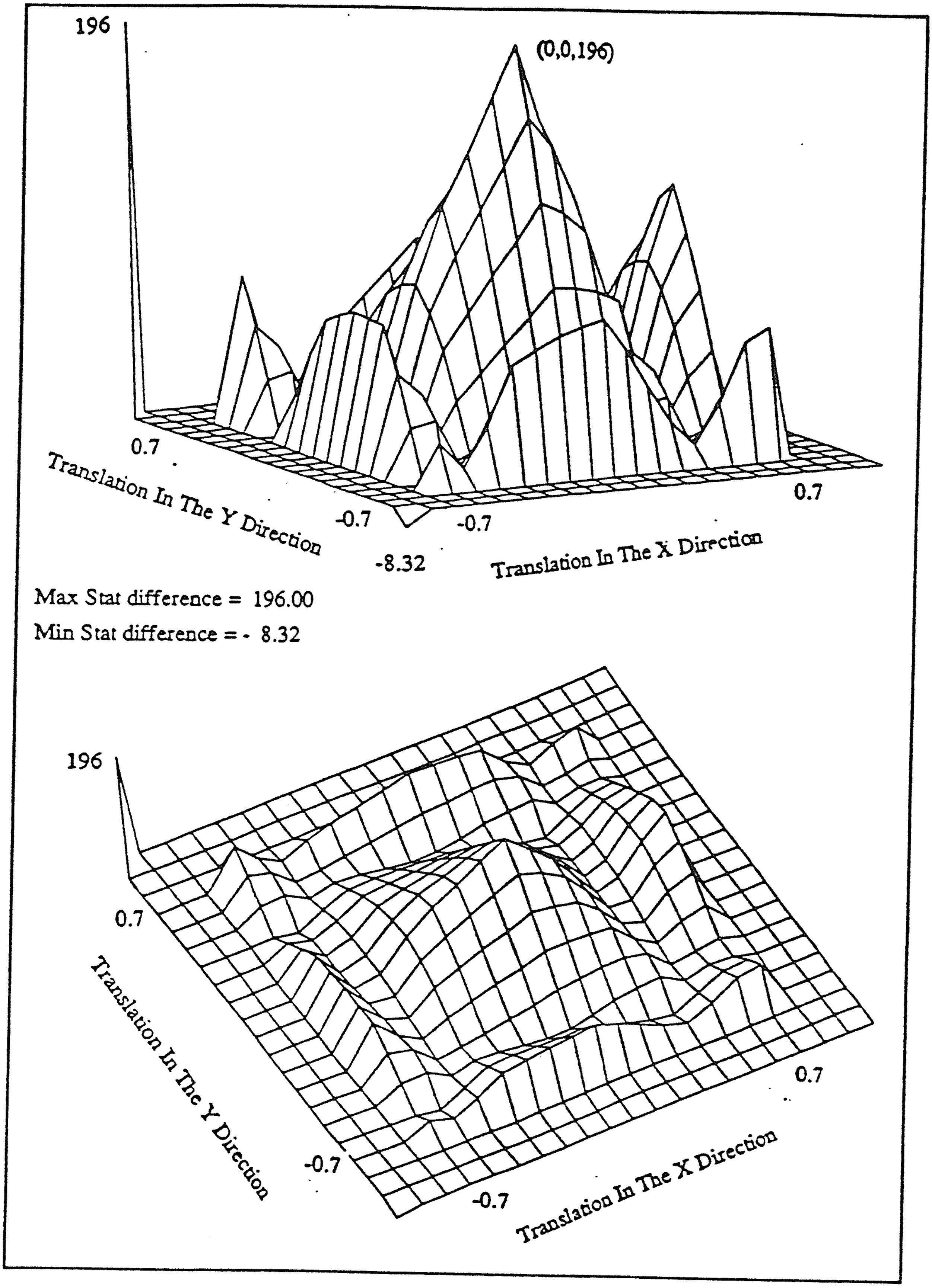


Figure 6-6a Mesh plot of difference between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of seven

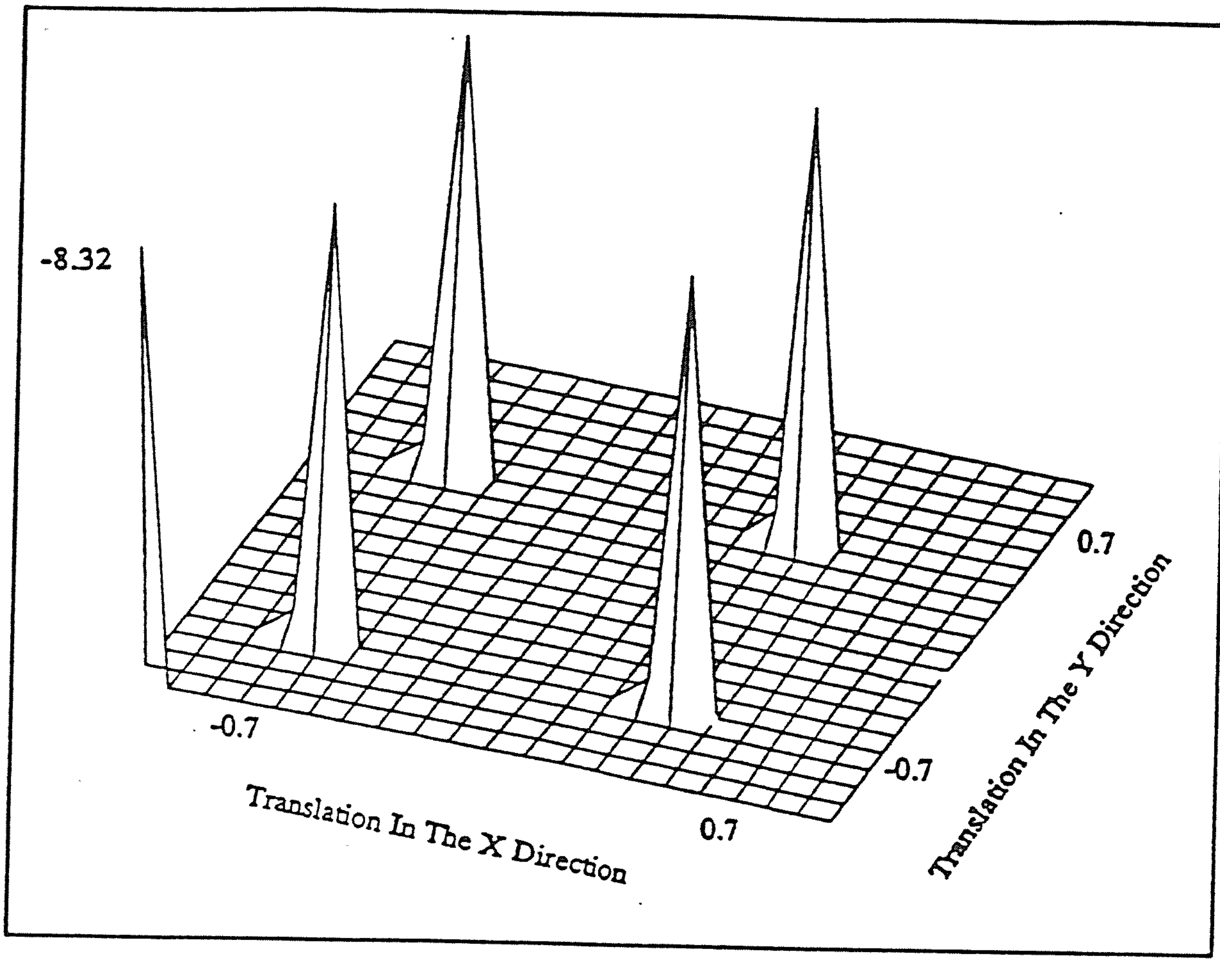


Figure 6-6b Mesh plot of the negative differences between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of seven

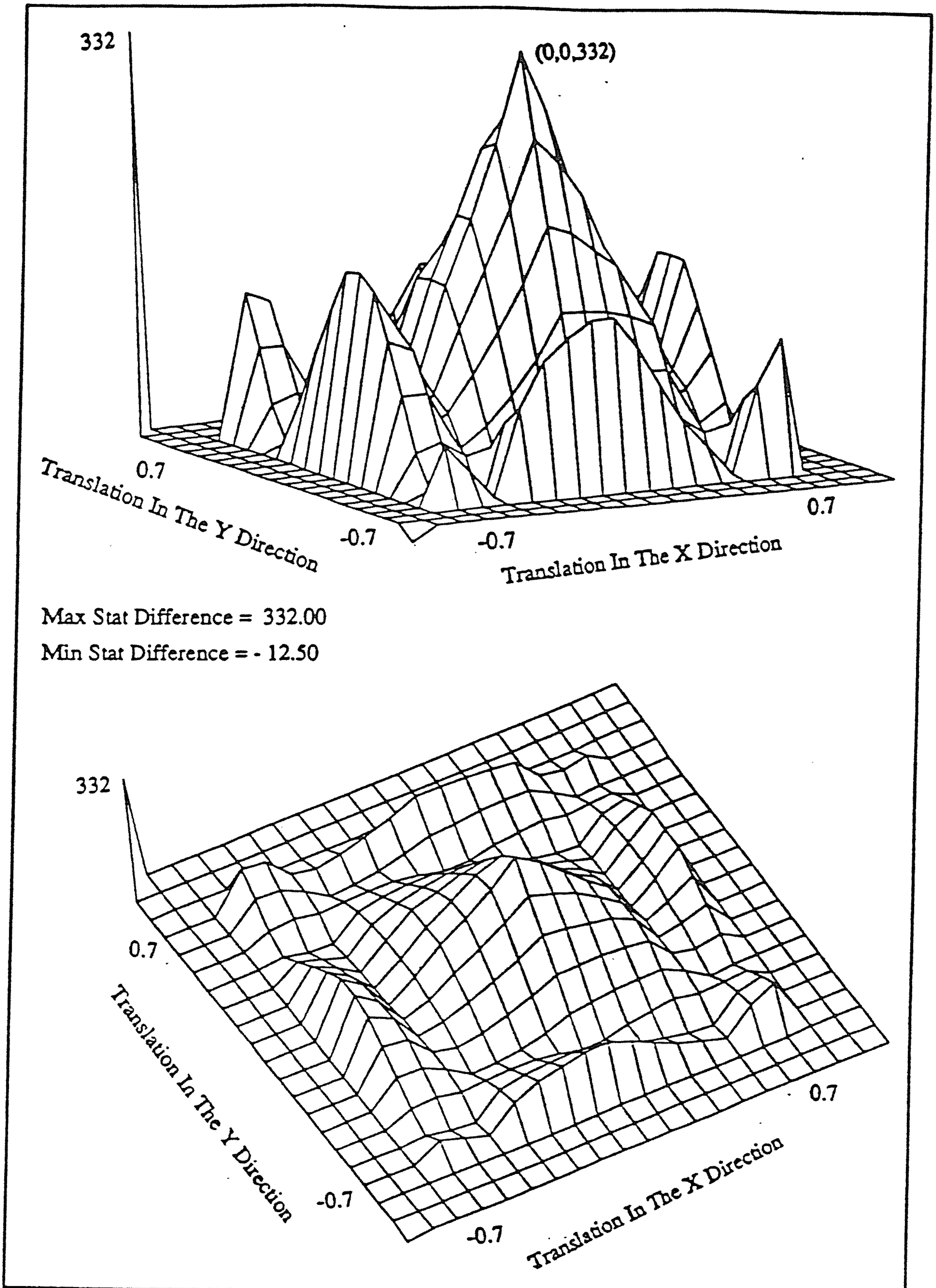


Figure 6-7a Mesh plot of difference between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of nine

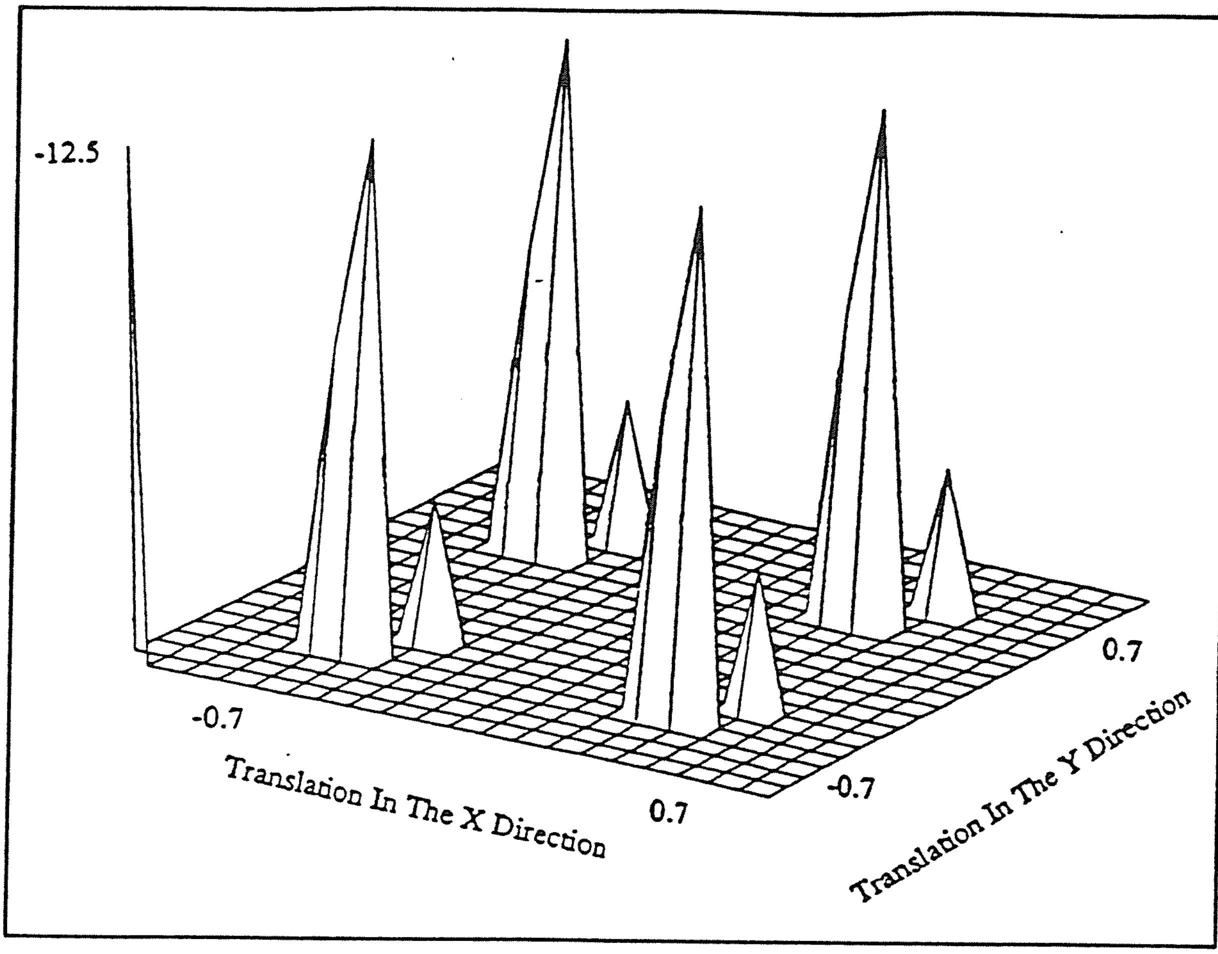


Figure 6-7b Mesh plot of the negative differences between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of nine

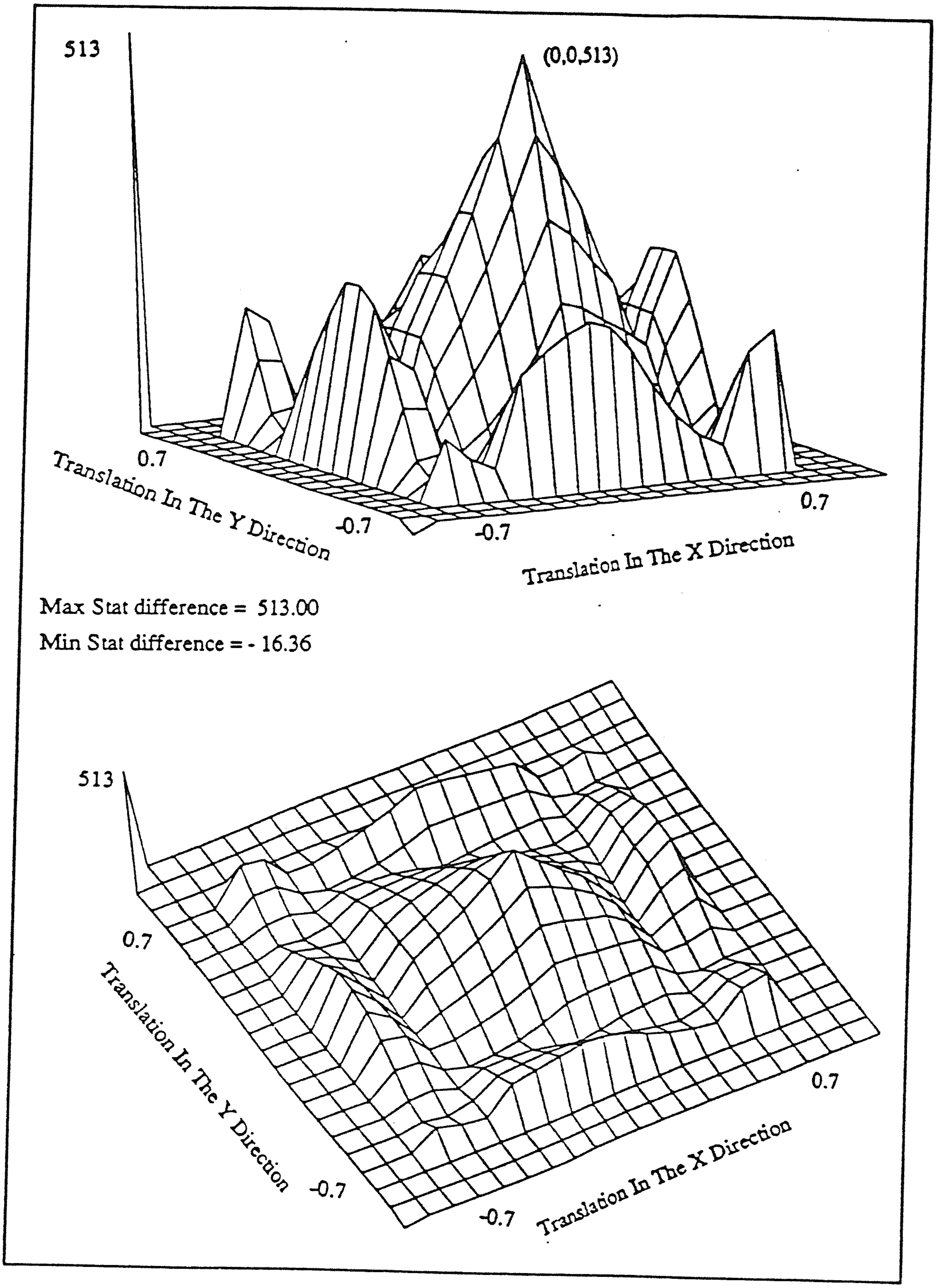


Figure 6-8a Mesh plot of difference between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of eleven

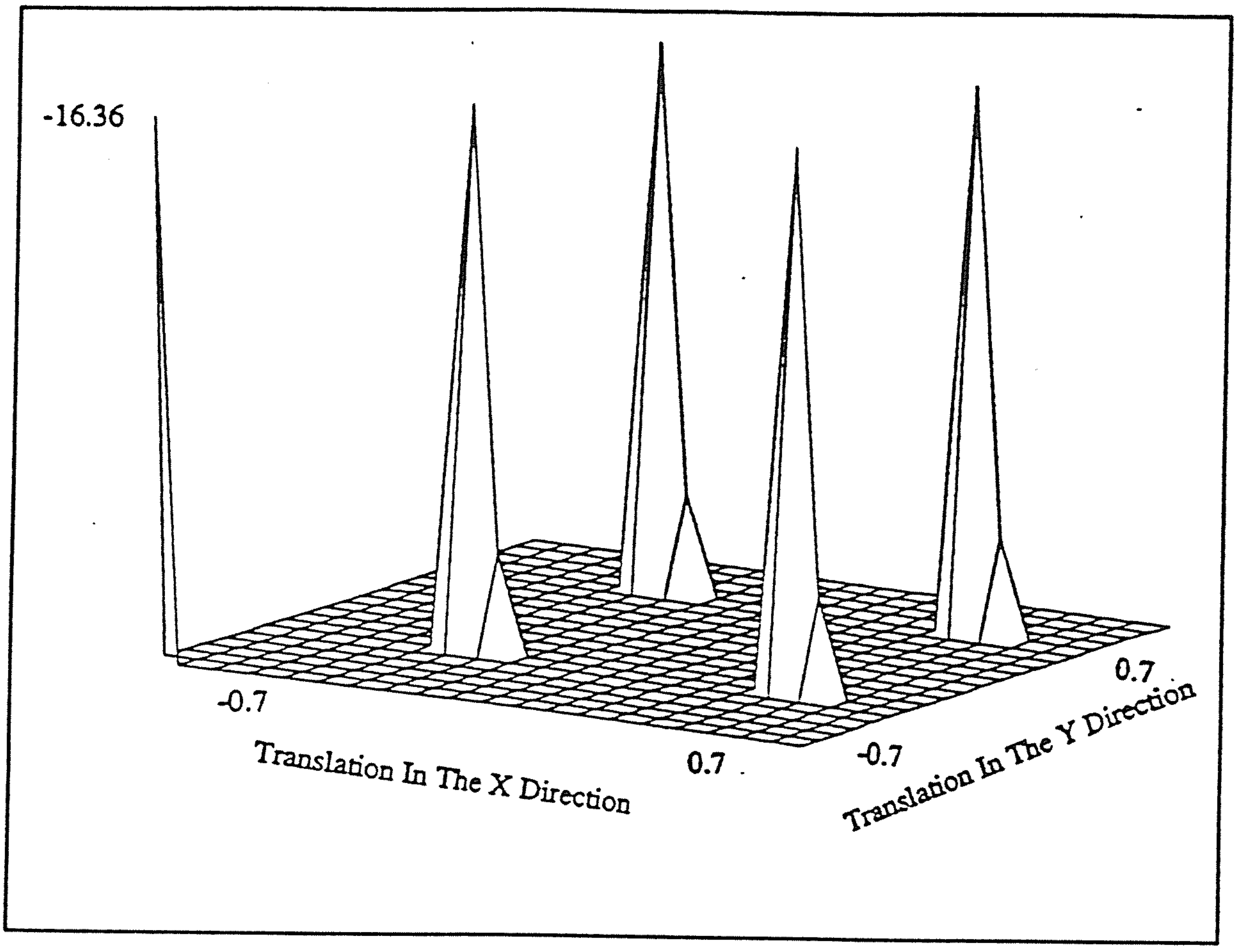


Figure 6-8b Mesh plot of the negative differences between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of eleven

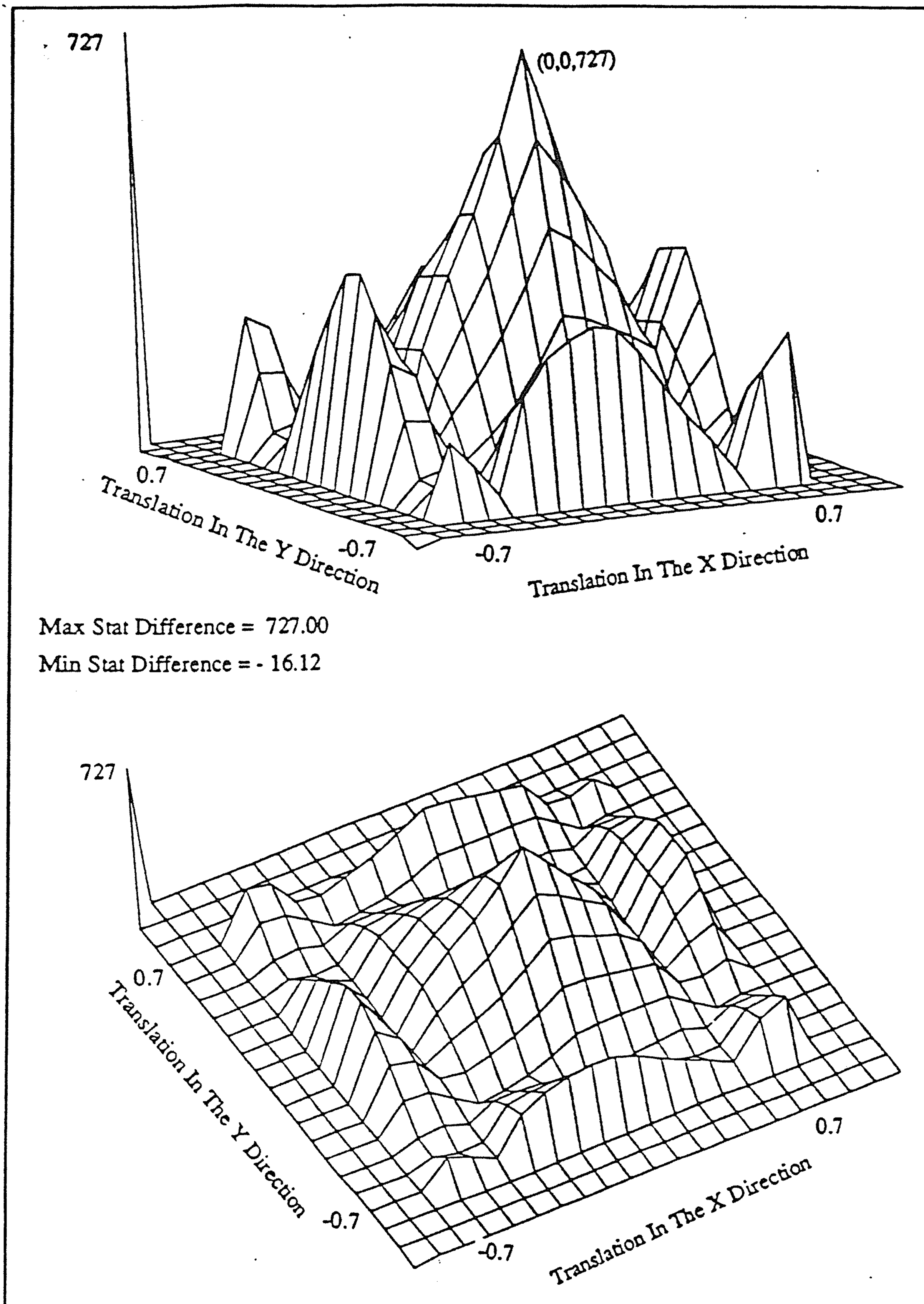


Figure 6-9a Mesh plot of difference between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of thirteen

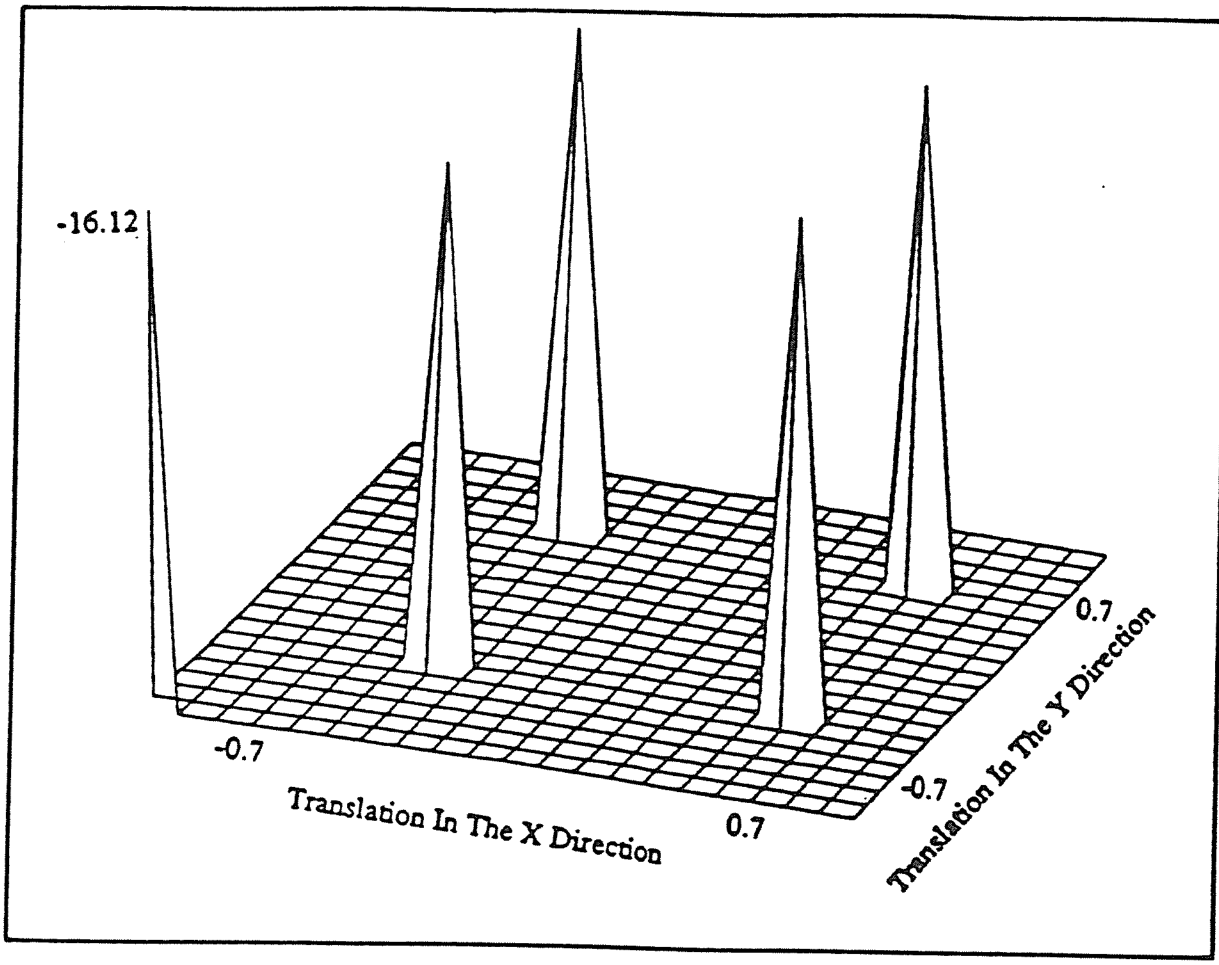


Figure 6-9b Mesh plot of the negative differences between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of thirteen

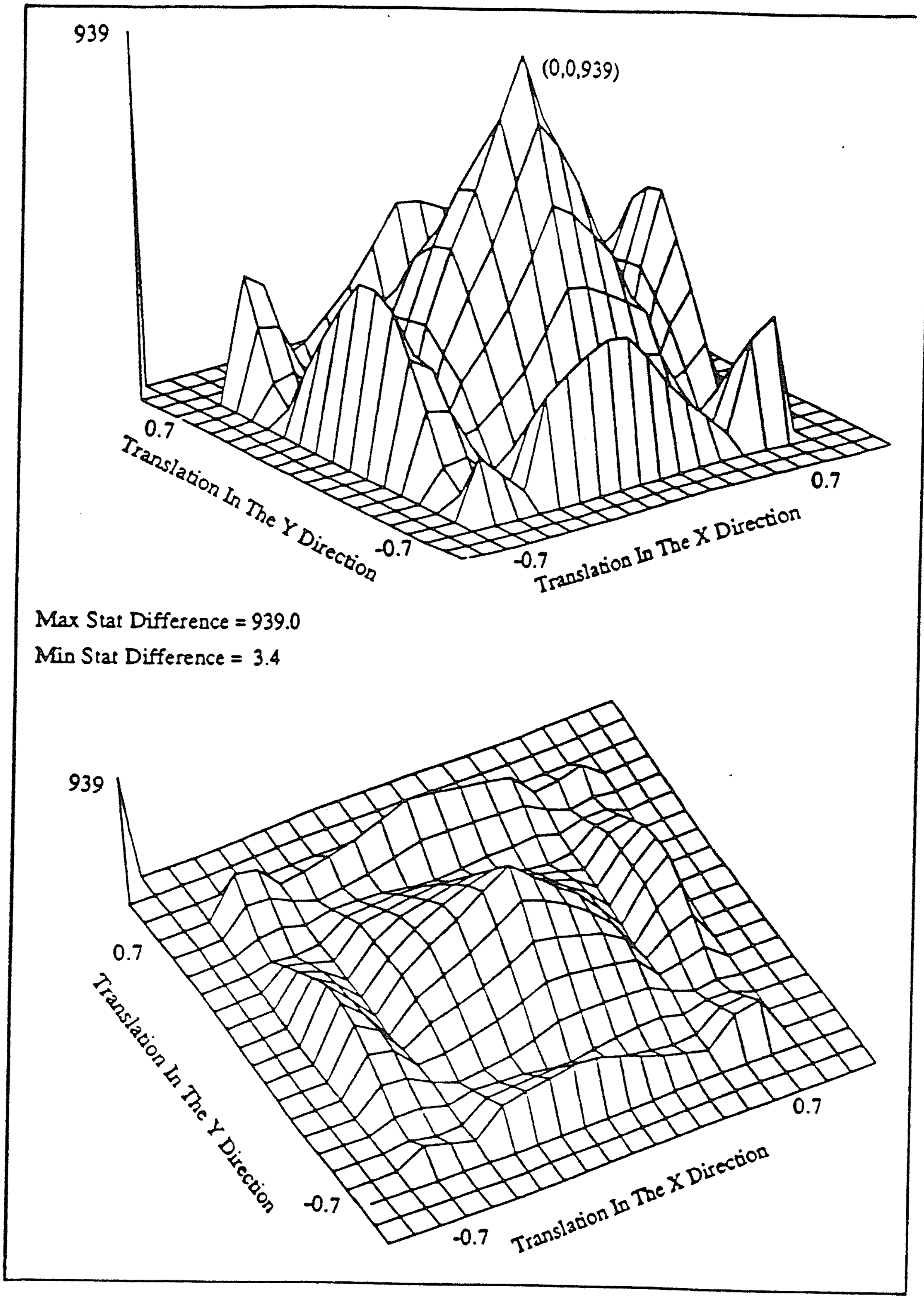


Figure 6-10 Mesh plot of difference between the SAD statistic for the correct location and next minimum for translations in the X and Y direction between -0.7 and 0.7 pixels for a window size of fifteen

6.3 Effect of results on current prototype

The algorithm in the current prototype uses a window size of eleven. When a translation around half a pixel occurs in the x and y direction the algorithm often selected an incorrect location. This occurs because the correct pixel location does not line up with the discretised 'grid', but is actually half way between the nearest two or four pixels. The nearest whole pixel location will not give as good a statistic as the actual location, increasing the chance of other locations with low SAD statistics being chosen.

There are three solutions to this problem. The first two solutions are preventative and they are to either use a bigger window size or include sub-pixel location calculations in the algorithm. The other solution is to perform a regular automatic or manual check on the tracking results to enable correction if a point has strayed from the correct location.

As can be seen from the results in this chapter the window size has serious affects on the results and must be chosen carefully.

Chapter 7
Selection of the window size used by the SAD statistic

The selection of the window size to be used by the SAD statistic has a significant effect on the performance of a statistic. When selecting window sizes consideration has to be given to the following :

1. If the window is too small there will not be sufficient detail to provide a significant comparison of the two images.

2. A very large window size has two major drawbacks:
   i) There may be significant movement between components of the image within the overall image, so that no single measure of translation or rotation exists.
   For example two vertebra may be within a window of the image yet they are moving in different directions, i.e. the movement of a window cannot be summarised by one vector and rotation because there is independent movement of items within that window.

   ii) The computation time increases as the square of the window size increases, and rapidly becomes excessive.

3. Rotation becomes increasingly significant as the window size is increased.

The 'ideal' window size will be dependant on the application, the statistic chosen and the method used to track. For tracking points through a sequence of X-rayed

Cervical spine movement using the sum of absolute difference (SAD) statistic the window size can be selected by examining how the window size effects the performance of the statistic when the frame being tracked to has been subjected to rotation and translation.

Table 7-1 summarises the results found in chapter 5 and 6. For rotation the level of confidence with which the correct position was chosen is given by the difference between the SAD statistic for the correct location and the next minimum when tracking from a frame to the same frame rotated by various amounts. The effect of sub-pixel translation can be shown by recording the number of incorrect locations chosen by the statistic when the frame being tracked to is the same frame translated between -0.7 and 0.7 pixels in the x and y direction.

Table 7-1 Effect of rotation and translation on performance of SAD statistic for various window sizes

Window Size	Rotation (degrees) at which there is a 90% decrease in confidence	Number of incorrect locations chosen when frame is translated +0.7 to -0.7 pixels
3	30	76
5	20 - 25	8
7	15 - 20	8
9	15	12
11	10 - 15	8
13	8 - 9	4
15	8 - 9	0

From this information a window size of eleven was chosen as the effects of rotation was easily within the tolerance level that was required of around 3° as well as being acceptably robust to sub-pixel translations particularly when it has been shown that if a SAD algorithm is used some compensation for sub-pixel translation must be employed. A window size of eleven will safely be within the defined limits if sub-pixel translation is compensated for. It allows for an adequate area of comparison while not being large enough to incorporate more than a few pixels of a neighbouring independent vertebra and is not excessively computationaly intensive (involves the summation of 121 short integer subtractions for a window comparison). For the rest of this report a window size of eleven is used for all SAD algorithm results.

Chapter 8
Allowing for sub-pixel translation

When the sum of absolute difference (SAD) statistic is used to find the new position of a point the value of the best (lowest) statistic corresponds to a window centred on the position where the new location of the point will be taken to be. Often the difference between the lowest statistic and the next lowest statistic is only one or two and usually less than ten. For a window size of eleven the difference between the two lowest statistics may be less than 5% of the statistic value. As shown in Table 5-3 the two lowest statistics often correspond to adjacent pixels and this led to the notion that the correct position of the tracked point may be between the pixels that had the lowest statistics.

When conceptualising the tracking procedure there are three 'views' to be considered. They are :

i the real object that is being filmed,
ii the real camera view as recorded on video,
iii the mathematically calculated view.

What the algorithm is attempting to do is track a point in the camera view between consecutive frames. This relates to a point on a real object which is moving continuously but because of the camera and the frame grabber the frames are effectively stills taken at discrete time intervals.

Given a first frame, a translating program can be used to calculate what the camera is expected to see if the object had translated by a certain known amount. This calculated view is called the 'mathematically calculated view'. What the camera actually sees is called the 'real camera view'.

To track a point on the 'real object' a pixel is moused on the 'real camera view'. From this point there are two possible ways of going about sub-pixel tracking, the backward or forward mapping method.

8.1 The backward mapping method

The 'backward' mapping method is to take the 'real camera view' that is being tracked to (usually the next in sequence after the frame we have specified the point in) and to translate this by different amounts and then search for the best match of the pixel window centred on the point that was specified in the first frame. To compare the windows the SAD statistic is used. When the minimum is found the position of the point tracked to the second frame is taken as the coordinate of the central pixel in the best match window in the translated second frame corrected by the translation undergone by the second frame.

8.2 The forward mapping method

The 'forward' mapping method is to take the 'real camera view', in which the point has been specified, and to translate this by different amounts and find the best match of the pixel window centred on the coordinates of the point specified in these translated views in the frame we are tracking to. The actual position of the tracked point in the second frame can be taken as the coordinates of the central pixel in the window of the best match in the second frame corrected by the translation undergone by the first frame.

8.3 Definition of six basic sub-pixel tracking strategies

Six strategies were defined and then tested on a mathematically simulated video of twenty frames. The definitions of the six strategies were :

Strategy 0 - no sub-pixel tracking, tracked whole pixel to whole pixel.

Strategy 1 - equivalent to the backward method. The frame being tracked to is translated and matched back to the frame the point is already defined in. Resolution is 0.125 pixels.

Strategy 2 - strategy 1, i.e. backward method, but resolution is 0.5 pixels.

Strategy 3 - equivalent to the forward method. The frame being tracked from, i.e. the frame the point is already defined in, is translated and matched to the next frame in the sequence. Resolution is 0.125 pixels.

Strategy 4 - strategy 3, i.e. forward method, but resolution is 0.5 pixels.

Strategy 5 - the frame that the points were originally moused in is repeatedly translated to match the frame being tracked to. Resolution is 0.125.

Strategy 6 - strategy 5, but resolution is 0.5 pixels.

8.4 Performance of the strategies

A series of images was simulated, by a computer program, of an X-ray of the cervical region of a spine moving at two pixels per frame at an angle of 23 degrees. The window diameter was set at eleven pixels and each strategy was run on the picture series. The error, ERR, used to measure the performance of each strategy was calculated for each frame as the distance from the correct location of the point, $(X,Y)_{CORR}$, to the actual location tracked to, $(X,Y)_{TRAC}$:

$$ERR = \sqrt{(X_{CORR}-X_{TRAC})^2 + (Y_{CORR}-Y_{TRAC})^2}$$

The performance of each strategy is shown in Tables 8-1 to 8-7 and graphed in Figure 8-1. The accumulated sum of these errors are graphed in Figure 8-2. Table 8-8 shows the final location tracked in the twentieth frame, the radial displacement from the correct location and the accumulated error over the twenty frames. From Table 8-8 it can be seen that strategy five and six (i.e. tracking from the moused frame) performed the best. However these strategies can only be used on real patient data if rotation is taken into account. This is possible but including rotation calculations in the tracking algorithm greatly increases the computational requirements. The other strategy which also performed well was strategy three. This strategy tracked the moused point to within a pixel for the first nineteen frames. Although strategy one performed well the accumulated error was over 200% greater than strategy three. Strategy two and four, which were strategy one and three to 0.5 resolution instead of 0.125 pixel resolution, performed no better than strategy one, which is the whole pixel tracking algorithm.

From these results it appears strategy three should be implemented for the sub-pixel mapping prototype. An alternative algorithm that could give good results is strategy five but with the alteration that every few frames the original moused frame is rotated to realign with the current frame and tracking then proceeds using the new rotated moused frame.

For a more realistic result noise must be added to the simulated spinal motion video.

Table 8-1 Tracking results using strategy zero

frame	dist	neg	should be (real)		should be		tracked to		ind Err	Sum Err
0	0	23	54	48	54	48	54	48	0	0
1	2	23	55.8410	48.7815	56	49	56	49	0.270	0.270
2	4	23	57.6820	49.5629	58	50	58	50	0.541	0.811
3	6	23	59.5230	50.3444	60	50	60	51	0.811	1.622
4	8	23	61.3640	51.1258	61	51	62	52	1.081	2.703
5	10	23	63.2050	51.9073	63	52	64	53	1.351	4.054
6	12	23	65.0461	52.6888	65	53	66	54	1.621	5.675
7	14	23	66.8871	53.4702	67	53	68	55	1.892	7.567
8	16	23	68.7281	54.2517	69	54	70	56	2.162	9.729
9	18	23	70.5691	55.0332	71	55	72	57	2.432	12.16
10	20	23	72.4101	55.8146	72	56	74	58	2.703	14.86
11	22	23	74.2511	56.5961	74	57	76	59	2.973	17.84
12	24	23	76.0921	57.3775	76	57	78	60	3.243	21.08
13	26	23	77.9331	58.159	78	58	80	61	3.513	24.59
14	28	23	79.7741	58.9405	80	59	82	62	3.784	28.38
15	30	23	81.6151	59.7219	82	60	84	63	4.054	32.43
16	32	23	83.4562	60.5034	83	61	86	64	4.324	36.75
17	34	23	85.2972	61.2849	85	61	88	65	4.594	41.35
18	36	23	87.1382	62.0663	87	62	90	66	4.865	46.21
19	38	23	88.9792	62.8478	89	63	92	67	5.135	51.35
20	40	23	90.8202	63.6292	91	64	94	68	5.405	56.75

Table 8-2 Tracking results using strategy one

frame	dist	deg	should be(real)		should be		tracked to		Ind Err	Sum Err
0	0	23	54	48	54	48	54	48	0	0
1	2	23	55.841	48.7815	56	49	56	49	0.27	0.27
2	4	23	57.682	49.5629	58	50	58	50	0.541	0.811
3	6	23	59.523	50.3444	60	50	59.875	50.75	0.537	1.348
4	8	23	61.364	51.1258	61	51	61.625	51.5	0.456	1.804
5	10	23	63.205	51.9073	63	52	63.5	52.375	0.553	2.357
6	12	23	65.0461	52.6888	65	53	65.375	53.25	0.65	3.007
7	14	23	66.8871	53.4702	67	53	67.375	54.25	0.92	3.927
8	16	23	68.7281	54.2517	69	54	69.25	55.125	1.017	4.945
9	18	23	70.5691	55.0332	71	55	71.125	56	1.115	6.06
10	20	23	72.4101	55.8146	72	56	73	56.875	1.213	7.273
11	22	23	74.2511	56.5961	74	57	74.875	57.75	1.312	8.585
12	24	23	76.0921	57.3775	76	57	76.75	58.5	1.301	9.886
13	26	23	77.9331	58.159	78	58	78.625	59.25	1.292	11.18
14	28	23	79.7741	58.9405	80	59	80.625	60.125	1.458	12.64
15	30	23	81.6151	59.7219	82	60	82.625	61.125	1.729	14.37
16	32	23	83.4562	60.5034	83	61	84.5	62	1.825	16.19
17	34	23	85.2972	61.2849	85	61	86.375	62.75	1.819	18.01
18	36	23	87.1382	62.0663	87	62	88.25	63.5	1.814	19.82
19	38	23	88.9792	62.8478	89	63	90.125	64.375	1.909	21.73
20	40	23	90.8202	63.6292	91	64	92.125	65.375	2.18	23.91

Table 8-3 Tracking results using strategy two

frame	dist	deg	should be (real)		should be		tracked to		ind Err	Sum Err
0	0	23	54	48	54	48	54	48	0	0
1	2	23	55.841	48.7815	56	49	56	49	0.27	0.27
2	4	23	57.682	49.5629	58	50	58	50	0.541	0.811
3	6	23	59.523	50.3444	60	50	60	51	0.811	1.622
4	8	23	61.364	51.1258	61	51	62	52	1.081	2.703
5	10	23	63.205	51.9073	63	52	64	53	1.351	4.054
6	12	23	65.0461	52.6888	65	53	66	54	1.621	5.675
7	14	23	66.8871	53.4702	67	53	68	55	1.892	7.567
8	16	23	68.7281	54.2517	69	54	70	56	2.162	9.729
9	18	23	70.5691	55.0332	71	55	72	57	2.432	12.16
10	20	23	72.4101	55.8146	72	56	74	58	2.703	14.86
11	22	23	74.2511	56.5961	74	57	76	59	2.973	17.84
12	24	23	76.0921	57.3775	76	57	78	60	3.243	21.08
13	26	23	77.9331	58.159	78	58	80	61	3.513	24.59
14	28	23	79.7741	58.9405	80	59	82	62	3.784	28.38
15	30	23	81.6151	59.7219	82	60	84	63	4.054	32.43
16	32	23	83.4562	60.5034	83	61	86	64	4.324	36.75
17	34	23	85.2972	61.2849	85	61	88	65	4.594	41.35
18	36	23	87.1382	62.0663	87	62	90	66	4.865	46.21
19	38	23	88.9792	62.8478	89	63	92	67	5.135	51.35
20	40	23	90.8202	63.6292	91	64	94	68	5.405	56.75

Table 8-4 Tracking results using strategy three

frame	dist	deg	should be (real)		should be		tracked to		ind Err	Sum Err
0	0	23	54	48	54	48	54	48	0	0
1	2	23	55.841	48.7815	56	49	55.875	48.75	0.046	0.046
2	4	23	57.682	49.5629	58	50	57.625	49.5	0.085	0.131
3	6	23	59.523	50.3444	60	50	59.5	50.25	0.097	0.228
4	8	23	61.364	51.1258	61	51	61.375	51.125	0.011	0.239
5	10	23	63.205	51.9073	63	52	63.25	52	0.103	0.342
6	12	23	65.0461	52.6888	65	53	65.125	52.75	0.1	0.442
7	14	23	66.8871	53.4702	67	53	67	53.5	0.117	0.559
8	16	23	68.7281	54.2517	69	54	68.875	54.375	0.192	0.751
9	18	23	70.5691	55.0332	71	55	70.75	55.25	0.282	1.033
10	20	23	72.4101	55.8146	72	56	72.625	56.125	0.378	1.411
11	22	23	74.2511	56.5961	74	57	74.5	56.875	0.374	1.785
12	24	23	76.0921	57.3775	76	57	76.375	57.625	0.376	2.16
13	26	23	77.9331	58.159	78	58	78.375	58.625	0.642	2.803
14	28	23	79.7741	58.9405	80	59	80.25	59.5	0.735	3.537
15	30	23	81.6151	59.7219	82	60	82.125	60.25	0.734	4.271
16	32	23	83.4562	60.5034	83	61	83.875	61	0.65	4.921
17	34	23	85.2972	61.2849	85	61	85.75	61.75	0.649	5.57
18	36	23	87.1382	62.0663	87	62	87.75	62.75	0.917	6.487
19	38	23	88.9792	62.8478	89	63	89.75	63.75	1.187	7.674
20	40	23	90.8202	63.6292	91	64	91.625	64.5	1.186	8.86

Table 8-5 Tracking results using strategy four

frame	dist	deg	should be (real)		should be		tracked to		ind Err	Sum Err
0	0	23	54	48	54	48	54	48	0	0
1	2	23	55.841	48.7815	56	49	56	49	0.27	0.27
2	4	23	57.682	49.5629	58	50	58	50	0.541	0.811
3	6	23	59.523	50.3444	60	50	60	51	0.811	1.622
4	8	23	61.364	51.1258	61	51	62	52	1.081	2.703
5	10	23	63.205	51.9073	63	52	64	53	1.351	4.054
6	12	23	65.0461	52.6888	65	53	66	54	1.621	5.675
7	14	23	66.8871	53.4702	67	53	68	55	1.892	7.567
8	16	23	68.7281	54.2517	69	54	70	56	2.162	9.729
9	18	23	70.5691	55.0332	71	55	72	57	2.432	12.16
10	20	23	72.4101	55.8146	72	56	74	58	2.703	14.86
11	22	23	74.2511	56.5961	74	57	76	59	2.973	17.84
12	24	23	76.0921	57.3775	76	57	78	60	3.243	21.08
13	26	23	77.9331	58.159	78	58	80	61	3.513	24.59
14	28	23	79.7741	58.9405	80	59	82	62	3.784	28.38
15	30	23	81.6151	59.7219	82	60	84	63	4.054	32.43
16	32	23	83.4562	60.5034	83	61	86	64	4.324	36.75
17	34	23	85.2972	61.2849	85	61	88	65	4.594	41.35
18	36	23	87.1382	62.0663	87	62	90	66	4.865	46.21
19	38	23	88.9792	62.8478	89	63	92	67	5.135	51.35
20	40	23	90.8202	63.6292	91	64	94	68	5.405	56.75

Table 8-6 Tracking results using strategy five

frame	dist	deg	should be (real)		should be		tracked to		ind Err	Sum Err
0	0	23	54	48	54	48	54	48	0	0
1	2	23	55.841	48.7815	56	49	55.875	48.75	0.046	0.046
2	4	23	57.682	49.5629	58	50	57.625	49.5	0.085	0.131
3	6	23	59.523	50.3444	60	50	59.5	50.375	0.038	0.17
4	8	23	61.364	51.1258	61	51	61.375	51.125	0.011	0.181
5	10	23	63.205	51.9073	63	52	63.25	52	0.103	0.284
6	12	23	65.0461	52.6888	65	53	65	52.625	0.079	0.362
7	14	23	66.8871	53.4702	67	53	66.875	53.5	0.032	0.394
8	16	23	68.7281	54.2517	69	54	68.75	54.25	0.022	0.416
9	18	23	70.5691	55.0332	71	55	70.5	55	0.077	0.493
10	20	23	72.4101	55.8146	72	56	72.375	55.75	0.074	0.567
11	22	23	74.2511	56.5961	74	57	74.25	56.625	0.029	0.596
12	24	23	76.0921	57.3775	76	57	76.125	57.375	0.033	0.629
13	26	23	77.9331	58.159	78	58	77.875	58.125	0.067	0.696
14	28	23	79.7741	58.9405	80	59	79.75	58.875	0.07	0.766
15	30	23	81.6151	59.7219	82	60	81.625	59.75	0.03	0.795
16	32	23	83.4562	60.5034	83	61	83.5	60.5	0.044	0.839
17	34	23	85.2972	61.2849	85	61	85.25	61.25	0.059	0.898
18	36	23	87.1382	62.0663	87	62	87.125	62.125	0.06	0.958
19	38	23	88.9792	62.8478	89	63	89	62.875	0.034	0.992
20	40	23	90.8202	63.6292	91	64	90.875	63.625	0.055	1.047

Table 8-7 Tracking results using strategy six

frame	dist	deg	should be (real)		should be		tracked to		ind Err	Sum Err
0	0	23	54	48	54	48	54	48	0	0
1	2	23	55.841	48.7815	56	49	56	49	0.27	0.27
2	4	23	57.682	49.5629	58	50	57.5	49.5	0.193	0.463
3	6	23	59.523	50.3444	60	50	59.5	50.5	0.157	0.62
4	8	23	61.364	51.1258	61	51	61.5	51	0.185	0.805
5	10	23	63.205	51.9073	63	52	63	52	0.225	1.03
6	12	23	65.0461	52.6888	65	53	65	52.5	0.194	1.225
7	14	23	66.8871	53.4702	67	53	67	53.5	0.117	1.341
8	16	23	68.7281	54.2517	69	54	68.5	54	0.34	1.681
9	18	23	70.5691	55.0332	71	55	70.5	55	0.077	1.758
10	20	23	72.4101	55.8146	72	56	72.5	56	0.206	1.964
11	22	23	74.2511	56.5961	74	57	74.5	56.5	0.267	2.231
12	24	23	76.0921	57.3775	76	57	76	57.5	0.153	2.384
13	26	23	77.9331	58.159	78	58	78	58	0.173	2.556
14	28	23	79.7741	58.9405	80	59	80	59	0.234	2.79
15	30	23	81.6151	59.7219	82	60	81.5	59.5	0.25	3.04
16	32	23	83.4562	60.5034	83	61	83.5	60.5	0.044	3.084
17	34	23	85.2972	61.2849	85	61	85.5	61.5	0.296	3.38
18	36	23	87.1382	62.0663	87	62	87	62	0.153	3.533
19	38	23	88.9792	62.8478	89	63	89	63	0.154	3.686
20	40	23	90.8202	63.6292	91	64	91	63.5	0.221	3.908

Table 8-8 Final position tracked by each tracking strategy

Summary Table

strategy	Tracked to		ind RMS	Sum RMS
should be	90.8202	63.6292	0	0
0	94	68	5.405	56.753
1	92.125	65.375	2.179	23.912
2	94	68	5.405	56.753
3	91.625	64.5	1.185	8.859
4	94	68	5.405	56.753
5	90.875	63.625	0.054	1.047
6	91	63.5	0.221	3.907

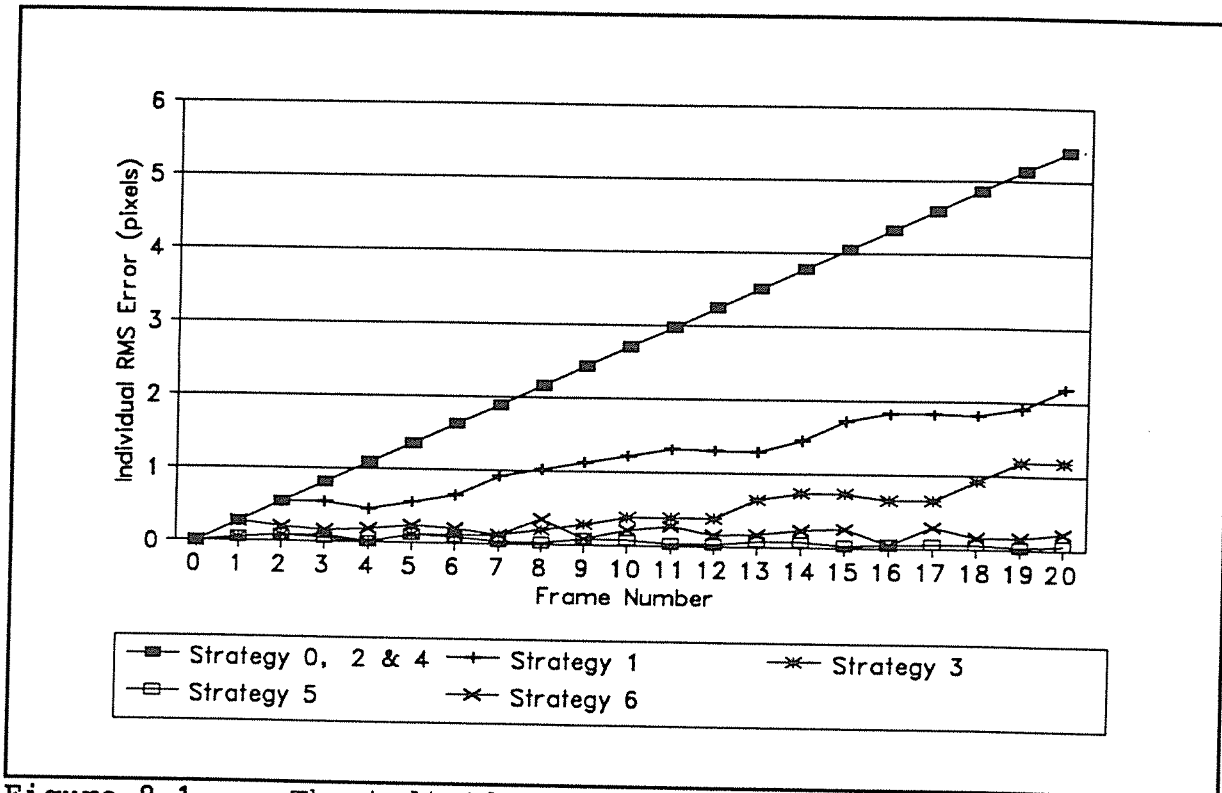


Figure 8-1 The individual frame radial error for the various tracking strategies

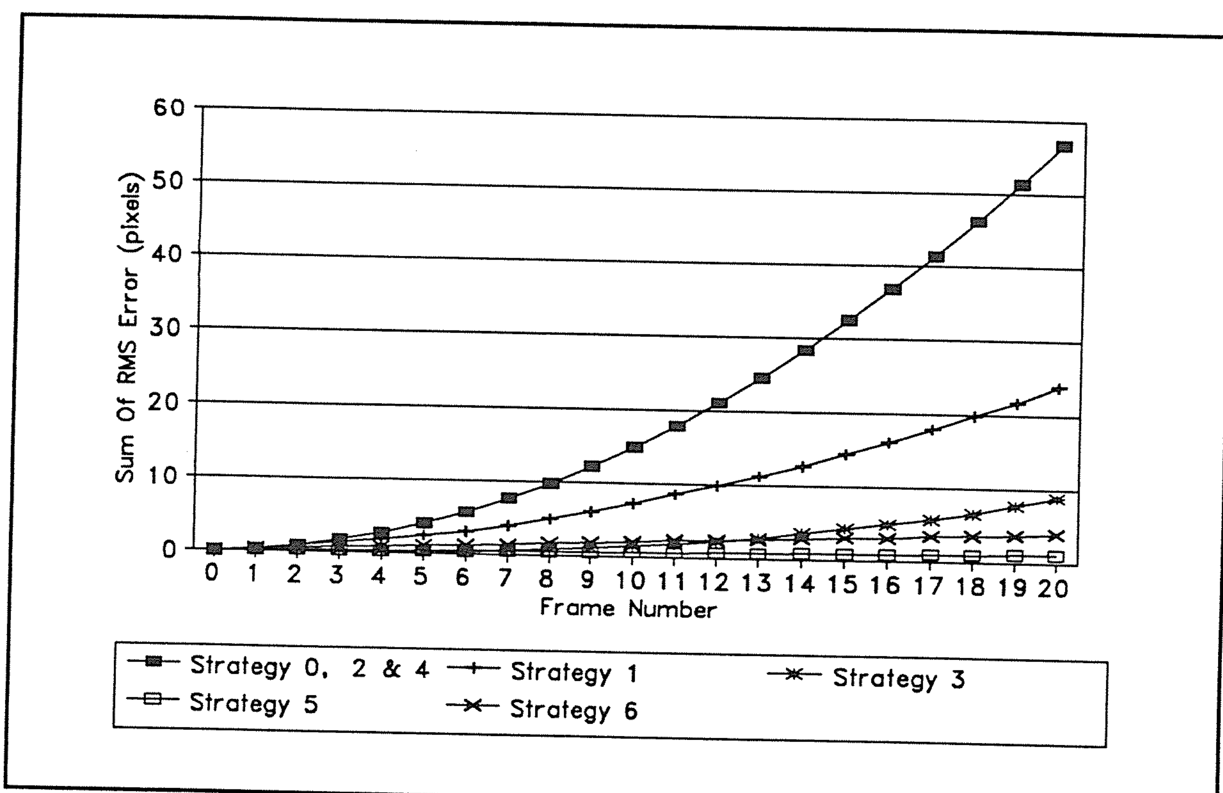


Figure 8-2 The cumulative radial error for the various strategies

8.5 Performance of the strategies when the picture is subjected to noise

An array of random noise was created for each frame using a random number generator with levels set between -255 and +255. To give an awareness of the effect of noise on the performance of each strategy these unique arrays were scaled between 0 and 10% and then added to the original simulated movie used in the first part of this chapter. This means the range of noise added when noise was scaled by 1% is -2.55 to +2.55 and when noise is scaled by 10% is -25 to +25. If adding the designated noise caused a grey level out of the possible range, i.e. below 0 or above 255, then the number was set at the limit being exceeded. This was unlikely to occur within a window of interest. The results of tracking through these noisy images with strategies zero, one, three and five are shown in Tables 8-9 to 8-12 and the resulting error is graphed in Figures 8-3 to 8-10. The experiment was only carried out on these four strategies from the six because one is the whole pixel algorithm (strategy zero) and the others are the sub-pixel algorithms tracking to a 0.125 pixel resolution. In section 8-4 of this chapter it was shown that a 0.5 pixel resolution shows no improvement over whole pixel tracking.

The summary of effects of the added noise on each strategy are:

Strategy Zero :

Noise scaled by 1, 2 and 8% had little or no affects on the tracking performance and for noise scaled by 3 and 4% the average radial error actually improved by 16.7% and 38.7% respectively.

Noise scaled between 5, 6, 7, 9 and 10% increased the average error between 130.6 and 402.4%

Strategy One :

Noise scaled by 1, 2 and 3% had little affect on the tracking performance and in all cases decreased the average radial error by between 25.9 and 82.4%.

For added noise scaled by 4, 5 and 6% average radial error increased by between 28.6 and 106.5%.

Noise scaled by 7, 8, 9 and 10% increased the average error by 584.9 to 736.7%.

The average radial error over the twenty frames for every case of noise added, except when scaled by 7% and 8%, was lower for strategy one than the corresponding strategy zero result.

Strategy Three :

Noise scaled by 1% decreased the average radial error by 5.9%.

For added noise scaled by 2 and 3% average radial error increased by 108.9 and 282.8% respectively.

Noise scaled by 4 and 10% increased the average error by 1493.1 to 3783.0%.

The average radial error over the twenty frames for cases of noise added scaled by 1, 2, 3, 5, and 9%, was lower for strategy three than the corresponding strategy zero result.

Strategy Five :

For added noise scaled by 1, 2, 3 and 4% the average radial error increased by between 40.6 and 573.6%.

Noise scaled by 5 to 10% increased the average error by 3086.0 to 18924.5%.

The average radial error over the twenty frames for every case of noise added, except when scaled by 8%, was lower for strategy five than the corresponding strategy zero result.

Table 8-13 and 8-14 show summary tables of the error for the point tracked to the 20th frame and the accumulated error over the 20 frames respectively. These tables are graphed in bar graphs in Figures 8-11 and 8-12. The bar graphs clearly show that although strategy three performs better than strategy one when there is no noise, strategy one actually performs remarkably better when noise scaled by 1, 3, 4, 5, 6, 8 and 10% were added. Also when noise scaled by 9 and 10% are added to the simulated video the sub-pixel algorithms, strategies one and three, perform equally well or better than strategy five. Based on these results the decision was made to implement both strategy one and strategy three in the chiropractic tracking prototype and examine their performance on real data from an X-ray spine video.

Table 8-9 Performance of strategy zero when noise is added to each frame

STRATEGY ZERO
(WINDOW RADIUS OF 5)

Frame	Dist	Deg	Should Be (real)		(rounded)	
0	0	23	54	48	54	48
1	2	23	55.841	48.7815	56	49
2	4	23	57.682	49.5629	58	50
3	6	23	59.523	50.3444	60	50
4	8	23	61.364	51.1258	61	51
5	10	23	63.205	51.9073	63	52
6	12	23	65.0461	52.6888	65	53
7	14	23	66.8871	53.4702	67	53
8	16	23	68.7281	54.2517	69	54
9	18	23	70.5691	55.0332	71	55
10	20	23	72.4101	55.8146	72	56
11	22	23	74.2511	56.5961	74	57
12	24	23	76.0921	57.3775	76	57
13	26	23	77.9331	58.159	78	58
14	28	23	79.7741	58.9405	80	59
15	30	23	81.6151	59.7219	82	60
16	32	23	83.4562	60.5034	83	61
17	34	23	85.2972	61.2849	85	61
18	36	23	87.1382	62.0663	87	62
19	38	23	88.9792	62.8478	89	63
20	40	23	90.8202	63.6292	91	64

Noise Scaled By 0.00

Tracked To		Ind RMS	Sum RMS
54	48	0.000	0.000
56	49	0.270	0.270
58	50	0.541	0.811
60	51	0.811	1.622
62	52	1.081	2.703
64	53	1.351	4.054
66	54	1.621	5.675
68	55	1.892	7.567
70	56	2.162	9.729
72	57	2.432	12.161
74	58	2.703	14.864
76	59	2.973	17.837
78	60	3.243	21.080
80	61	3.513	24.593
82	62	3.784	28.377
84	63	4.054	32.431
86	64	4.324	36.755
88	65	4.594	41.349
90	66	4.865	46.213
92	67	5.135	51.348
94	68	5.405	56.753

Noise Scaled By 0.01

Tracked To		Ind RMS	Sum RMS
54	48	0.000	0.000
56	49	0.270	0.270
58	50	0.541	0.811
60	51	0.811	1.622
62	52	1.081	2.703
64	53	1.351	4.054
66	54	1.621	5.675
68	55	1.892	7.567
70	56	2.162	9.729
72	57	2.432	12.161
74	58	2.703	14.864
76	59	2.973	17.837
78	60	3.243	21.080
80	61	3.513	24.593
82	62	3.784	28.377
84	63	4.054	32.431
86	64	4.324	36.755
88	65	4.594	41.349
90	66	4.865	46.213
92	67	5.135	51.348
94	68	5.405	56.753

Noise Scaled By 0.02

Tracked To		Ind RMS	Sum RMS
54	48	0.000	0.000
56	49	0.270	0.270
58	50	0.541	0.811
60	51	0.811	1.622
62	52	1.081	2.703
64	53	1.351	4.054
66	54	1.621	5.675
68	55	1.892	7.567
70	56	2.162	9.729
72	57	2.432	12.161
74	58	2.703	14.864
76	59	2.973	17.837
78	60	3.243	21.080
80	61	3.513	24.593
82	62	3.784	28.377
84	63	4.054	32.431
86	64	4.324	36.755
88	65	4.594	41.349
90	66	4.865	46.213
92	67	5.135	51.348
94	68	5.405	56.753

Noise Scaled By 0.03

Tracked To		Ind RMS	Sum RMS
54	48	0.000	0.000
56	49	0.270	0.270
58	50	0.541	0.811
59	51	0.839	1.649
61	52	0.947	2.596
64	55	3.193	5.790
65	55	2.312	8.101
67	56	2.532	10.634
69	57	2.762	13.395
71	58	2.998	16.393
73	59	3.240	19.633
75	60	3.485	23.118
76	60	2.624	25.742
78	60	1.842	27.584
80	61	2.072	29.656
82	62	2.310	31.967
84	63	2.555	34.522
86	64	2.805	37.326
88	65	3.058	40.384
90	66	3.313	43.697
92	67	3.571	47.269

Noise Scaled By 0.04

Tracked To		Ind RMS	Sum RMS
54	48	0.000	0.000
56	49	0.270	0.270
58	49	0.647	0.917
59	50	0.626	1.543
61	50	1.183	2.726
64	53	1.351	4.077
65	53	0.315	4.392
67	54	0.542	4.934
69	55	0.796	5.730
71	56	1.058	6.788
73	56	0.618	7.407
75	57	0.851	8.258
77	58	1.101	9.358
79	58	1.079	10.437
81	59	1.227	11.664
83	60	1.413	13.077
85	61	1.622	14.699
87	62	1.847	16.546
86	58	4.223	20.768
88	59	3.970	24.739
88	54	10.034	34.772

Noise Scaled By 0.05

Tracked To		Ind RMS	Sum RMS
54	48	0.000	0.000
56	49	0.270	0.270
56	44	5.812	6.082
57	46	5.024	11.106
59	46	5.645	16.750
61	48	4.487	21.237
62	47	6.453	27.690
64	49	5.321	33.011
66	49	5.918	38.929
66	44	11.942	50.871
68	45	11.679	62.551
69	45	12.730	75.280
71	45	13.384	88.664
72	46	13.529	102.194
74	46	14.170	116.364
76	48	12.997	129.361
76	45	17.203	146.564
78	46	16.937	163.502
81	50	13.538	177.040
83	52	12.387	189.426
85	55	10.409	199.835

Noise Scaled By 0.06

Tracked To		Ind RMS	Sum RMS
54	48	0.000	0.000
55	49	0.869	0.869
55	44	6.176	7.045
57	48	3.444	10.489
58	48	4.592	15.081
60	51	3.331	18.412
61	50	4.858	23.270
64	50	4.514	27.784
65	49	6.440	34.224
66	50	6.798	41.022
66	45	12.572	53.594
69	48	10.073	63.667
72	50	8.436	72.103
73	49	10.403	82.506
77	54	5.666	88.172
80	56	4.057	92.229
82	57	3.794	96.023
85	58	3.298	99.322
85	53	9.315	108.637
87	55	8.094	116.730
87	50	14.154	130.885

Noise Scaled By 0.07

Tracked To		Ind RMS	Sum RMS
54	48	0.000	0.000
55	49	0.869	0.869
55	44	6.176	7.045
56	46	5.593	12.638
58	47	5.323	17.961
60	50	3.730	21.691
61	49	5.475	27.166
64	49	5.321	32.488
65	47	8.154	40.642
66	43	12.871	53.513
68	44	12.611	66.124
70	48	9.590	75.714
73	50	7.999	83.713
74	49	9.968	93.681
78	53	6.200	99.880
81	55	4.762	104.642
83	50	10.513	115.156
85	52	9.290	124.445
87	53	9.067	133.513
89	54	8.848	142.360
89	49	14.742	157.102

Noise Scaled By 0.08

Tracked To		Ind RMS	Sum RMS
54	46	2.000	2.000
57	51	2.503	4.503
57	52	2.531	7.034
58	53	3.061	10.095
60	53	2.318	12.413
63	58	6.096	18.509
68	63	10.726	29.235
68	58	4.665	33.900
68	55	1.044	34.944
70	56	1.122	36.066
72	55	0.912	36.978
73	55	2.028	39.006
74	54	3.973	42.979
79	59	1.359	44.337
83	64	6.000	50.338
83	61	1.885	52.222
85	62	2.150	54.372
85	61	0.412	54.784
85	56	6.432	61.216
87	58	5.236	66.452

Noise Scaled By 0.09

Tracked To		Ind RMS	Sum RMS
55	47	1.414	1.414
58	52	3.876	5.290
58	53	3.452	8.742
59	54	3.693	12.434
61	53	1.909	14.344
64	58	6.144	20.488
69	63	11.043	31.531
74	68	16.177	47.709
77	72	19.581	67.290
79	77	23.529	90.819
80	74	19.706	110.525
81	75	19.602	130.127
83	75	18.928	149.055
87	78	21.815	170.870
90	76	19.890	190.759
90	79	21.023	211.782
93	84	25.361	237.143
93	79	19.317	256.460
93	75	14.200	270.660
95	76	14.465	285.125

Noise Scaled By 0.10

Tracked To		Ind RMS	Sum RMS
55	46	2.236	2.236
58	51	3.096	5.332
63	56	8.350	13.681
65	55	7.188	20.870
68	59	10.298	31.167
70	60	10.567	41.734
69	56	5.157	46.892
72	56	5.705	52.596
70	51	3.492	56.088
71	53	2.078	58.166
74	54	2.413	60.579
77	54	3.781	64.360
76	51	6.378	70.738
78	47	11.159	81.897
82	52	7.289	89.186
81	47	12.737	101.923
82	44	16.568	118.490
80	39	22.906	141.396
80	37	26.063	167.459
77	32	33.092	200.551

Table 8-10 Performance of strategy one when noise is added to each frame

STRATEGY ONE
(WINDOW RADIUS OF 5)

Frame	Dist	Deg	Should Be (real)		(rounded)		Noise Scaled By 0.00 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.01 Tracked To		Ind RMS	Sum RMS
0	0	23	54	48	54	48	54	48	0.000	0.000	54	48	0.000	0.000
1	2	23	55.841	48.7815	56	49	56	49	0.270	0.270	56	49	0.270	0.270
2	4	23	57.682	49.5629	58	50	58	50	0.541	0.811	57.875	49.625	0.203	0.473
3	6	23	59.523	50.3444	60	50	59.875	50.75	0.537	1.348	59.625	50.375	0.106	0.579
4	8	23	61.364	51.1258	61	51	61.625	51.5	0.456	1.804	61.5	51.125	0.136	0.715
5	10	23	63.205	51.9073	63	52	63.5	52.375	0.553	2.357	63.25	52.125	0.222	0.938
6	12	23	65.0461	52.6888	65	53	65.375	53.25	0.650	3.007	65	52.875	0.192	1.130
7	14	23	66.8871	53.4702	67	53	67.375	54.25	0.920	3.927	67	53.75	0.302	1.431
8	16	23	68.7281	54.2517	69	54	69.25	55.125	1.017	4.945	68.875	54.5	0.289	1.720
9	18	23	70.5691	55.0332	71	55	71.125	56	1.115	6.060	70.625	55.25	0.224	1.944
10	20	23	72.4101	55.8146	72	56	73	56.875	1.213	7.273	72.5	56.125	0.323	2.267
11	22	23	74.2511	56.5961	74	57	74.875	57.75	1.312	8.585	74.25	56.875	0.279	2.546
12	24	23	76.0921	57.3775	76	57	76.75	58.5	1.301	9.886	76	57.625	0.264	2.810
13	26	23	77.9331	58.159	78	58	78.625	59.25	1.292	11.178	77.875	58.25	0.108	2.918
14	28	23	79.7741	58.9405	80	59	80.625	60.125	1.458	12.637	79.875	59.125	0.210	3.128
15	30	23	81.6151	59.7219	82	60	82.625	61.125	1.729	14.365	81.75	60	0.309	3.437
16	32	23	83.4562	60.5034	83	61	84.5	62	1.825	16.190	83.625	60.75	0.299	3.736
17	34	23	85.2972	61.2849	85	61	86.375	62.75	1.819	18.009	85.375	61.375	0.119	3.855
18	36	23	87.1382	62.0663	87	62	88.25	63.5	1.814	19.823	87.125	62.125	0.060	3.915
19	38	23	88.9792	62.8478	89	63	90.125	64.375	1.909	21.732	89	63	0.154	4.069
20	40	23	90.8202	63.6292	91	64	92.125	65.375	2.180	23.912	90.875	63.75	0.133	4.201

Noise Scaled By 0.02 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.03 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.04 Tracked To		Ind RMS	Sum RMS
54.125	47.875	0.177	0.177	54.25	47.75	0.354	0.354	54.375	48.25	0.451	0.451
56	48.875	0.184	0.361	55.875	48.625	0.160	0.514	56	49.375	0.614	1.065
57.875	49.5	0.203	0.564	57.625	49.125	0.442	0.955	57.75	49.875	0.319	1.385
59.5	50.125	0.221	0.785	59.125	49.875	0.615	1.571	59.125	50.5	0.427	1.812
61.25	50.75	0.393	1.178	60.875	50.375	0.896	2.467	60.75	51	0.627	2.439
63	52	0.225	1.403	63.375	52.875	0.983	3.449	63.5	54.5	2.609	5.048
64.75	52.625	0.303	1.705	65	53.5	0.813	4.262	65	54	1.312	6.360
66.75	53.75	0.312	2.017	67.25	54.875	1.451	5.713	67.25	55.5	2.062	8.422
68.5	54.375	0.259	2.276	68.875	55.375	1.133	6.846	68.625	55	0.755	9.177
70.25	55	0.321	2.597	70.625	56	0.968	7.814	70.25	55.625	0.672	9.850
72	55.625	0.452	3.049	72.25	56.5	0.704	8.518	71.875	56	0.566	10.416
73.625	56.25	0.715	3.764	73.75	57.125	0.729	9.246	73.25	56.375	1.025	11.441
75.25	56.75	1.050	4.814	75.375	57.75	0.808	10.054	74.75	56.625	1.539	12.980
77.125	57.375	1.126	5.940	77.125	58.375	0.836	10.891	76.375	57.25	1.804	14.784
78.875	58.125	1.214	7.154	79.5	59.625	0.737	11.628	79.875	61	2.062	16.846
81	59	0.948	8.103	81.25	60.875	1.210	12.838	81.5	62.375	2.656	19.501
82.75	59.5	1.227	9.330	83	61.375	0.984	13.822	83.75	63.125	2.638	22.139
84.5	60	1.512	10.842	84.75	61.875	0.805	14.626	86	63.875	2.684	24.823
86.25	60.75	1.588	12.430	86.375	62.625	0.946	15.572	87.5	64.5	2.460	27.284
88.125	61.5	1.596	14.025	88.125	63.25	0.944	16.516	89.25	65.75	2.915	30.198
89.75	62	1.949	15.975	89.625	63.625	1.195	17.712	90.625	66	2.379	32.577

Noise Scaled By 0.05 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.06 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0 .07 Tracked To		Ind RMS	Sum RMS
54.375	47.625	0.530	0.530	54.5	47.625	0.625	0.625	54.5	47.5	0.707	0.707
55.875	49	0.221	0.751	55.875	49	0.221	0.846	58	51	3.096	3.803
56.25	44.375	5.382	6.133	56.25	43.75	5.987	6.833	59.625	51.5	2.744	6.546
58.75	49.875	0.904	7.038	58.75	49.25	1.340	8.173	63.125	57	7.568	14.114
60.375	50.375	1.242	8.279	60.25	49.75	1.770	9.943	65.625	56.375	6.761	20.875
62.875	53	1.141	9.421	63.75	54.25	2.405	12.348	69.25	60	10.101	30.976
64.375	53.5	1.053	10.474	65.25	53.5	0.836	13.185	70.75	64.625	13.229	44.205
66.625	55	1.552	12.026	67.5	55.25	1.882	15.067	72.375	61.875	10.038	54.243
68.125	54.5	0.652	12.678	69.875	57.875	3.800	18.867	70.875	56.375	3.020	57.263
69.75	55	0.820	13.498	71.625	58.25	3.386	22.253	68.5	50.875	4.645	61.907
71.375	54.625	1.577	15.075	73.25	58.5	2.814	25.067	70.875	50.375	5.652	67.559
72.75	54.875	2.284	17.358	74.75	59	2.455	27.522	72.375	52	4.964	72.524
74.125	54.5	3.486	20.844	76	59.625	2.249	29.771	73.875	51.625	6.165	78.689
75.5	55	3.987	24.831	77.625	59.125	1.014	30.785	74.5	50.125	8.737	87.425
79	58.75	0.797	25.629	80.125	60.5	1.598	32.384	76	47.5	12.047	99.472
81.375	60.375	0.696	26.324	81.625	62.125	2.403	34.787	78.375	49.25	10.962	110.434
83.625	61	0.525	26.849	84.125	62.625	2.225	37.011	76.875	43.75	18.000	128.434
86.125	61.625	0.895	27.744	86.5	63.375	2.411	39.423	79.5	46.375	15.997	144.431
87.625	63	1.053	28.797	87.875	64	2.069	41.492	79.125	44.625	19.194	163.625
89.375	64.375	1.578	30.375	91.5	69.5	7.114	48.606	80.75	46.25	18.526	182.151
90.75	64	0.377	30.752	90.125	64	0.788	49.394	84.125	47	17.926	200.077

Noise Scaled By 0.08 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.09 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.10 Tracked To		Ind RMS	Sum RMS
54.5	46.5	1.581	1.581	54.5	46.5	1.581	1.581	54.5	46.5	1.581	1.581
57.125	51.25	2.782	4.364	57.125	51.25	2.782	4.364	57.125	51.25	2.782	4.364
55.625	48.875	2.169	6.533	55.625	48.75	2.212	6.575	55.625	48.75	2.212	6.575
57	50.75	2.555	9.088	57	50.625	2.539	9.114	56.875	50.5	2.653	9.228
58.5	51.25	2.867	11.955	58.5	51.125	2.864	11.978	58.375	51	2.992	12.220
61	54.625	3.500	15.454	61	54.5	3.404	15.382	58.75	50.625	4.636	16.855
61.625	54.375	3.814	19.268	58.625	49	7.405	22.787	59.25	50	6.389	23.245
61.125	49	7.293	26.561	60	51.25	7.236	30.023	60.5	52.25	6.503	29.748
62.5	47.25	9.371	35.932	62.375	50.875	7.195	37.218	62.875	51.875	6.317	36.065
65	46.75	9.981	45.913	65	51.375	6.663	43.881	65.5	52.375	5.724	41.789
66.375	47.25	10.477	56.391	65.5	45.875	12.106	55.986	66.125	46.875	10.928	52.716
67.75	46.875	11.695	68.085	68.125	49.375	9.470	65.456	67.5	46.5	12.145	64.862
69.125	44.375	14.751	82.837	69.5	47	12.294	77.750	68.125	42.875	16.547	81.409
70.625	44.875	15.162	97.998	70.875	45.75	14.276	92.026	70.75	43.25	16.549	97.958
72.875	45.375	15.219	113.217	72.375	42.125	18.371	110.397	72.875	43.625	16.798	114.755
75.25	50.875	10.899	124.116	74.875	47.625	13.848	124.245	75.25	49.125	12.362	127.117
75	47.625	15.407	139.523	75.125	44.875	17.710	141.956	75.375	46.375	16.276	143.393
77.625	50.125	13.543	153.065	77.75	48.125	15.170	157.126	78.875	49	13.862	157.256
80.375	53.75	10.719	163.785	82.375	53.625	9.692	166.819	83.625	54.5	8.342	165.598

Table 8-11 Performance of strategy three when noise is added to each frame

STRATEGY THREE
(WINDOW RADIUS OF 5)

Frame	Dist	Deg	Should Be (real)		(rounded)		Noise Scaled By 0.00 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.01 Tracked To		Ind RMS	Sum RMS
0	0	23	54	48	54	48	54	48	0.000	0.000	54	48	0.000	0.000
1	2	23	55.841	48.7815	56	49	55.875	48.75	0.046	0.046	55.875	48.75	0.046	0.046
2	4	23	57.682	49.5629	58	50	57.625	49.5	0.085	0.131	57.625	49.375	0.196	0.243
3	6	23	59.523	50.3444	60	50	59.5	50.25	0.097	0.228	59.25	50.125	0.350	0.593
4	8	23	61.364	51.1258	61	51	61.375	51.125	0.011	0.239	61	50.75	0.523	1.116
5	10	23	63.205	51.9073	63	52	63.25	52	0.103	0.342	62.875	51.625	0.434	1.550
6	12	23	65.0461	52.6888	65	53	65.125	52.75	0.100	0.442	64.75	52.375	0.431	1.982
7	14	23	66.8871	53.4702	67	53	67	53.5	0.117	0.559	66.625	53.25	0.342	2.324
8	16	23	68.7281	54.2517	69	54	68.875	54.375	0.192	0.751	68.375	54	0.434	2.758
9	18	23	70.5691	55.0332	71	55	70.75	55.25	0.282	1.033	70.25	55	0.321	3.079
10	20	23	72.4101	55.8146	72	56	72.625	56.125	0.378	1.411	72.125	55.75	0.292	3.371
11	22	23	74.2511	56.5961	74	57	74.5	56.875	0.374	1.785	73.875	56.5	0.388	3.759
12	24	23	76.0921	57.3775	76	57	76.375	57.625	0.376	2.160	75.625	57.25	0.484	4.243
13	26	23	77.9331	58.159	78	58	78.375	58.625	0.642	2.803	77.625	58	0.347	4.590
14	28	23	79.7741	58.9405	80	59	80.25	59.5	0.735	3.537	79.5	58.75	0.334	4.924
15	30	23	81.6151	59.7219	82	60	82.125	60.25	0.734	4.271	81.25	59.625	0.378	5.302
16	32	23	83.4562	60.5034	83	61	83.875	61	0.650	4.921	83	60.25	0.522	5.823
17	34	23	85.2972	61.2849	85	61	85.75	61.75	0.649	5.570	84.75	60.875	0.684	6.507
18	36	23	87.1382	62.0663	87	62	87.75	62.75	0.917	6.487	86.5	61.75	0.712	7.219
19	38	23	88.9792	62.8478	89	63	89.75	63.75	1.187	7.674	88.5	62.75	0.489	7.708
20	40	23	90.8202	63.6292	91	64	91.625	64.5	1.186	8.860	90.25	63.375	0.624	8.333

Noise Scaled By 0.02 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.03 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.04 Tracked To		Ind RMS	Sum RMS
54.125	48	0.125	0.125	54.125	48	0.125	0.125	54.25	48	0.250	0.250
55.875	48.75	0.046	0.171	55.875	48.75	0.046	0.171	55.875	48.625	0.160	0.410
57.625	49.375	0.196	0.368	57.625	49.25	0.318	0.489	56.25	44.375	5.382	5.792
59.125	50.125	0.454	0.822	59.125	50	0.526	1.016	57.75	45.625	5.041	10.834
60.75	50.625	0.792	1.615	60.75	50.5	0.877	1.892	59.375	46	5.498	16.332
62.5	52	0.711	2.326	62.5	52.125	0.738	2.630	61.75	48.5	3.705	20.037
64.125	52.5	0.940	3.266	63.25	51.5	2.154	4.784	62.5	47.875	5.446	25.482
66.375	53.625	0.535	3.801	65.5	52.875	1.509	6.294	64.25	49.25	4.976	30.459
68	54.25	0.728	4.529	67.125	53.375	1.827	8.121	65.875	49.75	5.330	35.788
69.75	55.25	0.847	5.376	68.75	54.25	1.981	10.101	67.375	50.375	5.648	41.436
71.625	56	0.807	6.183	70.375	54.75	2.297	12.398	69	49.875	6.849	48.285
73.375	56.75	0.890	7.072	71.75	55.125	2.902	15.300	70.625	50.375	7.201	55.486
75.125	57.375	0.967	8.040	73.5	55.625	3.129	18.429	73.125	51.875	6.251	61.738
77	58.125	0.934	8.973	75.25	56.5	3.155	21.583	73.875	50.375	8.778	70.516
78.75	58.875	1.026	9.999	77.5	57.875	2.511	24.095	75.375	48	11.792	82.308
80.375	59.625	1.244	11.243	79.875	59.25	1.803	25.897	77.75	49.5	10.928	93.236
82.25	60.5	1.206	12.450	81.625	59.875	1.936	27.833	79.125	50	11.361	104.597
84	61.125	1.307	13.757	84.125	61.25	1.173	29.006	81	51.125	11.031	115.629
85.75	61.875	1.401	15.158	85.75	61.875	1.401	30.408	82.625	51.75	11.260	126.889
87.375	62.625	1.620	16.777	87.375	62.625	1.620	32.027	86	57.25	6.341	133.230
89.125	63.25	1.737	18.515	89	63.125	1.889	33.916	87.375	56.5	7.918	141.148

Noise Scaled By 0.05 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.06 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.07 Tracked To		Ind RMS	Sum RMS
54.25	48	0.250	0.250	54.375	47.5	0.625	0.625	54.375	47.5	0.625	0.625
55.875	48.625	0.160	0.410	56	48.875	0.184	0.809	57.625	52	3.680	4.305
56.375	44.375	5.350	5.760	56.375	44.5	5.229	6.038	62.125	57.5	9.096	13.401
57.75	45.625	5.041	10.802	57.75	45.875	4.808	10.847	64.625	62.125	12.838	26.239
59.125	42	9.396	20.198	59.25	42.375	9.003	19.849	66.125	61.125	11.075	37.314
61.375	44.5	7.630	27.828	61.375	43.875	8.238	28.087	67.5	61.75	10.739	48.053
62	42.75	10.395	38.223	62.75	43.375	9.593	37.680	70.125	64.25	12.628	60.680
64.625	40.5	13.166	51.389	64.375	45.5	8.357	46.037	74.625	67.875	16.352	77.032
67.125	46	8.406	59.795	65.875	46	8.731	54.768	77	71.375	19.017	96.048
68.5	45.5	9.755	69.550	66.5	41.5	14.132	68.899	79.5	73	20.064	116.112
70.125	44.125	11.911	81.461	68.25	42	14.427	83.327	78.25	68.5	13.965	130.078
71.625	43.5	13.357	94.818	70.5	45.5	11.713	95.040	77.75	63.625	7.852	137.929
75	49	8.448	103.266	73.125	47.75	10.074	105.114	78.125	64.25	7.167	145.096
76.5	45.5	12.740	116.006	73.875	47.125	11.757	116.871	79.75	63.5	5.642	150.738
80.875	51	8.016	124.023	78.5	52.375	6.688	123.559	83.25	68.125	9.820	160.558
80.25	46.5	13.292	137.315	81	54	5.755	129.313	82.75	64.625	5.033	165.591
82.5	46.125	14.410	151.725	82.625	49.5	11.035	140.348	85.125	66.125	5.864	171.455
85.875	51.625	9.677	161.402	84.875	51.125	10.169	150.517	87.375	66.5	5.614	177.068
87.625	52.5	9.579	170.981	87.25	52.75	9.317	159.834	88.625	67.625	5.754	182.823
90	54.125	8.782	179.763	89.625	54.375	8.497	168.331	90.125	67.125	4.428	187.251
89.625	48.75	14.927	194.690	89.25	49	14.713	183.044	90.5	65.75	2.145	189.395

Noise Scaled By 0.08 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.09 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.10 Tracked To		Ind RMS	Sum RMS
54.5	46.5	1.581	1.581	54.625	46.5	1.625	1.625	54.875	46.5	1.737	1.737
57.75	51.125	3.023	4.604	57.875	51.125	3.103	4.728	58.125	51.125	3.272	5.009
62.375	56.625	8.479	13.083	62.5	56.625	8.549	13.277	62.75	56.625	8.692	13.701
64.875	62.125	12.939	26.022	65	62.125	12.992	26.269	65.25	62.125	13.099	26.800
69.5	66.5	17.394	43.417	66.5	61	11.130	37.399	66.625	60.875	11.078	37.878
71	67	16.987	60.403	67	60.375	9.279	46.678	67.125	60.25	9.218	47.096
69.625	62.5	10.827	71.231	69.5	62.75	11.003	57.681	69.625	62.625	10.940	58.037
75.125	68	16.703	87.933	75	68.25	16.860	74.541	75.125	68	16.703	74.739
77.375	71.5	19.294	107.228	77.375	72.625	20.306	94.847	77.375	71.5	19.294	94.034
79.875	73.125	20.345	127.572	79.75	74.875	21.863	116.710	79.875	73.125	20.345	114.379
84.5	78.625	25.816	153.389	84.5	80.375	27.375	144.085	84.5	78.625	25.816	140.195
85.875	80.125	26.244	179.632	84.875	81	26.616	170.701	88.25	81.25	28.351	168.546
87.375	80.625	25.841	205.473	86.625	81.5	26.322	197.023	89.75	81	27.287	195.833
87.25	75.875	20.017	225.490	86.875	76.25	20.180	217.203	93	83.625	29.589	225.422
85.75	70.375	12.902	238.391	87.5	71.5	14.746	231.949	94.625	86.375	31.196	256.618
86.25	69.125	10.483	248.875	88	70	12.100	244.049	95	82.75	26.635	283.253
87.875	70	10.474	259.349	92.5	70.625	13.573	257.622	95.375	84.25	26.570	309.823
91.5	75.5	15.509	274.859	91.875	65.125	7.617	265.239	95.125	78.75	20.040	329.864
95	77	16.877	291.735	93	65.75	6.923	272.162	94.375	74.25	14.171	344.035

Table 8-12 Performance of strategy five when noise is added to each frame

STRATEGY FIVE
(WINDOW RADIUS OF 5)

Frame	Dist	Deg	Should Be (real)		(rounded)		Noise Scaled By 0.00 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.01 Tracked To		Ind RMS	Sum RMS
0	0	23	54	48	54	48	54	48	0.000	0.000	54	48	0.000	0.000
1	2	23	55.841	48.7815	56	49	55.875	48.75	0.046	0.046	55.875	48.875	0.099	0.099
2	4	23	57.682	49.5629	58	50	57.625	49.5	0.085	0.131	57.75	49.5	0.093	0.192
3	6	23	59.523	50.3444	60	50	59.5	50.375	0.038	0.170	59.5	50.375	0.038	0.230
4	8	23	61.364	51.1258	61	51	61.375	51.125	0.011	0.181	61.375	51.125	0.011	0.241
5	10	23	63.205	51.9073	63	52	63.25	52	0.103	0.284	63.25	52	0.103	0.344
6	12	23	65.0461	52.6888	65	53	65	52.625	0.079	0.362	65	52.75	0.077	0.421
7	14	23	66.8871	53.4702	67	53	66.875	53.5	0.032	0.394	66.875	53.5	0.032	0.453
8	16	23	68.7281	54.2517	69	54	68.75	54.25	0.022	0.416	68.75	54.375	0.125	0.578
9	18	23	70.5691	55.0332	71	55	70.5	55	0.077	0.493	70.5	55	0.077	0.655
10	20	23	72.4101	55.8146	72	56	72.375	55.75	0.074	0.567	72.5	55.875	0.108	0.763
11	22	23	74.2511	56.5961	74	57	74.25	56.625	0.029	0.596	74.25	56.625	0.029	0.792
12	24	23	76.0921	57.3775	76	57	76.125	57.375	0.033	0.629	76.125	57.5	0.127	0.919
13	26	23	77.9331	58.159	78	58	77.875	58.125	0.067	0.696	77.875	58.125	0.067	0.987
14	28	23	79.7741	58.9405	80	59	79.75	58.875	0.070	0.766	79.75	58.875	0.070	1.056
15	30	23	81.6151	59.7219	82	60	81.625	59.75	0.030	0.795	81.625	59.75	0.030	1.086
16	32	23	83.4562	60.5034	83	61	83.5	60.5	0.044	0.839	83.5	60.5	0.044	1.130
17	34	23	85.2972	61.2849	85	61	85.25	61.25	0.059	0.898	85.25	61.25	0.059	1.189
18	36	23	87.1382	62.0663	87	62	87.125	62.125	0.060	0.958	87.125	62.125	0.060	1.249
19	38	23	88.9792	62.8478	89	63	89	62.875	0.034	0.992	89	63	0.154	1.403
20	40	23	90.8202	63.6292	91	64	90.875	63.625	0.055	1.047	90.75	63.625	0.070	1.473

Noise Scaled By 0.02 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.03 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.04 Tracked To		Ind RMS	Sum RMS
54.125	48	0.125	0.125	54.125	48	0.125	0.125	54.25	48	0.250	0.250
55.875	48.875	0.099	0.224	56.125	49.125	0.446	0.571	56.125	49.125	0.446	0.696
57.75	49.625	0.092	0.317	57.875	49.5	0.203	0.774	57.875	49.5	0.203	0.899
59.375	50.5	0.215	0.531	59.375	50.5	0.215	0.988	59.25	50.5	0.314	1.213
61.375	51	0.126	0.658	61.5	51.5	0.398	1.387	61.5	51.5	0.398	1.611
63.25	52.125	0.222	0.880	63.375	52.125	0.276	1.663	63.5	52.25	0.452	2.063
65	52.75	0.077	0.957	65	52.875	0.192	1.855	65	52.875	0.192	2.255
67.125	53.625	0.284	1.240	67.125	53.75	0.367	2.222	67.25	54.25	0.860	3.115
68.625	54.375	0.161	1.401	68.625	54.5	0.269	2.491	68.625	54.5	0.269	3.384
70.5	55	0.077	1.478	70.5	55	0.077	2.567	70.5	55	0.077	3.461
72.5	55.875	0.108	1.586	72.5	55.875	0.108	2.676	72.5	55.875	0.108	3.569
74.25	56.625	0.029	1.615	74.25	56.625	0.029	2.705	74.5	56.625	0.251	3.820
75.875	57.5	0.249	1.864	75.75	57.5	0.363	3.068	75.625	57.5	0.483	4.302
77.875	58.125	0.067	1.932	77.75	58	0.243	3.310	77.875	58.5	0.346	4.648
79.75	58.875	0.070	2.001	79.75	58.875	0.070	3.380	79.5	58.5	0.519	5.167
81.5	59.875	0.192	2.193	81.625	60.5	0.778	4.158	81.5	60.5	0.787	5.954
83.5	60.5	0.044	2.237	83.5	60.5	0.044	4.202	83.5	60.5	0.044	5.998
85.25	61.25	0.059	2.296	85.25	61.25	0.059	4.261	85.25	61.125	0.167	6.164
87.125	62	0.068	2.363	87.125	61.875	0.192	4.453	87.125	61.875	0.192	6.356
88.875	63.125	0.296	2.659	88.875	63.25	0.415	4.868	88.75	63.25	0.463	6.819
90.75	63.5	0.147	2.806	90.625	63.5	0.234	5.102	90.625	63.5	0.234	7.053

Noise Scaled By 0.05 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.06 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.07 Tracked To		Ind RMS	Sum RMS
54.25	48	0.250	0.250	54.375	47.5	0.625	0.625	54.375	47.5	0.625	0.625
56.125	49.125	0.446	0.696	56.5	49.125	0.743	1.368	56.5	49.125	0.743	1.368
56.5	44.5	5.199	5.895	56.5	44.5	5.199	6.567	56.5	44.5	5.199	6.567
59.25	49.5	0.887	6.782	59.25	49.5	0.887	7.455	59.25	49.5	0.887	7.455
59.5	46.5	4.987	11.769	59.5	46.5	4.987	12.442	59.5	46.5	4.987	12.442
63.5	51.5	0.503	12.272	63.5	51.5	0.503	12.945	63.625	51.5	0.585	13.027
65	52.875	0.192	12.464	65	53	0.315	13.259	65	53	0.315	13.342
67.25	54.375	0.975	13.439	67.25	54.5	1.092	14.351	67.5	54.625	1.307	14.649
68.375	53.5	0.831	14.270	66.625	49.375	5.311	19.662	66.5	49.625	5.135	19.784
70.5	55	0.077	14.346	67.5	48.625	7.105	26.767	67.5	48.5	7.218	27.002
72.5	56	0.206	14.552	69.75	48.375	7.901	34.668	69.875	48.375	7.860	34.862
74.5	56.625	0.251	14.803	72.5	53	4.000	38.668	72.375	52.75	4.279	39.141
75.625	57.5	0.483	15.286	75.625	57.5	0.483	39.151	75.5	57.5	0.605	39.746
77.75	58.5	0.387	15.673	77.75	58.5	0.387	39.538	77.75	58.5	0.387	40.133
79.5	58.5	0.519	16.192	79.5	58.5	0.519	40.057	81.625	61.5	3.159	43.292
81.5	60.625	0.910	17.102	81.5	60.875	1.159	41.216	81.5	61.5	1.782	45.073
83.5	60.5	0.044	17.146	81.5	57.125	3.904	45.119	81.5	57.25	3.796	48.870
85.25	61.125	0.167	17.313	85.25	61.125	0.167	45.286	84.25	57.5	3.927	52.797
84.5	55.5	7.076	24.389	84.5	55.5	7.076	52.363	84.5	55.25	7.309	60.106
86.5	56.5	6.815	31.204	86.5	56.5	6.815	59.177	86.5	56.5	6.815	66.921
90.5	61.5	2.153	33.357	89.875	61.5	2.330	61.507	89.875	61.5	2.330	69.250

Noise Scaled By 0.08 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.09 Tracked To		Ind RMS	Sum RMS	Noise Scaled By 0.10 Tracked To		Ind RMS	Sum RMS
54.5	46.5	1.581	1.581	54.625	46.5	1.625	1.625	54.875	46.5	1.737	1.737
56.5	49.125	0.743	2.324	56.5	49	0.694	2.319	56.5	49	0.694	2.431
56.625	44.25	5.417	7.741	56.875	44.5	5.127	7.446	56.875	44.375	5.250	7.681
58.75	49.5	1.145	8.886	58.625	49.5	1.233	8.679	58.5	49.5	1.326	9.008
59.5	46.375	5.103	13.989	59.5	46.375	5.103	13.782	59.5	46.375	5.103	14.111
61.5	45.5	6.630	20.620	61.5	45.5	6.630	20.412	61.625	45.5	6.599	20.710
60.5	42.125	11.500	32.120	60.5	42.25	11.386	31.798	60.5	42.375	11.271	31.981
63.625	47.25	7.024	39.144	63.625	47.25	7.024	38.822	63.625	47.375	6.913	38.895
66.5	49.875	4.911	44.055	66.5	49.875	4.911	43.733	66.5	50	4.800	43.695
67.375	48.5	7.272	51.327	67.25	48.5	7.328	51.061	67.25	48.5	7.328	51.023
67.75	43.5	13.167	64.494	67.625	43.5	13.212	64.273	67.625	43.5	13.212	64.234
71.25	49.5	7.705	72.199	71.125	49.5	7.754	72.027	69.5	47.5	10.262	74.497
74.625	55.5	2.383	74.582	74.625	55.5	2.383	74.410	72.5	44.5	13.369	87.866
77.625	58.5	0.460	75.041	77.625	58.5	0.460	74.869	72.5	44.375	14.816	102.682
81.625	61.5	3.159	78.200	81.75	61.5	3.233	78.103	73.125	40.625	19.485	122.167
81.5	61.5	1.782	79.982	81.5	61.5	1.782	79.884	77.5	46.5	13.847	136.014
81.375	57.375	3.757	83.739	81.375	57.5	3.654	83.538	78.5	44.75	16.515	152.529
84.25	57.375	4.048	87.787	84.125	57.25	4.202	87.740	74.5	41.5	22.539	175.068
84.5	55	7.543	95.329	84.5	55	7.543	95.283	78.625	39.5	24.119	199.187

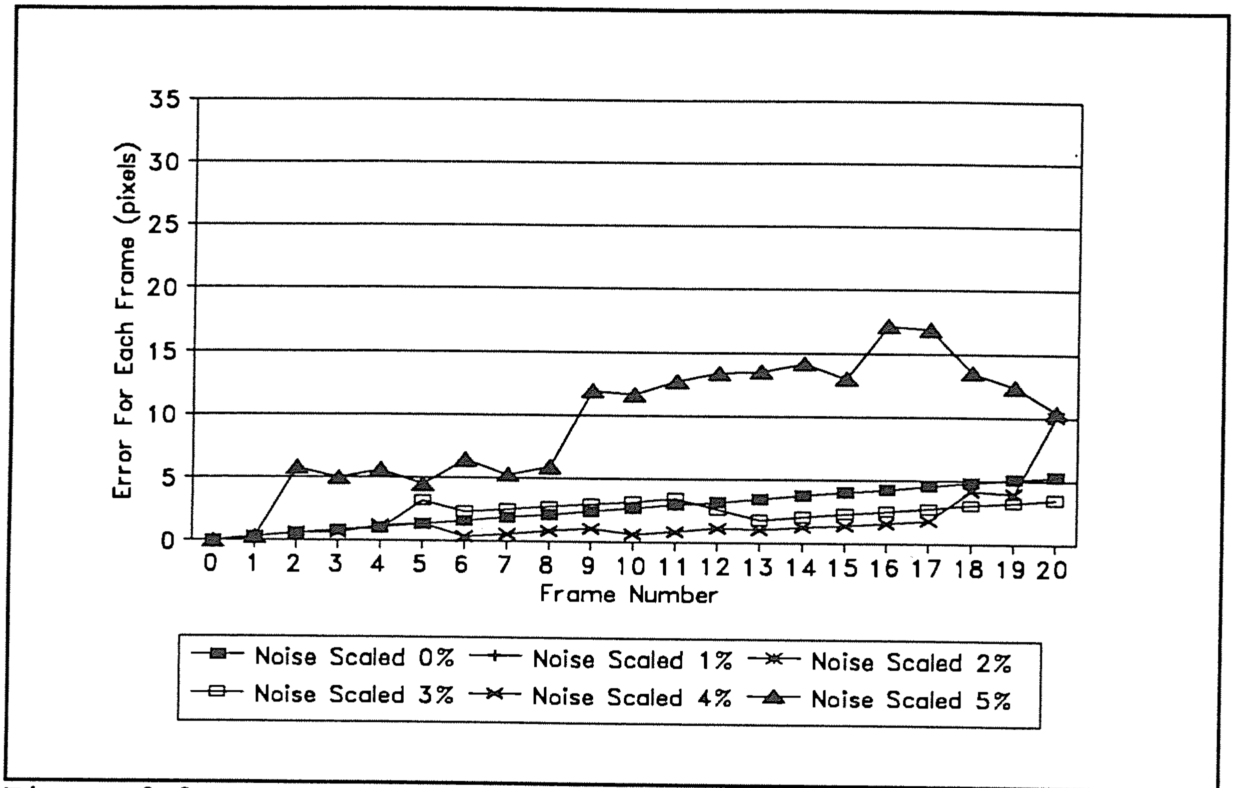


Figure 8-3a Performance of strategy zero when noise is added to each frame

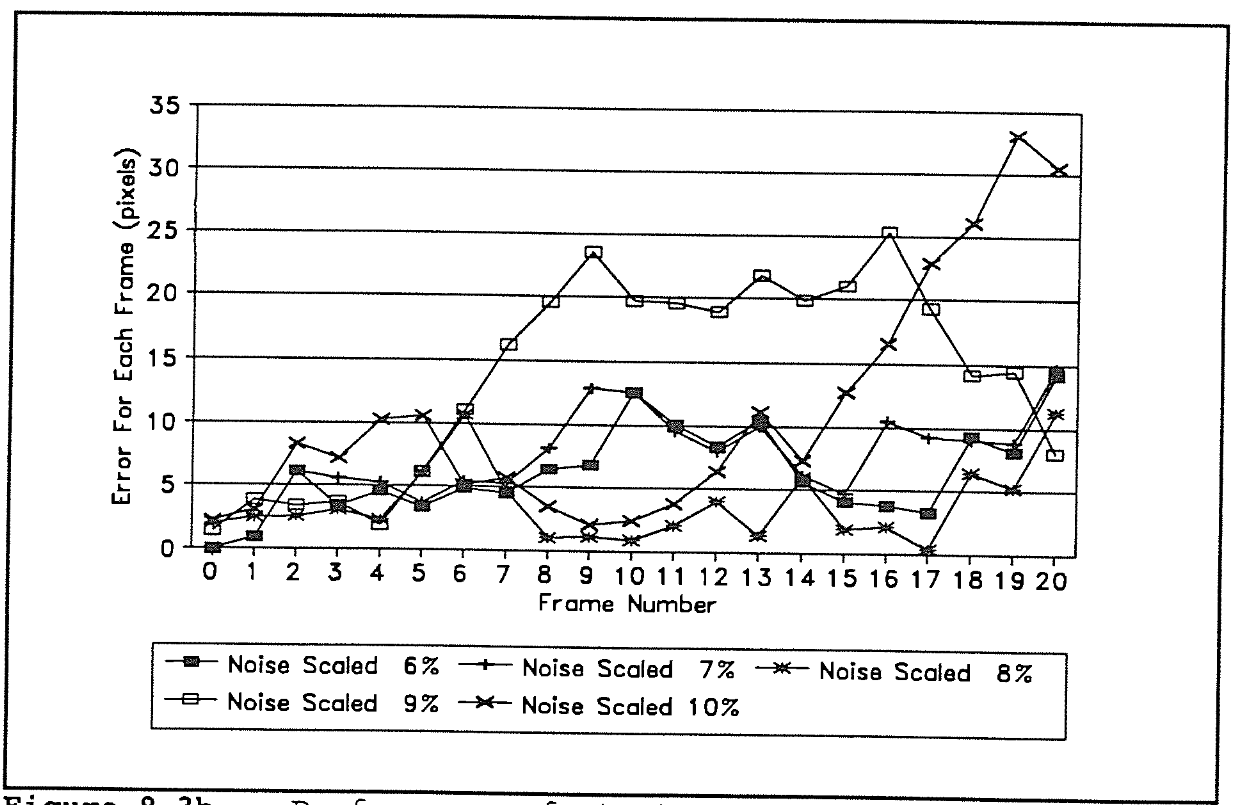


Figure 8-3b Performance of strategy zero when noise is added to each frame

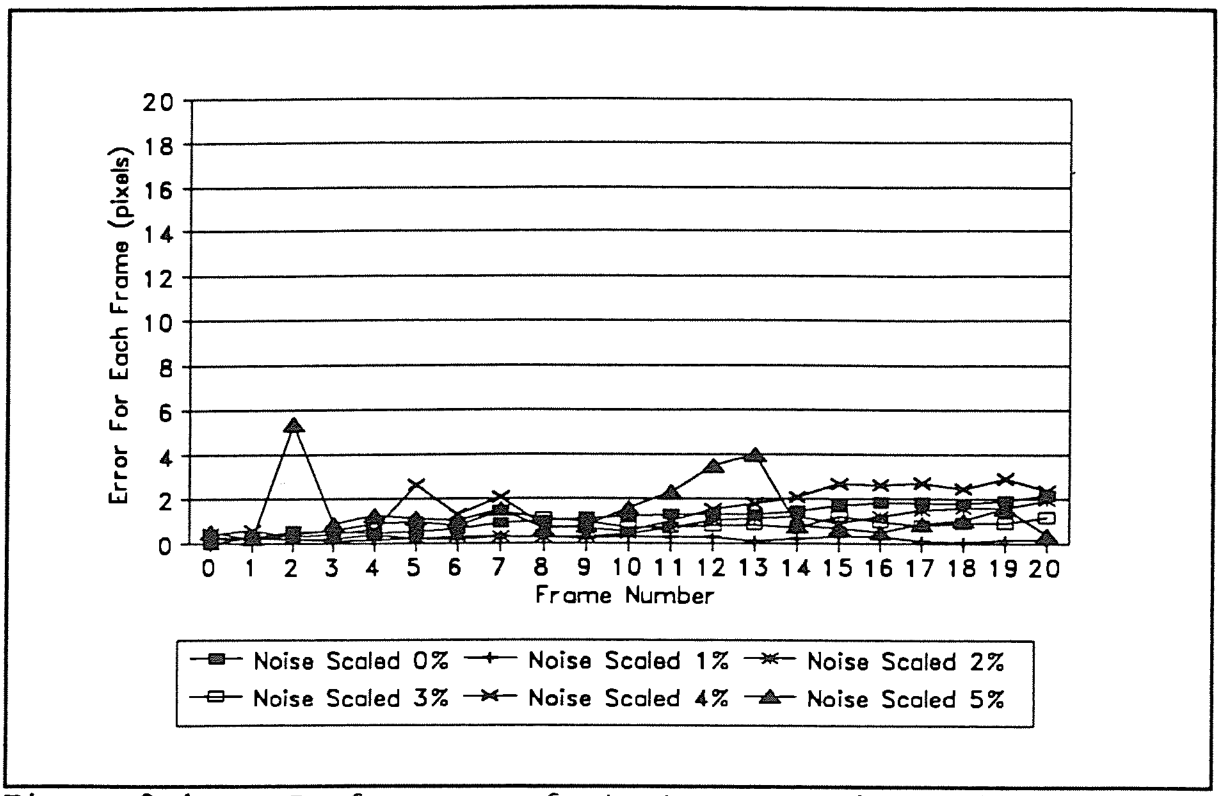


Figure 8-4a Performance of strategy one when noise is added to each frame

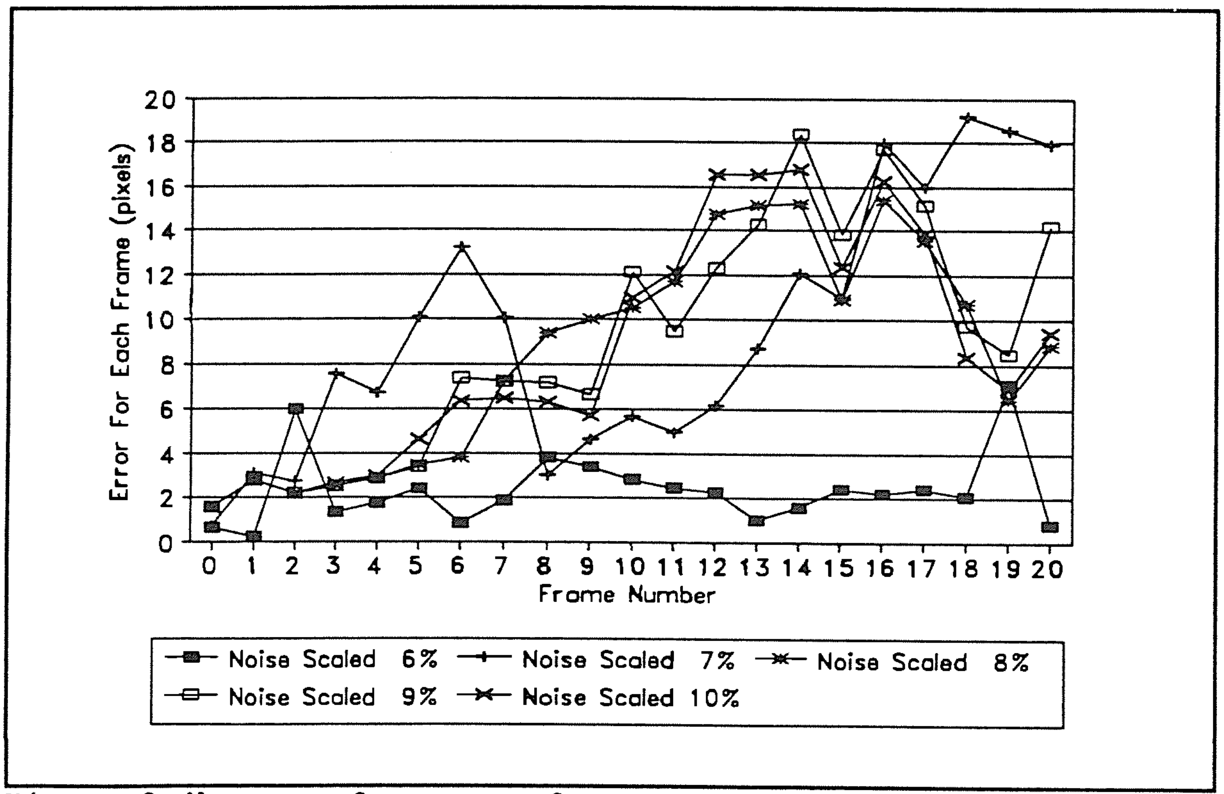


Figure 8-4b Performance of strategy one when noise is added to each frame

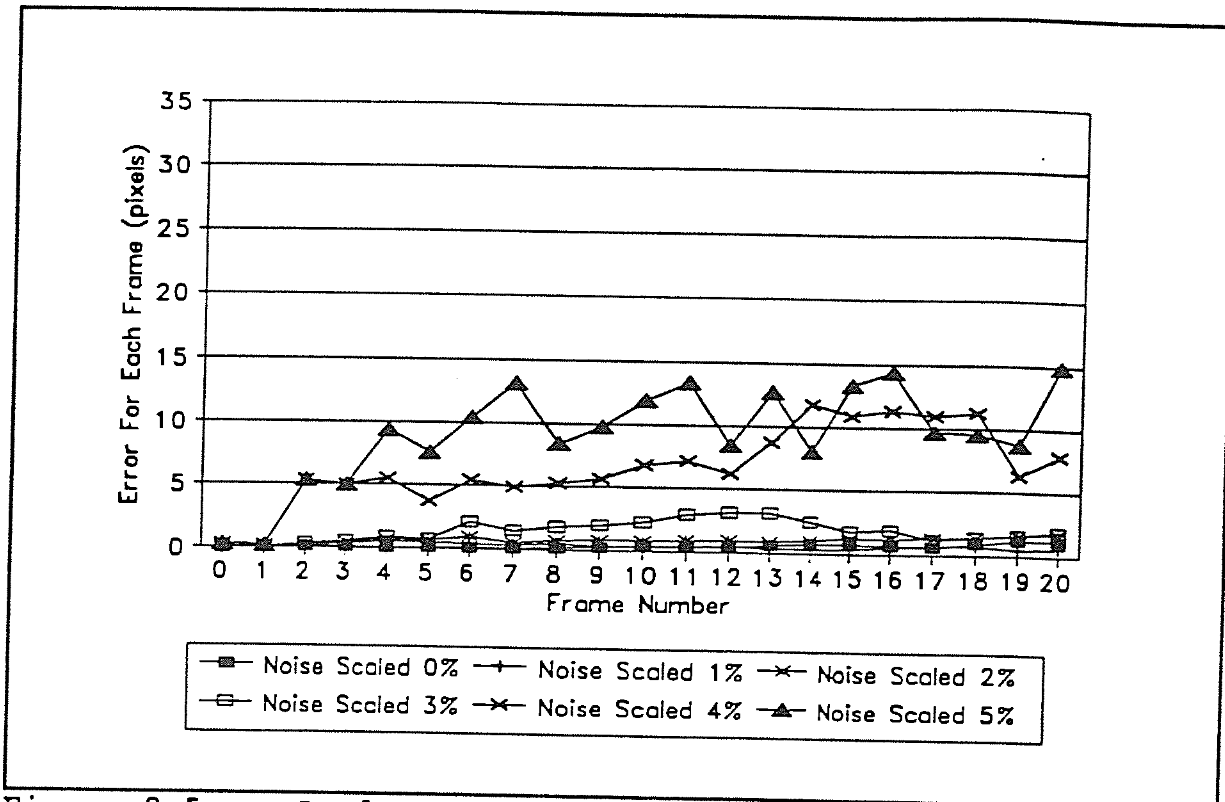


Figure 8-5a Performance of strategy three when noise is added to each frame

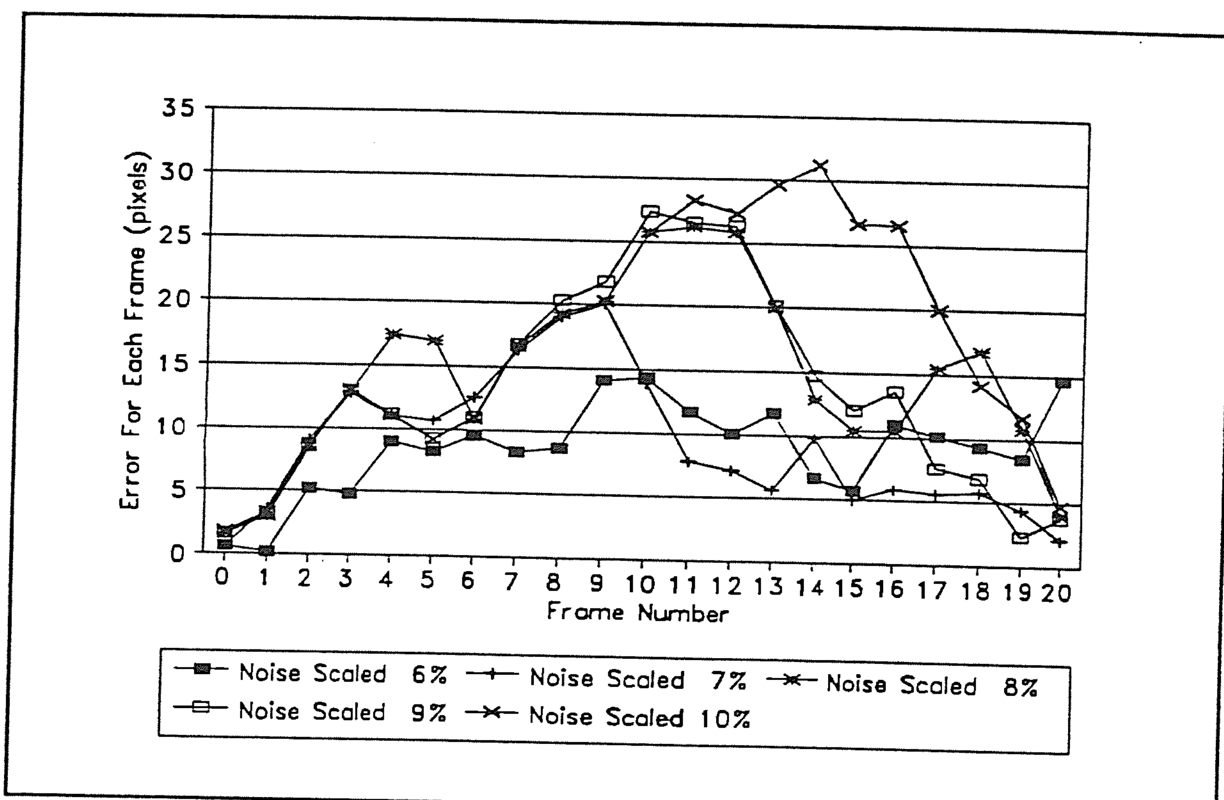


Figure 8-5b Performance of strategy three when noise is added to each frame

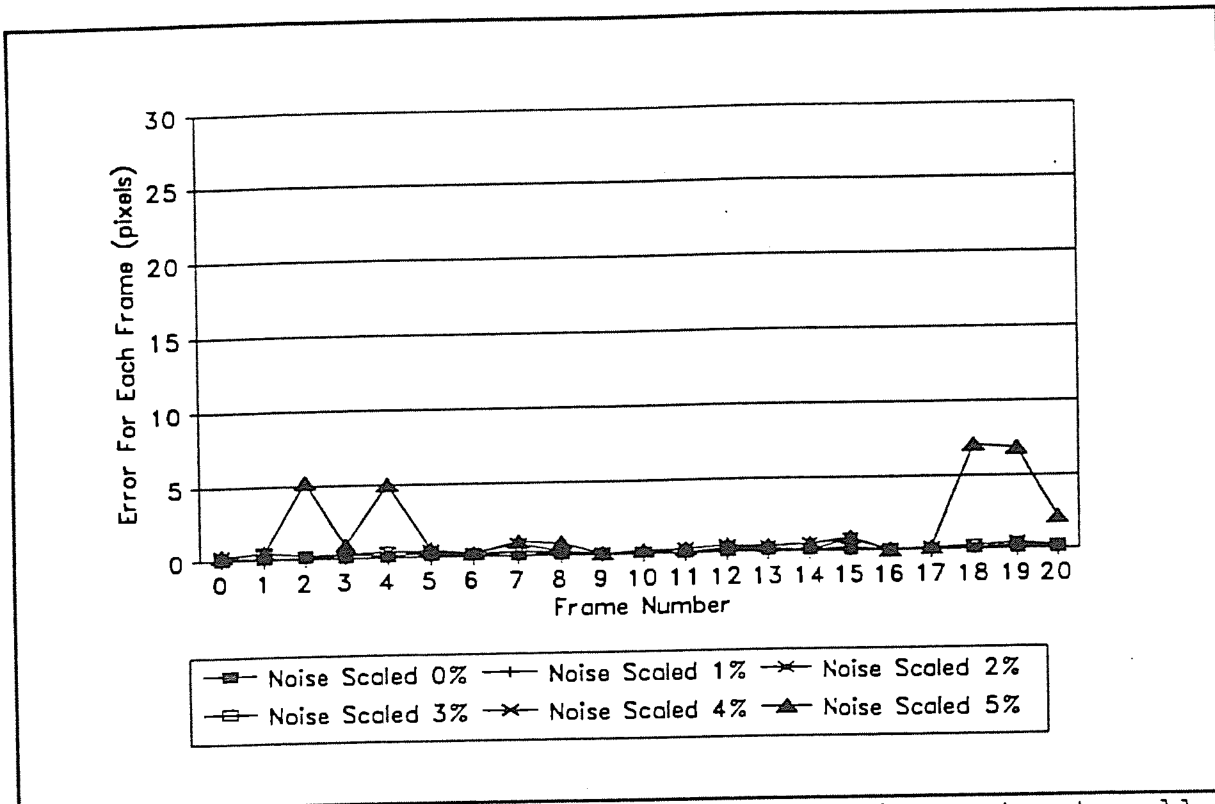


Figure 8-6a Performance of strategy five when noise is added to each frame

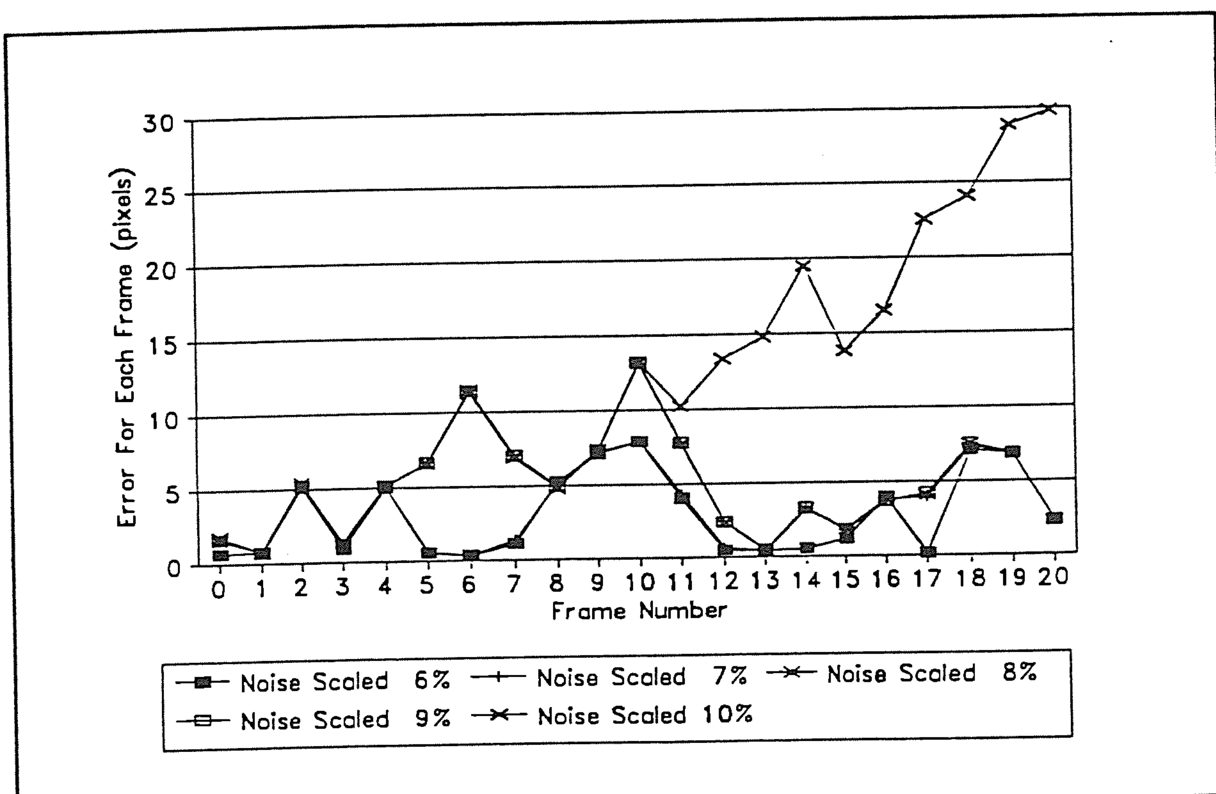


Figure 8-6b Performance of strategy five when noise is added to each frame

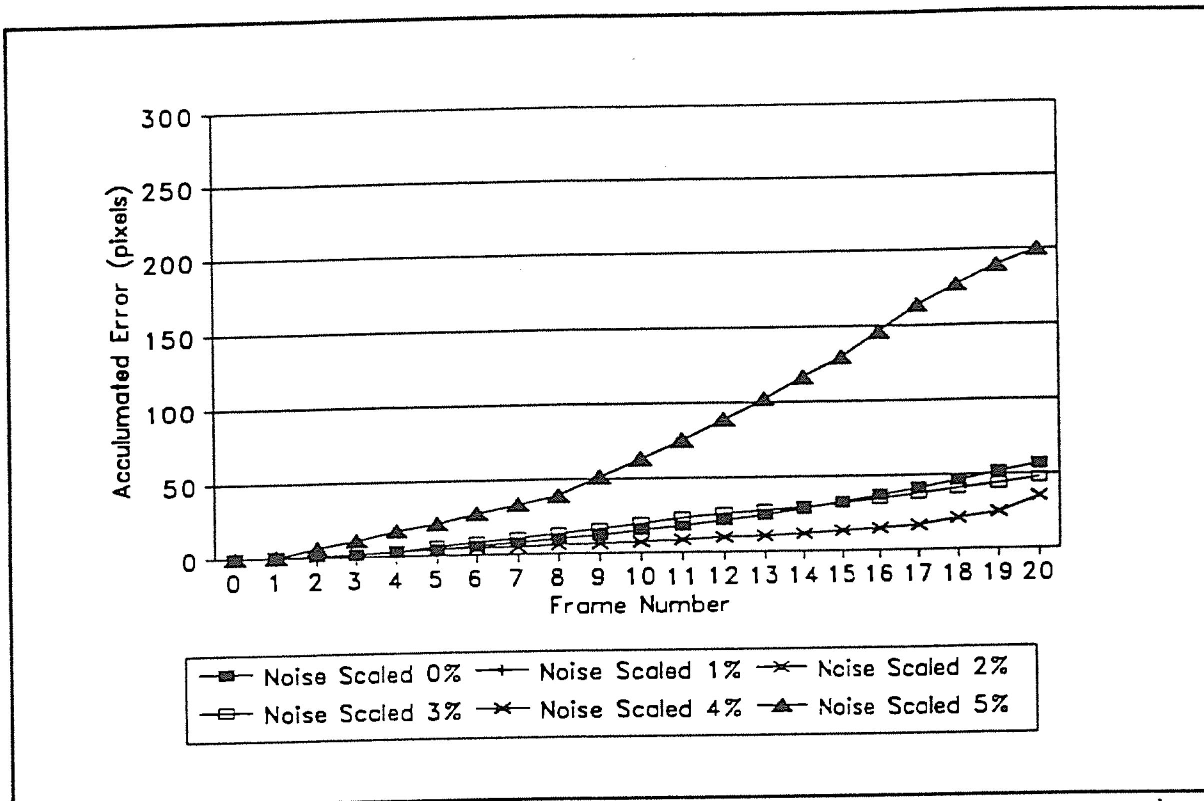


Figure 8-7a   Accumulated tracking error of strategy zero when noise is added to each frame

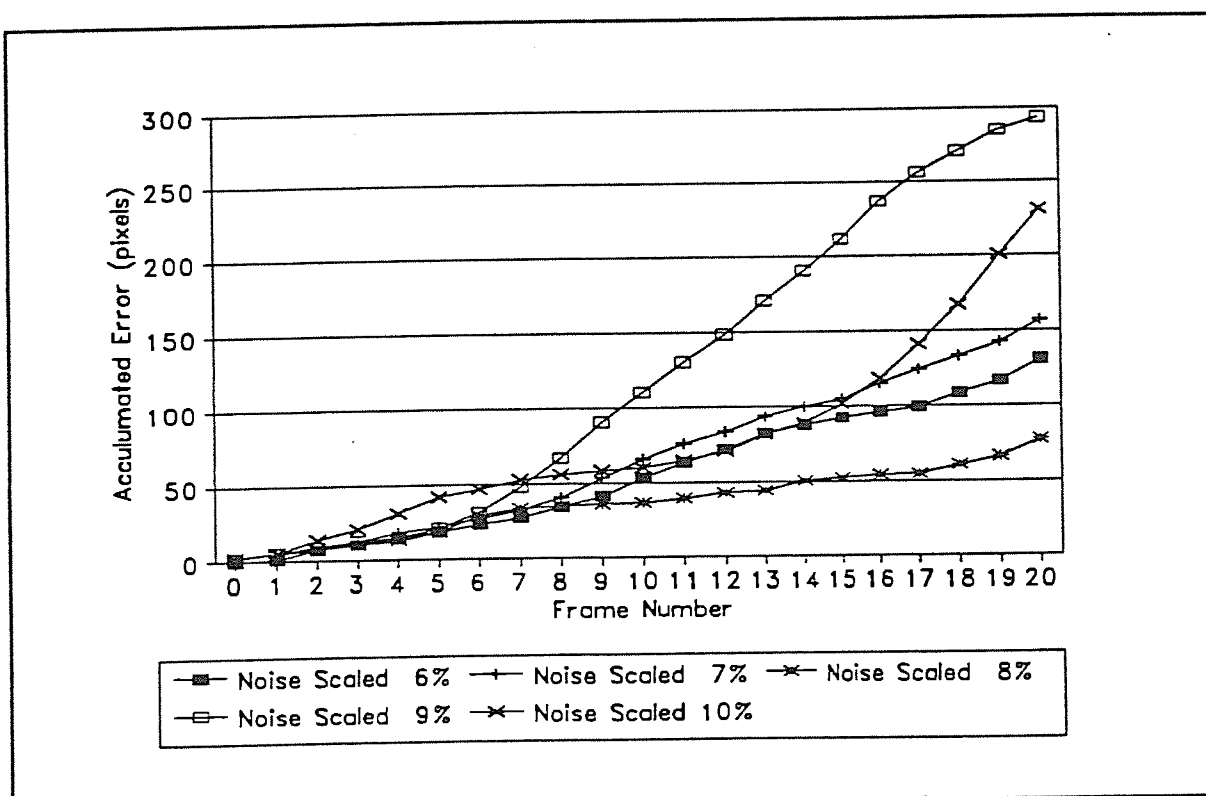


Figure 8-7b   Accumulated tracking error of strategy zero when noise is added to each frame

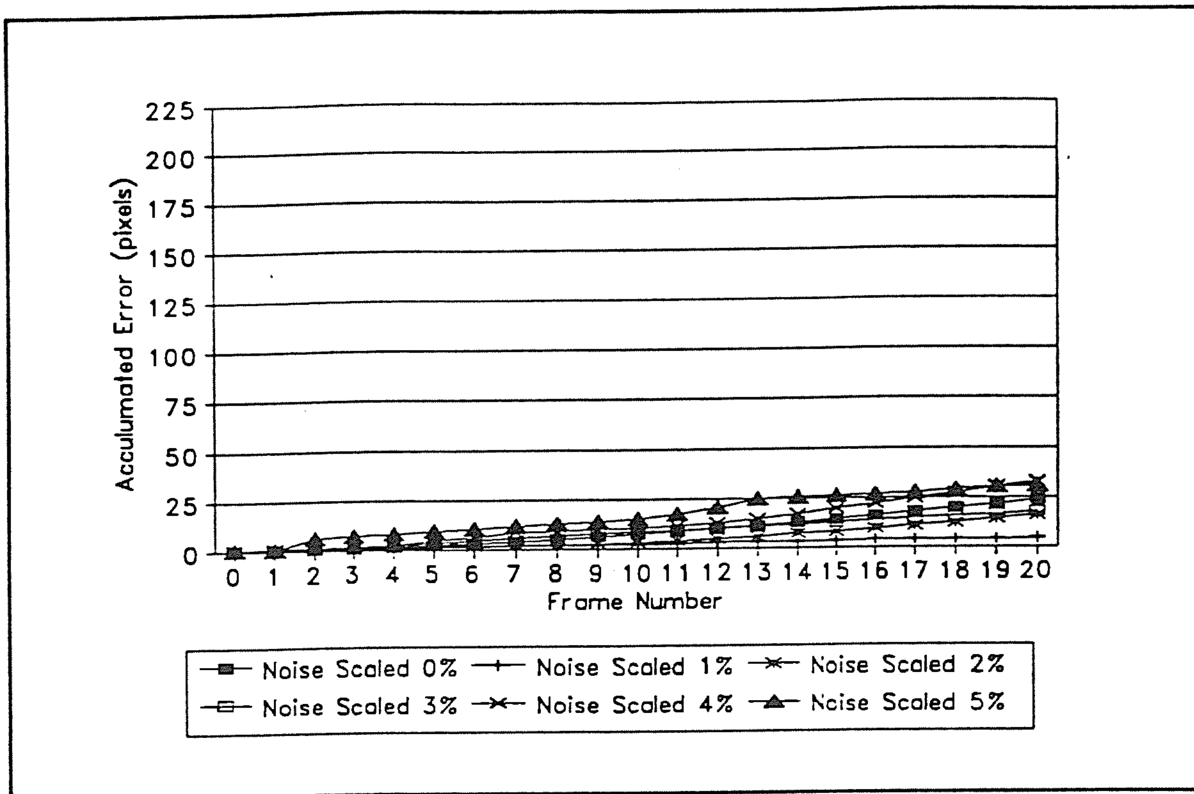


Figure 8-8a Accumulated tracking error of strategy one when noise is added to each frame

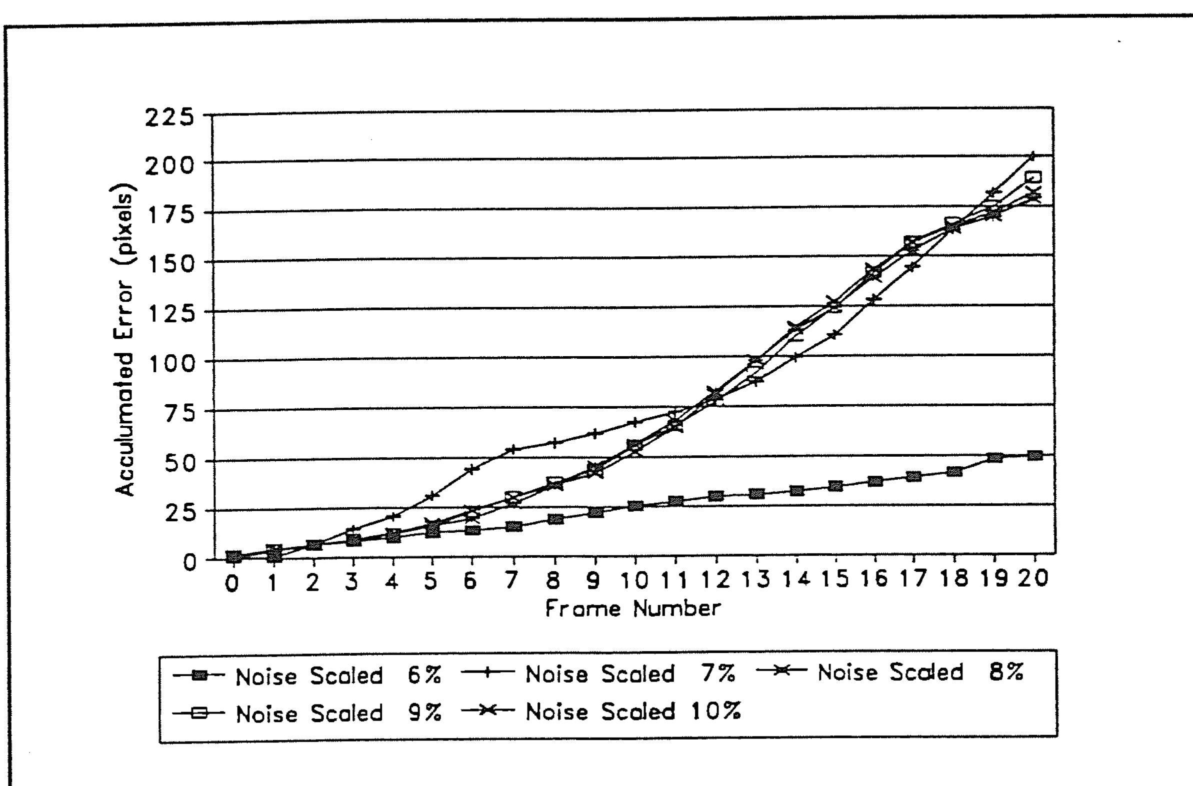


Figure 8-8b Accumulated tracking error of strategy one when noise is added to each frame

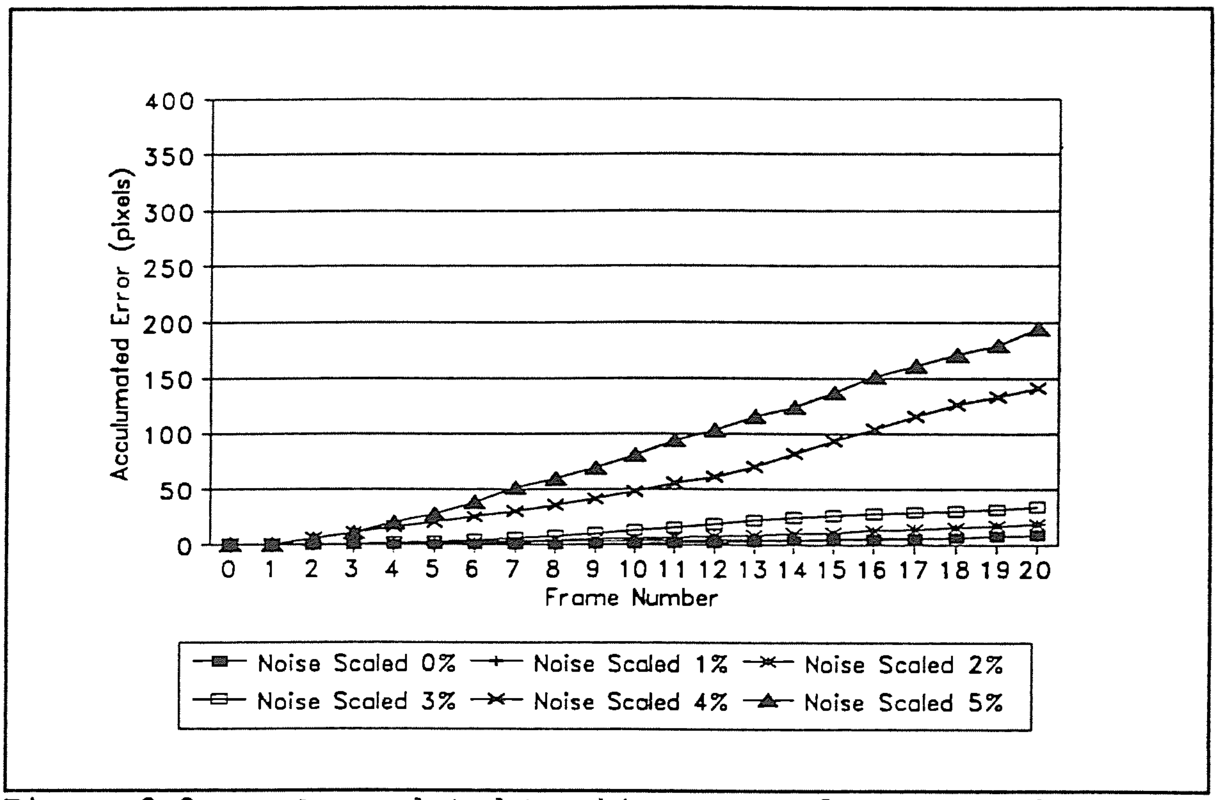


Figure 8-9a Accumulated tracking error of strategy three when noise is added to each frame

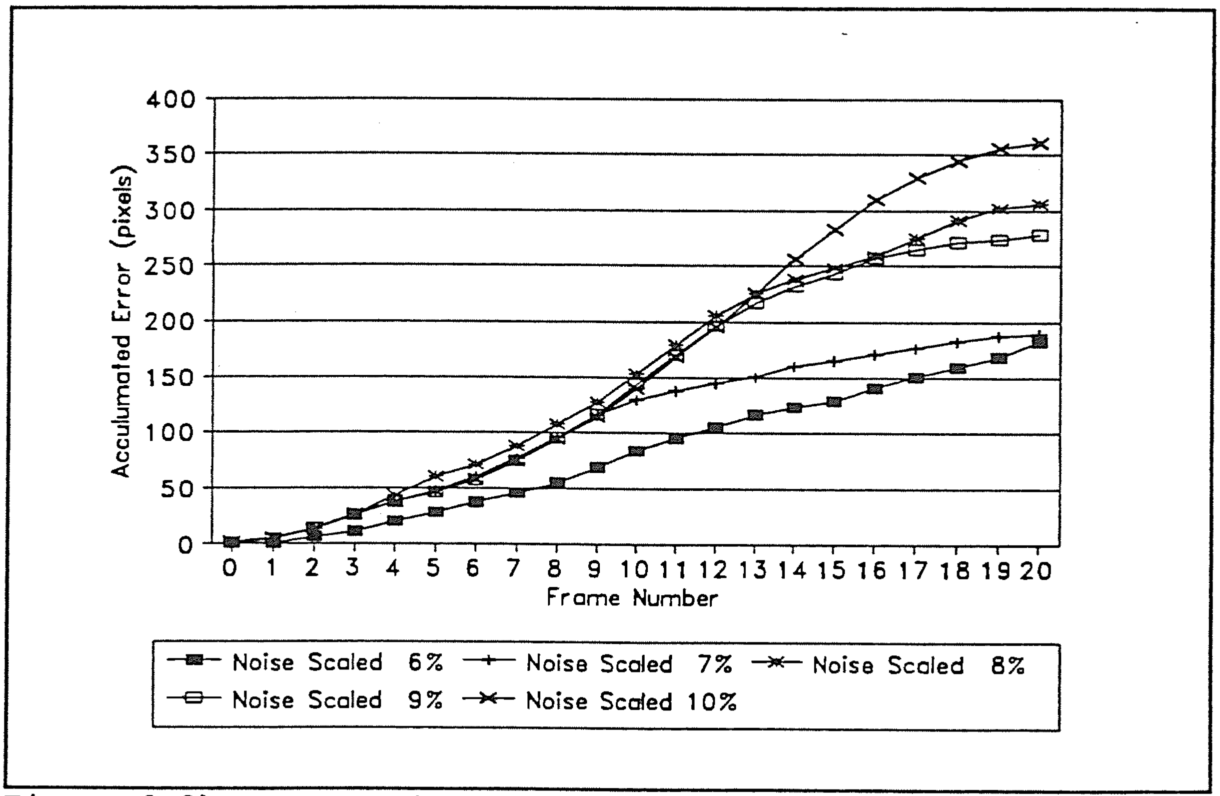


Figure 8-9b Accumulated tracking error of strategy three when noise is added to each frame

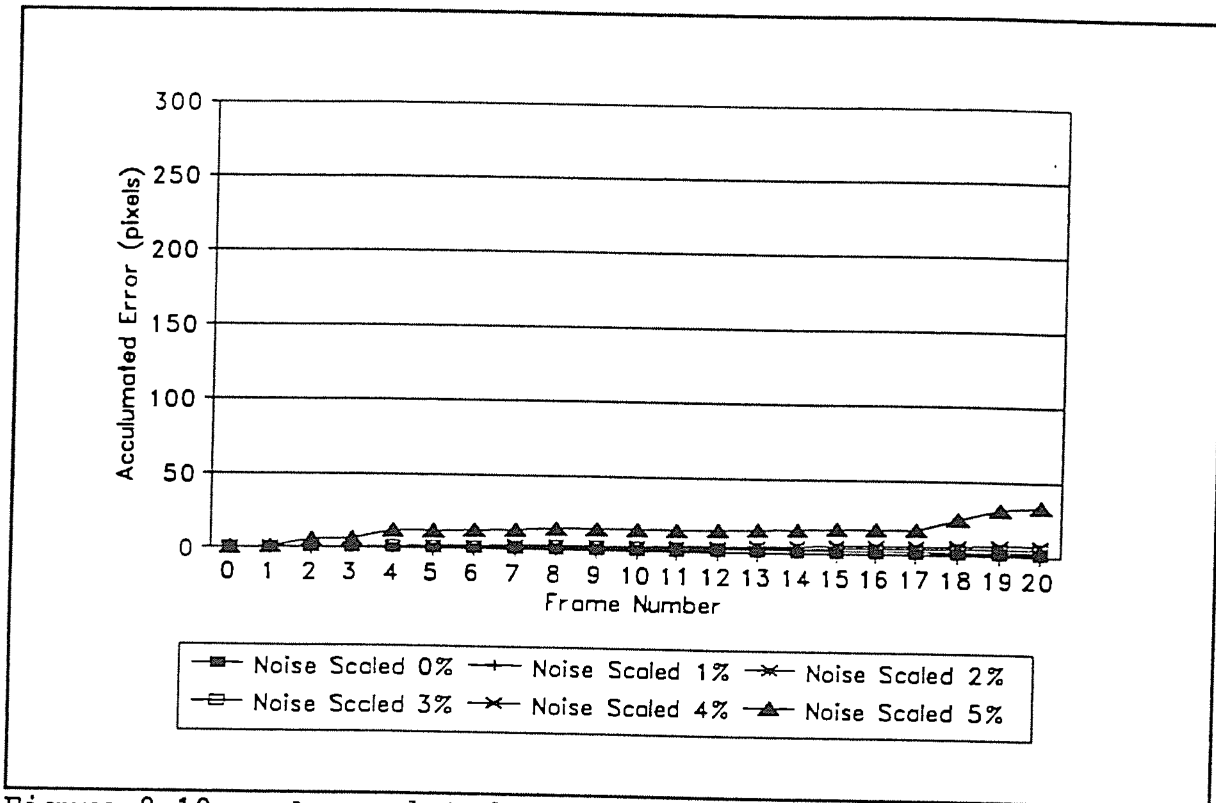


Figure 8-10a Accumulated tracking error of strategy five when noise is added to each frame

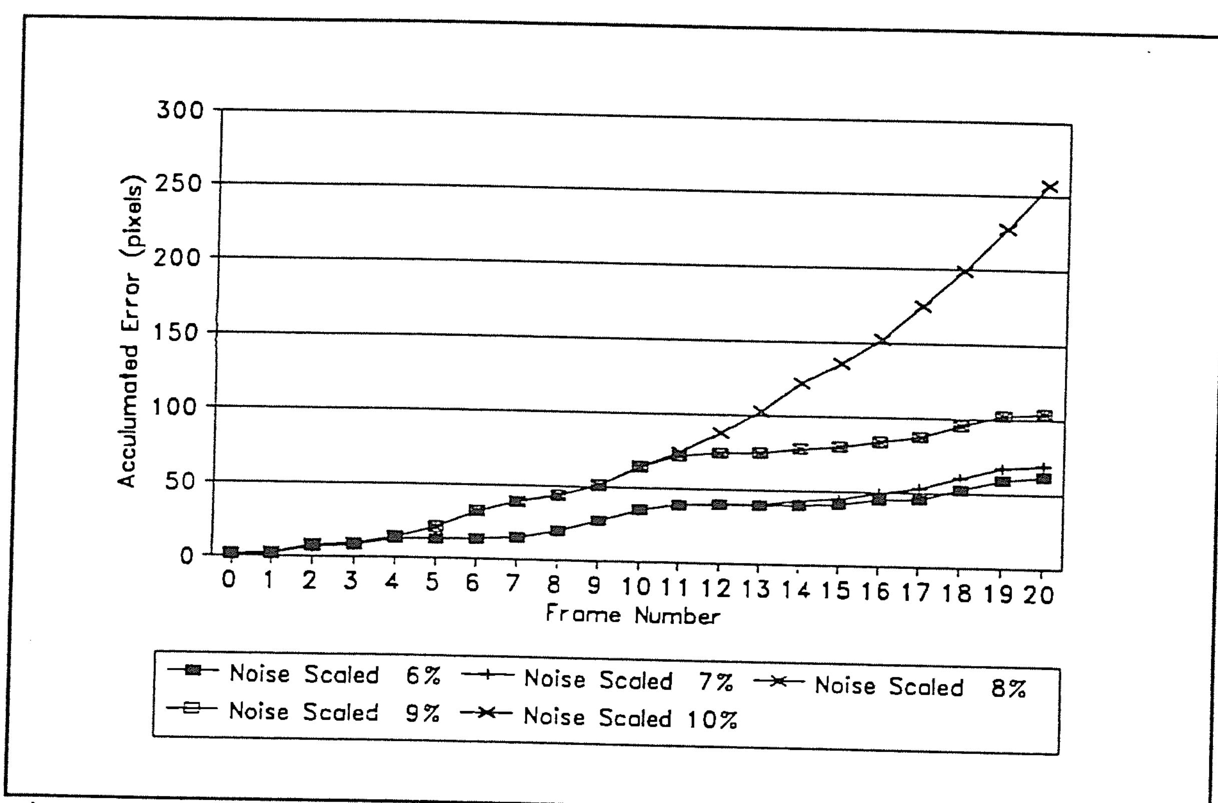


Figure 8-10b Accumulated tracking error of strategy five when noise is added to each frame

Table 8-13 Error (pixels) of the point tracked to the 20th frame

	Noise scaling factor										
strategy	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0	5.405	5.405	5.405	3.571	10.034	10.409	14.154	14.742	5.236	14.465	33.092
1	2.18	0.133	1.949	1.195	2.379	0.377	0.788	17.926	10.719	9.692	8.342
3	1.186	0.624	1.737	1.889	7.918	14.927	14.713	2.145	16.877	6.923	14.171
5	0.055	0.07	0.147	0.234	0.234	2.153	2.33	2.33	7.543	7.543	24.119

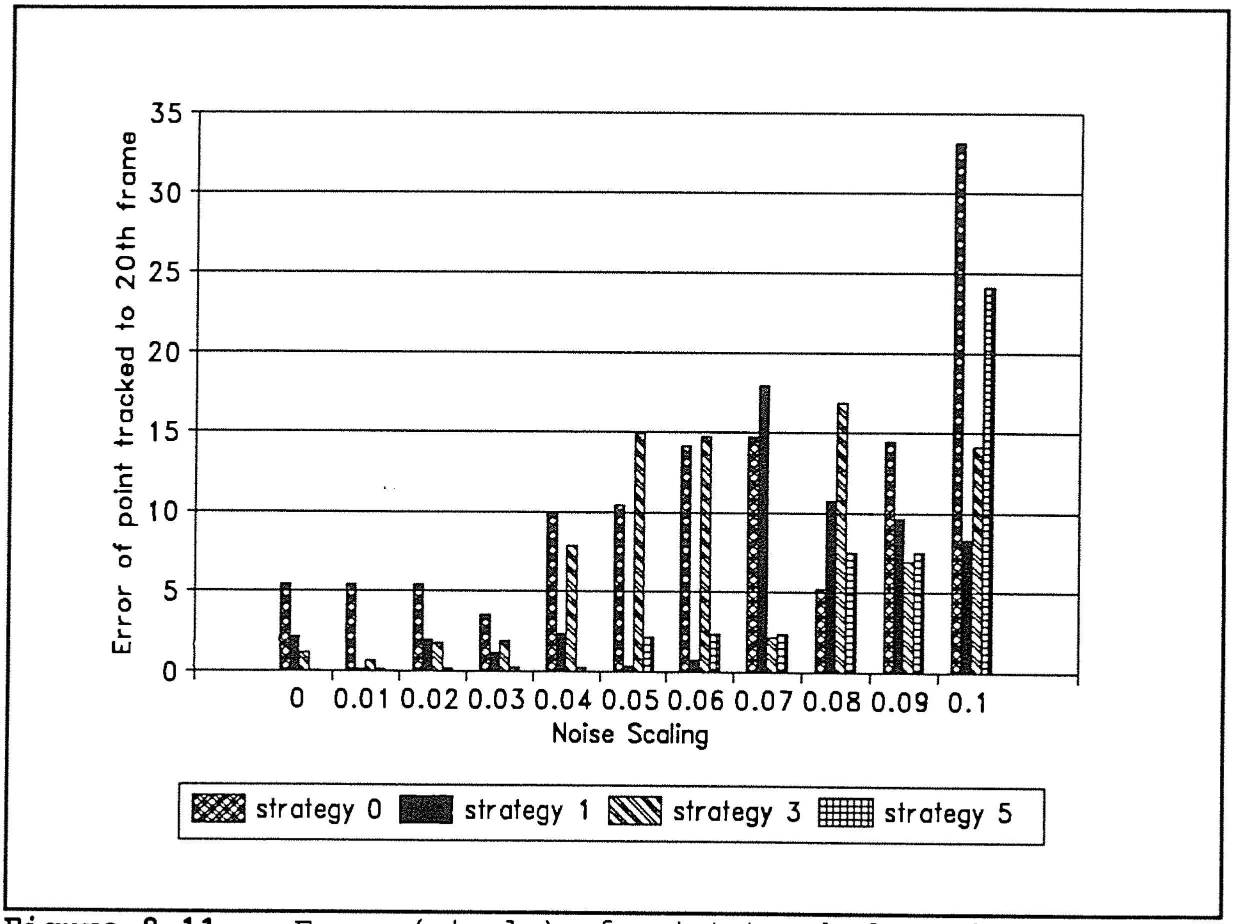


Figure 8-11 Error (pixels) of point tracked to the 20th frame

Table 8-14 Accumulated Error (pixels) over the 20 frames

	Noise scaling factor										
strategy	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0	56.753	56.753	56.753	47.269	34.772	199.835	130.885	157.102	66.452	285.125	200.551
1	23.912	4.201	15.975	17.712	32.577	30.752	49.394	200.077	163.785	166.819	165.598
3	8.86	8.333	18.515	33.916	141.148	194.690	183.044	189.395	291.735	272.162	344.035
5	1.047	1.473	2.806	5.102	7.053	33.357	61.507	69.250	95.329	95.283	199.187

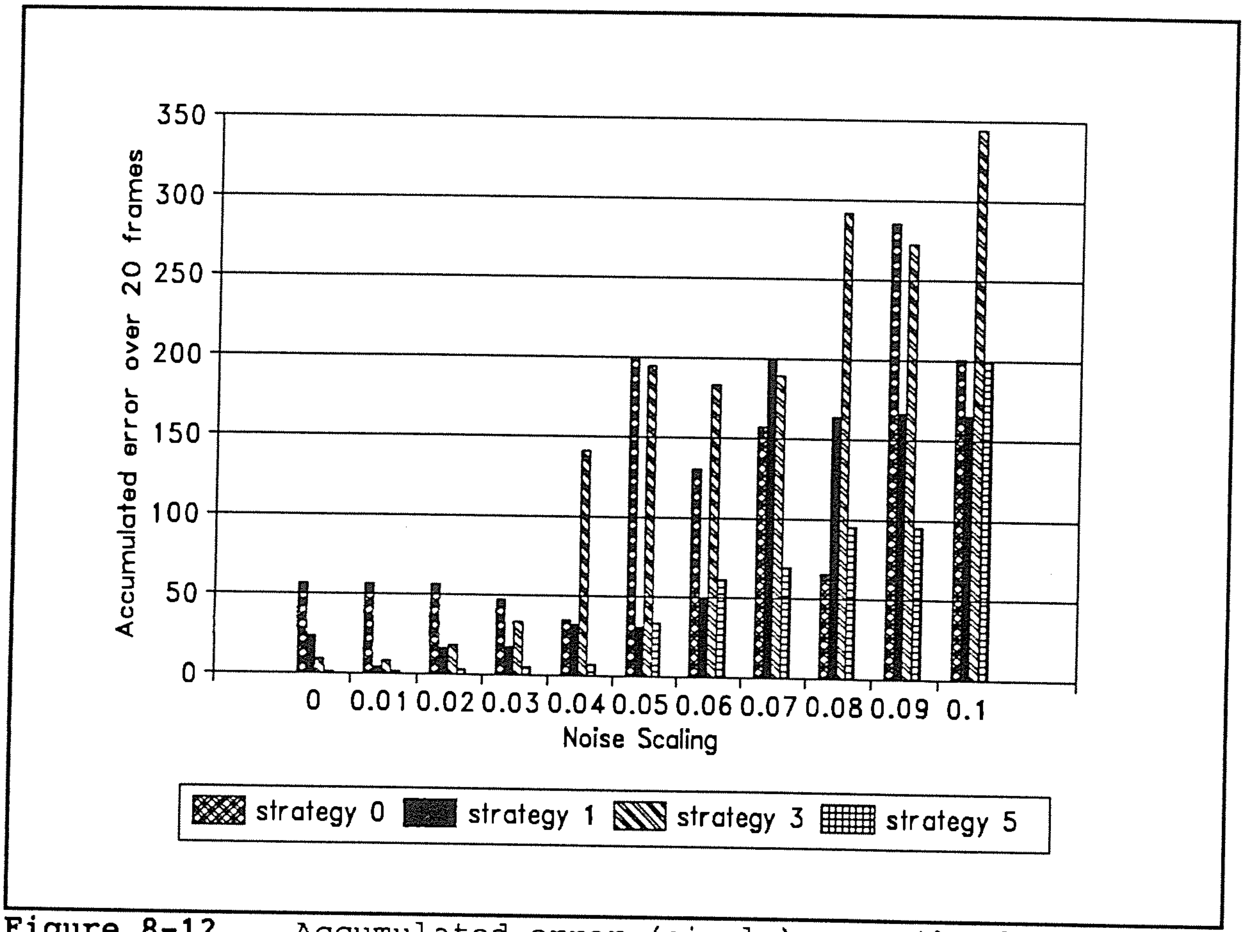


Figure 8-12 Accumulated error (pixels) over the 20 frames

Chapter 9
Practical results

9.1 Including a sub-pixel location search in the image subtraction statistic

Strategy 1 (sub-pixel tracking by translating a frame and matching to the previous frame) and strategy 3 (sub-pixel tracking by translating a frame and matching to the next frame) were implemented in the prototype package to track spinal movement. Strategy 0 (whole pixel tracking) was the original tracking strategy used by the prototype package so was evaluated with no modifications.

The sum of radial errors for each vertebra for each frame and each strategy is shown in Table 9-1 and graphed in Figures 9-1 to 9-6. The smoothing filter used to reduce the noise from secondary emission from the image intensifier was a median filter rank 5 [Jähne, 1991]. The smoothing filter 'blurs' the image which effectively reduces the noise in the image but also reduced the image clarity.

The major difference between the runs reported in this Chapter and those performed in Chapter 8 was that the image sequence being tracked through was a real captured cineagraphic sequence not a computer simulated sequence. This meant features in the image would be moving in an independent fashion, undergoing both translation and rotation, and interacting with other features (e.g. two vertebrae jamming together while the other vertebrae in the spine rotate freely).

The results show that smoothing the image does little if anything to improve the results of the tracking algorithm. It may indeed be that the loss of image definition outweighs the benefit gained from smoothing or blurring the noise. Despite the promising performance of strategy 1 in Chapter 8 it in fact made no improvement on the original strategy 0 and often performed markedly worse for the real image sequence. During the tests in Chapter 8 the test sequence images had translation components only, no rotation and no feature interaction. It is thought that strategy 1 no longer performed as well as strategy 0 because the window match would no longer be as 'pure' as in the test sequence and when windows centred on the outskirts of the search window underwent sub-pixel translation and were compared back to the window being searched for they would, in some cases, give a better match than the correct result. Strategy 3 performed well giving for most frames and vertebrae errors of 4 pixels or less. This may be because only the window being searched for was translated reducing the chance of an outlying point being manipulated and becoming a better match. As strategy 3 performed as well as, and in most cases a lot better than, strategy 0 the next step was to examine the difference in spinal measurements to determine whether it was necessary to permanently update the prototype tracking algorithm.

Table 9-1 Sum of radial errors for each vertebra

RADIAL ERROR FOR EACH VERTEBRA WHEN TRACKED USING STRATEGY ZERO (WHOLE-PIXEL) AND SMOOTHING

Frame No	C1	C2	C3	C4	C5	C6
0	6.479	9.359	10.797	10.358	9.641	10.472
1	8.715	10.521	11.723	10.256	10.084	11.035
2	8.122	11.729	10.227	9.398	5.650	9.537
3	6.886	11.768	9.301	9.398	8.243	10.870
4	8.122	8.236	9.301	9.049	5.064	9.944
5	7.236	9.162	7.650	7.708	5.064	15.180
6	4.650	8.414	7.886	5.472	9.078	8.000
7	5.650	11.064	6.886	6.472	5.657	6.657
8	6.064	8.414	7.064	6.064	6.064	7.886
9	5.650	9.064	6.064	5.243	6.064	8.799
10	5.236	7.236	5.414	4.414	5.650	6.606
11	3.414	8.650	5.000	4.414	3.414	7.414
12	3.236	7.236	5.000	3.414	2.000	4.650
13	4.828	4.000	4.243	3.000	5.000	1.414
14	1.414	3.414	4.828	2.000	2.000	2.000
15	0.000	1.414	3.000	0.000	3.000	3.000
16	0.000	0.000	0.000	0.000	0.000	0.000
17	1.000	1.414	0.000	0.000	0.000	6.385
18	4.828	3.000	2.000	2.414	2.000	8.000
19	2.828	3.828	3.414	3.414	3.000	8.062
20	5.576	6.886	4.414	5.243	4.414	9.000
21	7.236	4.650	7.064	5.243	3.000	10.476
22	5.414	3.236	6.650	6.064	3.236	9.414
23	6.236	6.162	4.828	5.650	3.236	6.385
24	8.650	6.479	6.064	5.064	4.243	7.325
25	8.749	4.650	7.479	6.472	2.828	6.799
26	11.020	7.064	7.893	7.886	3.000	7.325
27	10.780	8.122	8.301	7.398	3.828	10.544
28	10.143	9.492	9.227	8.634	3.000	9.544
29	10.487	9.049	10.677	7.236	4.828	10.476
30	6.768	8.301	12.609	7.236	6.236	12.713
31	6.768	7.479	12.609	10.236	3.414	10.625

RADIAL ERROR FOR EACH VERTEBRA WHEN TRACKED USING STRATEGY ZERO (WHOLE-PIXEL) AND NO SMOOTHING

Frame No	C1	C2	C3	C4	C5	C6
0	5.828	10.602	14.556	13.078	8.256	125.517
1	7.236	10.182	13.690	12.913	10.625	104.241
2	7.650	10.521	12.469	11.498	4.650	81.630
3	6.650	10.768	11.055	10.906	8.434	66.610
4	5.828	7.813	13.291	10.906	3.650	46.357
5	4.650	8.739	12.759	8.944	4.650	35.108
6	2.414	7.886	13.262	8.122	8.064	19.539
7	4.828	10.301	12.425	8.715	5.243	14.728
8	3.414	6.650	9.708	7.479	3.828	14.595
9	4.000	8.064	8.479	5.243	6.064	11.899
10	3.000	5.650	5.828	4.828	5.650	7.243
11	4.828	7.650	5.828	4.828	3.414	8.414
12	4.236	6.236	4.414	3.828	2.000	6.243
13	4.414	4.000	2.828	3.000	4.236	1.000
14	2.236	3.414	4.414	2.000	2.000	3.000
15	0.000	1.414	3.000	0.000	3.000	1.000
16	0.000	0.000	0.000	0.000	0.000	0.000
17	4.000	1.414	0.000	0.000	0.000	1.000
18	5.657	3.000	2.000	3.000	2.000	2.000
19	5.414	3.414	3.414	3.000	1.000	7.325
20	3.828	6.064	3.414	4.828	5.650	8.708
21	6.236	4.650	5.650	4.414	2.414	14.460
22	5.828	5.650	7.708	3.828	2.000	13.414
23	8.236	8.405	6.657	7.886	2.000	12.000
24	7.236	6.472	5.243	6.472	5.064	16.866
25	8.398	5.414	7.893	7.886	2.414	16.343
26	8.606	7.650	7.256	7.064	5.657	12.630
27	10.479	9.301	8.434	7.813	4.650	13.050
28	7.606	11.596	9.122	8.064	4.243	12.050
29	9.576	10.700	10.886	5.414	6.657	15.453
30	8.842	11.078	8.886	7.236	4.236	17.272
31	8.842	11.256	8.886	8.715	2.000	15.450

RADIAL ERROR FOR EACH VERTEBRA WHEN TRACKED USING STRATEGY ONE (SUB-PIXEL) AND SMOOTHING

Frame No	C1	C2	C3	C4	C5	C6
0	13.166	15.214	18.188	16.565	14.282	32.213
1	12.882	12.040	15.165	17.940	16.518	24.621
2	13.091	11.492	15.303	18.016	15.060	24.638
3	13.091	12.405	13.227	19.321	14.045	25.848
4	13.310	11.064	11.670	16.595	14.156	27.756
5	12.074	12.434	11.004	13.062	14.710	29.430
6	12.842	10.906	10.004	14.610	17.088	21.534
7	12.246	9.670	9.405	14.088	15.092	21.372
8	12.166	9.625	7.243	12.180	13.227	26.225
9	11.981	9.434	7.064	7.405	11.962	27.387
10	7.064	10.906	5.650	7.991	11.335	24.977
11	6.479	8.071	5.064	6.650	9.561	25.191
12	4.414	7.893	5.576	6.064	7.236	23.857
13	3.000	5.650	4.828	3.000	7.000	15.119
14	3.414	5.828	5.414	3.000	3.414	10.153
15	0.000	1.000	3.000	0.000	4.000	3.000
16	0.000	0.000	0.000	0.000	0.000	0.000
17	3.414	2.000	2.000	1.000	1.414	2.000
18	5.657	6.000	5.000	3.828	4.000	3.000
19	6.064	5.236	4.243	4.828	6.000	9.122
20	9.398	6.606	7.479	6.434	6.650	8.485
21	10.227	9.842	8.606	8.071	7.842	11.109
22	11.927	10.434	7.236	8.828	7.634	12.677
23	12.705	10.842	7.893	10.886	9.004	9.211
24	13.807	13.842	8.893	10.628	12.359	9.848
25	14.424	14.725	7.472	11.717	10.729	9.078
26	14.392	13.714	7.886	13.211	9.078	11.442
27	14.914	11.901	7.842	9.831	9.071	15.416
28	18.389	11.659	7.576	13.801	11.508	17.329
29	20.675	15.280	7.472	14.634	16.516	19.483
30	19.098	11.901	10.078	16.680	21.323	18.378
31	19.545	11.245	11.071	16.886	21.477	17.360

RADIAL ERROR FOR EACH VERTEBRA WHEN TRACKED USING STRATEGY ONE (SUB-PIXEL) AND NO SMOOTHING

Frame No	C1	C2	C3	C4	C5	C6
0	18.484	24.408	22.100	14.609	9.301	34.162
1	16.705	22.387	22.797	13.861	8.301	30.089
2	15.733	23.308	19.683	15.646	7.828	29.560
3	14.705	20.757	17.457	15.232	6.576	26.105
4	12.634	20.658	17.594	11.828	10.064	19.453
5	13.123	19.679	15.741	10.359	8.472	20.419
6	14.547	18.485	14.657	9.285	10.965	13.842
7	14.049	16.642	14.252	11.797	9.842	15.496
8	11.314	15.228	10.831	12.877	6.650	14.505
9	11.424	11.405	11.153	10.768	5.162	13.166
10	10.078	10.813	6.886	8.991	3.236	13.177
11	7.991	11.285	5.886	6.472	3.828	10.870
12	8.472	9.670	4.828	7.301	4.414	8.842
13	5.650	6.472	2.414	6.162	5.000	4.650
14	3.414	3.828	3.414	5.236	3.414	4.828
15	0.000	1.414	3.414	2.000	3.000	1.000
16	0.000	0.000	0.000	0.000	0.000	0.000
17	3.000	1.000	0.000	0.000	2.000	7.083
18	5.657	6.000	2.000	4.828	6.479	12.470
19	3.414	6.650	5.657	6.243	7.708	15.912
20	7.576	9.064	7.301	6.434	6.236	16.533
21	8.576	10.657	11.055	8.670	7.650	16.134
22	11.359	12.064	8.670	8.020	5.650	15.857
23	12.472	13.487	7.650	7.479	6.991	21.353
24	12.472	15.828	5.414	5.243	5.162	25.170
25	14.678	18.668	5.414	8.236	7.576	24.125
26	13.339	19.733	9.398	7.773	7.236	25.136
27	15.029	20.694	10.606	12.182	11.004	25.353
28	16.843	24.486	10.358	12.245	9.842	28.050
29	21.657	27.758	10.991	18.543	10.243	29.055
30	19.245	24.678	13.526	16.807	15.061	39.590
31	20.487	26.371	16.810	21.239	14.222	43.646

RADIAL ERROR FOR EACH VERTEBRA WHEN TRACKED USING STRATEGY THREE (SUB-PIXEL) AND SMOOTHING

Frame No	C1	C2	C3	C4	C5	C6
0	9.153	4.828	3.000	4.414	3.000	5.099
1	5.828	4.236	3.000	5.650	5.000	9.083
2	5.000	4.000	4.650	3.236	6.243	6.123
3	5.000	7.414	4.828	3.828	5.991	7.083
4	4.000	4.414	5.886	3.414	2.828	7.497
5	5.000	5.828	2.000	1.414	3.000	14.185
6	3.000	4.414	2.414	2.236	4.243	7.403
7	5.243	6.236	2.414	2.828	2.000	10.232
8	2.000	3.414	3.000	4.236	0.000	9.708
9	1.000	4.650	3.828	4.236	2.000	9.153
10	3.000	3.000	1.000	2.414	5.414	7.000
11	3.828	4.414	1.000	3.414	4.828	7.245
12	3.000	4.414	1.000	2.000	2.000	4.236
13	2.000	2.000	4.000	4.414	3.000	2.236
14	2.000	2.414	3.414	3.000	0.000	0.000
15	0.000	1.414	3.000	0.000	3.000	1.000
16	0.000	0.000	0.000	0.000	0.000	0.000
17	2.000	1.000	0.000	0.000	0.000	2.000
18	2.000	3.236	3.828	2.414	3.414	0.000
19	0.000	2.000	1.000	1.000	3.414	2.236
20	6.000	3.000	4.000	3.000	4.000	2.414
21	4.162	2.000	3.828	3.000	4.650	3.000
22	6.123	3.000	5.000	2.236	5.576	2.000
23	7.000	3.414	5.414	6.236	5.414	2.000
24	8.828	4.414	6.828	4.414	3.650	3.650
25	9.749	8.434	8.064	7.162	6.064	2.414
26	9.828	7.472	7.479	7.123	5.236	3.000
27	8.099	7.708	7.064	7.236	5.576	3.000
28	10.733	8.236	9.020	8.162	6.414	3.000
29	14.639	11.004	9.078	8.700	4.414	5.650
30	10.555	8.301	11.071	8.576	8.634	3.414
31	8.650	10.561	11.606	11.571	7.479	3.414

RADIAL ERROR FOR EACH VERTEBRA WHEN TRACKED USING STRATEGY THREE (SUB-PIXEL) AND NO SMOOTHING

Frame No	C1	C2	C3	C4	C5	C6
0	8.819	3.236	3.650	1.000	5.650	2.000
1	4.650	4.236	6.650	3.414	6.236	1.414
2	3.828	3.650	5.650	2.828	6.576	2.414
3	3.414	3.000	5.650	1.414	6.472	0.000
4	3.236	1.000	6.064	1.414	5.236	0.000
5	2.236	2.000	4.414	0.000	4.000	1.414
6	3.000	4.000	2.000	1.000	3.236	3.414
7	4.414	4.414	4.000	2.236	3.414	4.236
8	2.000	3.000	3.414	2.236	2.000	1.414
9	1.000	3.650	3.828	1.414	3.414	1.414
10	2.000	1.414	1.000	1.414	3.000	0.000
11	4.414	4.162	1.414	2.414	1.000	1.414
12	3.414	3.162	1.000	0.000	3.000	2.000
13	4.414	2.000	3.000	1.000	4.000	0.000
14	2.414	1.000	3.414	0.000	1.000	0.000
15	0.000	2.236	3.000	0.000	3.000	0.000
16	0.000	0.000	0.000	0.000	0.000	0.000
17	3.000	1.000	0.000	0.000	0.000	1.000
18	1.000	2.236	3.414	1.414	2.414	1.000
19	1.000	1.000	0.000	1.000	2.414	1.000
20	5.414	1.414	1.414	0.000	5.650	0.000
21	4.000	0.000	1.000	0.000	2.828	0.000
22	7.414	0.000	4.828	1.000	5.472	0.000
23	5.000	0.000	2.000	3.650	4.414	1.000
24	8.236	1.000	4.414	3.000	7.064	0.000
25	10.099	3.606	4.414	2.236	3.000	0.000
26	9.236	3.162	5.472	1.000	4.828	0.000
27	7.359	2.236	6.064	3.828	5.064	0.000
28	9.236	2.000	3.414	1.414	3.828	1.000
29	12.083	3.162	3.000	6.064	5.414	0.000
30	10.398	4.162	4.650	3.414	6.064	0.000
31	11.561	4.162	6.991	4.650	4.414	2.000

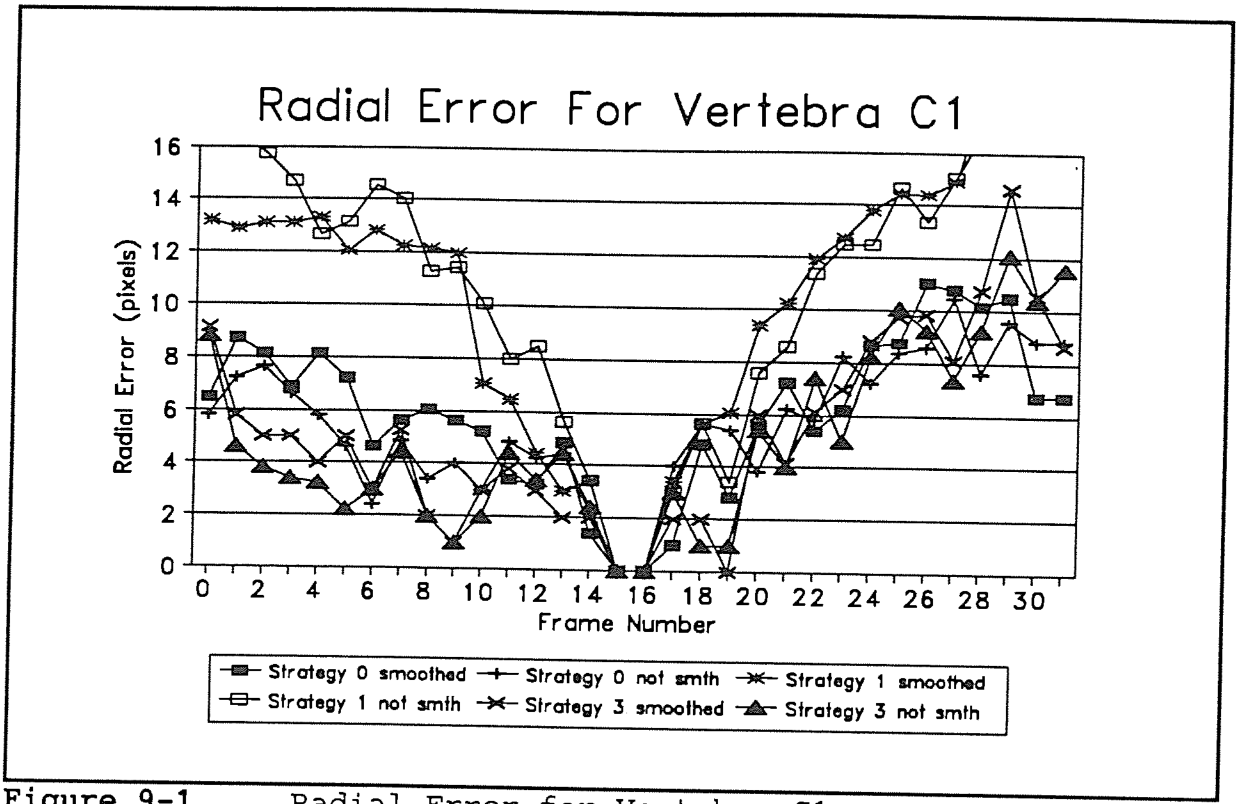


Figure 9-1 Radial Error for Vertebra C1

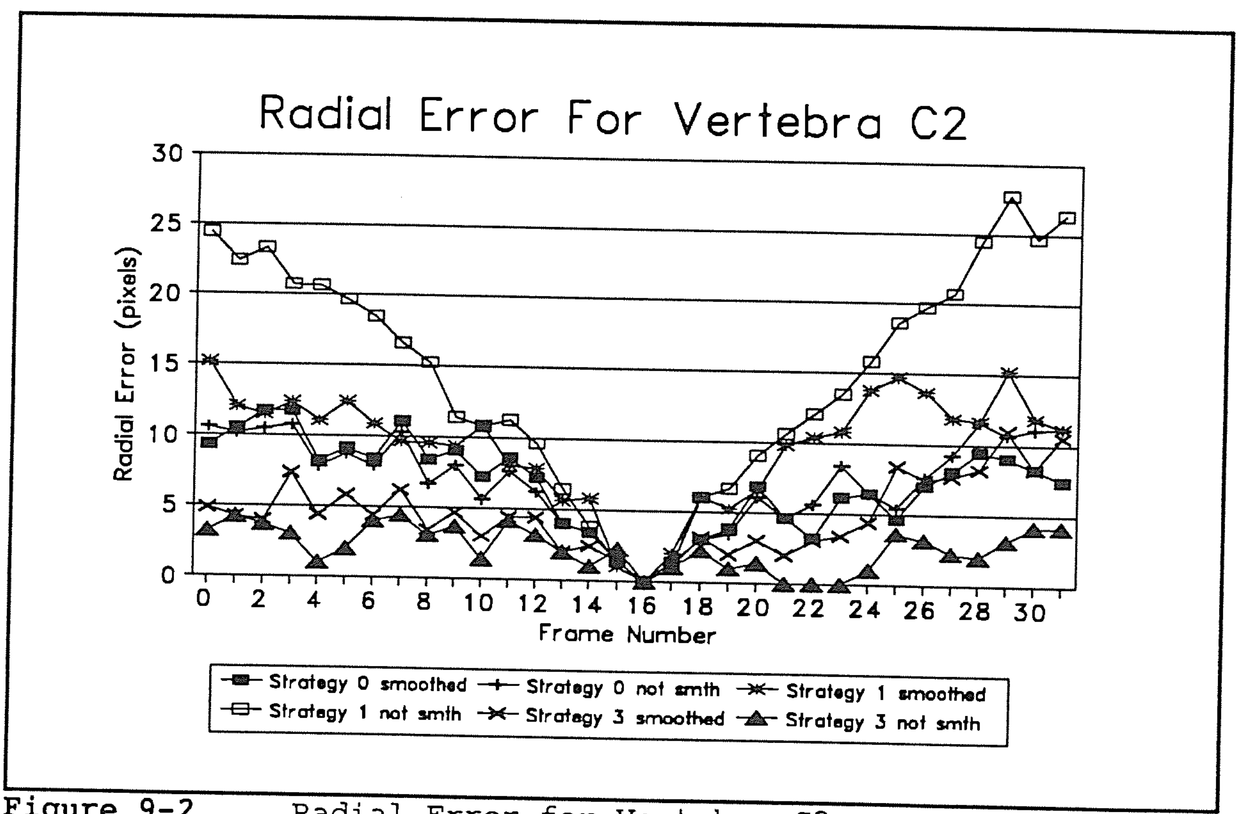


Figure 9-2 Radial Error for Vertebra C2

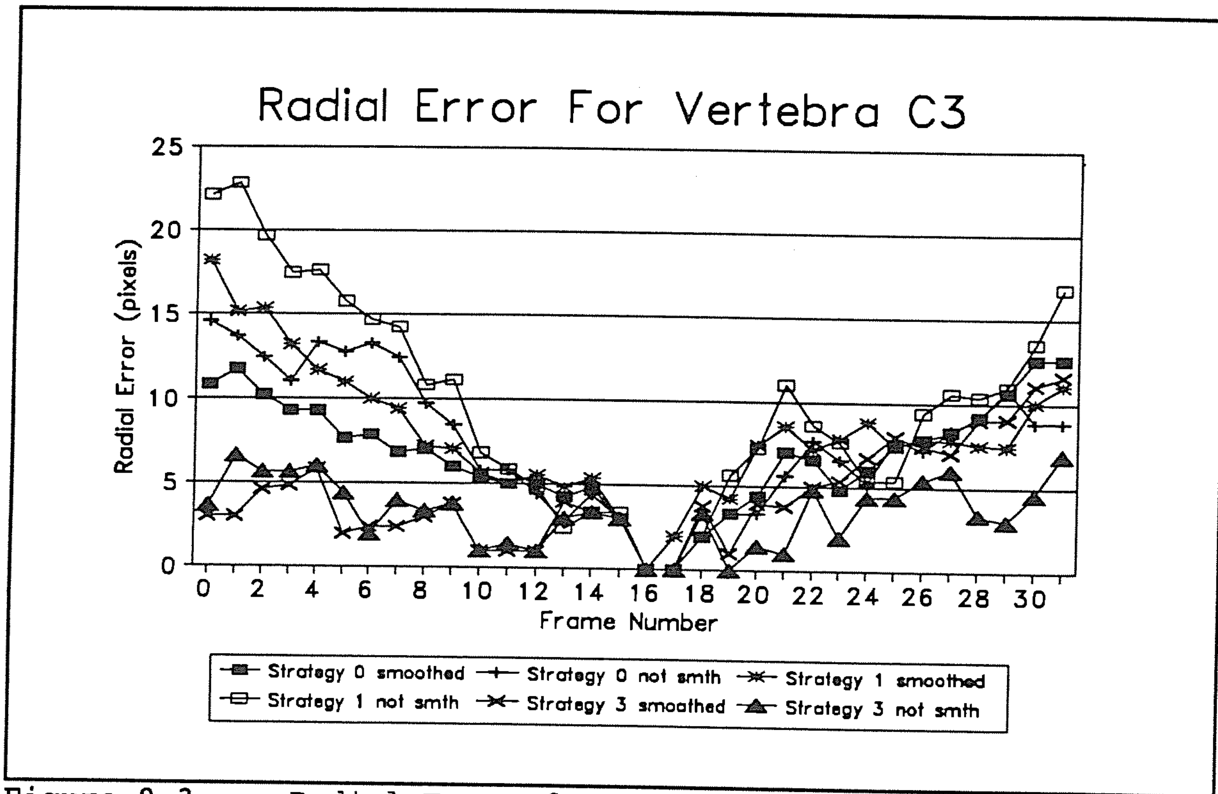


Figure 9-3 Radial Error for Vertebra C3

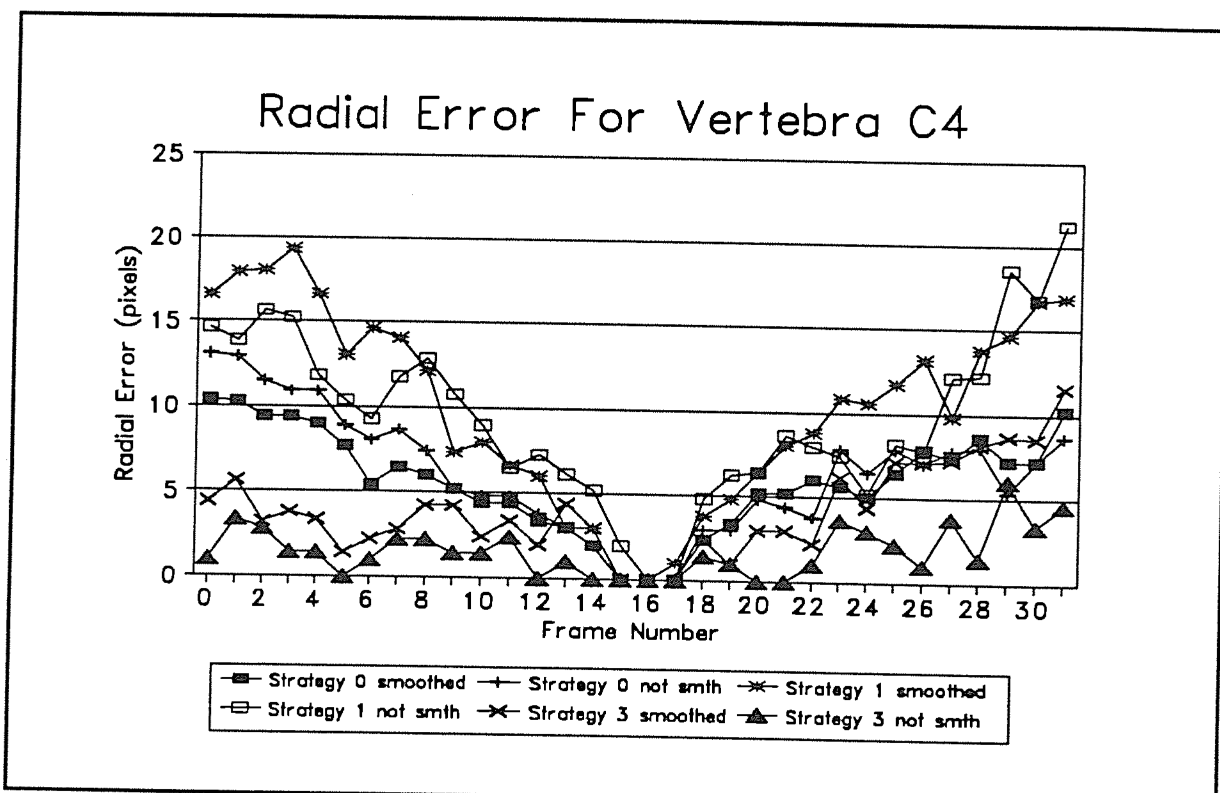


Figure 9-4 Radial Error for Vertebra C4

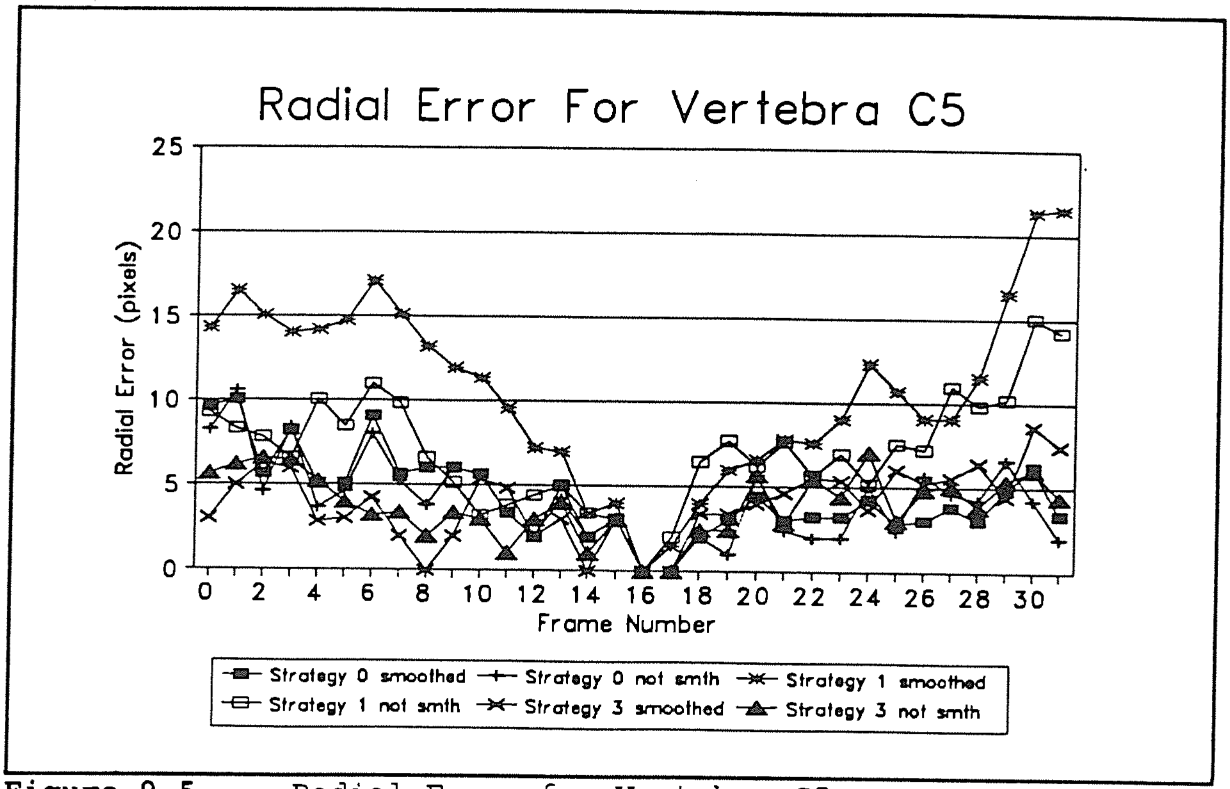


Figure 9-5 Radial Error for Vertebra C5

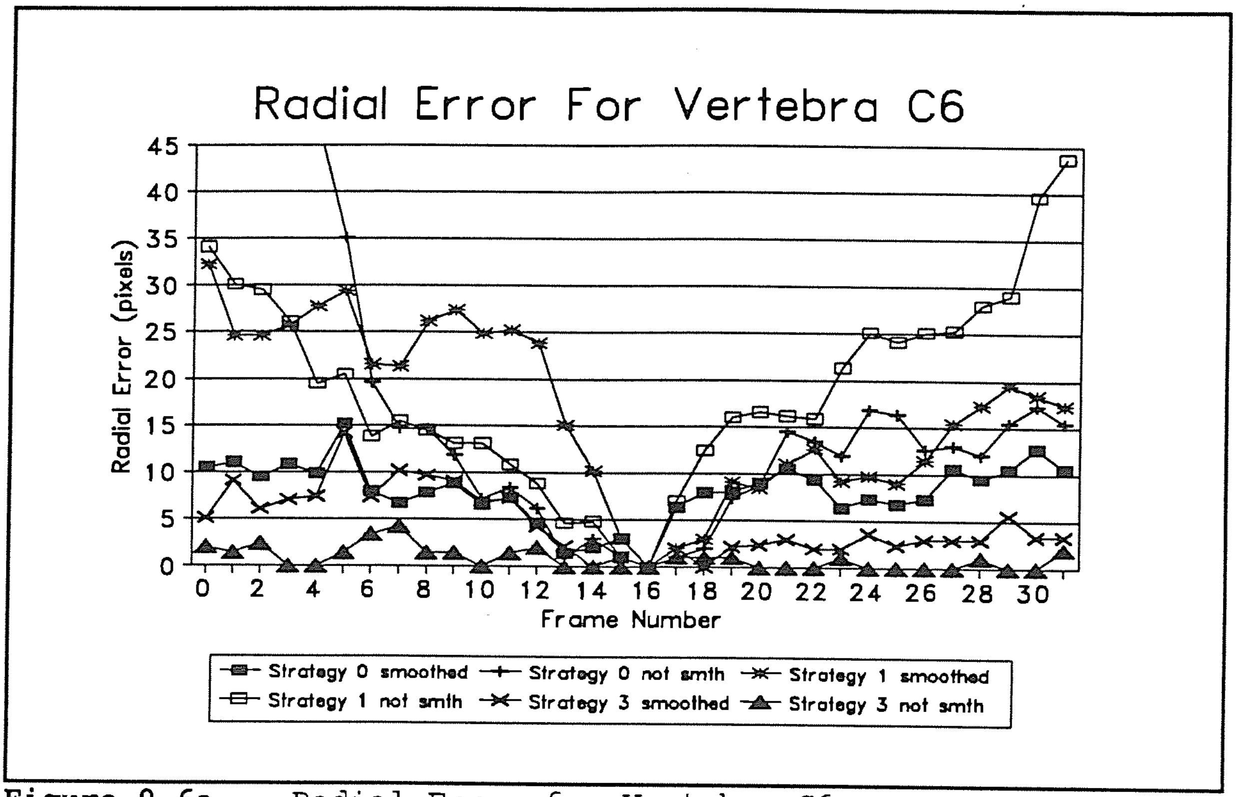


Figure 9-6a Radial Error for Vertebra C6

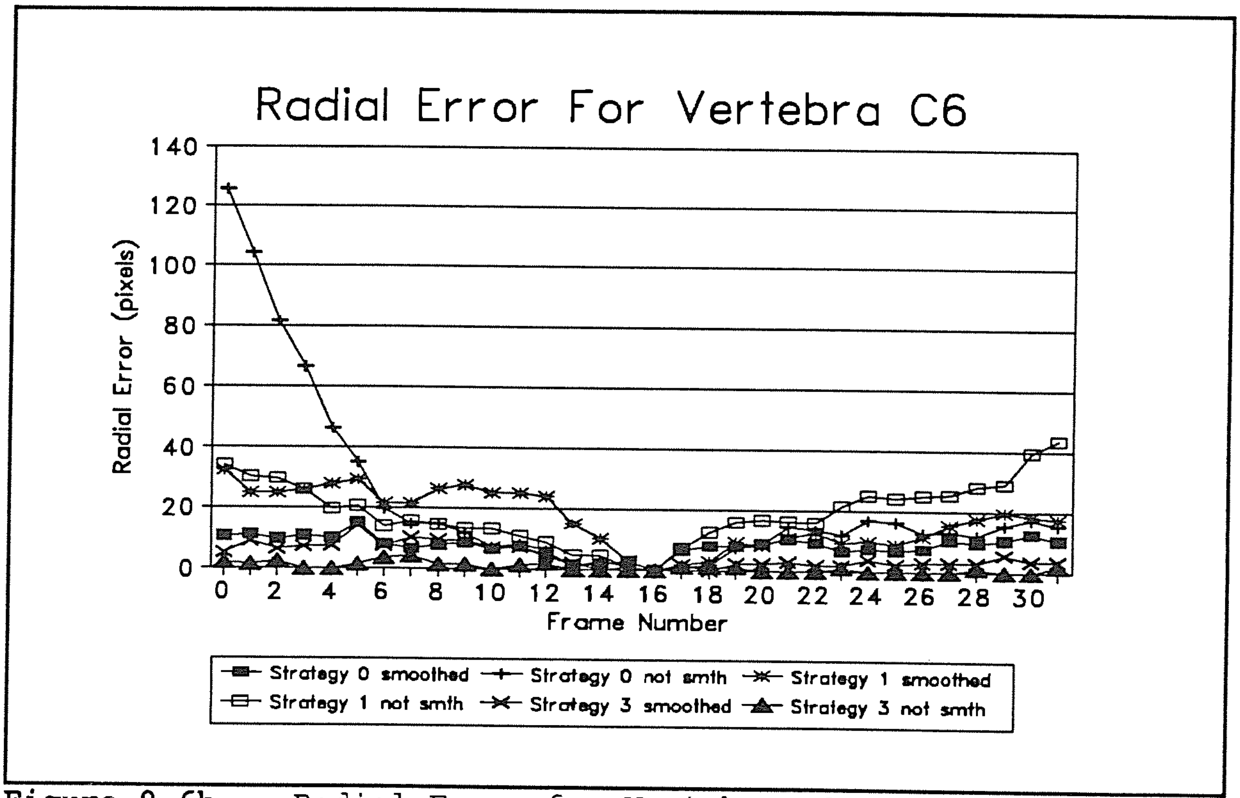


Figure 9-6b Radial Error for Vertebra C6

9.2 Effect of sub-pixel tracking on the measurements of the spine

Table 9-2 shows the spinal measurements calculated using the manually tracked data. These measurements were compared to the measurements given by the tracked points obtained using strategies 0, 1 and 3. The spinal measurements are given in Table 9-3, 9-4 and 9-5 for strategies 0, 1 and 3 respectively. The graphs of the data in these Tables are in Figures 9-7 to 9-16.

Figure 9-7 to 9-12 are graphs of the angles of the vertebrae given for each frame by each of strategies 0, 1 and 3 and the manual data. In every case strategy 3 gives a measurement far closer to the measurement from the manual data than either strategy 0 or 1. The measurements from strategy 0 data were often 5 degrees different than the manual measurement by the end or the beginning of a sequence. This may not be significant for vertebrae C1 and C2, where the vertebrae rotate between 35 and 45 degrees throughout the sequence, meaning the error is usually below 10%, but for the remainder of the vertebrae this is a significant error.

Figure 9-13 to 9-17 are the graphs of the planar distance between the vertebrae. The graphs appear very erratic due to the small scale of the distance which is never more than 5 pixels and in most cases less. Again strategy 3 appears to give the best measurements giving very similar measurements to the manual data.

The graphs of the axial distance between vertebrae are given in Figures 9-18 to 9-22. Once again measurements from strategy 3 data are close to the manual measurement.

From graphs 9-7 to 9-22 it can be seen that although Strategy 0 and 1 give reasonably good measurements for part of a sequence on some of the vertebrae it is strategy 3 data that gives measurements consistently closer to the manual data measurements.

In conclusion it would be beneficial, not only to the tracking results but also to the spinal measurements to implement strategy 3, as a means of sub-pixel location finding, in the Chiropractic prototype.

Table 9-2 Spine measurements from manually tracked data

FRAME NO	ANGLE OF VERTEBRA						PLANAR DISTANCE BETWEEN VERTEBRAE					AXIAL DISTANCE BETWEEN VERTEBRAE				
	C1	C2	C3	C4	C5	C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6
0	-19.5	-15.2	-26.1	-25.3	-21.5	-24.6	3.3	1.8	3.4	1.0	5.6	27.0	19.1	19.7	20.0	19.6
	-20.2	-14.5	-25.3	-24.9	-22.3	-25.4	2.7	1.5	3.2	1.1	5.4	26.9	18.3	20.3	19.4	19.7
2	-20.2	-15.6	-23.9	-24.1	-19.2	-23.4	2.1	1.7	3.0	0.1	5.8	26.4	19.0	20.3	19.4	19.7
	-19.1	-15.1	-22.7	-23.3	-19.5	-25.4	2.0	2.2	2.6	1.0	5.6	27.1	18.8	19.5	20.3	19.3
4	-17.7	-12.0	-22.0	-23.0	-19.5	-24.6	1.6	1.3	3.1	0.2	6.6	26.7	18.9	19.8	19.6	19.7
	-17.7	-11.8	-21.3	-20.4	-19.5	-24.6	2.1	1.5	2.6	0.3	6.2	26.1	19.4	20.1	19.1	19.9
6	-13.8	-10.1	-20.9	-19.6	-22.3	-19.3	1.3	1.3	2.5	1.0	5.7	27.1	18.8	19.9	19.4	20.6
	-13.8	-10.1	-20.3	-20.4	-20.4	-18.4	1.2	1.3	2.8	0.5	5.3	26.6	18.5	19.8	19.3	20.4
8	-12.7	-7.5	-17.5	-18.7	-19.2	-20.2	1.0	0.9	1.8	0.5	6.2	26.5	19.5	19.1	19.9	20.3
	-11.8	-6.5	-16.2	-17.0	-19.2	-21.1	1.3	0.7	1.9	0.5	6.3	26.4	19.0	20.3	19.3	20.1
10	-9.1	-3.8	-15.0	-16.7	-18.4	-20.6	0.6	0.4	2.0	0.7	6.1	26.1	19.4	20.0	19.5	19.6
	-5.6	-4.7	-15.0	-16.7	-17.1	-21.1	1.0	0.9	2.0	0.0	6.2	27.5	18.6	20.0	19.9	19.5
12	-5.7	-3.7	-14.3	-15.8	-15.9	-19.3	0.6	0.6	2.1	0.1	5.9	26.3	18.6	20.4	19.0	20.2
	-5.6	-0.9	-12.1	-12.9	-13.4	-19.3	0.9	0.2	1.7	0.5	5.1	27.0	18.9	20.1	19.9	19.1
14	-2.3	0.9	-11.1	-13.1	-15.9	-19.3	0.5	0.6	2.2	0.4	6.2	26.3	19.6	19.7	19.2	19.5
	-1.1	1.9	-10.0	-12.9	-15.7	-20.9	0.5	0.4	2.5	1.0	7.0	26.7	19.1	19.9	19.3	19.0
16	-1.1	2.8	-10.1	-12.9	-15.9	-20.6	0.5	0.1	1.8	0.2	6.5	26.5	19.3	20.0	19.2	19.2
	-1.1	3.8	-10.1	-12.9	-15.9	-20.2	0.0	0.0	1.8	0.2	6.7	26.3	19.5	20.0	19.2	19.1
18	3.4	2.8	-10.1	-11.8	-14.9	-20.6	0.4	0.1	2.3	0.6	6.7	27.0	18.8	19.7	19.2	19.4
	3.3	6.5	-6.1	-11.1	-15.7	-21.8	0.6	0.4	1.4	0.4	7.1	26.6	19.2	20.5	18.6	18.7
20	5.7	10.2	-6.1	-9.2	-13.2	-22.2	2.0	0.6	0.7	1.4	7.5	26.0	19.2	20.1	18.9	18.9
	7.8	10.4	-3.9	-9.2	-14.9	-20.0	2.0	0.7	1.3	0.5	6.6	26.5	19.7	19.5	19.3	18.6
22	10.0	11.3	-4.0	-9.0	-15.7	-21.3	2.1	1.0	1.9	0.2	7.3	25.8	18.5	20.4	19.0	18.9
	12.4	14.3	-4.0	-8.5	-15.7	-21.8	2.9	0.8	1.4	0.0	7.1	26.0	19.3	19.7	18.7	18.7
24	14.6	14.3	-4.0	-8.0	-12.5	-21.8	2.8	0.3	1.4	0.9	6.6	26.1	18.8	20.2	18.7	18.4
	15.2	14.0	-2.0	-8.5	-14.9	-21.3	1.7	0.8	2.1	0.1	7.1	26.4	18.8	19.1	18.9	18.7
26	19.5	17.2	-1.0	-5.5	-12.5	-20.9	2.9	0.7	0.6	0.8	6.7	26.2	18.3	19.2	18.9	19.1
	22.0	18.1	0.0	-6.4	-14.3	-20.0	2.8	0.3	0.9	0.5	6.9	26.1	18.9	19.7	18.1	19.3
28	22.6	20.2	1.0	-5.6	-13.2	-20.9	3.8	0.8	0.6	0.1	6.5	25.6	18.7	18.7	19.0	18.9
	24.0	19.7	2.0	-7.5	-11.5	-21.3	3.4	1.2	2.0	1.7	7.0	25.5	18.1	19.4	18.8	18.8
30	24.6	20.4	2.0	-6.4	-14.0	-22.2	3.3	0.2	0.6	0.4	6.6	25.7	18.3	19.2	19.0	18.9
	26.6	20.8	2.0	-5.8	-15.9	-20.9	2.9	0.0	0.7	0.4	7.3	25.5	18.4	19.0	18.9	18.9

Table 9-3 Spine measurements from data tracked using strategy 0

STRATEGY ZERO WITH SMOOTHING FILTER

FRAME	ANGLE OF VERTEBRA						PLANAR DISTANCE BETWEEN VERTEBRAE					AXIAL DISTANCE BETWEEN VERTEBRAE				
NO	C1	C2	C3	C4	C5	C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6
0	-16.3	-6.5	-19.3	-19.0	-16.8	-18.4	0.7	0.1	0.5	2.7	6.0	26.9	19.8	19.7	19.3	19.9
	-15.9	-6.7	-19.3	-18.2	-16.8	-18.1	0.7	0.2	0.3	2.8	5.7	26.9	19.7	19.4	19.5	19.9
2	-15.6	-5.7	-18.4	-18.2	-16.8	-18.1	0.8	0.1	0.2	2.8	5.7	26.6	19.7	19.7	19.5	19.9
	-14.6	-5.7	-18.4	-17.3	-16.8	-18.1	0.5	0.4	0.0	2.9	5.7	26.7	19.6	19.4	19.7	19.9
4	-12.4	-4.8	-17.5	-16.7	-16.5	-18.1	1.2	0.0	0.4	2.9	5.9	26.4	19.6	19.2	19.9	19.9
	-11.3	-3.8	-16.6	-15.8	-16.5	-16.1	1.5	0.1	0.1	2.0	6.5	26.4	19.6	19.2	19.9	20.5
6	-9.3	-2.8	-15.9	-15.8	-16.3	-16.1	1.9	0.1	0.3	1.8	6.7	26.6	19.5	19.4	19.8	20.4
	-8.1	-1.0	-15.0	-15.8	-16.5	-16.1	2.1	0.0	0.2	1.6	6.5	26.2	19.7	19.6	19.7	20.5
8	-7.1	0.0	-14.0	-15.8	-16.5	-15.8	2.1	0.1	0.1	1.6	6.7	26.2	19.7	19.9	19.7	20.4
	-5.9	1.0	-13.1	-16.1	-16.5	-15.8	2.0	0.5	0.8	1.3	6.7	26.1	19.7	19.9	19.7	20.4
10	-3.5	2.9	-11.1	-15.2	-16.5	-16.8	2.3	0.7	0.9	1.4	6.8	26.0	19.6	20.1	19.9	20.1
	-3.5	2.8	-10.9	-14.0	-16.5	-16.8	2.0	0.2	0.7	1.4	6.8	26.0	19.6	20.3	19.6	20.1
12	-2.3	2.8	-10.9	-14.0	-17.0	-20.6	1.7	0.2	1.2	0.8	6.4	26.2	19.6	20.2	19.4	18.9
	-1.2	2.8	-10.9	-13.1	-17.0	-20.6	1.5	0.2	1.4	0.9	6.4	26.5	19.6	19.9	19.7	18.9
14	-1.2	2.8	-11.1	-13.1	-17.0	-20.6	1.0	0.1	1.7	0.9	6.4	26.5	19.5	19.8	19.7	18.9
	-1.1	2.8	-10.1	-12.9	-16.4	-20.6	0.5	0.1	1.8	0.6	6.2	26.5	19.3	20.0	19.3	19.3
16	-1.1	2.8	-10.1	-12.9	-15.9	-20.6	0.5	0.1	1.8	0.2	6.5	26.5	19.3	20.0	19.2	19.2
	-1.1	2.8	-10.1	-12.9	-15.9	-16.5	0.3	0.1	1.8	0.2	6.7	26.5	19.3	20.0	19.2	20.4
18	-1.1	2.9	-10.5	-13.1	-15.9	-13.1	0.8	0.2	2.2	0.1	5.9	26.5	19.3	20.0	19.1	21.5
	1.1	4.8	-9.3	-13.3	-15.5	-14.5	0.6	0.1	2.4	0.2	6.9	27.1	19.0	19.7	19.2	20.9
20	3.1	4.8	-8.6	-13.3	-14.7	-13.5	0.9	0.3	2.9	0.3	6.6	26.6	18.8	19.8	19.4	21.0
	5.2	6.7	-8.6	-13.3	-14.7	-14.3	0.5	0.1	3.4	0.3	6.4	26.2	18.8	19.7	19.4	20.8
22	7.3	8.4	-7.4	-13.3	-14.7	-14.3	0.1	0.1	3.4	0.3	6.4	26.2	18.5	19.5	19.4	20.8
	10.5	9.3	-6.2	-12.0	-14.7	-16.3	0.8	0.1	3.2	0.1	5.8	26.3	18.5	19.5	19.0	20.2
24	12.3	11.1	-5.1	-12.0	-14.9	-16.6	0.9	0.2	3.2	0.2	5.9	26.4	18.3	19.2	19.0	20.1
	14.0	12.9	-4.2	-12.9	-14.9	-16.6	0.9	0.1	4.0	0.3	5.9	26.5	18.1	19.1	18.7	20.1
26	15.9	14.7	-4.2	-12.0	-15.1	-15.7	1.3	0.4	3.6	0.0	6.1	26.6	17.6	18.9	19.0	20.4
	17.8	15.6	-3.1	-11.7	-15.1	-13.6	1.9	0.4	3.6	0.3	5.6	26.3	17.7	19.2	18.4	21.0
28	20.0	16.4	-2.1	-11.7	-15.1	-13.6	2.4	0.2	3.4	0.3	5.6	26.6	17.2	19.0	18.4	21.0
	20.9	17.3	-2.1	-11.8	-14.9	-16.4	2.6	0.2	4.0	0.2	4.9	26.1	17.5	18.9	18.6	20.2
30	21.3	17.3	-2.1	-11.8	-14.9	-16.4	2.9	0.2	4.0	0.2	4.9	26.0	17.5	18.9	18.6	20.2
	23.1	18.4	-2.1	-12.7	-14.9	-16.7	2.7	0.2	4.2	0.1	5.2	25.5	17.8	19.1	18.3	20.1

STRATEGY ZERO WITHOUT SMOOTHING FILTER

FRAME	ANGLE OF VERTEBRA						PLANAR DISTANCE BETWEEN VERTEBRAE					AXIAL DISTANCE BETWEEN VERTEBRAE				
NO	C1	C2	C3	C4	C5	C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6
0	-20.6	-6.7	-13.8	-15.1	-17.3	38.3	0.6	2.0	0.4	1.8	30.3	26.8	18.4	20.5	18.8	38.8
	-20.2	-6.7	-13.8	-15.1	-16.8	31.2	0.8	2.0	0.4	2.4	22.4	26.8	18.4	20.5	18.9	37.0
2	-19.8	-6.7	-13.6	-15.1	-17.0	24.9	0.9	2.3	0.2	2.1	15.9	26.7	18.4	20.4	18.8	34.7
	-17.7	-6.7	-13.3	-15.1	-17.0	19.0	1.1	2.1	0.0	2.1	10.7	27.1	18.4	20.4	18.8	32.6
4	-15.2	-5.7	-11.5	-14.3	-17.0	10.4	0.3	2.6	0.1	1.7	4.4	27.2	18.1	20.5	18.9	29.4
	-14.0	-4.7	-10.6	-13.6	-17.4	2.5	0.6	2.6	0.0	1.4	0.1	27.2	18.0	20.4	19.3	26.5
6	-11.8	-3.8	-10.0	-13.6	-17.4	-3.5	1.3	2.4	0.5	1.4	2.3	27.4	17.9	20.6	19.3	24.5
	-10.6	-1.9	-10.0	-13.6	-17.4	-8.6	1.3	2.4	0.5	1.4	2.7	27.0	18.3	20.6	19.3	23.0
8	-9.5	-1.9	-10.0	-13.6	-17.9	-12.2	1.3	2.2	0.5	0.8	3.0	27.2	18.3	20.6	19.2	22.0
	-8.1	-0.9	-10.1	-13.8	-16.5	-12.9	1.0	2.0	1.1	1.2	4.3	27.1	18.5	20.5	19.5	21.3
10	-5.7	0.9	-10.1	-13.8	-16.5	-15.4	1.8	1.3	1.1	1.2	5.4	26.6	18.9	20.5	19.5	20.6
	-5.6	1.9	-10.1	-13.8	-16.5	-15.4	1.3	0.9	1.1	1.2	5.4	26.3	19.1	20.5	19.5	20.6
12	-4.4	1.9	-10.1	-13.1	-16.5	-16.3	0.8	0.9	1.6	1.0	5.5	26.5	19.1	20.1	19.7	20.3
	-3.3	2.8	-10.1	-13.1	-16.5	-19.7	0.8	0.6	1.6	1.0	6.3	26.5	19.3	20.1	19.7	19.3
14	-3.3	2.8	-10.1	-13.1	-16.5	-19.7	0.3	0.1	1.6	1.0	6.3	26.5	19.3	20.1	19.7	19.3
	-1.1	2.8	-10.1	-12.9	-16.5	-20.6	0.5	0.1	1.8	0.3	6.3	26.5	19.3	20.0	19.5	19.0
16	-1.1	2.8	-10.1	-12.9	-15.9	-20.6	0.5	0.1	1.8	0.2	6.5	26.5	19.3	20.0	19.2	19.2
	-1.1	2.8	-10.1	-12.9	-15.9	-20.6	0.5	0.1	1.8	0.2	6.5	26.5	19.3	20.0	19.2	19.2
18	-1.1	2.9	-10.5	-12.7	-15.9	-19.9	0.2	0.2	2.1	0.0	6.0	26.5	19.3	20.0	19.1	19.4
	2.2	5.7	-9.3	-12.7	-15.9	-15.4	1.2	0.3	2.1	0.0	5.6	26.5	19.2	19.7	19.1	20.8
20	3.2	5.7	-9.3	-11.7	-16.2	-15.2	0.3	0.8	1.7	0.3	5.6	26.3	19.2	20.0	18.7	20.8
	5.3	6.7	-9.1	-10.8	-16.2	-11.1	0.3	0.8	2.2	0.2	5.9	26.6	19.0	19.7	19.0	22.0
22	7.4	6.7	-9.1	-9.9	-16.2	-11.1	0.2	1.3	2.0	0.1	5.9	26.2	19.0	19.4	19.2	22.0
	9.3	7.7	-9.3	-9.8	-16.2	-11.9	0.3	1.7	2.0	0.3	5.7	26.0	18.7	19.4	19.2	21.8
24	11.5	9.5	-8.1	-9.8	-16.2	-8.4	0.7	1.7	2.0	0.3	4.3	26.0	18.5	19.2	19.2	22.9
	13.5	11.1	-7.0	-9.8	-16.2	-8.4	1.3	2.2	2.1	0.3	4.3	26.1	18.4	18.9	19.2	22.9
26	15.5	12.2	-7.1	-9.8	-15.9	-12.5	1.2	2.1	2.3	0.5	4.0	26.0	18.1	18.9	19.1	21.7
	17.5	13.3	-7.3	-9.8	-15.9	-17.7	1.6	2.4	2.6	0.5	3.7	25.7	17.9	18.8	19.1	20.0
28	18.8	14.0	-6.2	-9.8	-15.9	-19.2	2.0	2.4	2.5	0.5	3.8	25.7	17.5	19.1	19.1	19.5
	19.1	14.0	-6.1	-9.8	-16.2	-18.4	2.3	3.2	2.2	0.3	3.3	25.6	17.5	19.1	19.2	19.7
30	20.1	14.0	-6.1	-9.8	-16.4	-18.2	2.5	3.2	2.2	0.6	2.8	25.4	17.5	19.1	19.1	19.8
	22.0	14.9	-6.1	-11.5	-16.4	-18.2	2.4	3.5	2.6	0.8	2.8	24.8	17.9	19.6	18.6	19.8

Table 9-4 Spine measurements from data tracked using strategy 1

STRATEGY ONE WITH SMOOTHING FILTER

FRAME	ANGLE OF VERTEBRA						PLANAR DISTANCE BETWEEN VERTEBRAE					AXIAL DISTANCE BETWEEN VERTEBRAE				
NO	C1	C2	C3	C4	C5	C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6
0	-17.8	-13.3	-22.3	-28.2	-19.8	-7.6	0.6	2.9	3.0	1.4	0.3	30.2	14.6	22.7	21.6	23.2
	-17.5	-12.4	-23.2	-27.0	-18.7	-11.0	0.4	3.0	3.1	1.8	1.5	29.9	15.1	22.6	21.6	21.6
2	-17.5	-10.7	-24.1	-28.5	-18.7	-11.4	0.6	2.9	4.2	1.7	2.0	29.2	15.9	22.5	21.2	21.2
	-16.5	-10.0	-22.3	-28.1	-20.7	-11.4	0.9	2.5	4.2	1.5	2.7	29.1	16.7	21.8	21.0	21.3
4	-15.1	-8.0	-19.1	-27.0	-20.5	-8.9	0.9	1.5	4.0	1.7	2.0	28.4	16.9	22.0	21.2	21.9
	-14.3	-5.9	-18.1	-25.4	-19.4	-7.6	0.5	0.8	4.5	2.0	1.4	27.6	17.2	21.9	20.8	22.3
6	-12.7	-4.8	-18.1	-25.8	-17.7	-8.8	0.5	1.0	5.0	2.5	2.1	27.2	17.4	22.3	20.3	21.9
	-11.5	-3.8	-18.4	-23.3	-17.4	-9.5	0.7	0.9	4.7	2.5	1.9	27.1	17.7	22.0	20.1	21.9
8	-10.4	-1.9	-17.2	-21.6	-16.9	-6.8	0.5	0.2	4.1	2.5	0.8	27.3	18.0	21.4	20.2	22.5
	-9.5	-0.9	-15.4	-18.7	-16.9	-6.9	0.0	0.1	3.4	2.2	0.6	27.3	18.5	20.4	20.1	22.6
10	-8.3	0.0	-13.1	-16.7	-15.0	-8.0	0.2	0.4	2.3	2.4	1.5	27.3	18.7	20.1	19.9	22.6
	-7.0	0.0	-11.3	-15.6	-14.4	-8.8	0.5	0.4	1.4	1.7	1.3	27.5	18.6	20.0	19.6	22.7
12	-4.6	-0.9	-10.6	-14.7	-14.9	-12.4	0.2	0.1	1.4	1.3	0.8	27.3	18.6	19.9	19.7	21.7
	-2.2	0.0	-9.8	-12.7	-14.9	-13.8	0.0	0.3	1.0	1.8	2.5	27.0	18.5	20.2	19.8	21.0
14	-1.1	0.9	-10.0	-12.9	-14.9	-17.4	0.3	0.2	1.5	1.6	4.3	27.0	18.8	20.1	19.7	20.4
	-1.1	1.9	-10.1	-12.9	-16.2	-21.1	0.2	0.0	1.8	0.4	6.3	26.7	19.0	20.0	19.2	19.5
16	-1.1	2.8	-10.1	-12.9	-15.9	-20.6	0.5	0.1	1.8	0.2	6.5	26.5	19.3	20.0	19.2	19.2
	0.0	3.7	-9.3	-12.7	-14.9	-20.9	0.9	0.1	1.7	0.9	6.8	26.5	19.3	20.3	19.5	18.8
18	1.1	4.5	-8.6	-12.0	-13.8	-20.9	0.9	0.0	1.7	1.0	6.3	26.5	19.3	20.3	20.0	18.7
	3.2	7.1	-7.5	-13.3	-13.2	-18.1	1.4	0.4	2.3	0.7	6.9	26.2	19.7	20.2	19.7	19.6
20	5.2	9.6	-5.4	-14.5	-12.4	-16.5	1.5	1.5	2.9	0.6	6.7	26.0	19.9	20.3	19.2	20.2
	7.1	12.2	-3.2	-13.8	-12.4	-16.2	1.7	2.4	3.1	1.0	7.0	25.8	20.1	20.0	19.0	20.7
22	10.1	14.3	-1.1	-12.6	-12.9	-15.4	2.3	2.8	2.5	1.0	7.3	25.6	20.1	20.3	18.4	20.3
	12.1	17.0	0.0	-11.7	-13.5	-18.7	2.5	3.3	2.1	0.9	7.0	25.4	20.5	20.4	17.9	19.8
24	14.0	19.5	2.0	-11.9	-11.8	-19.5	2.5	3.8	2.3	1.6	7.0	25.4	20.1	20.4	18.0	19.6
	15.9	21.2	2.9	-11.1	-11.9	-19.8	3.3	3.2	2.2	1.7	7.0	25.3	20.0	20.4	17.7	19.6
26	17.8	22.3	3.9	-10.1	-11.5	-17.2	3.6	2.8	1.8	2.0	7.1	25.0	20.2	20.4	17.7	20.9
	21.4	23.4	4.8	-10.0	-12.4	-16.4	3.8	2.6	1.6	1.5	6.4	24.9	20.2	20.7	16.9	21.9
28	23.6	25.0	4.7	-12.1	-13.7	-14.7	4.1	2.5	1.5	0.9	6.0	25.6	19.8	20.7	16.4	22.7
	24.4	25.8	3.7	-15.3	-15.0	-17.9	4.5	2.1	2.2	0.9	6.3	26.6	19.4	20.6	15.8	22.4
30	24.4	26.6	5.4	-15.3	-14.6	-16.6	4.8	2.1	1.8	1.5	6.7	26.8	19.7	20.2	15.9	22.3
	25.3	27.3	8.0	-14.5	-14.4	-16.6	4.5	2.3	0.9	1.6	6.9	26.5	20.3	20.2	15.7	22.2

STRATEGY ONE WITHOUT SMOOTHING FILTER

FRAME	ANGLE OF VERTEBRA						PLANAR DISTANCE BETWEEN VERTEBRAE					AXIAL DISTANCE BETWEEN VERTEBRAE				
NO	C1	C2	C3	C4	C5	C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6
0	-15.1	-17.0	-19.7	-26.6	-17.6	-14.0	4.2	2.2	9.1	1.3	0.4	29.0	16.5	18.2	21.3	26.4
	-15.6	-14.8	-18.2	-27.0	-17.6	-14.2	3.2	2.2	8.2	1.1	1.0	28.9	16.5	18.5	21.2	25.7
2	-15.6	-14.0	-16.9	-27.8	-16.8	-13.3	2.5	1.7	7.4	1.0	0.8	29.3	16.5	18.9	21.5	25.2
	-15.4	-12.5	-16.9	-26.6	-16.8	-14.2	2.3	1.3	6.9	0.4	0.5	29.3	16.9	18.8	21.4	24.5
4	-14.6	-11.2	-17.4	-26.2	-16.8	-16.5	2.1	1.1	7.1	0.2	1.9	28.1	17.3	18.7	21.3	22.9
	-14.9	-8.8	-16.5	-22.8	-16.2	-17.0	2.6	1.3	6.8	0.8	2.7	28.5	18.2	18.5	21.0	22.2
6	-14.0	-7.3	-15.3	-20.6	-15.6	-14.3	2.5	0.9	6.1	1.0	2.8	28.3	18.5	19.0	20.3	22.3
	-13.2	-7.3	-13.8	-19.1	-14.7	-12.3	2.0	0.4	5.6	1.8	2.7	28.1	18.9	19.2	20.1	22.0
8	-12.0	-5.7	-12.2	-19.1	-15.3	-13.3	1.3	0.6	5.1	2.0	4.3	27.9	19.3	19.3	20.0	21.6
	-7.3	-4.1	-11.5	-17.8	-15.9	-15.3	0.1	1.0	4.8	2.1	5.9	27.9	19.0	19.4	20.0	20.9
10	-2.4	-0.9	-11.7	-17.5	-15.7	-13.3	1.3	1.2	4.3	1.7	5.7	27.3	19.5	19.5	19.8	21.0
	-2.4	1.8	-11.9	-17.2	-16.8	-14.8	1.8	1.4	3.8	0.6	5.8	27.0	19.7	19.7	19.3	20.4
12	-2.3	2.7	-11.9	-17.0	-15.1	-17.8	2.0	0.8	3.6	0.3	4.9	26.7	19.9	19.7	19.1	19.2
	-1.1	2.7	-11.9	-15.6	-14.9	-17.5	1.2	0.5	2.9	0.3	4.8	26.5	19.8	20.1	18.7	19.2
14	-1.1	2.8	-11.9	-14.5	-14.9	-17.5	0.2	0.3	2.4	0.2	4.9	26.5	19.8	20.0	18.9	19.7
	-1.1	2.8	-11.1	-13.6	-15.9	-20.6	0.5	0.1	1.9	0.1	6.1	26.5	19.5	20.1	18.9	19.3
16	-1.1	2.8	-10.1	-12.9	-15.9	-20.6	0.5	0.1	1.8	0.2	6.5	26.5	19.3	20.0	19.2	19.2
	-1.1	3.8	-10.1	-12.9	-14.9	-15.3	0.4	0.2	1.8	0.6	6.3	26.2	19.5	20.0	19.5	20.6
18	-1.1	4.7	-10.1	-12.2	-14.5	-11.1	0.1	0.4	1.4	0.5	5.0	26.0	19.8	19.9	19.4	21.7
	2.2	5.7	-10.1	-11.5	-14.5	-10.0	0.4	0.3	1.0	0.3	4.6	25.6	20.0	19.7	19.6	22.0
20	4.2	6.7	-10.1	-9.6	-13.8	-8.8	1.6	0.5	0.6	0.1	4.4	25.5	20.3	19.7	19.2	23.1
	7.3	7.6	-10.0	-9.5	-13.8	-9.3	1.5	0.4	1.1	0.1	3.7	25.2	20.5	19.6	19.1	23.0
22	10.0	8.4	-8.8	-9.3	-13.6	-8.5	1.1	0.3	1.4	0.4	3.9	24.7	20.5	19.8	19.2	22.7
	10.7	11.0	-7.9	-8.3	-12.9	-4.0	0.8	0.7	1.3	0.7	4.1	24.9	21.0	19.8	19.2	23.7
24	12.6	13.2	-4.9	-8.1	-12.2	-0.9	0.1	1.3	1.6	0.4	2.1	24.1	21.5	20.0	18.9	24.4
	15.2	14.7	-3.0	-9.2	-11.3	-0.9	0.2	1.7	2.1	0.2	1.8	24.0	21.9	19.6	18.8	24.2
26	15.4	16.1	-1.0	-9.2	-10.4	1.0	0.5	2.5	2.0	0.2	2.0	24.0	22.4	19.1	19.0	24.4
	16.3	18.2	1.9	-9.9	-9.3	3.8	1.0	3.3	1.4	1.2	1.2	24.1	22.8	19.2	18.8	24.5
28	17.5	19.9	3.8	-11.5	-9.3	3.8	1.8	3.7	1.6	1.6	1.7	23.9	23.5	19.2	18.3	24.4
	18.7	21.1	3.8	-11.3	-9.3	3.9	2.4	4.2	1.6	1.7	2.7	24.1	23.9	19.2	17.7	24.9
30	18.7	23.2	3.8	-10.3	-9.3	11.7	3.0	4.6	1.7	1.8	0.3	24.3	23.6	18.9	17.9	27.8
	18.7	24.9	5.6	-8.6	-10.1	14.2	3.8	5.3	1.3	2.0	1.1	24.4	23.2	19.0	18.2	28.5

Table 9-5 Spine measurements from data tracked using strategy 3

STRATEGY THREE WITH SMOOTHING FILTER

FRAME	ANGLE OF VERTEBRA						PLANAR DISTANCE BETWEEN VERTEBRAE					AXIAL DISTANCE BETWEEN VERTEBRAE				
NO	C1	C2	C3	C4	C5	C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6
0	-22.4	-13.3	-24.9	-25.0	-23.9	-20.8	1.5	1.8	2.9	1.1	6.4	25.5	19.5	20.4	18.7	21.4
	-22.0	-14.0	-25.3	-25.0	-24.2	-19.9	1.1	2.3	3.0	0.7	6.9	25.9	19.1	19.9	19.5	20.9
2	-22.0	-13.1	-24.1	-24.6	-23.8	-21.6	1.3	1.9	2.8	0.9	6.8	25.5	19.7	20.1	19.1	20.7
	-21.0	-11.3	-23.7	-23.4	-22.3	-19.9	1.2	1.8	3.1	0.0	6.5	25.6	19.6	20.2	18.7	21.6
4	-19.9	-9.6	-22.8	-23.0	-21.2	-19.3	1.0	1.5	2.6	0.4	6.0	26.3	19.4	20.3	19.0	21.2
	-20.3	-7.6	-20.6	-21.0	-20.9	-16.5	0.9	0.8	2.4	0.0	7.2	25.7	19.6	20.5	19.0	21.8
6	-16.3	-7.6	-19.3	-21.0	-21.2	-15.5	0.7	0.9	2.8	0.2	7.2	25.9	19.8	20.0	19.1	21.6
	-15.5	-5.7	-18.7	-19.8	-20.4	-19.1	0.8	0.7	2.5	0.4	7.4	26.0	20.0	20.0	19.3	20.5
8	-14.0	-5.8	-17.8	-20.1	-19.2	-15.8	0.5	0.8	2.7	0.2	6.9	25.8	19.8	20.2	19.1	21.7
	-12.0	-3.8	-17.2	-19.6	-19.2	-16.8	0.2	0.8	2.8	0.1	6.8	26.2	19.9	19.6	19.4	20.9
10	-9.3	-2.9	-15.3	-17.6	-18.4	-17.1	0.2	0.5	2.6	0.9	7.5	26.9	19.2	20.1	19.2	21.0
	-8.0	-1.0	-15.3	-16.7	-18.7	-16.8	0.1	0.5	2.0	0.2	7.1	26.1	19.6	20.5	18.6	21.1
12	-6.8	-1.0	-14.0	-17.0	-17.0	-21.4	0.2	0.5	2.2	0.5	6.2	26.3	19.3	20.1	19.5	19.3
	-5.6	0.9	-14.0	-16.7	-15.9	-20.9	0.1	0.4	2.4	0.2	6.4	26.5	19.3	20.1	19.2	19.2
14	-2.3	1.9	-11.9	-15.8	-15.9	-19.3	0.8	0.2	2.1	0.1	6.2	26.5	19.6	19.8	19.0	19.5
	-1.1	2.8	-10.1	-12.9	-15.9	-20.6	0.5	0.1	1.8	0.2	6.1	26.5	19.3	20.0	19.2	19.3
16	-1.1	2.8	-10.1	-12.9	-15.9	-20.6	0.5	0.1	1.8	0.2	6.5	26.5	19.3	20.0	19.2	19.2
	-1.1	3.8	-10.1	-12.9	-15.9	-20.9	0.2	0.2	1.8	0.2	6.8	26.3	19.5	20.0	19.2	19.1
18	2.2	5.7	-7.1	-11.0	-14.3	-20.6	1.4	0.3	1.2	0.6	6.3	26.5	19.8	19.8	19.1	19.3
	3.3	7.6	-6.0	-11.0	-14.7	-19.7	1.1	0.6	1.3	0.6	6.6	26.8	19.0	20.5	18.8	19.1
20	7.4	10.4	-5.0	-10.8	-12.8	-21.3	2.2	0.7	1.4	0.9	7.2	26.5	19.0	20.8	18.8	18.7
	9.5	10.4	-4.2	-10.1	-13.8	-20.9	2.0	0.4	1.3	0.9	7.4	26.0	19.3	20.2	18.8	18.9
22	11.5	12.2	-2.0	-11.0	-12.8	-21.3	1.8	0.8	1.4	1.1	7.7	25.7	19.3	20.5	18.9	18.6
	14.5	14.9	-1.0	-11.0	-13.6	-20.0	3.2	1.0	1.3	0.7	7.5	26.0	19.3	20.7	18.1	19.1
24	16.2	15.8	0.0	-9.9	-12.3	-20.9	2.5	1.0	0.9	0.7	7.4	26.1	19.3	20.3	18.2	18.9
	18.1	19.0	3.1	-10.8	-12.2	-20.9	3.8	1.7	1.8	0.9	7.7	26.2	18.8	20.2	18.0	19.4
26	20.3	19.9	3.2	-9.9	-11.5	-20.0	3.5	1.5	1.1	0.9	7.7	25.6	19.1	20.5	18.0	19.4
	21.3	21.6	4.1	-9.0	-12.0	-20.9	3.6	1.5	1.3	1.1	7.9	25.5	19.4	20.5	17.5	19.3
28	23.6	21.0	5.2	-8.9	-12.0	-20.6	4.2	0.9	0.6	0.8	8.1	25.5	19.3	20.2	17.4	19.3
	24.4	23.7	6.2	-9.2	-12.0	-19.7	4.9	1.8	1.4	1.3	8.2	25.7	19.5	20.0	17.5	20.0
30	26.6	24.1	7.4	-9.8	-12.8	-21.4	4.9	1.9	1.2	0.6	8.4	25.6	19.0	20.5	17.5	19.1
	27.0	25.0	9.3	-9.0	-11.5	-21.4	4.6	2.2	0.7	0.7	8.2	25.0	19.2	20.7	17.2	19.2

STRATEGY THREE WITHOUT SMOOTHING FILTER

FRAME	ANGLE OF VERTEBRA						PLANAR DISTANCE BETWEEN VERTEBRAE					AXIAL DISTANCE BETWEEN VERTEBRAE				
NO	C1	C2	C3	C4	C5	C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6	C1-C2	C2-C3	C3-C4	C4-C5	C5-C6
0	-21.0	-14.7	-26.6	-25.0	-24.6	-25.8	1.2	2.2	3.4	1.0	6.8	25.3	19.5	19.7	19.4	19.7
	-20.2	-12.9	-26.6	-23.0	-24.2	-26.6	0.7	1.9	3.1	0.6	7.0	25.8	19.5	19.2	19.4	19.4
2	-22.0	-12.9	-24.9	-21.8	-23.8	-25.4	1.0	2.1	2.7	0.7	6.7	25.6	19.4	20.1	19.2	19.5
	-18.4	-12.5	-24.5	-22.1	-22.8	-25.4	0.5	1.7	3.1	0.8	6.8	25.8	19.4	19.9	19.7	19.4
4	-18.4	-11.1	-24.1	-21.8	-23.4	-24.6	0.8	1.6	3.1	0.9	6.8	26.1	19.2	20.1	19.1	19.7
	-17.4	-10.1	-23.2	-20.4	-21.2	-25.8	0.9	1.6	3.0	0.7	6.6	25.5	19.6	20.3	19.5	18.9
6	-16.3	-7.5	-20.9	-19.3	-21.2	-21.6	0.4	0.8	2.7	0.4	6.8	25.9	19.6	19.8	19.8	20.2
	-14.0	-6.5	-19.7	-18.4	-20.9	-21.6	0.2	0.7	2.7	0.3	6.3	26.1	19.6	20.2	19.5	19.8
8	-14.0	-5.7	-20.0	-16.7	-19.8	-21.4	0.2	1.1	2.1	0.2	6.8	25.8	19.7	19.8	19.7	19.6
	-11.5	-4.8	-19.1	-15.8	-18.7	-22.3	0.3	0.8	2.4	0.1	6.8	26.0	19.8	20.2	19.6	19.1
10	-10.4	-2.9	-15.9	-15.6	-17.7	-20.6	0.1	0.7	1.8	0.2	6.7	26.1	19.9	20.0	19.4	19.4
	-8.0	-1.9	-16.2	-15.8	-17.4	-20.6	0.6	1.1	1.8	0.1	6.1	26.4	19.6	20.0	19.5	19.3
12	-7.0	-0.9	-14.0	-15.8	-18.4	-19.3	0.3	0.7	1.8	0.4	6.2	26.3	19.3	20.5	19.2	19.8
	-4.5	0.9	-13.8	-13.8	-16.8	-19.3	0.8	0.1	1.9	0.1	5.8	26.3	19.4	20.4	19.2	19.4
14	-2.2	1.9	-11.9	-13.1	-15.7	-19.3	1.0	0.2	1.5	0.7	6.4	26.5	19.6	19.6	19.3	19.5
	-1.1	3.8	-10.1	-12.9	-15.9	-20.9	0.7	0.2	1.8	0.2	6.4	26.2	19.5	20.0	19.2	19.2
16	-1.1	2.8	-10.1	-12.9	-15.9	-20.6	0.5	0.1	1.8	0.2	6.5	26.5	19.3	20.0	19.2	19.2
	-1.1	3.8	-10.1	-12.9	-15.9	-20.6	0.5	0.2	1.8	0.2	6.5	26.2	19.5	20.0	19.2	19.2
18	2.2	4.8	-7.3	-11.1	-15.1	-19.7	0.9	0.4	1.7	0.6	6.1	26.7	19.5	19.7	19.4	18.8
	4.4	6.7	-6.1	-10.2	-14.9	-22.2	1.1	0.2	1.2	0.6	6.7	26.8	19.2	20.3	19.1	18.6
20	7.7	9.5	-5.0	-9.2	-15.1	-22.2	1.5	0.6	0.7	0.2	7.0	26.8	19.2	19.8	19.2	18.5
	8.4	10.4	-4.0	-9.2	-14.9	-20.0	1.4	0.9	1.5	0.5	6.6	26.4	19.7	19.5	19.3	18.6
22	11.7	11.3	-2.0	-8.1	-14.3	-21.3	1.7	0.9	0.8	0.7	7.1	27.0	19.5	19.3	19.8	18.5
	13.8	14.3	-2.0	-6.3	-14.9	-20.9	2.5	1.2	0.5	0.7	7.1	26.7	19.2	19.8	19.0	18.2
24	17.8	15.2	0.0	-6.3	-14.7	-21.8	2.8	1.3	0.4	0.5	7.1	27.1	19.0	19.8	18.9	18.5
	19.7	17.2	1.0	-7.4	-15.1	-21.3	2.8	1.4	0.9	0.1	7.4	26.7	19.3	19.7	18.6	18.6
26	22.8	20.2	3.9	-5.4	-14.9	-20.9	3.7	1.8	0.4	0.1	6.8	26.5	18.8	19.5	19.0	18.8
	23.9	20.2	4.9	-3.7	-13.0	-20.0	3.3	2.0	0.2	0.4	6.5	26.2	19.0	19.7	19.0	18.7
28	24.3	22.0	4.0	-4.6	-14.0	-20.0	3.7	1.9	0.1	0.1	6.5	26.7	18.8	19.3	19.0	18.9
	26.1	22.7	5.0	-3.8	-14.9	-21.3	3.7	2.2	0.2	0.2	7.1	26.3	19.0	19.2	18.5	18.7
30	28.9	23.7	4.9	-4.6	-14.0	-22.2	5.4	1.7	0.0	0.0	7.1	25.8	19.2	19.5	18.5	18.7
	30.6	24.1	7.9	-2.8	-15.1	-22.2	5.2	2.0	0.8	0.3	7.6	26.0	19.5	18.7	18.8	18.8

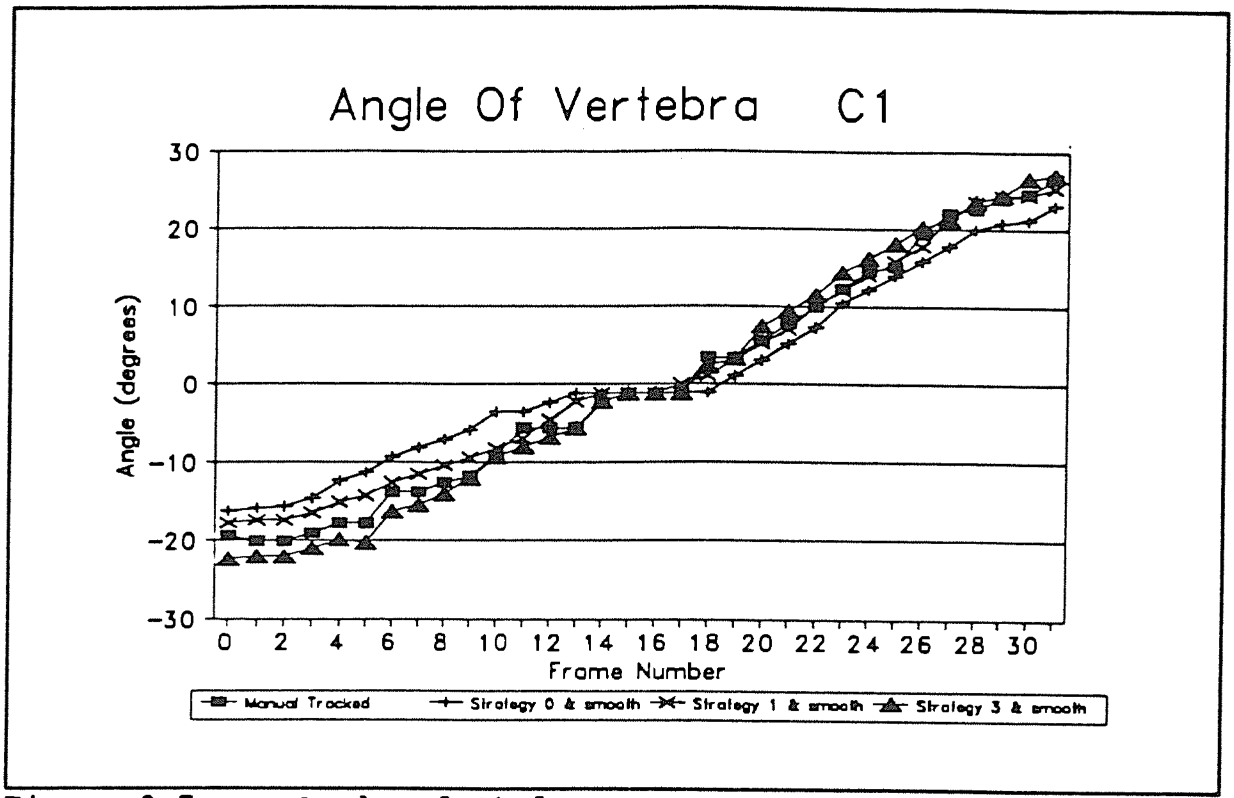


Figure 9-7a Angle of C1 from strategies and smoothing

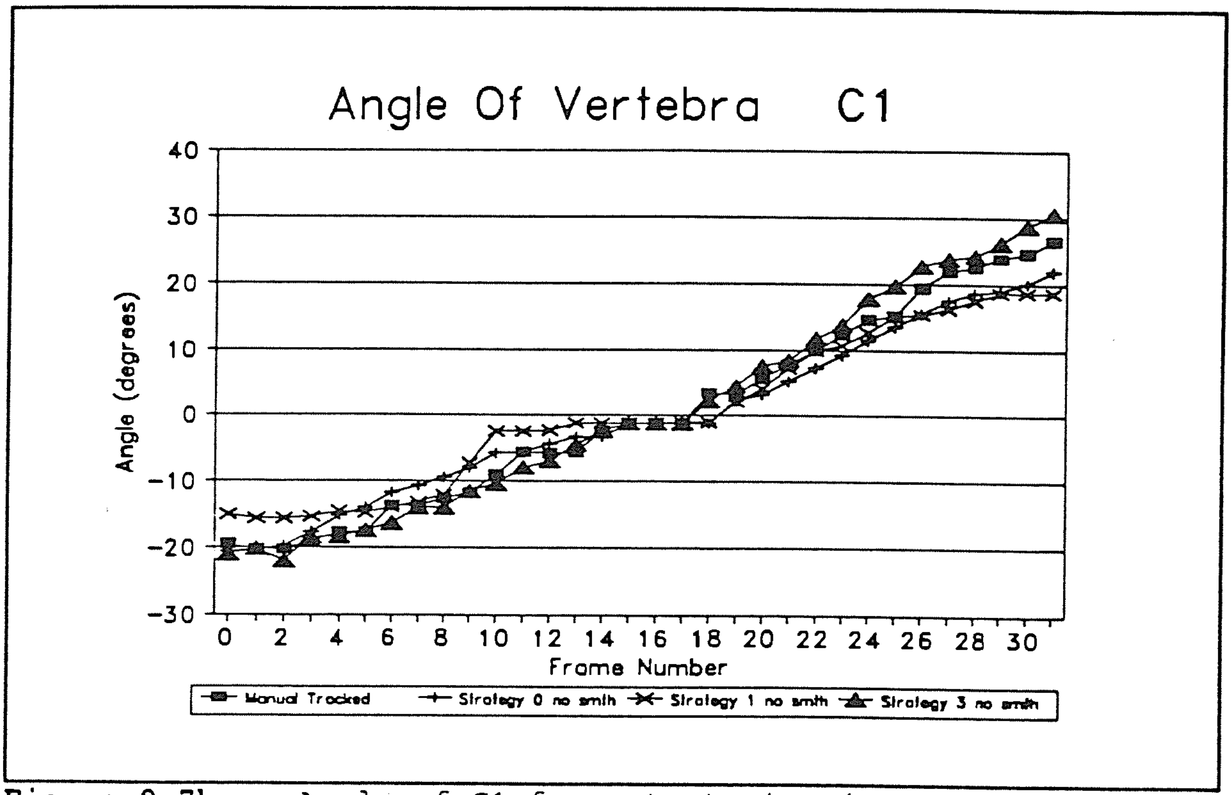


Figure 9-7b Angle of C1 from strategies (no smoothing)

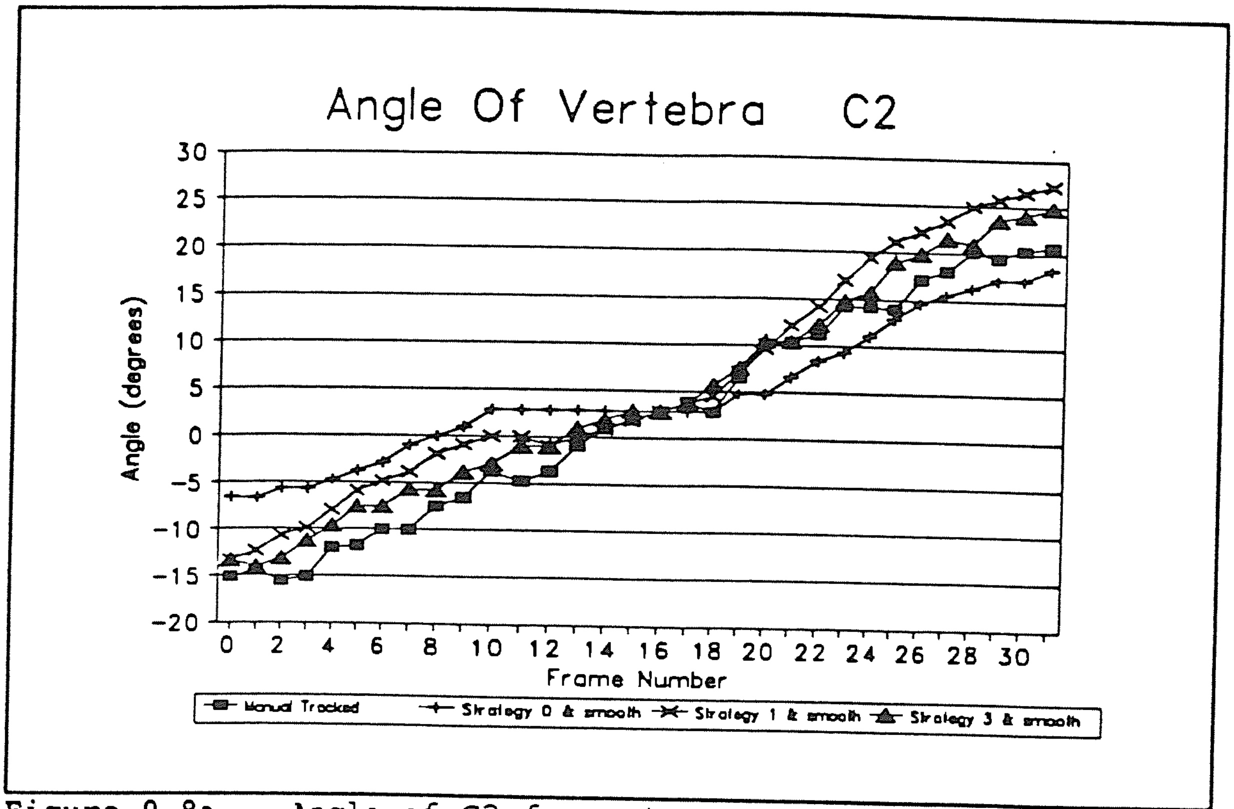


Figure 9-8a Angle of C2 from strategies and smoothing

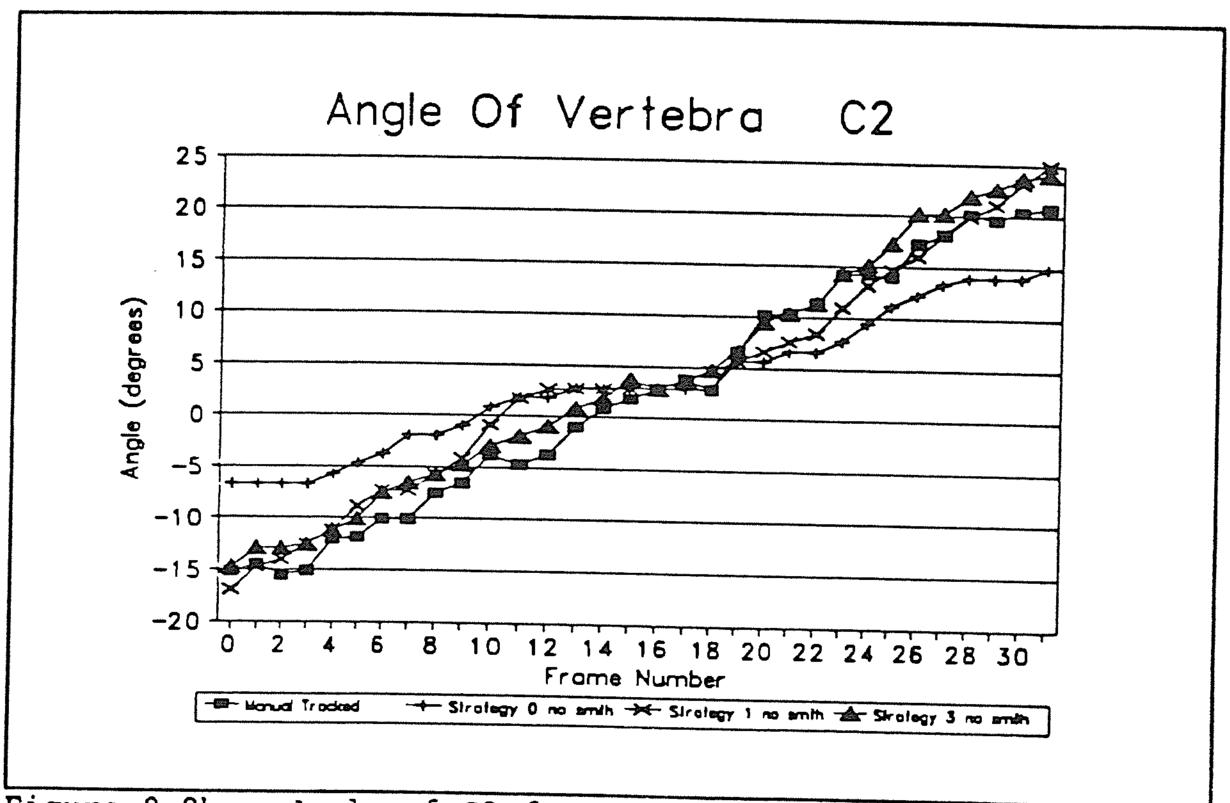


Figure 9-8b Angle of C2 from strategies (no smoothing)

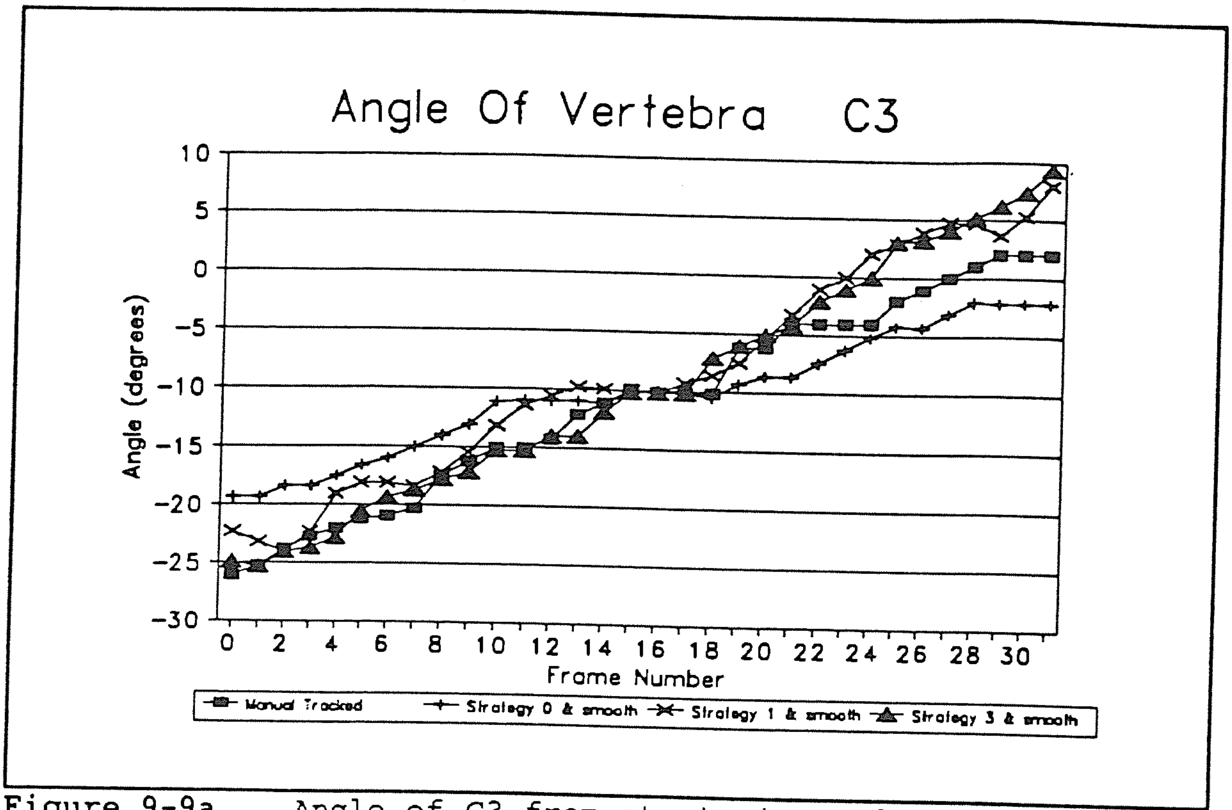


Figure 9-9a Angle of C3 from strategies and smoothing

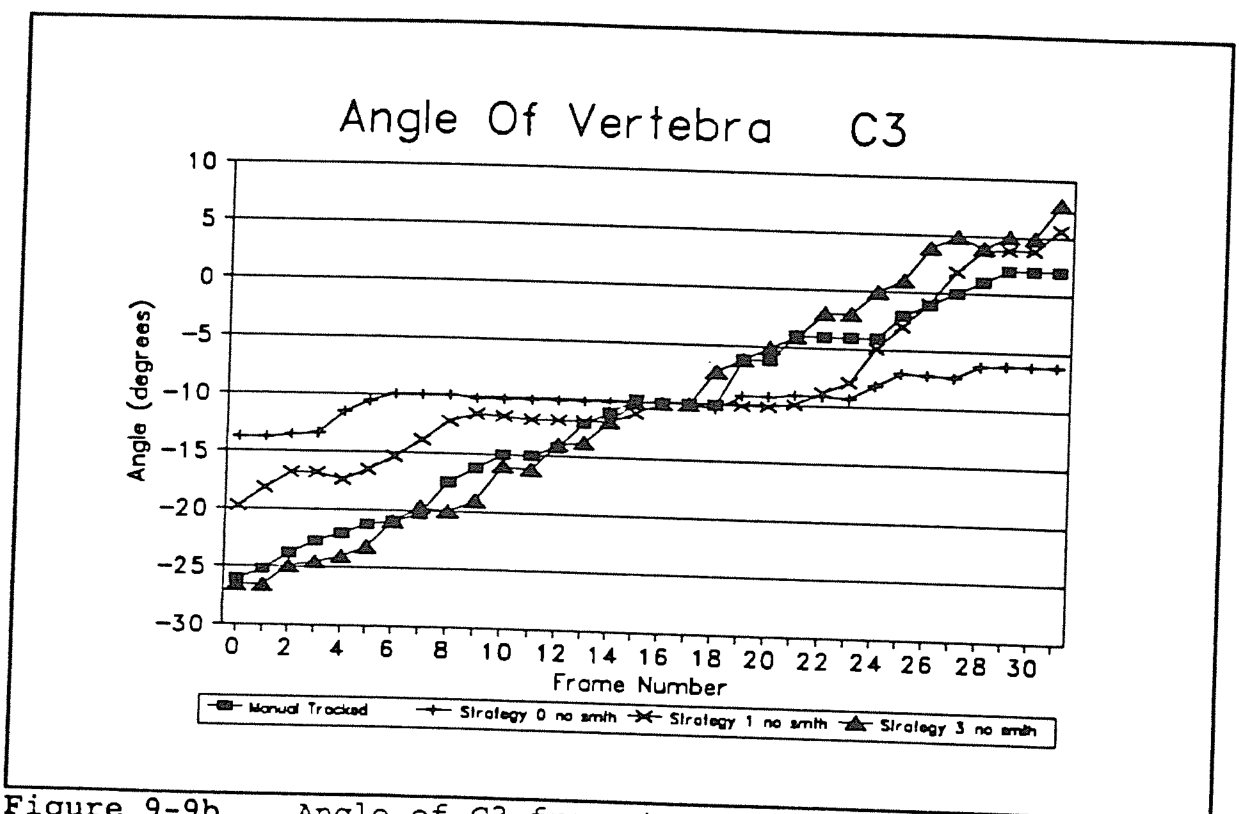


Figure 9-9b Angle of C3 from strategies (no smoothing)

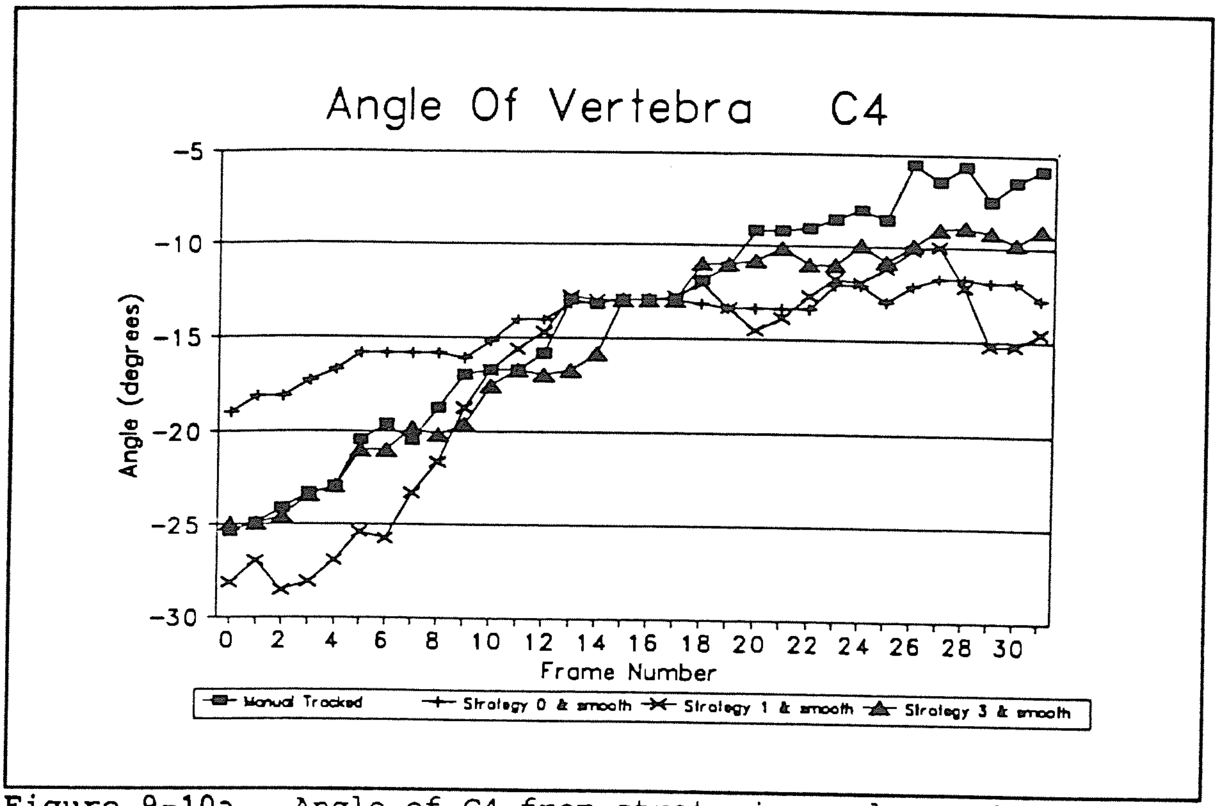


Figure 9-10a Angle of C4 from strategies and smoothing

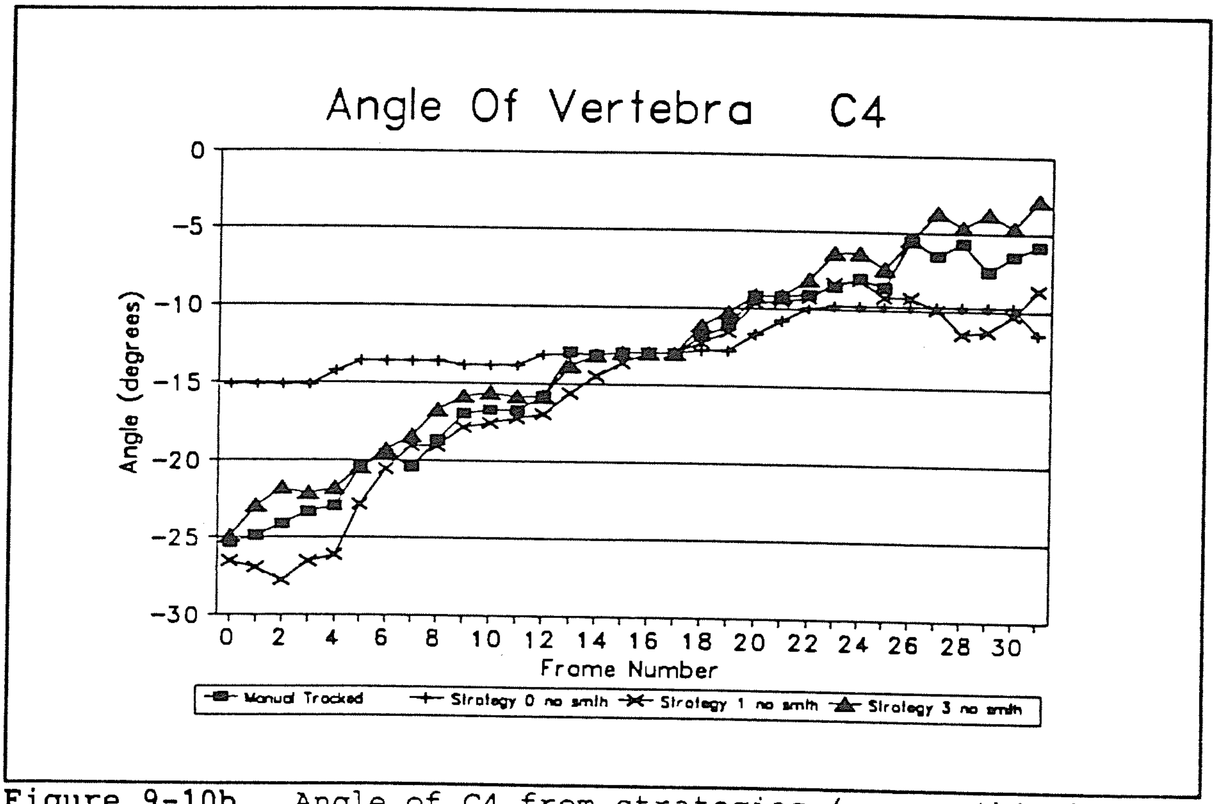


Figure 9-10b Angle of C4 from strategies (no smoothing)

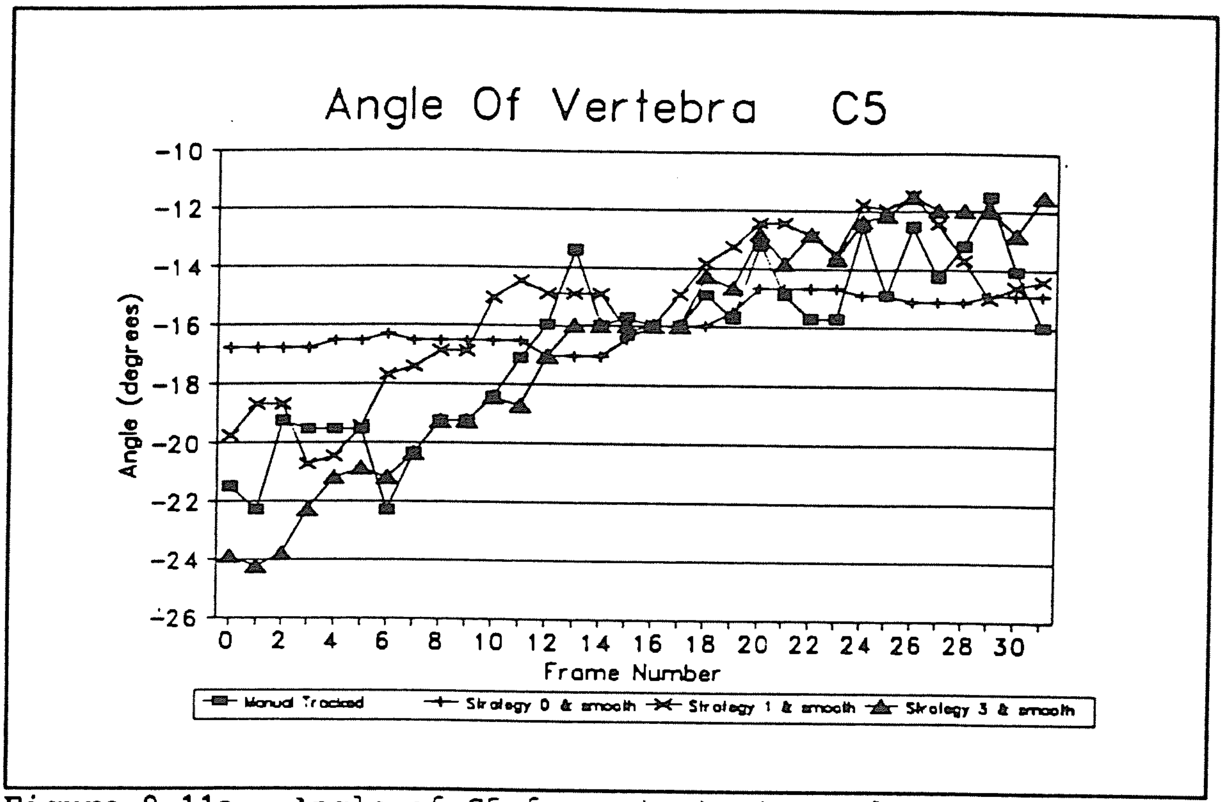


Figure 9-11a Angle of C5 from strategies and smoothing

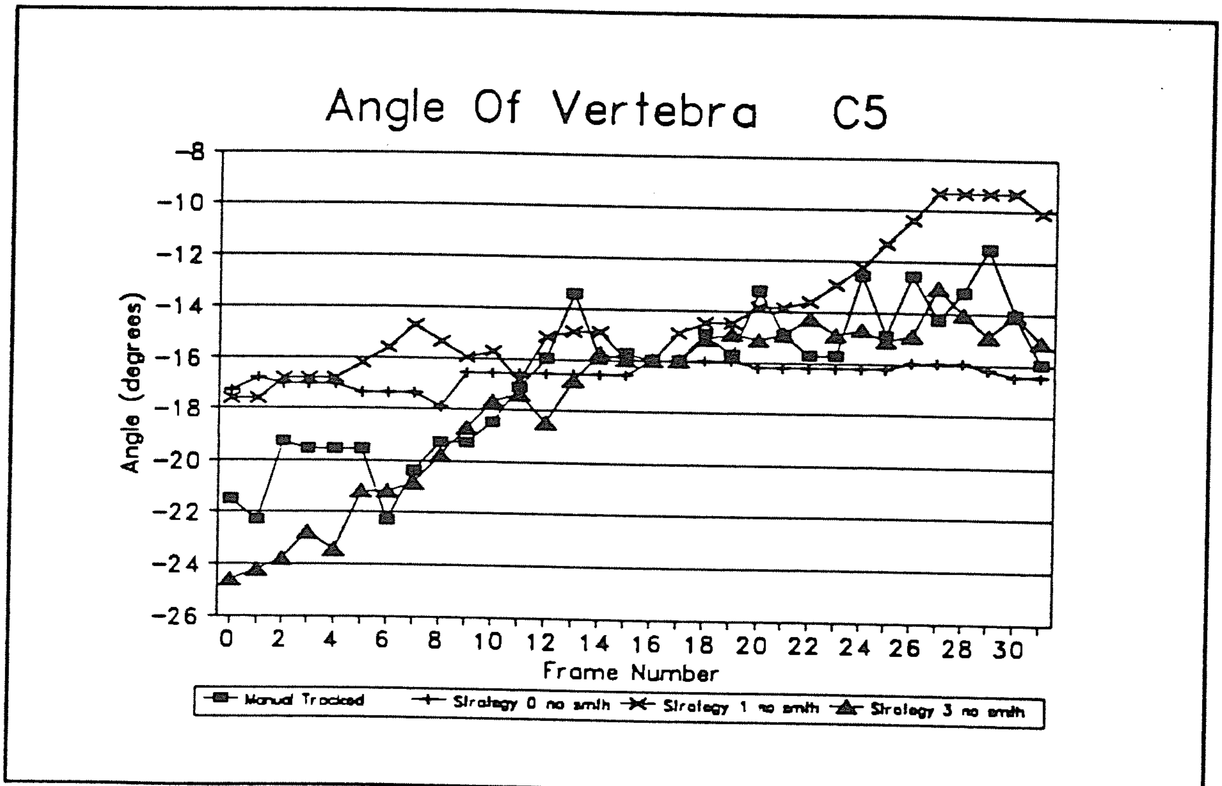


Figure 9-11b Angle of C5 from strategies (no smoothing)

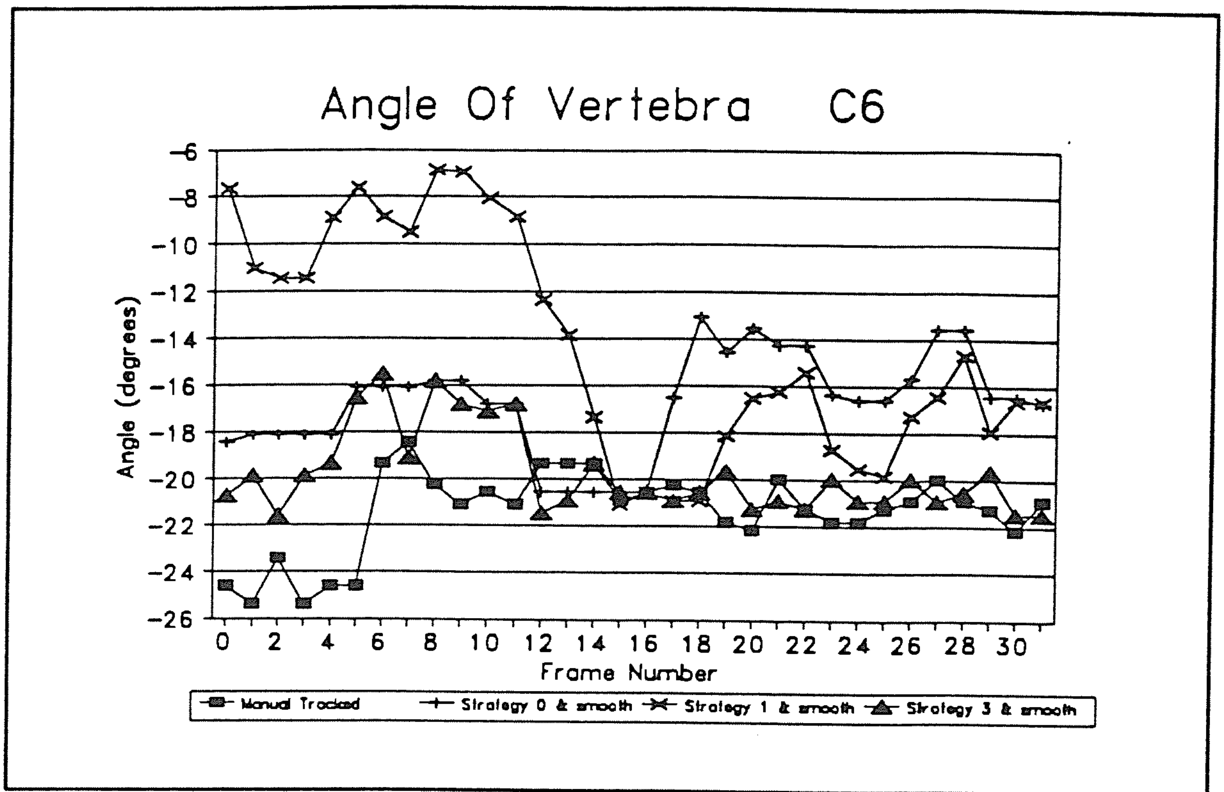


Figure 9-12a Angle of C6 from strategies and smoothing

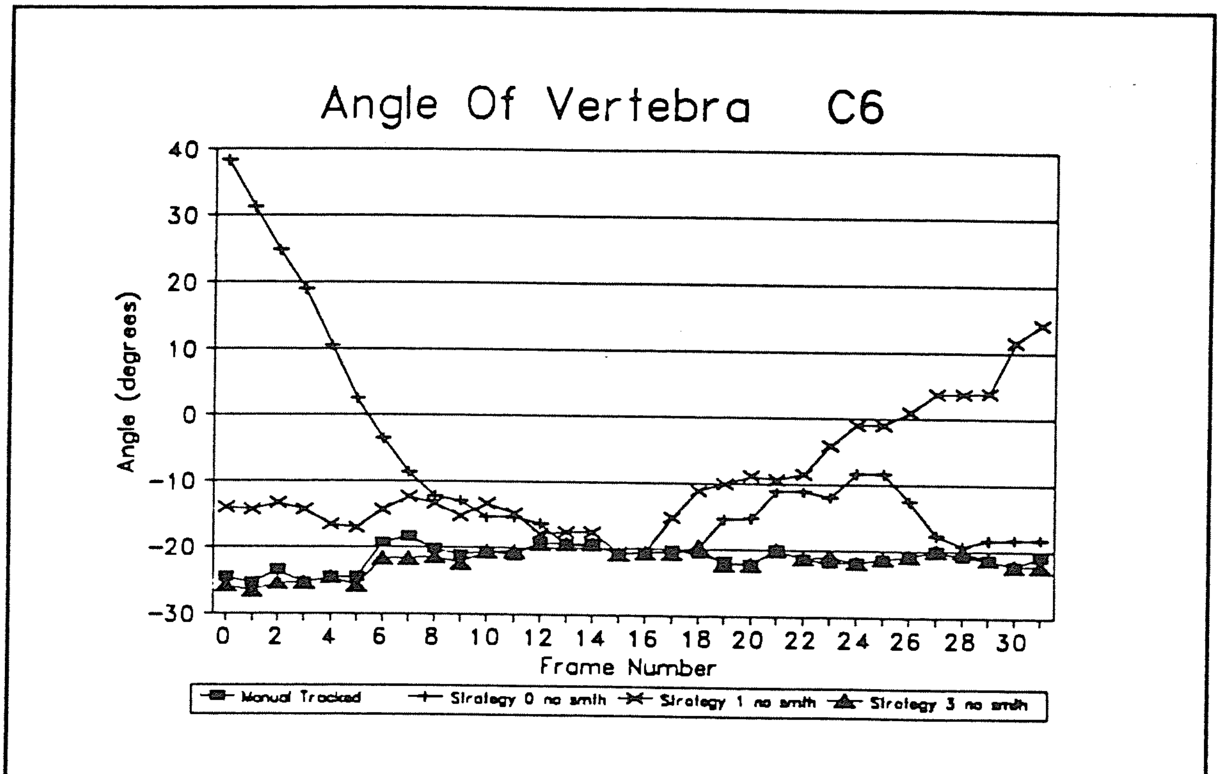


Figure 9-12b Angle of C6 from strategies (no smoothing)

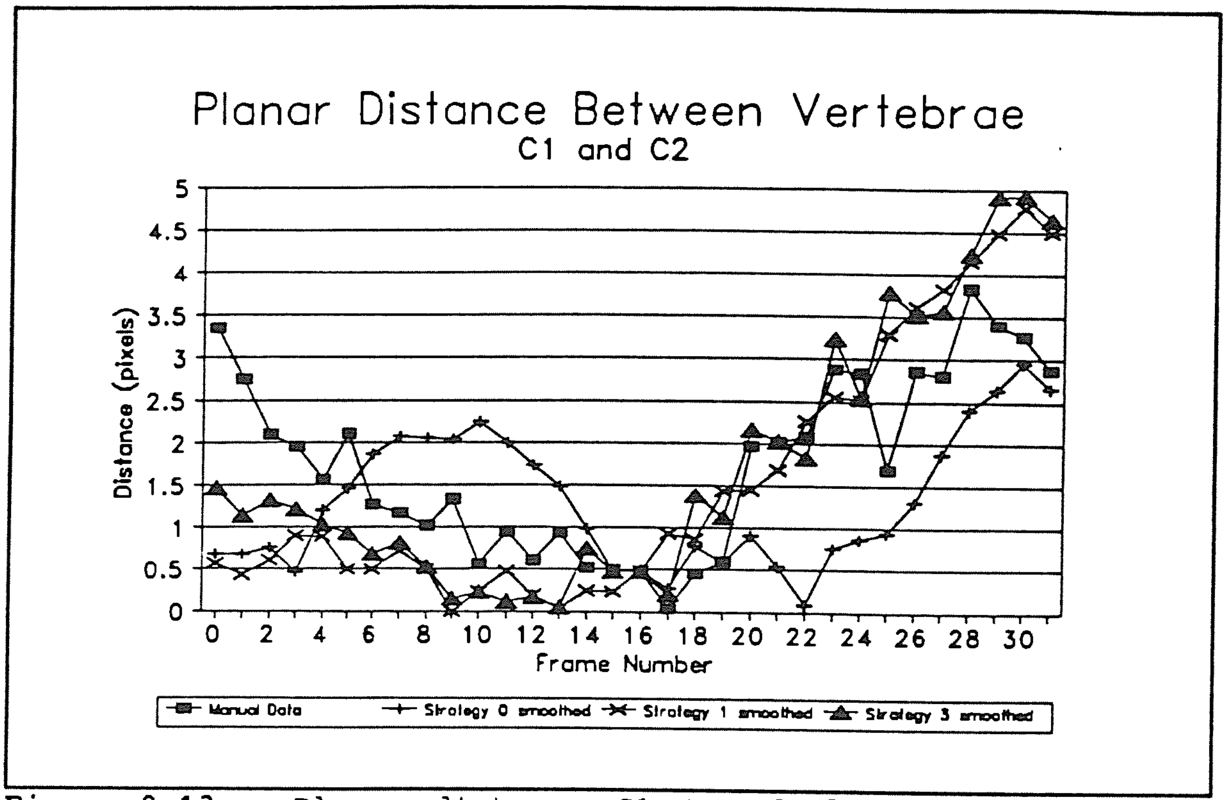


Figure 9-13a Planar distance C1 to C2 from strategies and smoothing

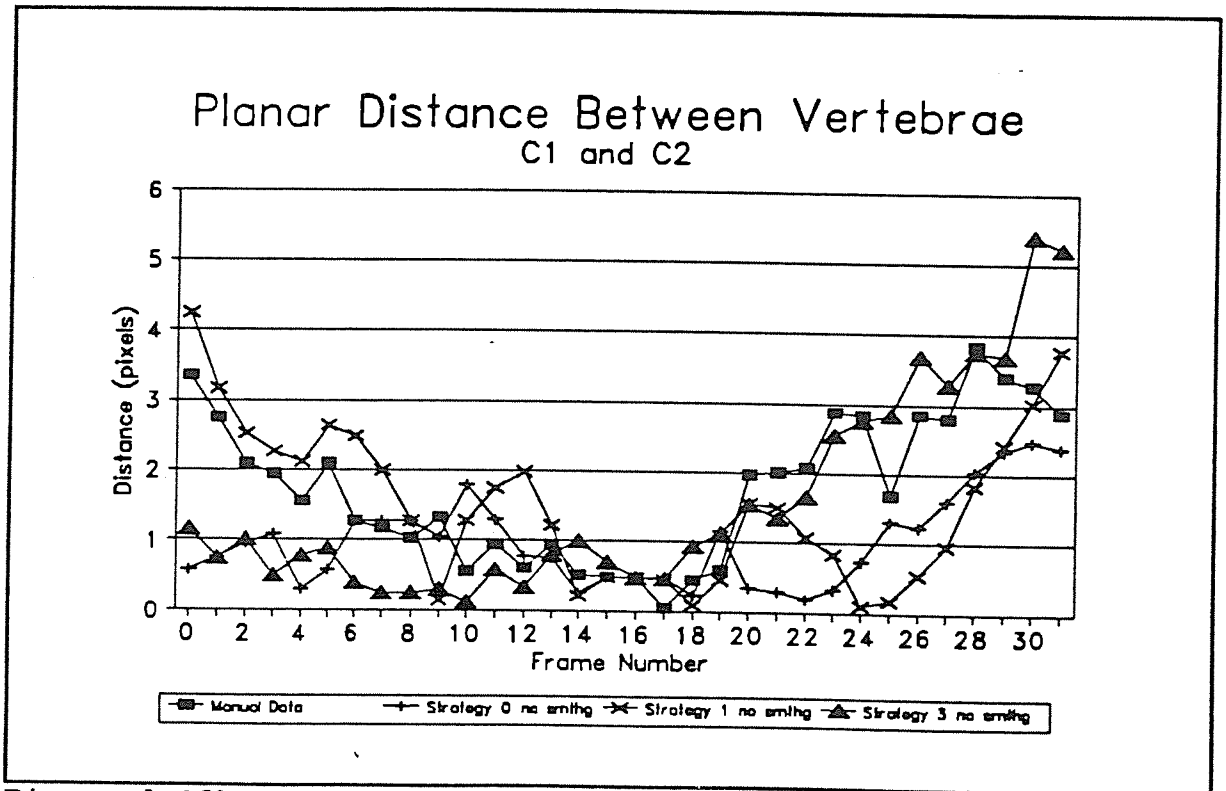


Figure 9-13b Planar distance C1 to C2 from strategies (no smoothing)

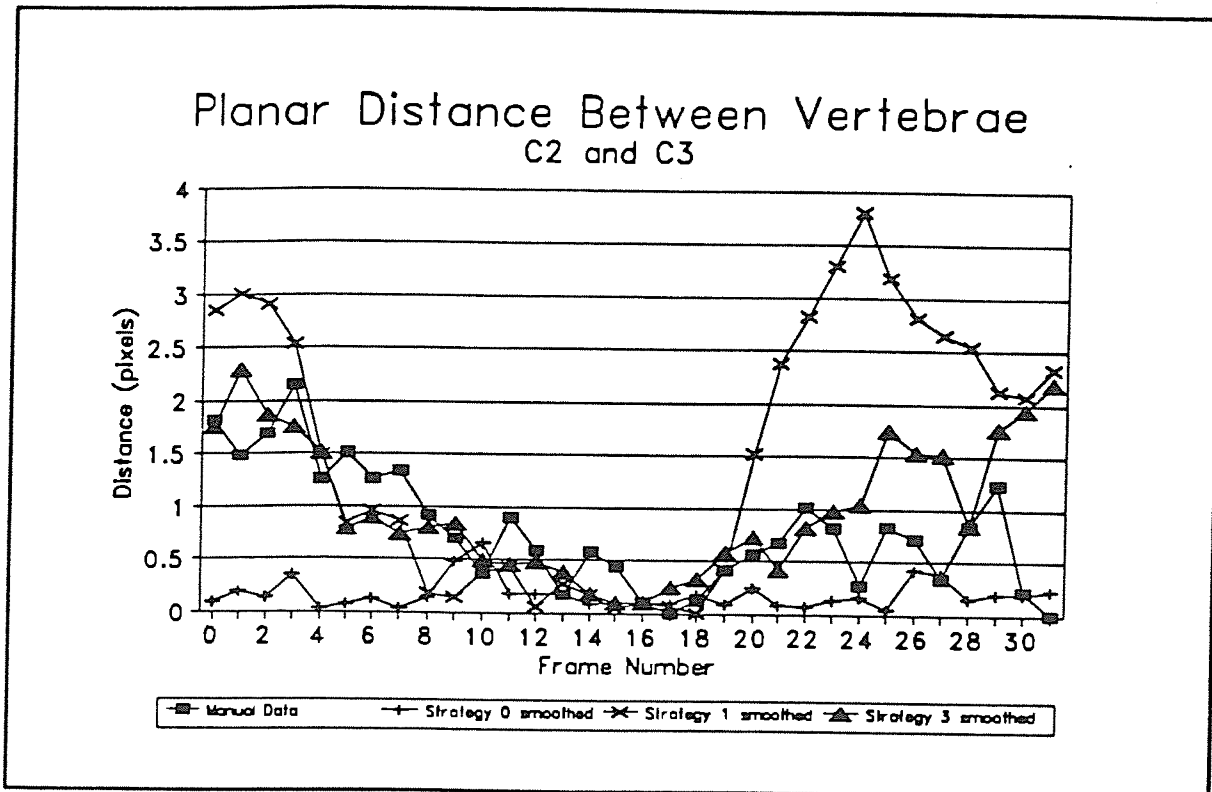


Figure 9-14a Planar distance C2 to C3 from strategies and smoothing

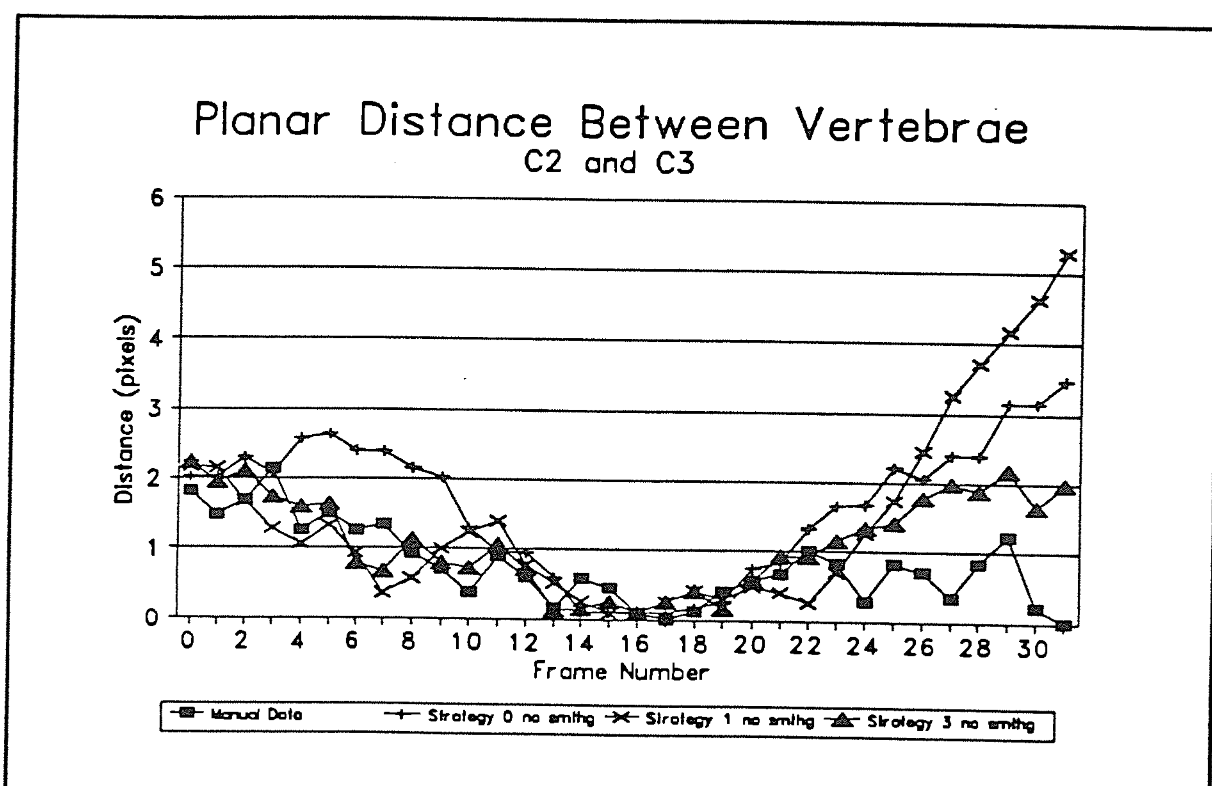


Figure 9-14b Planar distance C2 to C3 from strategies (no smoothing)

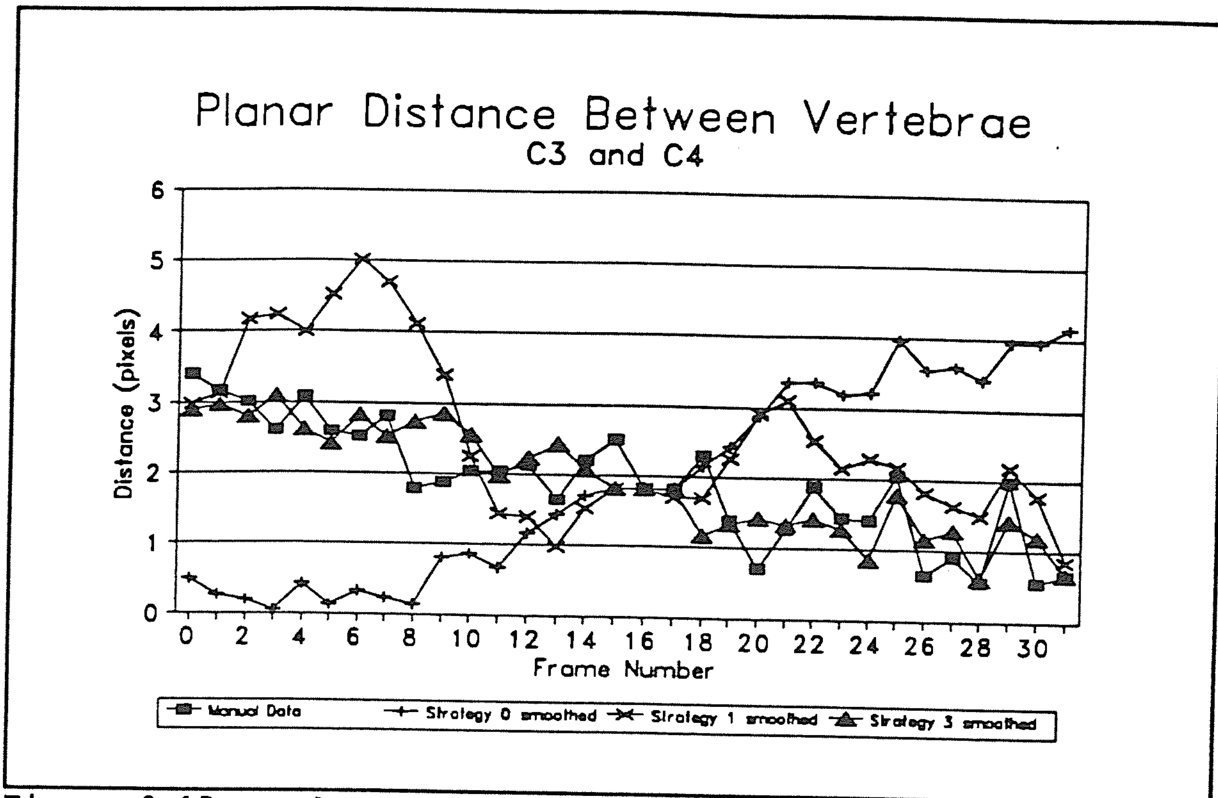


Figure 9-15a Planar distance C3 to C4 from strategies and smoothing

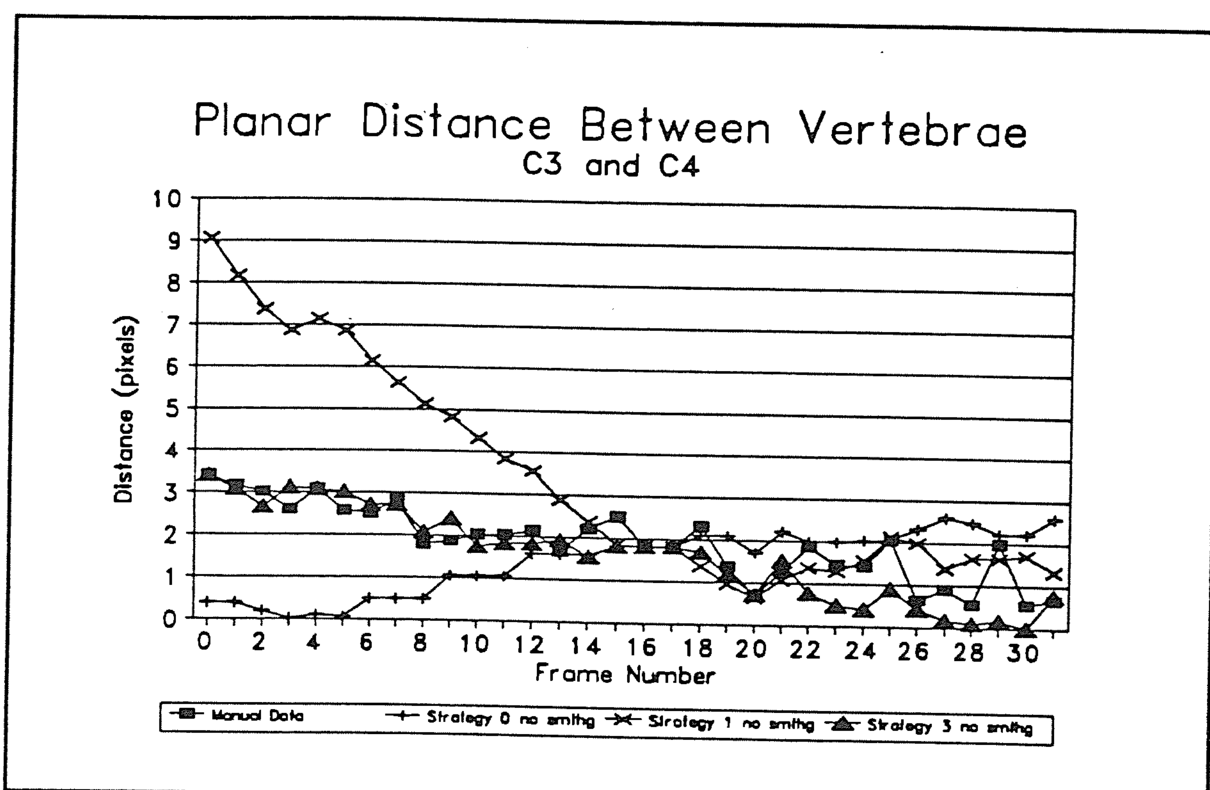


Figure 9-15b Planar distance C3 to C4 from strategies (no smoothing)

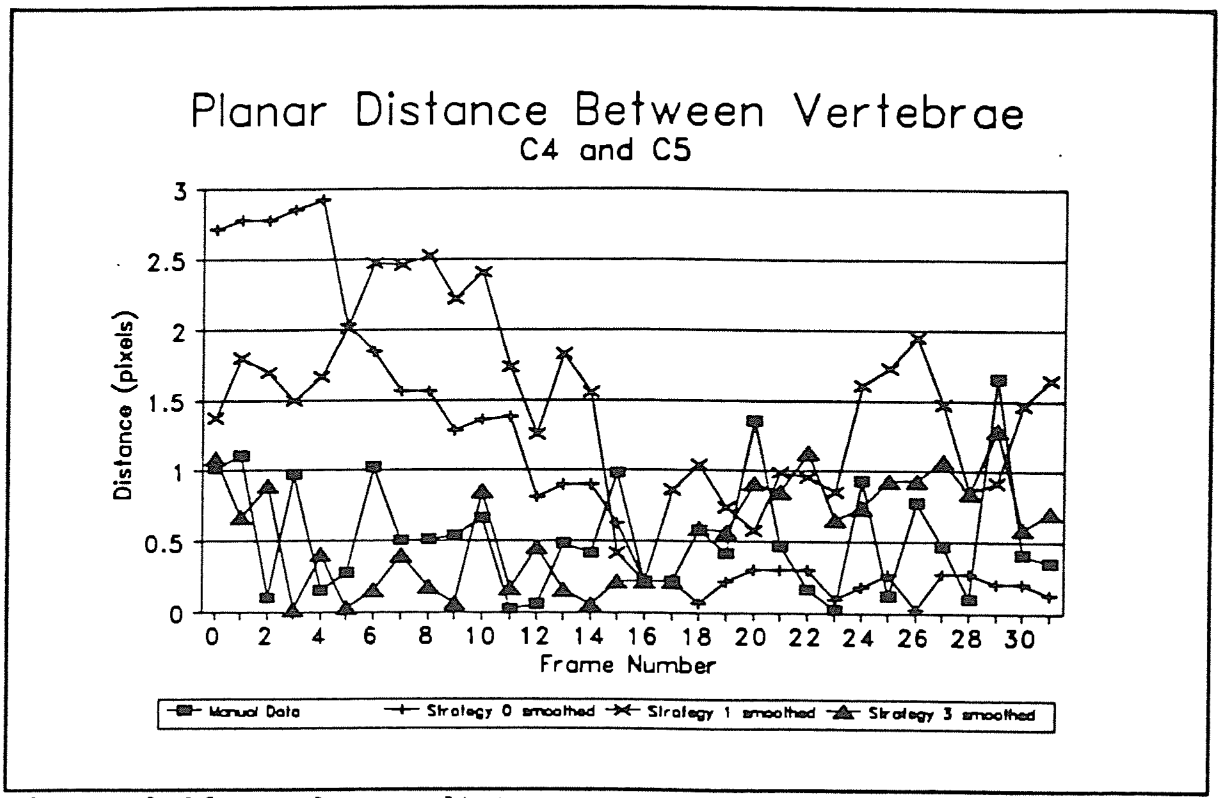


Figure 9-16a Planar distance C4 to C5 from strategies and smoothing

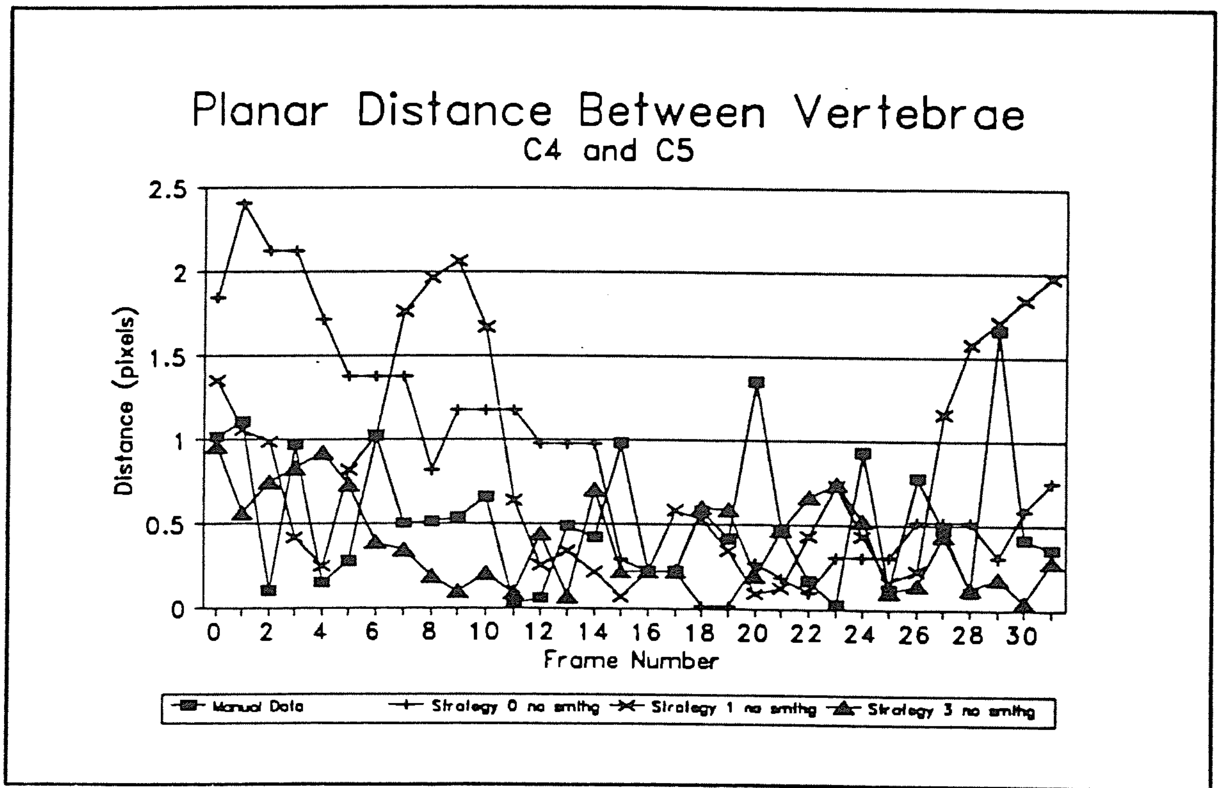


Figure 9-16b Planar distance C4 to C5 from strategies (no smoothing)

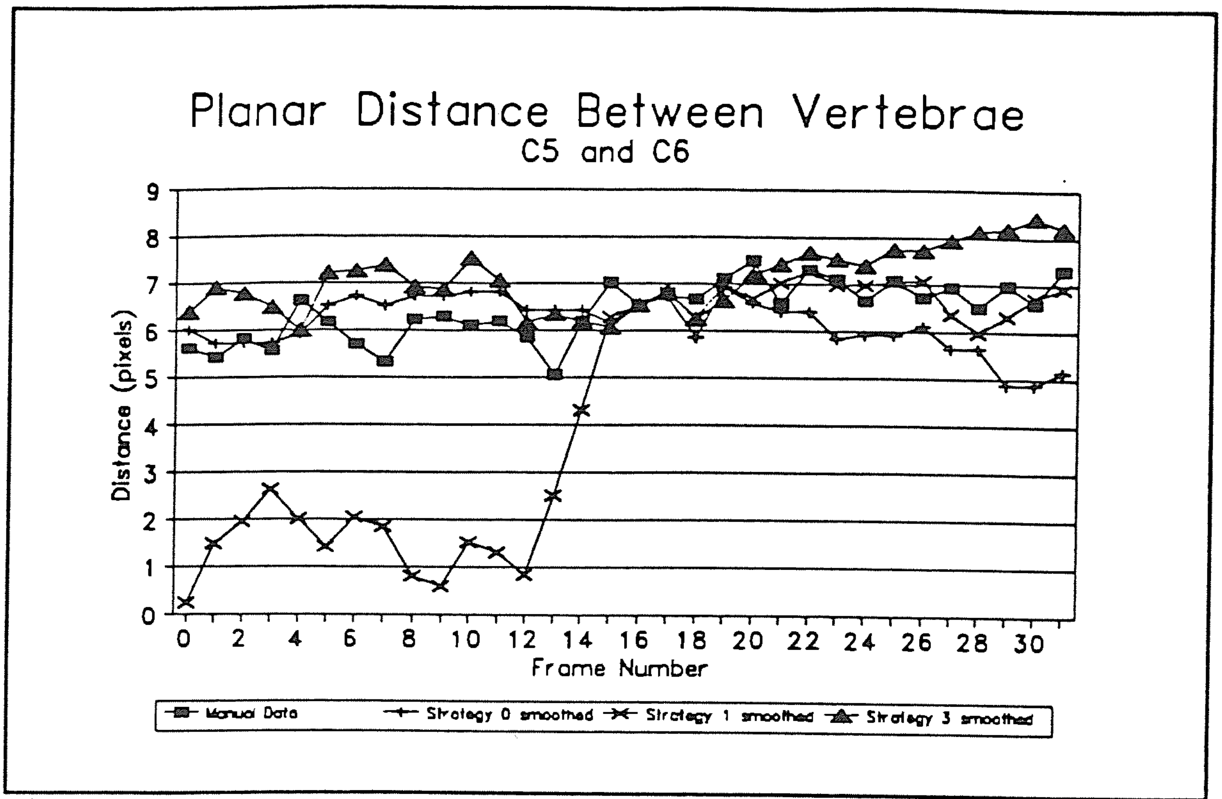


Figure 9-17a Planar distance C5 to C6 from strategies and smoothing

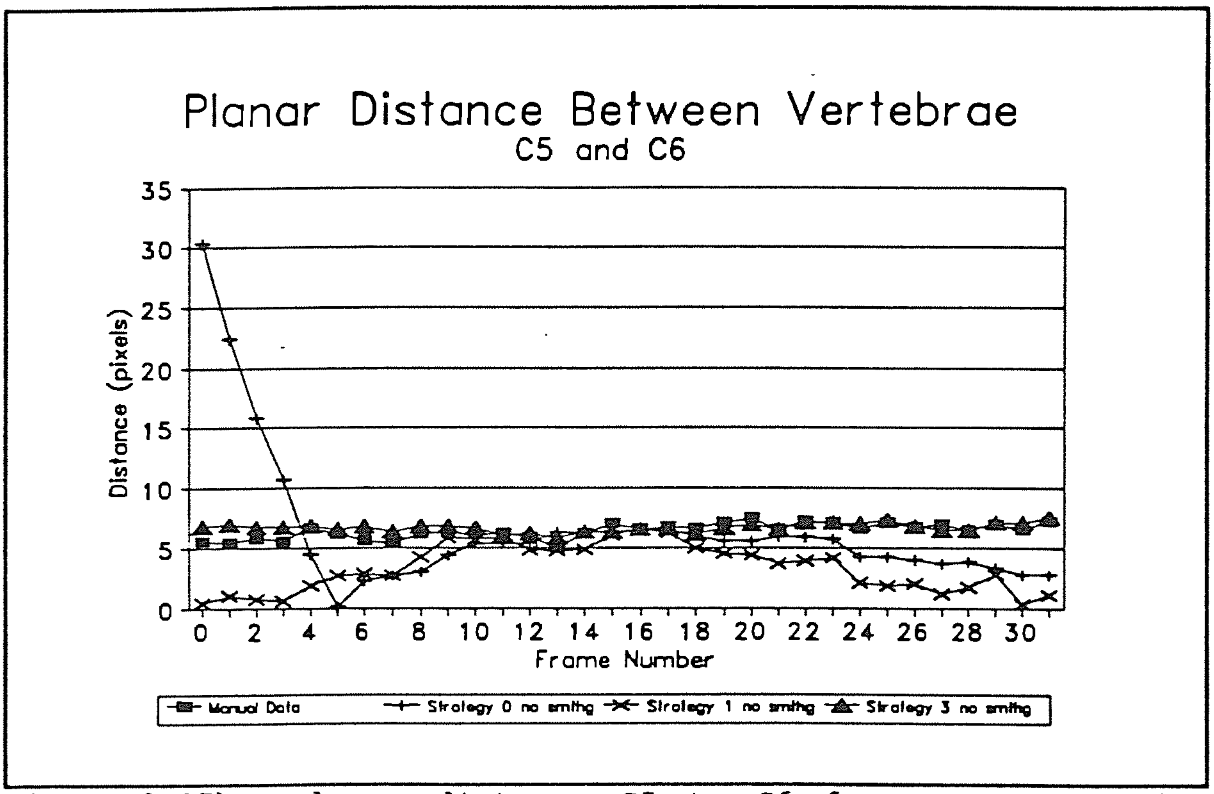


Figure 9-17b Planar distance C5 to C6 from strategies (no smoothing)

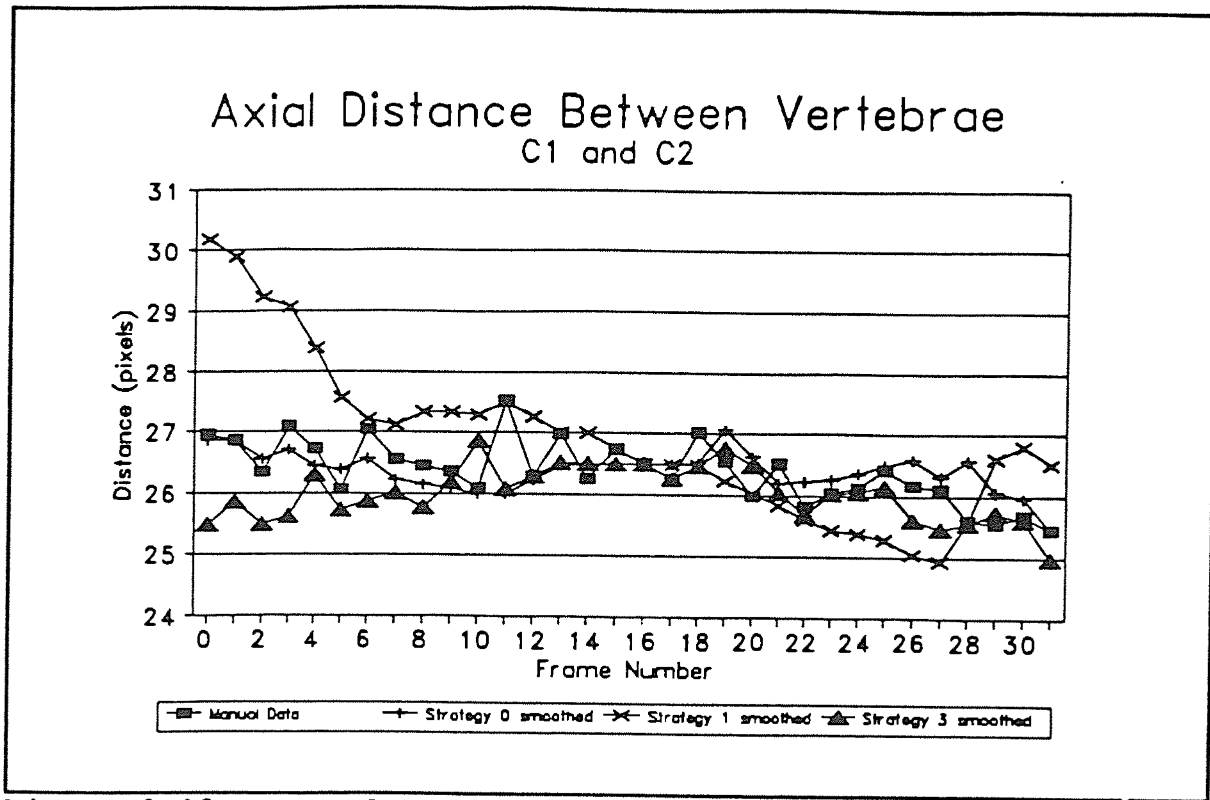


Figure 9-18a Axial distance C1 to C2 from strategies and smoothing

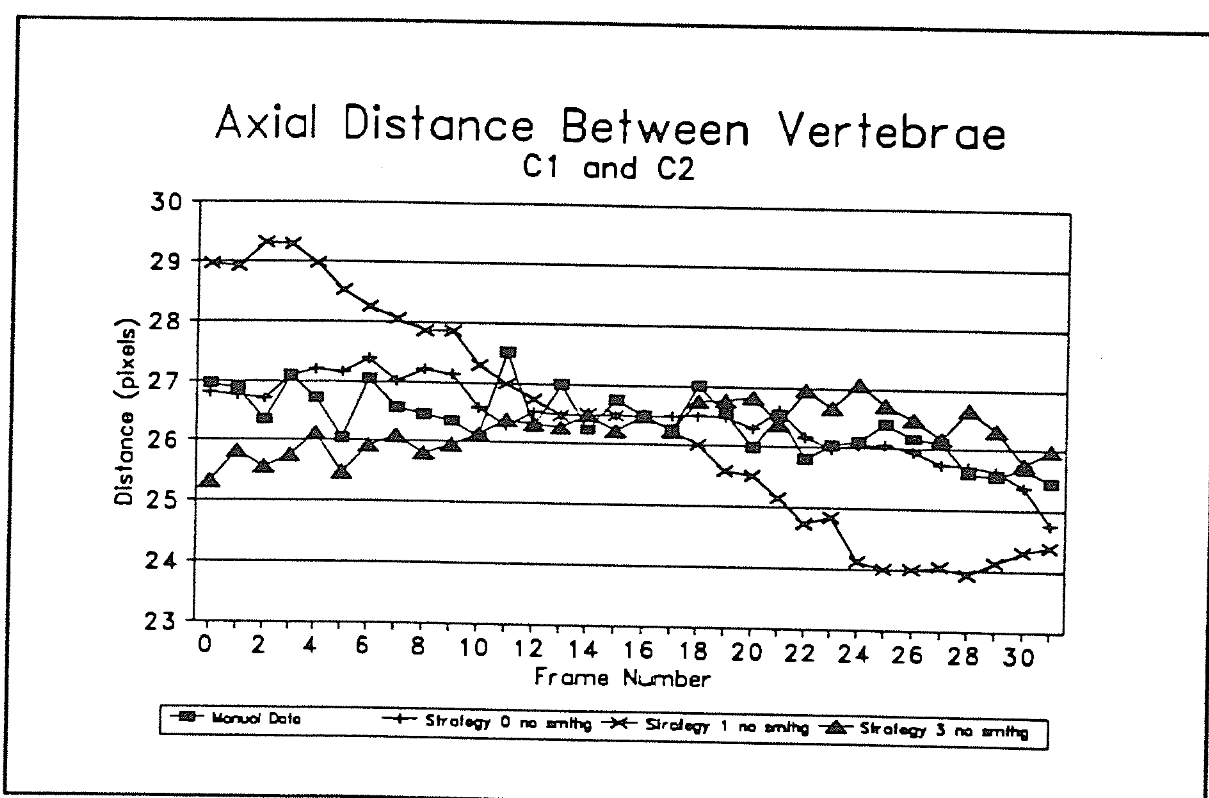


Figure 9-18b Axial distance C1 to C2 from strategies (no smoothing)

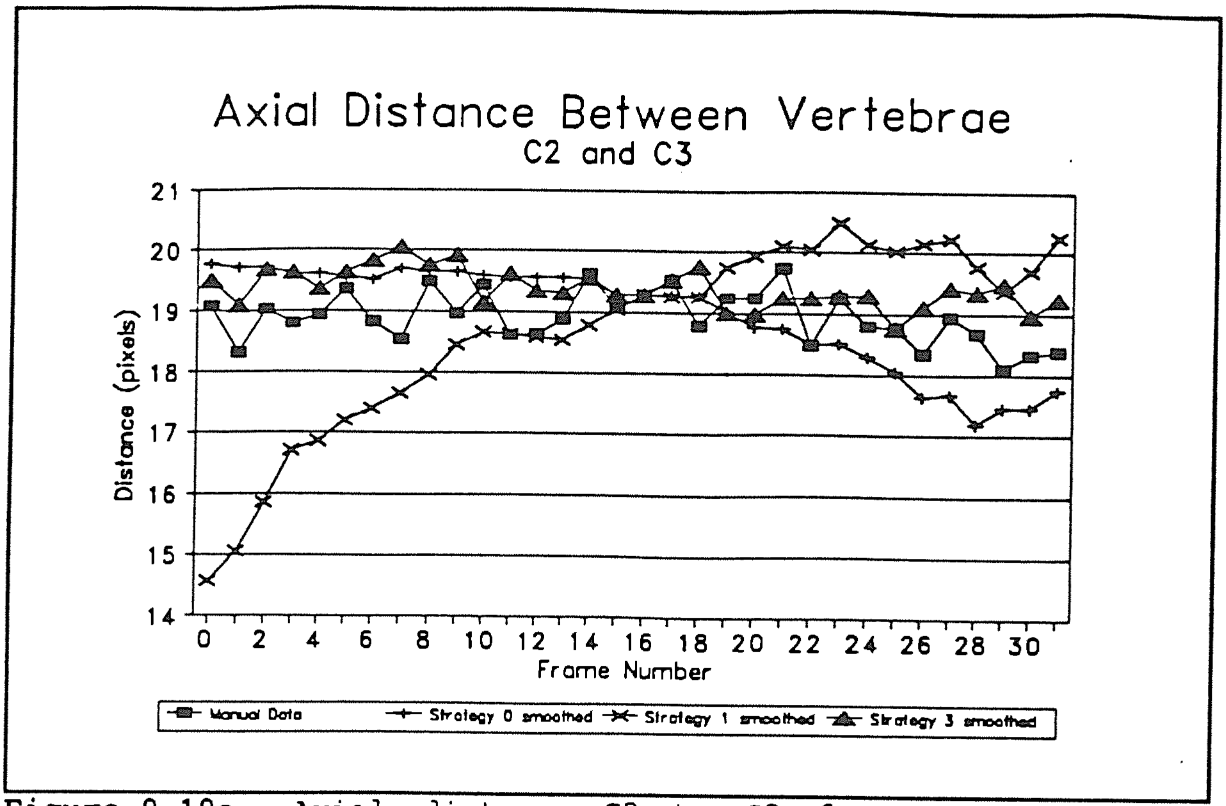


Figure 9-19a Axial distance C2 to C3 from strategies and smoothing

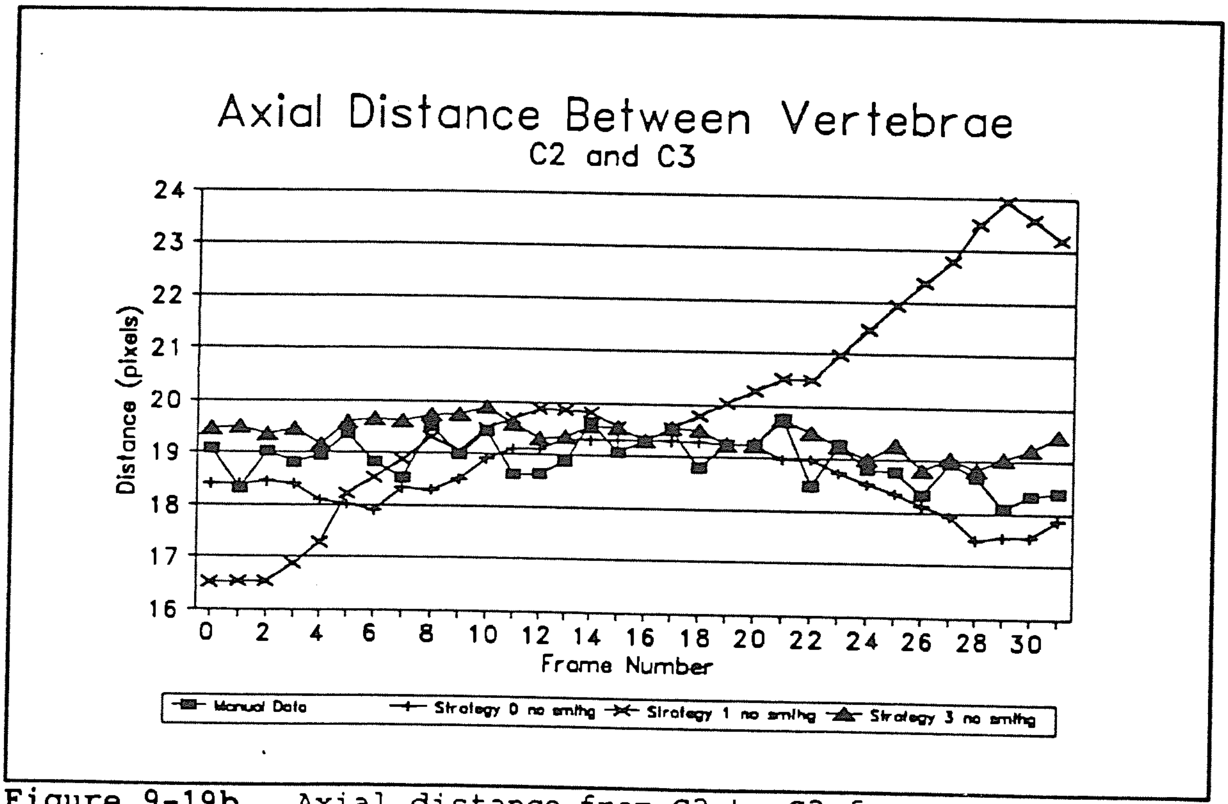


Figure 9-19b Axial distance from C2 to C3 from strategies (no smoothing)

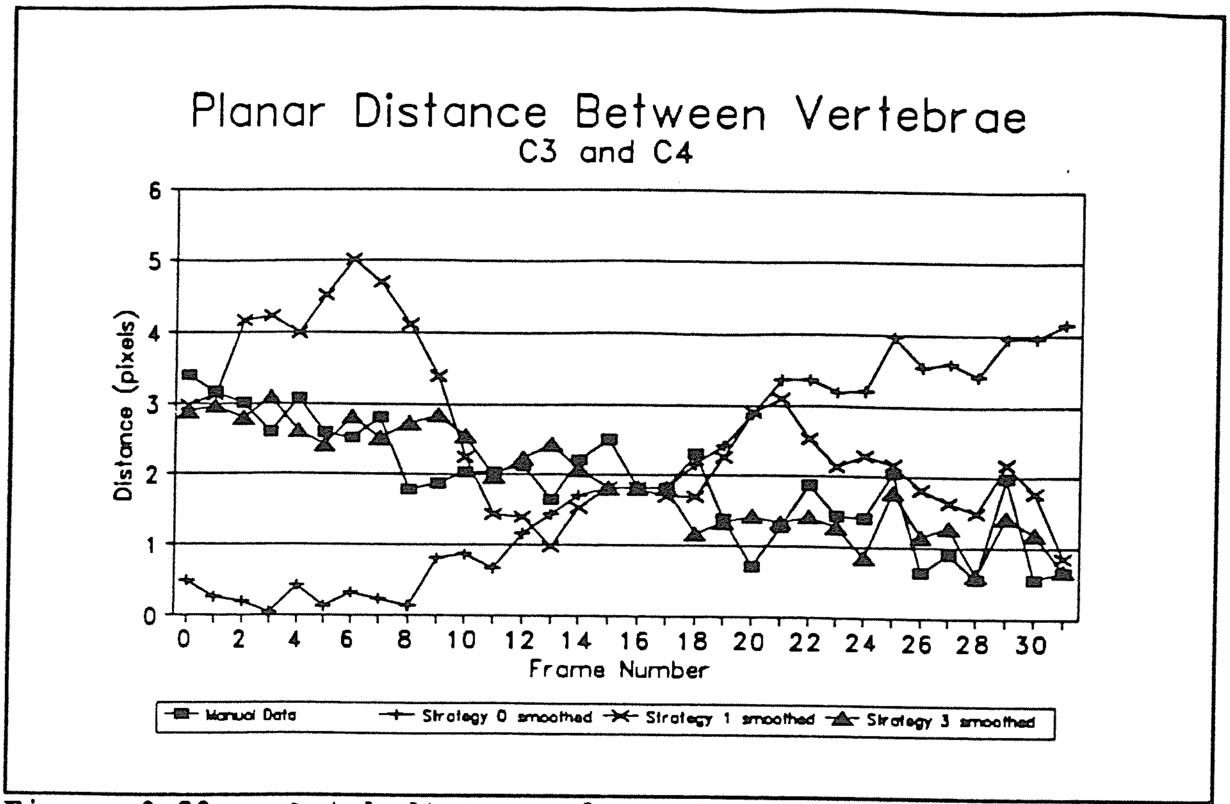


Figure 9-20a Axial distance from C3 to C4 from strategies and smoothing

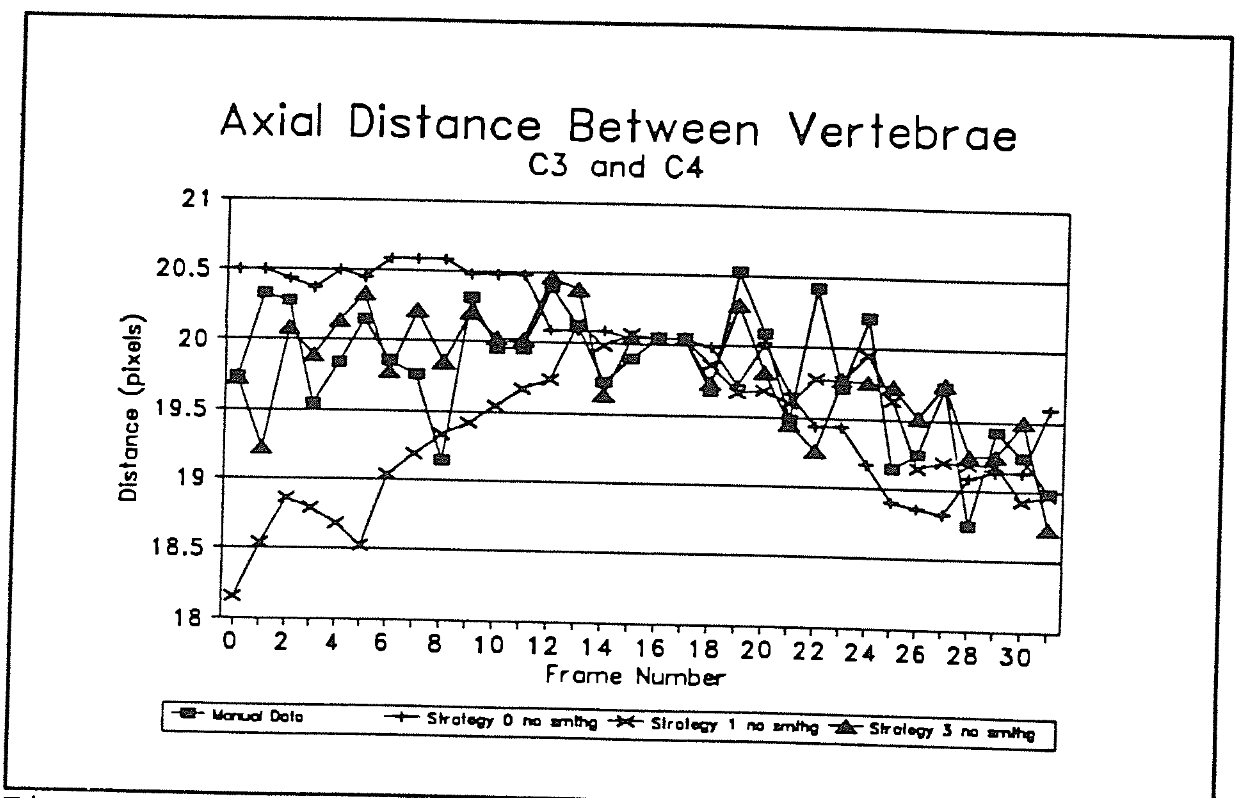


Figure 9-20b Axial distance C3 to C4 from strategies (no smoothing)

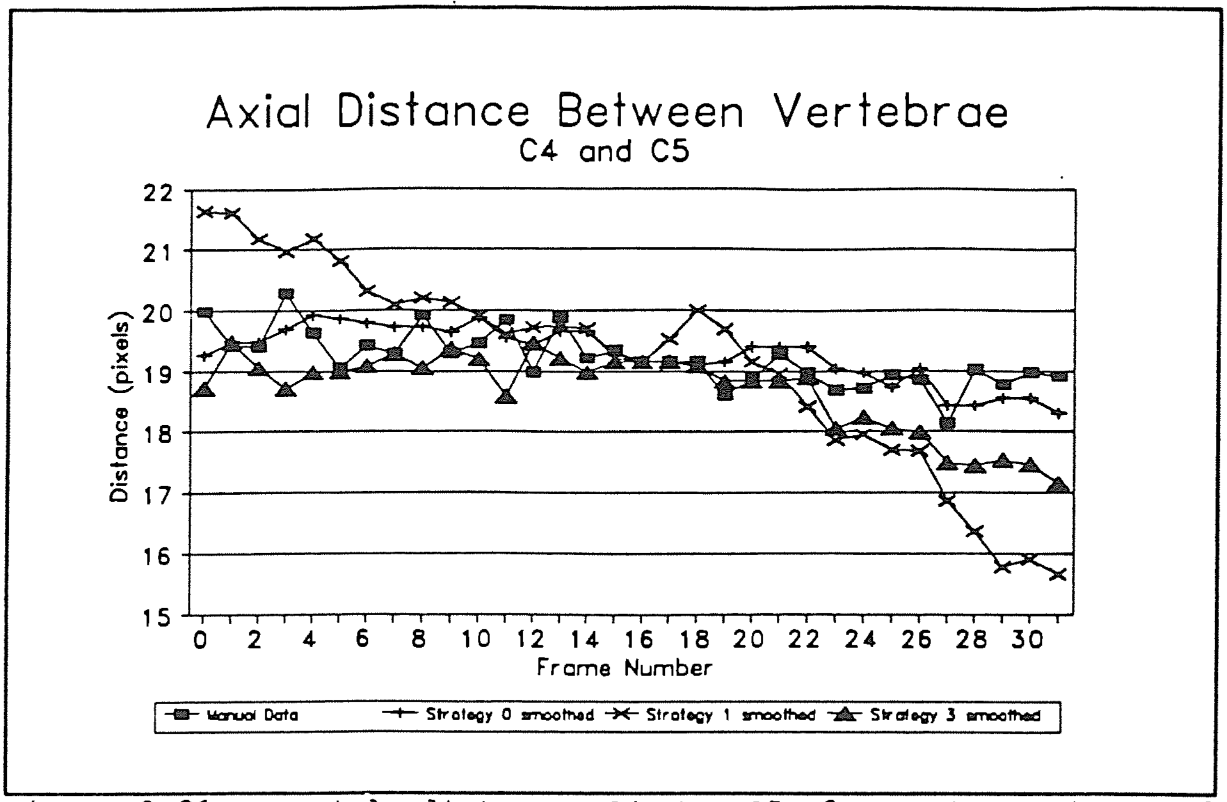


Figure 9-21a Axial distance C4 to C5 from strategies and smoothing

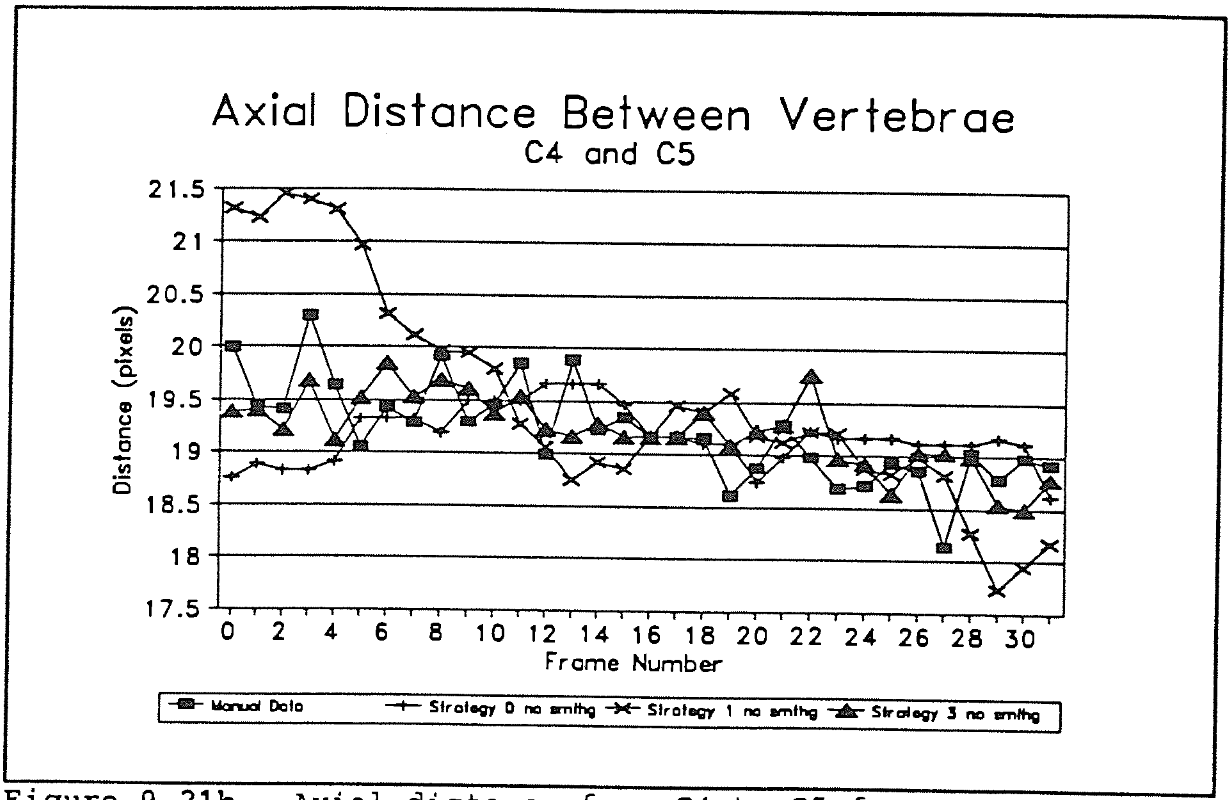


Figure 9-21b Axial distance from C4 to C5 from strategies (no smoothing)

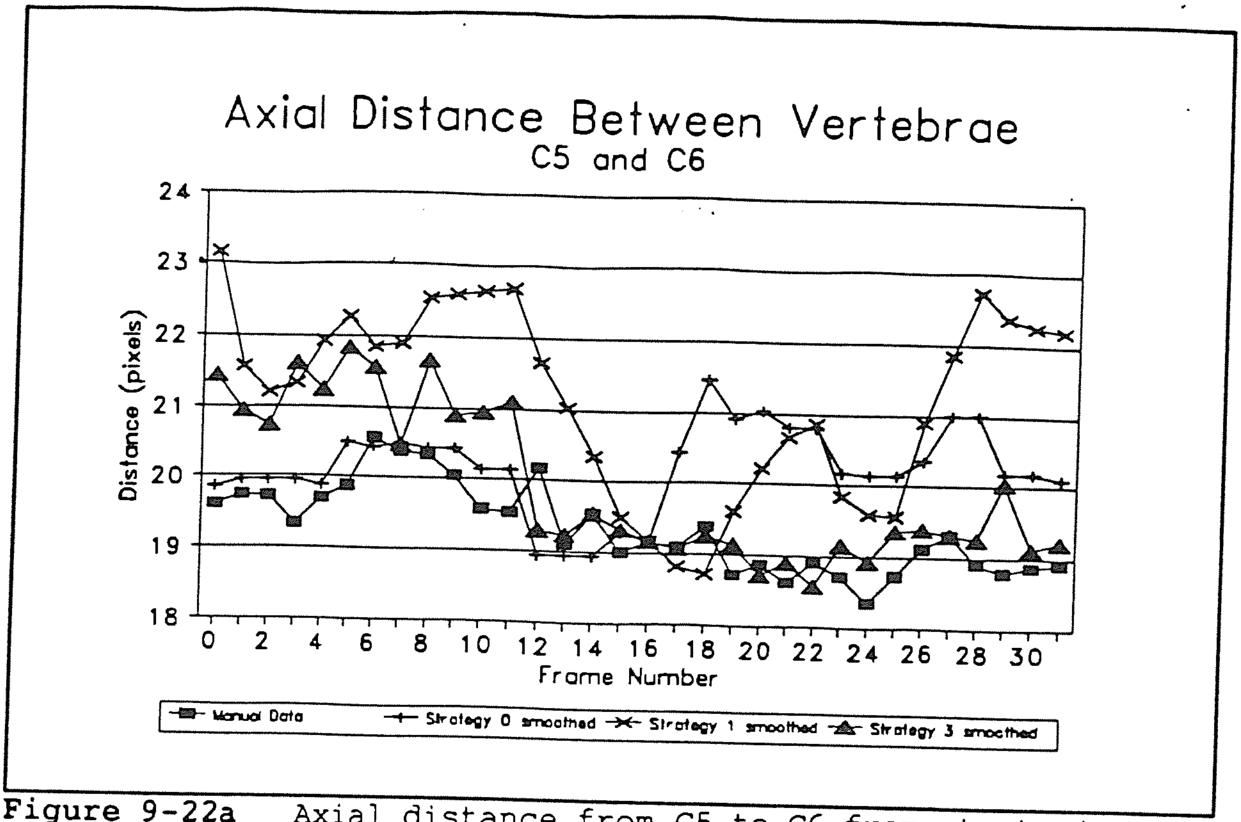


Figure 9-22a Axial distance from C5 to C6 from strategies and smoothing

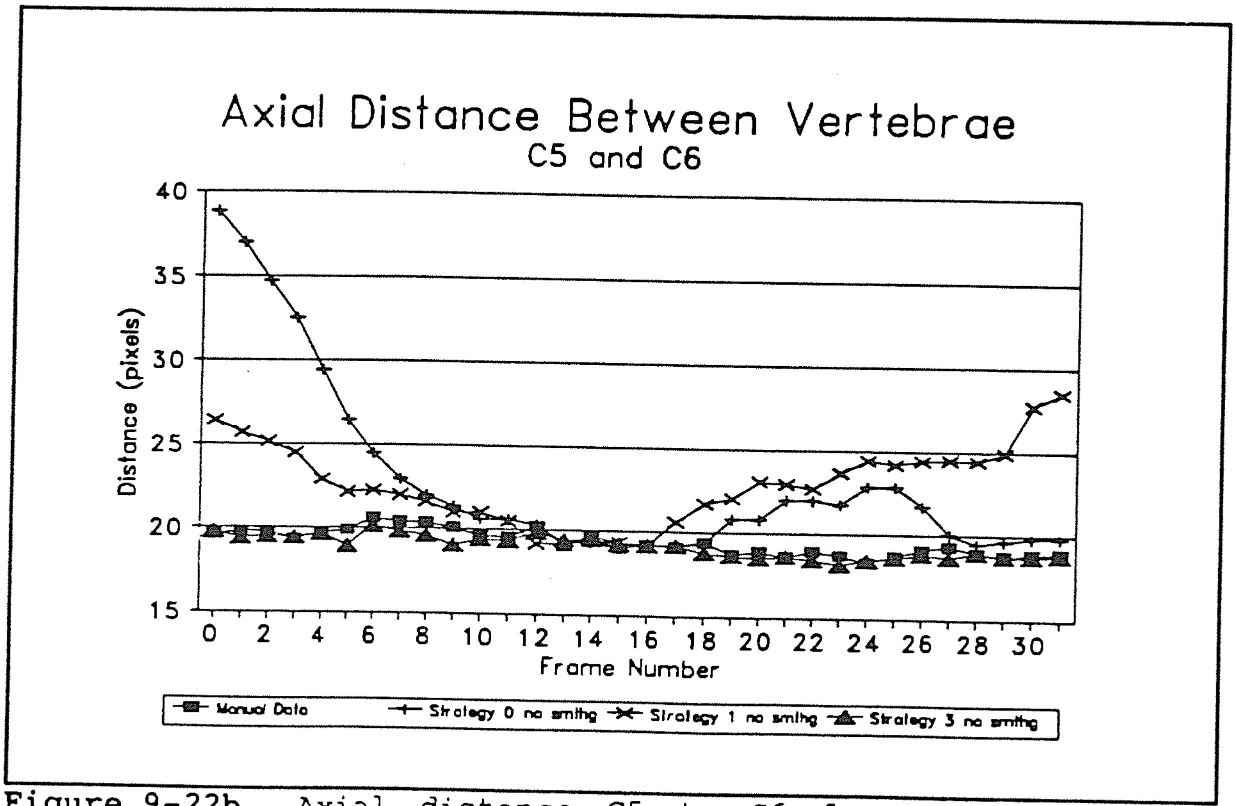


Figure 9-22b Axial distance C5 to C6 from strategies (no smoothing)

Chapter 10
Further work

10.1 Tracking methods to improve overall results

10.1.1 Tracking through a sequence

Normally a point is tracked from the first frame through the consecutive frames to the last frame. There are several variations on this. One is to start in the centre of the sequence of frames and work towards both ends, thus when using a poor algorithm the likely numbers of frames that are tracked within acceptable bounds is greatly increased. There is also a decision whether to make the comparison of windows interframe or with respect to the first frame in the sequence. When tracking interframe rotation is small. Tracking interframe is therefore advantageous if the algorithm does not allow for rotation, however a disadvantage is that the point may wander because of discretisation, and once the algorithm fails to find a correct location there is no chance of recovery unless this is provided by a supervising algorithm. The effect of discretisation may be significantly reduced by allowing for sub-pixel locations. The disadvantage of using the first frame to compare to the whole sequence is that rotation must be allowed for unless the algorithm is unaffected by rotation.

The best, or most robust method, may be a combination of both methods, for example tracking between consecutive frames but checking at regular frame intervals with the first frame (using a rotation and translation calculation), then adjusting if necessary.

10.1.2 Searching an image using the two dimensional logarithmic search

A direct search method discussed by Jain [Jain, 1991] that is very applicable to the tracking work discussed in this report was the two-dimensional logarithmic search. For a search radius of five pixels, which has been found sufficient for the nature of the movement in the video, the number of search locations for the SAD algorithm would be reduced from 121 to between 13 and 21. Currently when scanning for the correct new location the statistic for every pixel within the search window is calculated. A depression in the statistic value is discernable surrounding the correct location match with the correct location of course giving the lowest statistic. This search method utilises this characteristic by using the direction of minimum distortion as the search direction vector. This allows the area of minima to be quickly zoomed in on until the plane of search reduces to a 3 by 3 size and then in the final step all nine locations can be searched. It was found that when the search is sub-pixel this characteristic also holds. Therefore it is suggested that the next logical step when the final 3 by 3 search window has been scanned is to execute a sub-pixel search in the area surrounding the pixel with the lowest statistic - again using the direction of minimum distortion to highlight the particular area of interest.

10.1.3 Recovery of a lost point

10.1.3.1 Referencing to a moused frame

The most obvious way to recover from a lost point has been discussed in section 10.1 and that is to regularly compare the picture to the frame with the moused points using a rotation as well as a translation calculation. If there are discrepancies then the correct points can be recorded and tracking reversed in sequence, using these correct points, towards where the algorithm tracked inaccurately, stopping when the forward tracked and backward tracked locations were consistent. This recovery procedure would reduce the complete tracking speed if it was regularly necessary to re-track but would be a tidy check if the algorithm performed consistently well.

10.1.3.2 Using the Vertebra rigid body property

The vertebrae being tracked are of course bone and, although they are a three dimensional object, during the X-ray capture the z rotation is minimised making it almost ignorable for tracking purposes. A vertebra on the video can therefore be conceptualised as a rigid two dimensional roughly rectangular or trapezoidal shape. The shape outlined by the four points moused onto each vertebra can be compared to and/or enforced on the points reported back after the algorithm has tracked them to each frame. This will allow one point per vertebra per frame to be recovered if the other three points on the vertebra track satisfactory. This is more useful than it first appears as often only one point will deviate, usually due to X-ray shadow or spinal movement that causes two vertebra to touch. If one or more points do not conform to the original vertebra moused shape a signal or alarm could be used to notify the clinician that an error may have occurred in tracking and provide the option to move the points to the correct positions or accept the points as tracked by the algorithm, before tracking continues. If such a check is implemented consideration must be made of what bounds to place on the 'constant' relation because there must be an allowance for the possible minimal rotation in the z direction, towards the camera, which would change the perceived shape of a vertebra.

The vertebrae rigid body property was used successfully during testing to estimate the correct location of vertebra points for measuring the accuracy of the tested algorithms. This was implemented by first manually mousing points on the clearly perceivable features of a vertebra on a selected frame and then manually rotating and translating that outline to superimpose it on successive frames.

10.1.3.3 Prediction of location from motion trajectories

Fletcher [Fletcher, 1990] developed a method of predicting missing points on features given a long image sequence. This would be very useful in the vertebra tracking application as it is common for X-ray shadows to reduce point detection and vertebrae sometimes 'merge' on film which is detrimental when attempting to match image windows. Unfortunately the interpolation algorithms currently defined in literature ([Schalkoff, 1989], [Cipolla, 1990] and [Sethi, 1987]) would have to be modified as they rely on a smoothness of motion criteria that allows points to be linked together to form trajectories across an image sequence and this criteria would not always hold for vertebra motion.

10.2 Alternative statistics

10.21 Methods suggested in literature

Many feature tracking algorithms reported in the literature track lines, contours or curves and rely on clearly defined lines ([Rosin, 1989], [Brint ,1990] and [Deriche, 1989]). For the cinematic X-ray images of the spine it is difficult to locate a complete outline of the vertebra. It may be possible to edge detect [Jähne, 1991] or edge enhance [Castleman, 1979] the area of interest around a vertebra and then track the edges that were deemed trackable.

Other methods reported in literature attempt to match a two-dimensional scene to a three-dimensional predefined data base model of the object being searched for ([Stephen, 1990], [Binfold, 1982] and [Besl, 1985]) . Because vertebra shapes vary from vertebra to vertebra and patient to patient a three-dimensional model is not possible. As z rotation is kept to a minimum in the X-ray sequence is would be possible use a two-dimensional outline of the vertebra to aid tracking, if the majority of the points being tracked remained accurate.

10.2.2 Moment of inertia

One possible algorithm calculates the moment of inertia, fr^2, where f in the intensity value of the pixel and r is the distance from the central pixel in the window, for each point within some radius of a selected location. This statistic is independent of rotation and thus reduces computational overhead.

10.2.3 Rate of change of intensity statistic

Another type of statistic that may perform well is one that compares the rate of change of intensity across two windows. One advantage of this statistic is that it is based on locating a change of intensity pattern rather than a pixel to pixel comparison. Another benefit is that the change in overall intensity between frames has no effect as it is the changes across each window that are being compared. Two rate of change statistics were defined. The first statistic used the rate of change to weight the intensity difference. If windows being compared are of size n, from frames p and q, where $f_p(i,j)$ is the intensity of the pixel at position (i,j) in the current window somewhere in frame p, then the statistic is given by :

$$LS = \sum_{i,j}^{n} \left(f_p(i,j) - f_q(i,j) \right) * \\ [\left((f_p(i-1,j)-f_p(i+1,j)) - (f_q(i-1,j)-f_q(i+1;j)) \right) \\ \left((f_p(i,j-1)-f_p(i,j+1)) - (f_q(i,j-1)-f_q(i,j+1)) \right)$$

The spatial rate of change statistic (defined below) is very similar but the rate of change is taken across adjacent cells not cells one apart. It is rationalised by scaling the statistic by the difference between the average brightness of the two windows being compared where as the first statistic has a more localised scaling factor.

$$MS = n^2 * \sum_{i,j}^{n} \frac{1}{f_p(i,j) - f_q(i,j)} * \\ \left[\left((f_p(i,j)-f_p(i+1,j)) - (f_q(i,j)-f_q(i+1,j)) \right) + \\ \left((f_p(i,j)-f_p(i,j+1)) - (f_q(i,j)-f_q(i,j+1)) \right) \right]$$

10.2.4 Feature tracking

When viewing a frame of the video it is often easier to determine where a corner of the vertebra is by finding the intercept of the parts of the sides of the vertebra that are distinguishable. It is hypothesised that this may be a good foundation for another algorithm. In some frames the sides of the vertebra are clearer than in others depending on how the edge lined up with the camera senses and the effect of shadowing, however the true side can always be determined within two pixels. This algorithm would have the advantage that it could always recover itself and could compensate for rotation at a very low computational cost as the pixels that defined the side could be rotated instead of a large window.

Chapter 11
Conclusions

The accuracy of the measurements of the spine made by the spinal movement analyser depends on the tracking algorithm providing good mapping of the points of interest throughout the cine series. The sum of absolute difference algorithm used by the original prototype gave varied levels of performance meaning results could not be relied on for diagnosis. The more obvious possible causes of these varied results were the original point placement, and the effects of rotation and discretisation.

The first factor, the original point selection, can be compensated for by user training and a simple test algorithm that compares the moused point to those in the surrounding area to test the uniqueness of the point. A point selected on the edge of a feature will not be clearly distinguishable from other points along the same edge. When points are determined to be not unique or of marginal trackable quality the user can be informed by the program to pick another point on a more definite feature of the vertebra or in an area of higher contrast.

The effect of rotation on the algorithm was determined by tracking a point to the same frame rotated by computer program. Using a computer program to simulate rotation meant an exact measure of the rotation could be compared to the resultant performance of the algorithm. The experimental work concluded that because inter frame rotation was less than 3 degrees rotation did not have a significant effect on the performance of the sum of absolute difference statistic when tracking to the next frame in the sequence but when tracking to a frame which is significantly later in the sequence rotation must be included in the algorithm.

Discretisation was found to have a detrimental effect on the sum of absolute difference statistic performance. This can be compensated for by adding a sub-pixel location calculation into the algorithm which decreased the gross error by 17 pixels for a vertebra in one case.

In the original prototype a median filter (rank 5) was used to smooth the noise in the image to be searched. This was found to only slightly improve the performance of the statistic unless a sub-pixel location is sought in which case smoothing the picture had a detrimental effect due to the loss of image clarity and detail.

The algorithms defined in the literature were found to be unsuitable for the application in question as they tracked clearly discernable lines or searched for a shape matching a predefined three-dimensional model.

One algorithm that may prove to be a suitable alternative is the rate of change algorithm that compares changes in intensity across a window so is based on locating a change of intensity pattern rather than a pixel to pixel comparison.

There are additional features that could be included in the tracking procedure to make the algorithm more effective (the two dimensional logarithmic search) and provide checks and safeguards against points incrementally deviating from the correct location as tracking progresses (referencing a moused frame, using the vertebra's rigid body property). The benefit of incorporating the safe guard features would have to be weighed against the cost of extra computational time.

In conclusion, using the image subtraction technique to track from one frame was improved to within an acceptable error margin by tracking inter frame, that is from one frame to the next in the video sequence, and including a sub-pixel location calculation.

Chapter 12
References

1 Bajpai A C, Mustoe L R and Walker D: "Engineering Mathematics", Wiley, pp 143-149, 220-223.

2 Besl P J and Jain R C: "Three dimensional object recognition", ACM Compute. Surveys, vol17, (March 1985) pp 75-145.

3 Binfold : "Survey of model-based image analysis systems", Int. J. Robotics, vol 1, (1982), pp 18-64.

4 Bose N K: "Digital Filters - theory and applications", North-Holland, pp 16-22.

5 Brint A T and Brady M: "Stereo Matching Of Curves", Image And Vision Computing (Feb 1990) pp 50-56.

6 Browne R F and Monteith W: "Tracking Vertebrae In Cinegraphic Fluoroscopic X-rays", Proceedings Of The 5th N.Z. Image Processing Workshop (Aug 1990) pp 34-41.

7 Castleman K R: "Digital Image Processing", Prentice-Hall Inc, pp 157-158.

8 Cipolla R and Yamamoto M: "Steroscopic tracking of bodies in motion", Image and Vision Computing (Feb 1990) pp 85-90.

9 Deriche R and Faugeras O: "Tracking line segments", Image And Vision Computing (Nov 1989) pp 261-270.

10 Fletcher M J and Mitchell R J: "Predicting multiple feature locations for a class of dynamic image sequences", Image and Vision Computing (Aug 1990), pp 193-198.

11 Jähne B: "Digital image processing - concepts, algorithms and scientific applications", Springer-Verlag, pp 107-108, 134.

12 Jain A K: "Fundamentals Of Digital Image Processing", Prentice-Hall 1989, pp 400-407.

13 Kemp R: "Pascal for students", Edward Arnold.

14 Liu H and Srinath M: "Partial shape classification using contour matching in distance transformation", IEEE transactions on Pattern analysis and machine intelligence (Nov 1990), pp 1072-1079.

15 Page W H, Monteith W: "A System For The Analysis Of Bone Motion In Cinematic X-ray Images", Proceedings Of NELCON 91 (Aug 1991) pp 75-80.

16 Page W H, Monteith W: "Bone Movement Analysis From Computer Processing Of X-ray Cinematic Video Images", Proceedings Of The 4th International Conference On Image Processing And Its Applications (Apr 1992).

17 Pratt W K: "Digital Image Processing", second edition, Wiley-Interscience, pp 93-121.

18 Reynolds R J: "Cinematography By The Indiirect Method", Radiology, Vol. 31, pp. 177-182, 1938.

19 Rosin P L and West G A W: "Segmentation of edges into lines and arcs", Image And Vision Computing (May 1989) pp 109-114.

20 Schalkoff R J: "Digital Image Processing and Computer Vision - an introduction to theory and implementation", John Wiley and Sons Inc, pp 54-57, 213-218.

21 Schutte H, Frydrychowicz S and Schroder J: "Scene Matching With Translation Invariant Transforms", 5th International Conference Of Pattern Recognition Proceedings (1980) pp 195-198.

22 Sethi I K and Jain R: "Finding trajectories of feature points in a monoculer image sequence", IEEE transactions on Pattern analysis and machine intelligence (Jan 1987), pp56-73.

23 Stephens R S: "Real-time 3D object tracking", Image and Vision Computing (Feb 1990), pp 91-96.

Appendix A
Full derivation of spinal measurements

Derivation of rotational displacement

The point A1 is exactly halfway along a straight line between P1 and P4. A1 is given by the average of point P1 and P4.

$$(X, Y)_{A1} = \left(\frac{X_{P1}+X_{P4}}{2}, \frac{Y_{P1}+Y_{P4}}{2}\right)$$

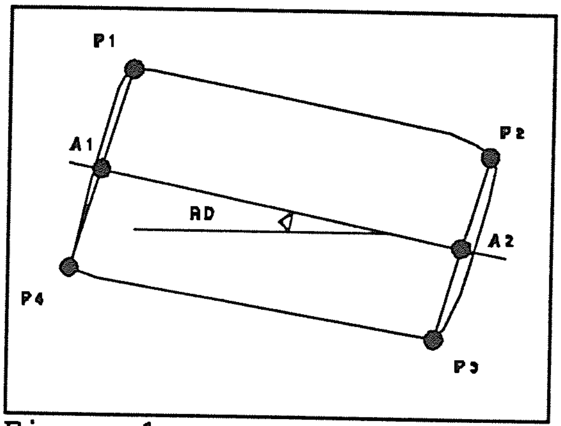


Figure 1

similarly A2 is the average of P2 and P3.

$$(X, Y)_{A1} = \left(\frac{X_{P2}+X_{P3}}{2}, \frac{Y_{P2}+Y_{P3}}{2}\right)$$

The rotational displacement, RD, is a measure of the vertebra angle to the horizontal plane. This angle may be found using trigonometry.

$$\theta_{RD} = \arctan\left(\frac{Y_{A1}-Y_{A2}}{X_{A1}-X_{A2}}\right)$$

Derivation of axial and planar displacement

The derivation of axial and planar displacement both use the following :

- a line that bisects the space between the two vertebrae,
- the centroid of each vertebra,
- the intercept of the line perpendicular to the bisecting line through the centroid of the vertebra and the bisecting line, for each vertebra.

If the points marked on the corners of the vertebrae are named as shown then the coordinate of AV1 is the average of the coordinates of PA1, PA4, PB1 and PB4. Similarly the coordinate of AV2 is the average of the coordinates of PA2, PA3, PB2 and PB3.

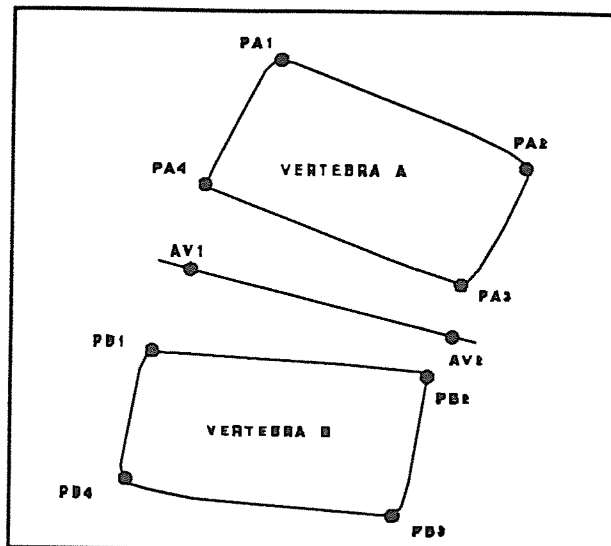


Figure 2

$$(X,Y)_{AV1} = \left(\frac{X_{A1}+X_{A4}+X_{B1}+X_{B4}}{4}, \frac{Y_{A1}+Y_{A4}+X_{B1}+X_{B4}}{4}\right)$$

$$(X,Y)_{AV2} = \left(\frac{X_{A2}+X_{A3}+X_{B2}+X_{B3}}{4}, \frac{Y_{A2}+Y_{A3}+X_{B2}+X_{B3}}{4}\right)$$

The equation of the bisecting line is :

$$Y = M_{BIS}*X+C_{BIS}$$

$$\text{where } M_{BIS} = \frac{\Delta Y}{\Delta X} = \frac{Y_{AV2}-Y_{AV1}}{X_{AV2}-X_{AV1}}$$

$$\text{and } C_{BIS} = Y_{AV1} - M_{BIS}*X_{AV1}$$

The coordinate of the centroid of a vertebra are given by the average of the coordinates of the four points where these are treated as having equal weighting on a plane.

Thus the coordinate of the centroid of vertebra A, $(X,Y)_{CENTA}$, are:

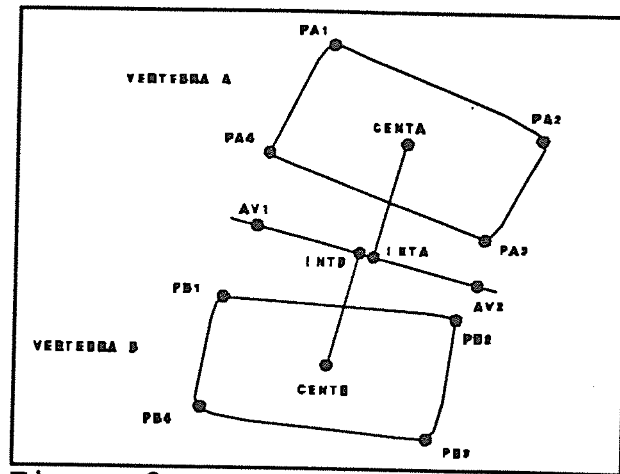


Figure 3

$$(X,Y)_{CENTA} = \left(\frac{X_{PA1}+X_{PA2}+X_{PA3}+X_{PA4}}{4},\ \frac{Y_{PA1}+Y_{PA2}+Y_{PA3}+Y_{PA4}}{4}\right)$$

and the coordinate of the centroid of vertebra B, $(X,Y)_{CENTB}$, are:

$$(X,Y)_{CENTB} = \left(\frac{X_{PB1}+X_{PB2}+X_{PB3}+X_{PB4}}{4},\ \frac{Y_{PB1}+Y_{PB2}+Y_{PB3}+Y_{PB4}}{4}\right)$$

To find the coordinates, $(X,Y)_{INTA}$ and $(X,Y)_{INTB}$, of the perpendicular intercept of a line from the vertebra centroid and the bisecting line the equation for the perpendicular line must be calculated. The gradient of a line perpendicular to the bisecting line is $-1/M_{BIS}$. The Y intercept of the perpendicular line can be calculated as the gradient and the coordinate of a point on the line is known :

$$C_{PERPA} = Y_{CENTA} + \frac{1}{M_{BIS}} * X_{CENTA}$$

$$C_{PERPB} = Y_{CENTB} + \frac{1}{M_{BIS}} * X_{CENTB}$$

The coordinates of the intercepts can be calculated by solving the following simultaneous equations:

$$Y = M_{BIS}*X + C_{BIS}$$

$$Y = \frac{-X}{M_{BIS}} + \frac{X_{CENTA}}{M_{BIS}} + Y_{CENTA}$$

to find $(X,Y)_{INTA}$, and :

$$Y = M_{BIS}*X + C_{BIS}$$

$$Y = \frac{-X}{M_{BIS}} + \frac{X_{CENTB}}{M_{BIS}} + Y_{CENTB}$$

to find $(X,Y)_{INTB}$.

Solving these equations gives :

$$X_{INTA} = \frac{X_{CENTA} + M_{BIS}*(Y_{CENTA}-C_{BIS})}{M_{BIS}^2+1}$$

$$Y_{INTA} = M_{BIS}*X_{INTA} + C_{BIS}$$

and

$$X_{INTB} = \frac{X_{CENTB} + M_{BIS}*(Y_{CENTB}-C_{BIS})}{M_{BIS}^2+1}$$

$$Y_{INTB} = M_{BIS}*X_{INTB} + C_{BIS}$$

The axial displacement, AD, is the sum of the distance from $(X,Y)_{CENTA}$ to $(X,Y)_{INTA}$ and the distance from $(X,Y)_{CENTB}$ to $(X,Y)_{INTB}$:

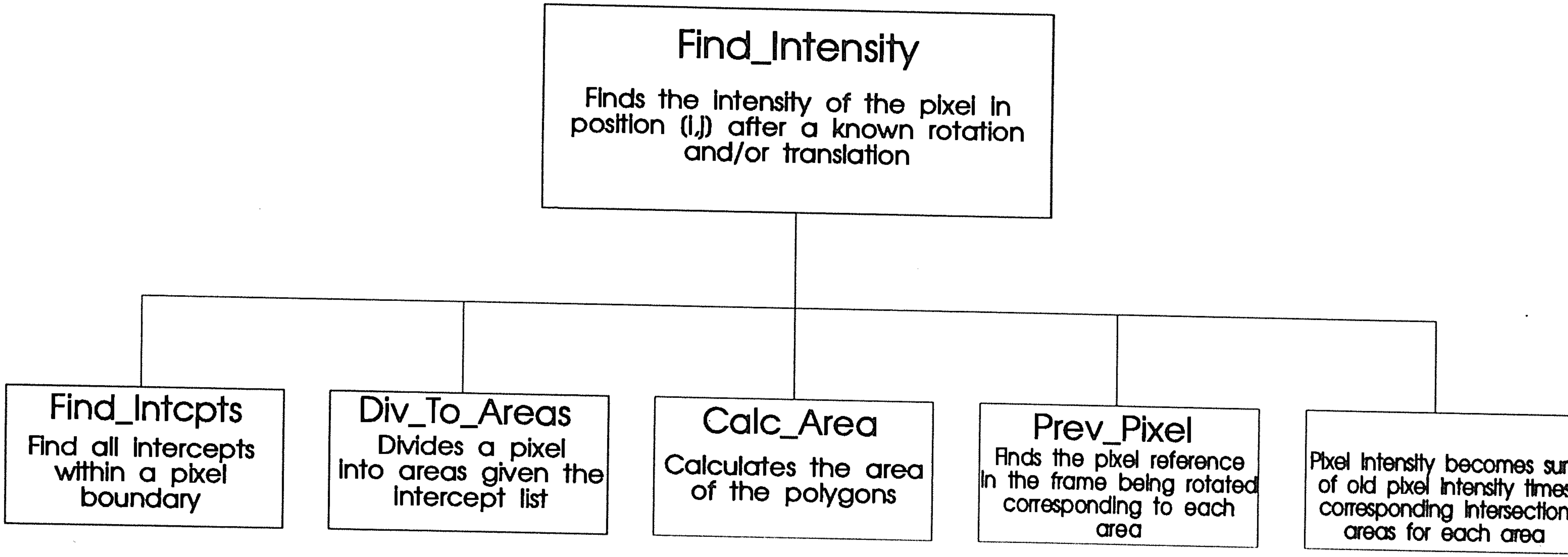
$$AD = \sqrt{(Y_{CENTA}-Y_{INTA})^2 + (X_{CENTA}-X_{INTA})^2} + \sqrt{(Y_{CENTB}-Y_{INTB})^2 + (X_{CENTB}-X_{INTB})^2}$$

The planar displacement, PD, is the distance between the two intercepts $(X,Y)_{INTA}$ and $(X,Y)_{INTB}$:

$$PD = \sqrt{(Y_{INTA}-Y_{INTB})^2 + (X_{INTA}-X_{INTB})^2}$$

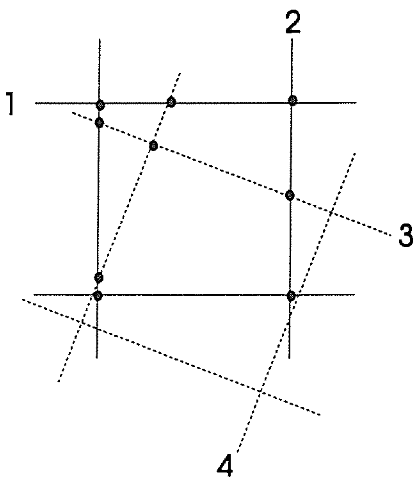
APPENDIX B
Computer program to simulate the rotation of an image

A Pascal program was written to rotate an image. The origin of rotation is taken as the centre of a pixel. The program first finds the equations of the lines within the area of interest that define the pixels before and after rotation. The intercepts of these lines are used to calculate the area of each of the previous pixels that contribute to the new pixel intensity. Using these areas and the previous pixel intensities the new intensity value of a pixel can be calculated.


Find_Intensity
Finds the intensity of the pixel in position (i,j) after a known rotation and/or translation
Find_Intcpts
Find all intercepts within a pixel boundary
Div_To_Areas
Divides a pixel into areas given the intercept list
Calc_Area
Calculates the area of the polygons
Prev_Pixel
Finds the pixel reference in the frame being rotated corresponding to each area
Pixel intensity becomes sum of old pixel intensity times corresponding intersection areas for each area

Find_Intcpts

Find all intercepts within a pixel boundary



1. Horizontal Grid ... $y = d * (k1 - 0.5)$
2. Vertical Grid ... $x = d * (k2 - 0.5)$
3. Rotated Horizontal Grid ... $y = -x * \tan(th) + d * (k3 - 0.5) / \cos(th)$
4. Rotated Vertical Grid ... $y = x * \tan(90 - th) + d * (0.5 - k4) / \sin(th)$

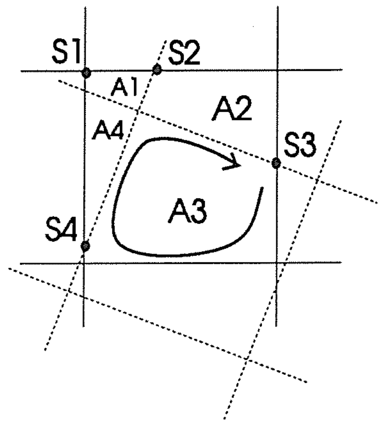
where rotation is specified by angle 'th' and 'd' is the width of the pixel

The central pixel is defined by line equations and intercepts of :

Top left hand corner : k1 = 0 k2 = 0
Top right hand corner : k1 = 1 k2 = 0
Bottom left hand corner : k1 = 0 k2 = 1
Bottom right hand corner : k1 = 1 k2 = 1

Div_To_Areas

Divides a pixel into areas given the intercept list

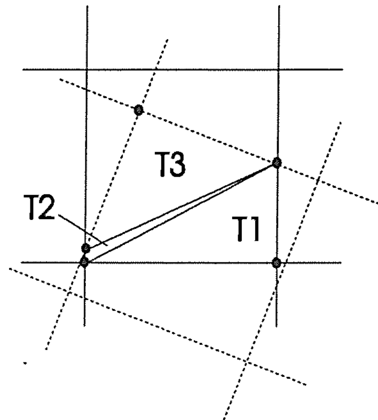


The algorithm segments the pixel into areas (eg. A1 to A4) by traversing the line in a clockwise direction (starting at S1 to S4) until encountering the nearest intercept along the line.

This procedure is repeated until the start point is reached.

Calc_Area

Calculates the area of the polygons



Calculates area by dividing the area into triangles, then summing the area of each triangle.

Prev_Pixel

Finds the pixel reference in the frame being rotated corresponding to each area

Pixel reference is relative to central pixel.

Pixel intensity becomes sum of old pixel intensity times corresponding intersection areas for each area

```
{home}
{       unit rottran

   This programm was written to work out the new values of pixels after
   a clockwise rotation of a known ammount.

   The rotated square must be an odd by odd number of pixels
   (i.e. there must be a central pixel)

EQUATION FOR LINES FOR ODD BY ODD SQUARE OF PIXELS.
Rotation is specified by angle `th' taken clockwise,
`d' is the width of the pixels,
coordinate (0,0) is in the centre of the central pixel [m,n],

1. HORIZONTAL GRID...            y = d * (k1 - 0.5)

2. VERTICAL GRID...              x = d * (k2 - 0.5)

3. ROTATED HORIZONTAL GRID...
        y = -x * tan(th) + d * (k3 - 0.5)/cos(th)

4. ROTATED VERTICAL GRID...
        y = x * tan(90-th) + d * (0.5 - k4)/sin(th)

e.g. The central pixel is defined by line eqns and intercepts of :
        (1, k1 = 0), (1, k1 = 1),
        (2, k2 = 0), (2, k2 = 1),
        (3, k3 = 0), (3, k3 = 1),
        (4, k4 = 0), (4, k4 = 1),

nomenclature :
   -  `i' is the vertical axis, `j' the horizontal.
   -  when specifying a pixel in the grid specify in relation to the central
      pixel [m,n], e.g. [m+3, n+1] is the pixel 3 below the central pixel
      and one to the right.
   -  when specifying a position in a matrix correct order is [i,j].
   -  [i,j] = [1,1] is the top left hand corner element of a matrix.
}
```

```
{******************************}
unit rottran;
{*******************INTERFACE****************}
interface

uses
    crt, vbles;

var
  xa,ya : integer;  {belong in int_in_scope - here for debug}


PROCEDURE int_in_scope;



{*****************IMPLEMENTATION*******************}
implementation

type
    index = 1..10;
    int_type = (i1_2, i1_3, i1_4, i2_3, i2_4, i3_4);
    type_pr = array[int_type] of string[4];

    ln_type = 1..4;

    ptr_itype = ^list_itype;
    list_itype = record
                 itype : int_type;
                 next: ptr_itype;
                end;

    intercept = record
                coords : point;
                itype  : int_type;
               end;

    ptr_int = ^int_list;
    int_list = record
               next : ptr_int;
               data : intercept;
              end;

    ptr_k = ^k_list;
    K_list = record
             next  : ptr_k;
             value : integer;
            end;

    ptr_coords = ^coord_list;
    coord_list = record
               coords : point;
               next : ptr_coords;
              end;

    coord = (x,y);
    coord_array = array[1..10, coord] of real;   {these coordinates specify
                                                   a polygon}
    areas_array = array[1..8] of coord_array;
    side_array = array[1..8] of integer;

const
     black = 0;
     white = 255;
     d = 1;          {width and height of individual pixels}

     print_type: type_pr= ('i1_2','i1_3','i1_4','i2_3','i2_4','i3_4');
     empty1: coord_array=((0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(
0,0),(0,0));
```

```
      empty2:areas_array =(
                ((0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)),
                ((0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)),
                ((0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)),
                ((0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)),
                ((0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)),
                ((0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)),
                ((0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)),
                ((0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0))
);

      empty3: side_array = (0,0,0,0,0,0,0,0);

var
   cor_1, cor_2, cor_3, cor_4 : point;

   test : ptr_int;

   test_co_ar : coord_array;
   test_rl_num : real;


FUNCTION TAN (angle : real) : real;

begin
     tan := sin(angle) / cos(angle);
end;


PROCEDURE SEARCH(list: ptr_k; kvalue:integer; var found:boolean);
{Procedure to search a list of k values to establish whether a value is
 already in the list}

var p: ptr_k;
begin
     p := list;
     found := false;
     while not found and (p<> nil) do
           if p^.value = kvalue then found := true
           else  p := p^.next
end;


PROCEDURE INS_INTCPT(info:intercept; var head : ptr_int);
{This procedure inserts an intercept into the front of the list of
 intercepts.}

 var p: ptr_int;

 begin

      new(p);
      p^.next := head;
      head    := p;
      p^.data := info;
 end;


FUNCTION MEMB(int: intercept; intcpt_list: ptr_int) : boolean;
{This function looks through a list of intercepts to check if the intercept
 is a memeber of the list.}

var
   found : boolean;
   p : ptr_int;

begin
   p:= intcpt_list;
   found := false;
```

```
   while not found and (p <> nil) do
   begin;
      if (p^.data.coords.x = int.coords.x) and
         (p^.data.coords.y = int.coords.y)
      { and (p^.data.itype = int.itype)   }
      then found := true
      else p := p^.next;
   end;

   memb := found
end;


FUNCTION REMOVE(int: intercept; intcpt_list: ptr_int) : ptr_int;
{This function removes an intercept from a list of intercepts.}

var
   p,
   p_last : ptr_int;

begin
   p := intcpt_list;
   p_last := nil;

   while (p <> nil) do
   begin
      if (p^.data.coords.x = int.coords.x) and
         (p^.data.coords.y = int.coords.y) and
         (p^.data.itype = int.itype)
      then begin
         if (p_last = nil) then begin
            intcpt_list := p^.next;
            p := p^.next;
            end
         else begin
            p_last^.next := p^.next;
            p := p_last^.next;
           end;
         end
      else begin
         p_last := p;
         p := p^.next;
         end;
   end;

remove := intcpt_list;
end;


PROCEDURE PRINT_LIST(list:ptr_int);
{Procedure to print out a list of intercepts.}

var p : ptr_int;
    stop,t : integer;

begin
     p := list;
     stop := 10;
     while (p <> nil) do
     begin
          stop := stop + 1;
          write('coords: (',p^.data.coords.x:6:3,',',p^.data.coords.y:6:3);
          writeln(') ,   type : ', print_type[p^.data.itype]);
          p := p^.next;
          if trunc(stop/22)=(stop/22) then readln;
     end;
     for t := (stop mod 22) to 22 do writeln;
end;
```

```
PROCEDURE INS_KLIST(kvalue: integer; var list : ptr_k );
{This procedure inserts a k value into the a list of ordered k values after
 checking it is not already in the list.}

var
   finished : boolean;
   p, p_last, p_pres: ptr_k;

begin
     p_last := nil;
     p_pres := list;
     finished := false;

     repeat
       begin
         if (p_pres^.value = kvalue) then finished := true else
            if (p_pres^.value > kvalue) or (p_pres = nil) then begin
               new(p);
               p^.next := p_pres;
               p^.value:= kvalue;
               if (p_last = nil) then list := p else p_last^.next:= p;
               finished := true;
               end
            else begin
               p_last := p_pres;
               p_pres := p_last^.next;
               end;
      end;
     until finished;

end;


PROCEDURE INS_CDLIST(coord:point; var list:ptr_coords);
{Checks if the coordinate is in the list and if it isn't then it inserts
 it in the front of the linked list}

var p : ptr_coords;
    found : boolean;

begin
     found := false;
     p := list;

     while (p <> nil) and not found do
        begin
           if p^.coords.x = coord.x then
              begin
                 if p^.coords.y = coord.y then found := true;
              end
           else p:= p^.next;
        end;
      if not found then
         begin
            new(p);
            p^.coords := coord;
            p^.next := list;
            list := p;
         end;
     end;


PROCEDURE INS_ITYPE(i_t : int_type; var list: ptr_itype);
{This procedure inserts an itype into a linked list of itypes}

var
   p : ptr_itype;

begin
```

```
    new(p);
    p^.itype := i_t;
    p^.next := list;
    list := p;
 end;


 FUNCTION LENGTH (pts_coord:coord_array; pt1, pt2 :index) : real;
 {Finds the length of a line between two points, pt1 and pt2 }

 var
    x1, y1,                     {coordinates of first point }
    x2, y2 : real;              {coordinates of second point}

 begin
      x1 :=  pts_coord[pt1,x];
      y1 :=  pts_coord[pt1,y];
      x2 :=  pts_coord[pt2,x];
      y2 :=  pts_coord[pt2,y];

      if y1 = y2 then length := abs(x1 - x2)
      else if x1 = x2 then length := abs(y1 - y2)
           else length := sqrt( sqr(x1 - x2) + sqr(y1 - y2) );
 end;


 PROCEDURE SWAP_VALS(var val1, val2 : real);
 {Function to swap two values}

 var temp : real;            {temporary store}

 begin
      temp := val1;
      val1 := val2;
      val2 := temp;
 end;


 FUNCTION INT_BTWN(num1,num2:real) : integer;
 {This function returns the integer between the two real numbers passed
  to the function}

 begin
      int_btwn := round((num1 + num2)/2);
 end
 ;

 FUNCTION ANGLE (pts_coord:coord_array; pt1, pt2, pt3 : index) : real;
 {Finds the angle between three points , where pt1 is the origin of the angle}

 var
    x1, y1,                   {coordinates of first point  }
    x2, y2,                                  { second point }
    x3, y3 : real;                           { third point  }
    adj : real;               {the length of the side adjacent to the angle}

 begin
      x1 :=  pts_coord[pt1,x];
      y1 :=  pts_coord[pt1,y];
      x2 :=  pts_coord[pt2,x];
      y2 :=  pts_coord[pt2,y];
      x3 :=  pts_coord[pt3,x];
      y3 :=  pts_coord[pt3,y];

      if (x1 = x2) then
         begin
              adj := abs(y3 - y1);
              if adj = 0 then angle:= rad90
                 else if y2 = y1 then angle := 0
```

```
                    else angle := arctan( abs(x3 - x2) / adj)
        end
        else if (x1 = x3) then
                begin
                   adj := abs(y2 - y1);
                   if adj = 0 then angle:= rad90
                      else if y3 = y1 then angle := 0
                          else angle := arctan( abs(x3 - x2) / adj)
                end
          else if ((x2>x1) and (x1>x3)) or ((x2<x1) and (x1<x3)) then
                angle := pi - (arctan( abs((y1 - y3) / (x1 - x3)) ) +
                       arctan( abs((y1 - y2) / (x1 - x2)) ))
                else if ((y2>y1) and (y3>y1)) or ((y2<y1) and (y3<y1)) then
                    angle := abs(arctan( abs((y1 - y3) / (x1 - x3)) ) -
                           arctan( abs((y1 - y2) / (x1 - x2)) ))
                    else angle := arctan( abs((y1 - y3) / (x1 - x3)) ) +
                              arctan( abs((y1 - y2) / (x1 - x2)) );

end;


FUNCTION CALC_AREA (poly_coord:coord_array; num_sides:index) : real;
{Function to find the area of a polygon.

The polygon is defined by the coordinates of each corner,
The information passed to this function consists of :
     -  an array of coordinates,
     -  the number of sides the polygon has.

The maxinum number of sides is set by the index value defined in the
main programmes types.

The polygon is divided into triangles and the area of each triangle
is calculated - these are summed to give the total area.
}

var
   sum_area     : real;
   height, ang  : real;
   base, lngth2 : real;
   count        : integer;

begin
     sum_area  := 0;
     for count := 3 to num_sides do
     begin
          base := length(poly_coord, 1, count);
          lngth2 := length(poly_coord, 1, (count-1) );
          if base < lngth2 then swap_vals(base, lngth2);
          ang := angle(poly_coord, 1, (count - 1), count);
          height := sin(ang) * lngth2;
          sum_area := sum_area + base * height;
     end;
     calc_area := 0.5 * sum_area

end;



FUNCTION REAL_K3(x,y : real): real;
{This function returns a k3 value of type real which would give as a point
 on it the x,y coordinates supplied to the function if it was used in the
 equation for the horizontal rotated line.}

begin
     if th <> 0 then
        real_k3 := 0.5 + (y - trans.y + (x-trans.x)*tan(th)) * cos(th)/d
     else real_k3 := (y - trans.y)/d + 0.5;
end;
```

```
FUNCTION REAL_K4(x,y : real): real;
{This function returns a k4 value of type real which would give as a point
 on it the x,y coordinates supplied to the function if it was used in the
 equation for the vertical rotated line.}

begin
     if th <> 0 then
        real_k4 := 0.5 - (sin(th)/d)*(y -trans.y - (x-trans.x)*tan(rad90-th))
     else real_k4 := (x - trans.x)/d + 0.5;
end;


PROCEDURE INTERCEPT_1_2(k1, k2 : integer; var int: point);
{This procedure finds the intercept of the horizontal and vertical line
 given k1 and k2, the variables in the equations. }

begin
     int.x := d * (k2 - 0.5);
     int.y := d * (k1 - 0.5);
end;


PROCEDURE INTCPT_1_3_GK(k1, k3 : integer; var int: point);
{This procedure finds the intercept of the horizontal and rotated horizontal
 grid given k1 and k3, the variables in the equations. }

begin
     int.y := d * (k1 - 0.5);
     int.x := (d * (k3-0.5)/cos(th) - int.y + trans.y)/tan(th) + trans.x;
end;


PROCEDURE INTCPT_1_4_GK(k1, k4 : integer; var int: point);
{This procedure finds the intercept of the horizontal and rotated vertical
 grid given k1 and k4, the variables in the equations. }

begin
     int.y := d * (k1 - 0.5);
     if th <> 0 then
 int.x := (int.y - trans.y - d * (0.5-k4)/sin(th))/tan(rad90 - th) + trans.x
     else int.x := d * (k4 - 0.5) + trans.x;
end;


PROCEDURE INTCPT_2_3_GK(k2, k3 : integer; var int: point);
{This procedure finds the intercept of the vertical and rotated horizontal
 grid given k2 and k3, the variables in the equations. }

begin
     int.x := d * (k2 - 0.5);
     if th <> 0 then
     int.y := - (int.x-trans.x) * tan(th) + d * (k3-0.5)/cos(th) + trans.y
     else int.y := d * (k3 - 0.5) + trans.y;
end;


PROCEDURE INTCPT_2_4_GK(k2, k4 : integer; var int: point);
{This procedure finds the intercept of the vertical and rotated vertical
 grid given k2 and k4, the variables in the equations. }

begin
     int.x := d* (k2 - 0.5);
     int.y :=  (int.x-trans.x)* tan(rad90 - th) + d * (0.5-k4)/sin(th) +
trans.y;
end;
```

```
PROCEDURE INTCPT_3_4_GK(k3, k4 : integer; var int: point);
{This procedure finds the intercept of the rotated horizontal and rotated
 vertical lines given k2 and k4, the variables in the equations. }

begin
     if th <> 0 then
     begin
       int.x := trans.x + d * ((k3-0.5)/cos(th) - (0.5-k4)/sin(th))
                  / (tan(rad90 - th)+tan(th));
       int.y := -(int.x-trans.x) * tan(th) + d * (k3-0.5)/cos(th) + trans.y;
     end
     else
     begin
       int.x := d * (k4 - 0.5) + trans.x;
       int.y := d * (k3 - 0.5) + trans.y;
     end;

end;


PROCEDURE INTERCEPT_1_3(corn1, corn3 :point; k1_hd :ptr_k;
                        var int_hd:ptr_int;var k3_hd:ptr_k);
{This procedure finds the intercepts of the horizontal grid lines and
 the rotated horizontal grid lines given the coordinates of corners
 one and three of the pixel of interest}

var
   k3_1, k3_2 : real;  {the values which k3 is between}
   k1,
   k3: integer;        {the value of k3 in the rotated horizontal grid eqn.}
   k1_ptr   : ptr_k;      {pointer to list of k1's}
   intcpt   : intercept;

begin
     k1_ptr  := k1_hd;
     intcpt.itype := i1_3;

     {finding intercepts with bottom horizontal line}
     k1 := k1_ptr^.value;
     k3_1 := real_k3(corn1.x, corn3.y);
     k3_2 := real_k3(corn3.x, corn3.y);
     k3 := int_btwn(k3_1, k3_2);
     intcpt_1_3_gk(k1,k3, intcpt.coords);
     intcpt.coords.x := round(1000000 * intcpt.coords.x) /1000000;
     intcpt.coords.y := round(1000000 * intcpt.coords.y) /1000000;
     if (intcpt.coords.x >= corn1.x) and (intcpt.coords.x <= corn3.x) and
        (intcpt.coords.y >= corn3.y) and (intcpt.coords.y <= corn1.y) then
        begin
          ins_intcpt(intcpt, int_hd);
          ins_klist(k3, k3_hd);
        end;

     {finding intercept with top horizontal grid line}
     k1_ptr := k1_ptr^.next;
     k1 := k1_ptr^.value;
     k3_1 := real_k3(corn1.x, corn1.y);
     k3_2 := real_k3(corn3.x, corn1.y);
     k3 := int_btwn(k3_1, k3_2);
     intcpt_1_3_gk(k1,k3, intcpt.coords);
     intcpt.coords.x := round(1000000 * intcpt.coords.x) /1000000;
     intcpt.coords.y := round(1000000 * intcpt.coords.y) /1000000;
     if (intcpt.coords.x >= corn1.x) and (intcpt.coords.x <= corn3.x) and
        (intcpt.coords.y >= corn3.y) and (intcpt.coords.y <= corn1.y) then
        begin
          ins_intcpt(intcpt, int_hd);
          ins_klist(k3, k3_hd);
        end;

end;
```

```
PROCEDURE INTERCEPT_2_3(corn1, corn3 :point; k2_hd:ptr_k;
                         var int_hd:ptr_int;var k3_hd:ptr_k);
{This procedure finds the intercepts of the vertical grid lines and
 the rotated horizontal grid lines given the coordinates of corners
 one and three of the pixel of interest}

var
   k3_1, k3_2 : real;  {the values which k3 is between}
   k2,
   k3: integer;        {the value of k3 in the rotated horizontal grid eqn.}
   k2_ptr  : ptr_k;       {pointer to list of k2's}
   intcpt  : intercept;


begin
     k2_ptr := k2_hd;
     intcpt.itype := i2_3;

     {finding intercepts with left vertical line}
     k2 := k2_ptr^.value;
     k3_1 := real_k3(corn1.x, corn1.y);
     k3_2 := real_k3(corn1.x, corn3.y);
     k3 := int_btwn(k3_1, k3_2);
     intcpt_2_3_gk(k2,k3, intcpt.coords);
     intcpt.coords.x := round(1000000 * intcpt.coords.x) /1000000;
     intcpt.coords.y := round(1000000 * intcpt.coords.y) /1000000;
     if (intcpt.coords.x >= corn1.x) and (intcpt.coords.x <= corn3.x) and
        (intcpt.coords.y >= corn3.y) and (intcpt.coords.y <= corn1.y) then
        begin
          ins_intcpt(intcpt, int_hd);
          ins_klist(k3, k3_hd);
        end;


     {finding intercept with right vertical grid line}
     k2_ptr := k2_ptr^.next;
     k2 := k2_ptr^.value;
     k3_1 := real_k3(corn3.x, corn1.y);
     k3_2 := real_k3(corn3.x, corn3.y);
     k3 := int_btwn(k3_1, k3_2);
     intcpt_2_3_gk(k2,k3, intcpt.coords);
     intcpt.coords.x := round(1000000 * intcpt.coords.x) /1000000;
     intcpt.coords.y := round(1000000 * intcpt.coords.y) /1000000;
     if (intcpt.coords.x >= corn1.x) and (intcpt.coords.x <= corn3.x) and
        (intcpt.coords.y >= corn3.y) and (intcpt.coords.y <= corn1.y) then
        begin
          ins_intcpt(intcpt, int_hd);
          ins_klist(k3, k3_hd);
        end;

end;


PROCEDURE INTERCEPT_1_4(corn1, corn3 :point; k1_hd:ptr_k;
                         var int_hd:ptr_int;var k4_hd:ptr_k);
{This procedure finds the intercepts of the horizontal grid lines and
 the rotated vertical grid lines given the coordinates of corners one
 and three of the pixel of interest}

var
   k4_1, k4_2 : real;  {the values which k4 is between}
   k1,
   k4: integer;        {the value of k4 in the rotated horizontal grid eqn.}
   k1_ptr  : ptr_k;       {pointer to list of k1's}
   intcpt  : intercept;

begin
     k1_ptr:= k1_hd;
```

```
      intcpt.itype := i1_4;

      {finding intercepts with bottom horizontal line}
      k1 := k1_ptr^.value;
      k4_1 := real_k4(corn1.x, corn3.y);
      k4_2 := real_k4(corn3.x, corn3.y);
      k4 := int_btwn(k4_1, k4_2);
      intcpt_1_4_gk(k1,k4, intcpt.coords);
      intcpt.coords.x := round(1000000 * intcpt.coords.x) /1000000;
      intcpt.coords.y := round(1000000 * intcpt.coords.y) /1000000;
      if (intcpt.coords.x >= corn1.x) and (intcpt.coords.x <= corn3.x) and
         (intcpt.coords.y >= corn3.y) and (intcpt.coords.y <= corn1.y) then
         begin
           ins_intcpt(intcpt, int_hd);
           ins_klist(k4, k4_hd);
         end;


      {finding intercept with top horizontal grid line}
      k1_ptr := k1_ptr^.next;
      k1 := k1_ptr^.value;
      k4_1 := real_k4(corn1.x, corn1.y);
      k4_2 := real_k4(corn3.x, corn1.y);
      k4 := int_btwn(k4_1, k4_2);
      intcpt_1_4_gk(k1,k4, intcpt.coords);
      intcpt.coords.x := round(1000000 * intcpt.coords.x) /1000000;
      intcpt.coords.y := round(1000000 * intcpt.coords.y) /1000000;
      if (intcpt.coords.x >= corn1.x) and (intcpt.coords.x <= corn3.x) and
         (intcpt.coords.y >= corn3.y) and (intcpt.coords.y <= corn1.y) then
         begin
           ins_intcpt(intcpt, int_hd);
           ins_klist(k4, k4_hd);
         end;

end;


PROCEDURE INTERCEPT_2_4(corn1, corn3 :point; k2_hd:ptr_k;
                        var int_hd:ptr_int;var k4_hd:ptr_k);

{This procedure finds the intercepts of the vertical grid lines and
 the rotated vertical grid lines given the coordinates of corners one
 and three of the pixel of interest}

var
   k4_1, k4_2 : real;  {the values which k4 is between}
   k2,
   k4: integer;        {the value of k4 in the rotated horizontal grid eqn.}
   k2_ptr   : ptr_k;     {pointer to list of k2's}
   intcpt   : intercept;


begin
     k2_ptr := k2_hd;
     intcpt.itype := i2_4;

     {finding intercepts with left vertical line}
     k2 := k2_ptr^.value;
     k4_1 := real_k4(corn1.x, corn1.y);
     k4_2 := real_k4(corn1.x, corn3.y);
     k4 := int_btwn(k4_1, k4_2);
     intcpt_2_4_gk(k2,k4, intcpt.coords);
     intcpt.coords.x := round(1000000 * intcpt.coords.x) /1000000;
     intcpt.coords.y := round(1000000 * intcpt.coords.y) /1000000;
     if (intcpt.coords.x >= corn1.x) and (intcpt.coords.x <= corn3.x) and
        (intcpt.coords.y >= corn3.y) and (intcpt.coords.y <= corn1.y) then
        begin
          ins_intcpt(intcpt,int_hd);
          ins_klist(k4, k4_hd);
```

```
          end;

       {finding intercept with right vertical grid line}
       k2_ptr := k2_ptr^.next;
       k2 := k2_ptr^.value;
       k4_1 := real_k4(corn3.x, corn1.y);
       k4_2 := real_k4(corn3.x, corn3.y);
       k4 := int_btwn(k4_1, k4_2);
       intcpt_2_4_gk(k2,k4, intcpt.coords);
       intcpt.coords.x := round(1000000 * intcpt.coords.x) /1000000;
       intcpt.coords.y := round(1000000 * intcpt.coords.y) /1000000;
       if (intcpt.coords.x >= corn1.x) and (intcpt.coords.x <= corn3.x) and
          (intcpt.coords.y >= corn3.y) and (intcpt.coords.y <= corn1.y) then
          begin
            ins_intcpt(intcpt, int_hd);
            ins_klist(k4, k4_hd);
          end;

end;


PROCEDURE INTERCEPT_3_4(corn1, corn3 :point; k3_hd, k4_hd :ptr_k;
                        var int_hd:ptr_int);

{This procedure finds the intercepts of the rotated vertical grid lines
 and the rotated horizontal grid lines given the coordinates of corners one
 and three of the pixel of interest and the list of k3's and k4's.}

var
   a, b : ptr_k;
   k3, k4: integer;
   k3_ptr, k4_ptr : ptr_k;       {pointer to list of k3's and k4's}
   intcpt  : intercept;


begin
     a := k3_hd;
     intcpt.itype := i3_4;

     while (a <> nil) do
        begin
           b := k4_hd;
           while (b <> nil) do
           begin
              intcpt_3_4_gk(a^.value, b^.value, intcpt.coords);
              intcpt.coords.x := round(1000000 * intcpt.coords.x) /1000000;
              intcpt.coords.y := round(1000000 * intcpt.coords.y) /1000000;
              if (intcpt.coords.x >= corn1.x) and
                 (intcpt.coords.x <= corn3.x) and
                 (intcpt.coords.y >= corn3.y) and
                 (intcpt.coords.y <= corn1.y)
                then ins_intcpt(intcpt, int_hd);
              b:= b^.next;
              end;
              a := a^.next;
           end;
end;


PROCEDURE CORNERS (i, j :integer ; var c1, c3 : point;
                    var int_hd : ptr_int; var k1_hd, k2_hd: ptr_k);
{Procedure finds the coordinates of the pixels four corners given
 its position with respect to the central pixel [m,n] is [m+i,n+j].
where  c1 = top left hand corner,
       c2 = top right hand corner,
       c3 = bottom right hand corner,
       c4 = bottom right hand corner.

}
```

```
var k1, k2 : integer;
   intcpt  : intercept;

begin;
     intcpt.itype := i1_2;

     k1 := 1 - i;
     k2 := j;
     ins_klist(k1, k1_hd);
     ins_klist(k2, k2_hd);
     intercept_1_2(k1, k2, c1);
     intcpt.coords := c1;
     ins_intcpt(intcpt, int_hd);

     k1 := -i;
     k2 := j + 1;
     ins_klist(k1, k1_hd);
     ins_klist(k2, k2_hd);
     intercept_1_2(k1, k2, c3);
     intcpt.coords := c3;
     ins_intcpt(intcpt, int_hd);

    intercept_1_2((1-i), (j+1), intcpt.coords);        {c2}
    ins_intcpt(intcpt, int_hd);

    intercept_1_2((-i), j, intcpt.coords);             {c4}
    ins_intcpt(intcpt, int_hd);

end;


PROCEDURE FIND_INTCPTS(i, j: integer; var list: ptr_int; var c1: point);
{Procedure to find all intercepts inside a square specified by [i,j],
 which is its position relative to the central pixel [m,n].}

var
   c3 : point;       {coordinates of corner3, i.e.BR corner (c1 is TL corner}
   k1_list, k2_list, k3_list, k4_list : ptr_k;       {lists of the k values}

begin
     list := nil;
     k1_list := nil;
     k2_list := nil;
     k3_list := nil;
     k4_list := nil;

     corners(i,j,c1,c3,list, k1_list, k2_list);
     if (th <> 0) or (trans.x <> 0) or (trans.y <> 0) then
     begin
        if th <> 0 then
          begin
             intercept_1_3(c1, c3, k1_list, list, k3_list);
             intercept_2_4(c1, c3, k2_list, list, k4_list);
          end;
        if (th <> 0) or (trans.y <> 0) then
                intercept_2_3(c1, c3, k2_list, list, k3_list);
        if (th <> 0) or (trans.x <> 0) then
                intercept_1_4(c1, c3, k1_list, list, k4_list);
        intercept_3_4(c1, c3, k3_list, k4_list, list);
     end;
{*********
print_list(list);
}
end;


FUNCTION ALL_ITYPE_INTS(list_it: ptr_itype; list:ptr_int):ptr_int;
{This function sorts through a list of intercepts and given the list of
 itypes that are possible, gives a list of all the intercepts that have
```

```
 a possible itype.}

var
   t: ptr_itype;
   int : ptr_int;
   ans : ptr_int;

begin
   t := list_it;
   ans := nil;

   while (t <> nil) do
   begin
      int := list;
      while (int <> nil) do
      begin

         if t^.itype = int^.data.itype then ins_intcpt(int^.data, ans);
         int:= int^.next;
      end;
      t := t^.next;
   end;
   all_itype_ints := ans;
end;


FUNCTION POS_INTCPTS(trav: ln_type; list: ptr_int): ptr_int;
{This function returns the posible intercepts given the current line type
 about to be traversed}

var
   it_list : ptr_itype;

begin
     it_list := nil;

     case trav of

     1:  begin
            ins_itype(i1_2, it_list);
            ins_itype(i1_3, it_list);
            ins_itype(i1_4, it_list);
         end;

     2:  begin
            ins_itype(i1_2, it_list);
            ins_itype(i2_3, it_list);
            ins_itype(i2_4, it_list);
         end;

     3:  begin
            ins_itype(i1_3, it_list);
            ins_itype(i2_3, it_list);
            ins_itype(i3_4, it_list);
         end;

     4:  begin
            ins_itype(i1_4, it_list);
            ins_itype(i2_4, it_list);
            ins_itype(i3_4, it_list);
         end;

     end;
     pos_intcpts := all_itype_ints(it_list, list);
end;


FUNCTION NEAREST_INT(last_coord:point; list:ptr_int) : ptr_int;
{This function finds the nearest intercept to the last coordinate}
```

```
var
   p : ptr_int;
   nrst_int : ptr_int;        {nearest intercept at the moment}
   f,                        {factor calculated to distuinguish nearest coord}
   nrst_f : real;             {factor of current nearest intercept}

begin
     nrst_int := nil;
     p := list;
     nrst_f := 999;

     while (p <> nil) do
     begin
        f := abs(p^.data.coords.x - last_coord.x) +
             abs(p^.data.coords.y - last_coord.y);
        if (f < nrst_f) and (f>0) then
           begin
              if (nrst_int<>nil) then nrst_int:= nil;
              ins_intcpt(p^.data, nrst_int);
              nrst_f := f;
           end
        else if (f = nrst_f) then ins_intcpt(p^.data, nrst_int);
        p := p^.next;
     end;
     nearst_int := nrst_int;
end;


FUNCTION NEXT_INTCPT(x_op, y_op:integer; coord:point; list:ptr_int): ptr_int;
{This function finds the next intercept given the relation to the last
 intercept by x_op and y_op, where -1 means 'less than', 0 means 'same as',
and 1
 means 'greater than'.}

var
   p : ptr_int;
   found : ptr_int;

begin
   p := list;
   found := nil;
   case x_op of
   -1 :  case y_op of
           -1 : begin
                   while (p <> nil) do
                   begin
                      if (p^.data.coords.x < coord.x) and
                         (p^.data.coords.y < coord.y)
                         then ins_intcpt(p^.data, found);
                      p:= p^.next;
                   end;
                end;
            0 : begin
                   while (p <> nil) do
                   begin
                      if (p^.data.coords.x < coord.x) and
                         (p^.data.coords.y = coord.y)
                         then ins_intcpt(p^.data, found);
                      p:= p^.next;
                   end;
                end;
            1 : begin
                   while (p <> nil) do
                   begin
                      if (p^.data.coords.x < coord.x) and
                         (p^.data.coords.y > coord.y)
                         then ins_intcpt(p^.data, found);
                      p:= p^.next;
                   end;
```

```
               end;
          end;
     0 :  case y_op of
            -1 : begin
                   while (p <> nil) do
                   begin
                      if (p^.data.coords.x = coord.x) and
                         (p^.data.coords.y < coord.y)
                         then ins_intcpt(p^.data, found);
                      p := p^.next;
                   end;
                 end;
             0 : begin
                   while (p <> nil) do
                   begin
                      if (p^.data.coords.x = coord.x) and
                         (p^.data.coords.y = coord.y)
                         then ins_intcpt(p^.data, found);
                      p:= p^.next;
                   end;
                 end;
             1 : begin
                   while (p <> nil) do
                   begin
                      if (p^.data.coords.x = coord.x) and
                         (p^.data.coords.y > coord.y)
                         then ins_intcpt(p^.data, found);
                      p := p^.next;
                   end;
                 end;
          end;
     1 :  case y_op of
            -1 : begin
                   while (p <> nil) do
                   begin
                      if (p^.data.coords.x > coord.x) and
                         (p^.data.coords.y < coord.y)
                         then ins_intcpt(p^.data, found);
                      p := p^.next;
                   end;
                 end;
             0 : begin
                   while (p <> nil) do
                   begin
                      if (p^.data.coords.x > coord.x) and
                         (p^.data.coords.y = coord.y)
                         then ins_intcpt(p^.data, found);
                      p:= p^.next;
                   end;
                 end;
             1 : begin
                   while (p <> nil) do
                   begin
                      if (p^.data.coords.x > coord.x) and
                         (p^.data.coords.y > coord.y)
                         then ins_intcpt(p^.data, found);
                      p := p^.next;
                   end;
                 end;
          end;
  end;
  if (found^.next <> nil) then next_intcpt := nearst_int(coord, found)
  else next_intcpt := found;

end;


FUNCTION NEXT_TRAV(trav: ln_type; nxt_itype: int_type): ln_type;
{This function gives the next line to be traversed given the last line
```

```
 type trasversed and the current intercept type.}

begin
case trav of
   1: begin
        case nxt_itype of
           i1_2 :  next_trav := 2;
           i1_3 :  next_trav := 3;
           i1_4 :  next_trav := 4;
        end;
      end;
   2: begin
        case nxt_itype of
           i1_2 :  next_trav := 1;
           i2_3 :  next_trav := 3;
           i2_4 :  next_trav := 4;
        end;
      end;
   3: begin
        case nxt_itype of
           i1_3 :  next_trav := 1;
           i2_3 :  next_trav := 2;
           i3_4 :  next_trav := 4;
        end;
      end;
   4: begin
        case nxt_itype of
           i1_4 :  next_trav := 1;
           i2_4 :  next_trav := 2;
           i3_4 :  next_trav := 3;
        end;
      end;
   end;
end;


PROCEDURE TRAV_LN_12(trav:ln_type; last_coord :point;
                c1 :point; intcpt_list: ptr_int;
                var nxt_intcpt:ptr_int);
{Procedure to 'traverse' a line1 or a line2 and find the next intercept
 while tracking an area.}

var
   pos_int : ptr_int;

begin

   pos_int := pos_intcpts(trav, intcpt_list);

   case trav of
   1: begin
        if last_coord.y = c1.y
          then nxt_intcpt := next_intcpt(1, 0, last_coord, pos_int)
          else  nxt_intcpt := next_intcpt(-1, 0, last_coord, pos_int);
        if (nxt_intcpt^.next <> nil) then {there is an i3_4 intercept}
           begin    {the next line type is an i1_4}
             if nxt_intcpt^.data.itype = i1_4 then nxt_intcpt^.next := nil
             else nxt_intcpt := nxt_intcpt^.next;
           end;
      end;

   2: begin
       if last_coord.x = c1.x then
          nxt_intcpt := next_intcpt(0, 1, last_coord, pos_int)
          else nxt_intcpt := next_intcpt(0, -1, last_coord, pos_int);
       if (nxt_intcpt^.next <> nil) then {there is an i3_4 intercept}
          begin    {the next line type is an i1_3}
            if nxt_intcpt^.data.itype = i2_3 then nxt_intcpt^.next := nil
            else nxt_intcpt := nxt_intcpt^.next;
```

```
            end;
        end;

      end;
  end;


PROCEDURE    TRAV_LN_34(trav:ln_type;    last_intcpt:intercept;
lst_lst_coords:point;
                          incpt_list: ptr_int; var nxt_intcpt:ptr_int);
{Procedure to 'traverse' a line3 or a line4 and find the next intercept
 while tracking an area.}

var
   temp,
   pos_int,
   pos_int1 : ptr_int;

begin
   pos_int := pos_intcpts(trav, incpt_list);

   if (last_intcpt.itype <> i3_4) then
      begin
          pos_int1 := remove(last_intcpt, pos_int);
          nxt_intcpt := nearst_int(last_intcpt.coords,pos_int1);
      end

   else    {itype = i3_4}
    begin
     if (th <> 0) then
     begin
       if (lst_lst_coords.x < last_intcpt.coords.x) then
           if (lst_lst_coords.y > last_intcpt.coords.y) then
              nxt_intcpt:=next_intcpt(-1, -1, last_intcpt.coords, pos_int)
           else nxt_intcpt:=next_intcpt(1, -1, last_intcpt.coords, pos_int)
       else
           if (lst_lst_coords.y > last_intcpt.coords.y) then
              nxt_intcpt:=next_intcpt(-1, 1, last_intcpt.coords, pos_int)
           else nxt_intcpt:=next_intcpt(1, 1, last_intcpt.coords, pos_int);
     end      {calculation if th <> 0}

     else      {th is 0 , i.e. there was no rotation}
       begin
         if (lst_lst_coords.x < last_intcpt.coords.x) then
            nxt_intcpt:=next_intcpt(0, -1, last_intcpt.coords, pos_int)
         else
           if (lst_lst_coords.x > last_intcpt.coords.x) then
              nxt_intcpt:=next_intcpt(0, 1, last_intcpt.coords, pos_int)
           else
             if (lst_lst_coords.y < last_intcpt.coords.y) then
                 nxt_intcpt:=next_intcpt(1, 0, last_intcpt.coords, pos_int)
             else    { lst_lst_coords.y > last_intcpt.coords.y  }
                   nxt_intcpt:=next_intcpt(-1,     0,     last_intcpt.coords,
pos_int);
       end;
     end;

   if (nxt_intcpt^.next <> nil) then {there is an i3_4 intercept}
   {a decision must be made on which intercept is the correct one.
    The rules for this decision are :
      if the line traversed was a 3 then
        -  if choice is an i1_3 or an i3_4 pick the i1_3,
        -  if choice is an i2_3 or i3_4 then take the i3_4.
      if the line traversed was a 4 then
        -  if choice is an i1_4 or an i3_4 pick the i3_4,
        -  if choice is an i2_4 or i3_4 then take the i2_4.
    }
     begin
       if nxt_intcpt^.data.itype = i3_4 then
```

```
          begin
           temp := nxt_intcpt;
           nxt_intcpt := nxt_intcpt^.next;
          end
        else temp:= nil;
        case nxt_intcpt^.data.itype of
          i1_3 : nxt_intcpt^.next := nil;
          i2_3 : if temp <> nil then
                   begin
                     temp^.next := nil;
                     nxt_intcpt := temp;
                   end
                 else nxt_intcpt := nxt_intcpt^.next;
          i1_4 : if temp <> nil then
                   begin
                     temp^.next := nil;
                     nxt_intcpt := temp;
                   end
                 else nxt_intcpt := nxt_intcpt^.next;
          i2_4 : nxt_intcpt^.next := nil;
        end;
      end;

end;


PROCEDURE NEXT_START(new_start,used_start:ptr_int ;c1:point;
                     var nxt_st:ptr_int ; var nxt_trav: ln_type);
{This procedure finds the next new start and the trav to start the
 tracking of the new area.}

var
   ns_next : ptr_int;
   member : boolean;

begin
   nxt_st := new_start;      {in case it is nil to start with}
   ns_next := new_start;
   member := true;

   while member and (nxt_st <> nil) do
   begin
      nxt_st := ns_next;
      member := memb(nxt_st^.data, used_start);
      ns_next := nxt_st^.next;
   end;
   if (nxt_st <> nil) then
   begin
      if (nxt_st^.data.itype = i1_3) or (nxt_st^.data.itype = i1_4) then
         nxt_trav := 1
      else if (nxt_st^.data.itype = i2_3) or (nxt_st^.data.itype = i2_4) then
              nxt_trav := 2
           else    {itype = i1_2}
              if (nxt_st^.data.coords.x = c1.x) then
                 if (nxt_st^.data.coords.y = c1.y) then nxt_trav := 1
                 else nxt_trav := 2
              else if (nxt_st^.data.coords.y = c1.y) then nxt_trav := 2
                   else nxt_trav := 1;
   end;
end;


PROCEDURE DIV_TO_AREAS(incpt_lst : ptr_int; c1:point;
                       var all_areas : areas_array; var num_sides: side_array;
                       var num_areas : integer);
{This procedure divides a pixel into areas given the list of all
 intercepts occuring in that pixel.}

var
```

```
   start_trav,                   {the first line type to be traversed in new
area}
   trav,                        {the line type about to be traversed}
   nxt_trav : ln_type;          {the line type last traversed}
   ns_ptr, us_ptr,
   new_start,                   {list of intercepts areas can be started from}
   used_start : ptr_int;        {list of 'starts' already used}
   area : coord_array;
   lst_lst_coords : point;
   last_intcpt : intercept;    {last coordinates visited}
   nxt_start,                  {next start - for another area}
   nxt_intcpt : ptr_int;       {next intercept to be visited : retuned by the
traverse procedure}
   finished : boolean;
   count : integer;            {counts number of sides}

begin
     all_areas := empty2;
     area := empty1;
     new_start  := nil;
     used_start := nil;
     new(nxt_start);
     nxt_start^.data.coords.x := c1.x;
     nxt_start^.data.coords.y := c1.y;
     nxt_start^.data.itype := i1_2;
     ins_intcpt(nxt_start^.data, new_start); {insert c1 as the first starting
point}
     start_trav := 1;                     {traverse a '1' line to start}
     num_areas := 0;

     while (nxt_start <> nil) do
     begin
        nxt_intcpt := nxt_start;
        count := 0;
        num_areas := num_areas + 1;
        nxt_trav  := start_trav;
        finished  := false;
        area := empty1;

        while not finished do
        begin
           last_intcpt := nxt_intcpt^.data;
           count := count + 1;
           area[count,x]:= last_intcpt.coords.x;
           area[count,y]:= last_intcpt.coords.y;
           trav := nxt_trav;
           if (trav = 1) or (trav = 2) then
             begin
               trav_ln_12(trav,last_intcpt.coords,c1,incpt_lst,nxt_intcpt);
               ins_intcpt(last_intcpt, used_start);
             end
           else begin    {trav = 3 or 4}
              lst_lst_coords.x := area[(count-1),x];
              lst_lst_coords.y := area[(count-1),y];
              if (last_intcpt.itype <> i3_4)
                then ins_intcpt(last_intcpt, new_start);

trav_ln_34(trav,last_intcpt,lst_lst_coords,incpt_lst,nxt_intcpt);
           end;
           nxt_trav := next_trav(trav, nxt_intcpt^.data.itype);
           if (nxt_intcpt^.data.coords.x = area[1,x]) and
              (nxt_intcpt^.data.coords.y = area[1,y]) then finished:= true;
        end;

                        num_sides[num_areas] := count;
        all_areas[num_areas] := area;
        next_start(new_start, used_start, c1, nxt_start, start_trav);
     end;
end;
```

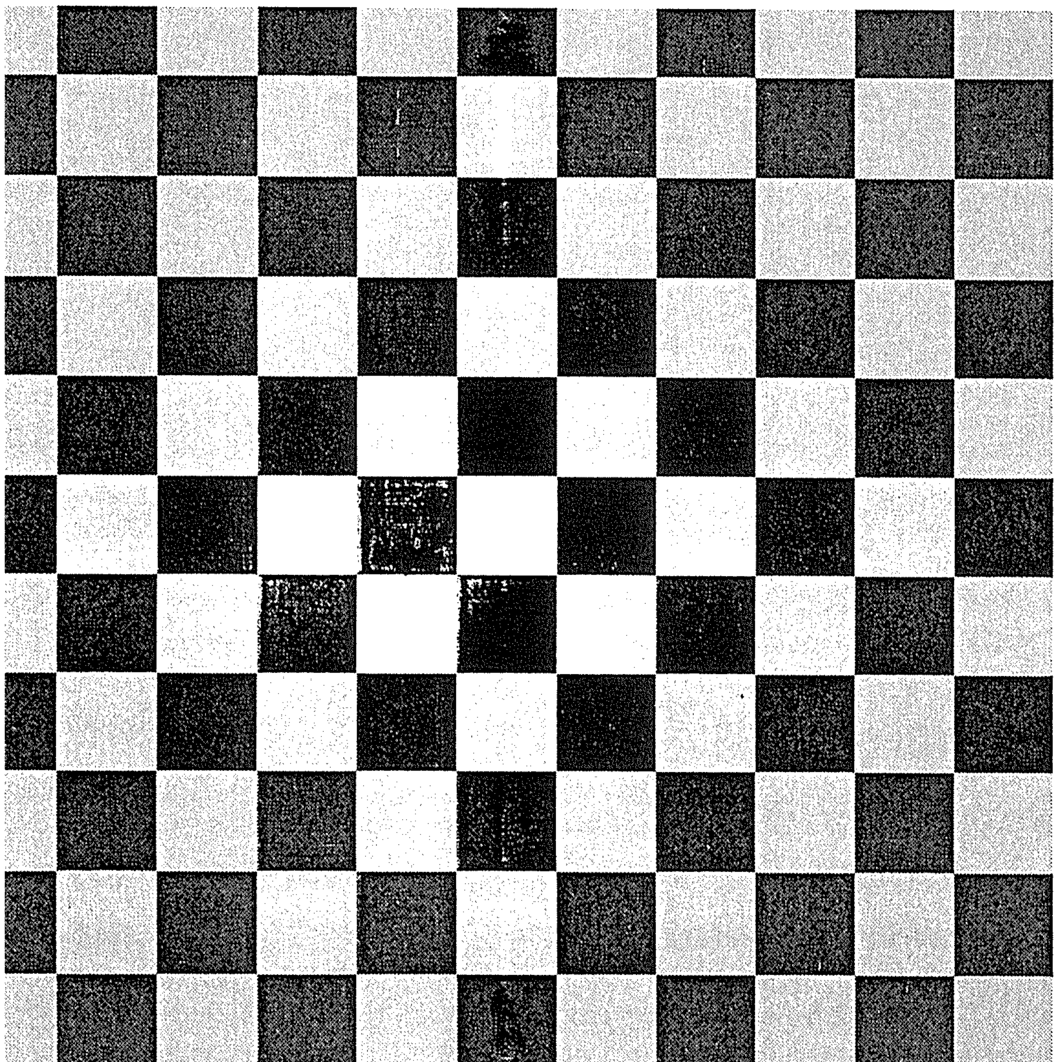
```
PROCEDURE PREV_PIXEL(area: coord_array; num_sides: integer;
                     var i_br, j_br:integer);
{This procedure finds the position of a pixel, in relation to the central
 pixel, given the coordinates of an area that marks the rotated pixel.}

var count : integer;
    x_av, y_av,            {average x and y}
    x_th, y_th,            {x across at theta, y across at theta}
    sum_x, sum_y : real;

begin
   sum_x := 0;
   sum_y := 0;

   for count := 1 to num_sides do
      begin
         sum_x := sum_x + area[count,x];
         sum_y := sum_y + area[count,y];
      end;

   x_av := sum_x / num_sides;
   y_av := sum_y / num_sides;

   if th = 0 then   {no rotation}
     begin
       x_th := x_av - trans.x;
       y_th := y_av - trans.y;
     end

   else   {there is rotation}
     begin
       {calculate x_th and y_th for rotation and no translation}
       x_th := x_av * cos(th) - y_av * sin(th);
       y_th := y_av * cos(th) + x_av * sin(th);

       {if there is translation then adjust x_th and y_th acccordingly]
       if (trans.x <> 0) and (trans.y = 0) then
         begin
           x_th := trans.x * cos(th) - x_th;
           y_th := y_th + trans.x * sin(th);
         end
         else if (trans.x = 0) and (trans.y <> 0) then
                begin
                  x_th := x_th + trans.y * sin(th);
                  y_th := trans.y * cos(th) - y_th;
                end
              else if (trans.x <> 0) and (trans.y <> 0) then
                     {both x and y translation and rotation}
                     {x_th is the same as for rotation only}
                      y_th := y_th - sqrt(sqr(trans.x) + sqr(trans.y))

     end;

   if y_th > 0 then i_br := -trunc((y_th + 0.5*d)/d)
               else i_br :=  trunc((abs(y_th) + 0.5*d)/d);
   if x_th < 0 then j_br := -trunc((abs(x_th) + 0.5*d)/d)
               else j_br :=  trunc((x_th + 0.5*d)/d);

end;


FUNCTION FIND_INTENSITY(i,j: integer) : integer;
{This function finds the intensity of a pixel, given its location [i,j],
 after a rotation of deg_th degrees for the whole range of pixels.}

var
   p : pointer;
```

```
    asb:string;
    c1 : point;
    test: ptr_int;
    div_areas : areas_array;
    num_sides : side_array;
    num_areas : integer;
    c_area : real;
    count1, count2 : integer;
    i_br, j_br : integer;        {the coordinates in the original image}
    sum_intensity : real;
    xi, yi : integer;        {the array index that gives intensity of [i,j]}

 begin
    mark(p);
    sum_intensity := 0;
    test:= nil;
    find_intcpts(i,j, test, c1);
    div_to_areas(test,c1, div_areas, num_sides, num_areas);
    for count1 := 1 to num_areas do
       begin
          c_area := calc_area(div_areas[count1] , num_sides[count1]);
          prev_pixel(div_areas[count1], num_sides[count1], i_br, j_br);
          xi := spec_pt.x + j_br;
          yi := spec_pt.y + i_br;
          sum_intensity := sum_intensity + (c_area*first_pic[xi,yi])
       end;
    find_intensity := round(sum_intensity);
    release(p);
 end;


FUNCTION CODE(deg,xdir, ydir: real) : string;
{works out a code for the file name}

VAR
   a, b, c : integer;
   sum : string;

begin
   if (deg  <= 0) then a:=0 else a:=1;
   if (xdir <= 0) then b:=0 else b:=1;
   if (ydir <= 0) then c:=0 else c:=1;
   str( ((4*a) + (2*b) + c), sum);
   code := sum;
end;



PROCEDURE INT_IN_SCOPE;
{Finds the new intensities after a specified rotation of the pixels needed
 to calculate the statistic for a specified point.
 The area is rotated around the pixel specified by 'spec_pt'.}

VAR
   x, y : integer;    {array index}
   i, j : integer;    {matrix coordinates}
   st1, st2 :string;

begin
   for xa := (spec_pt.x-win_rad)
            to  (spec_pt.x+win_rad) do
    begin
     j := xa - spec_pt.x;
     for ya:= (spec_pt.y-win_rad)
             to  (spec_pt.y+win_rad) do
       begin
        i := ya - spec_pt.y;
        calc_pic[xa,ya] := find_intensity(i,j);
       end;
```

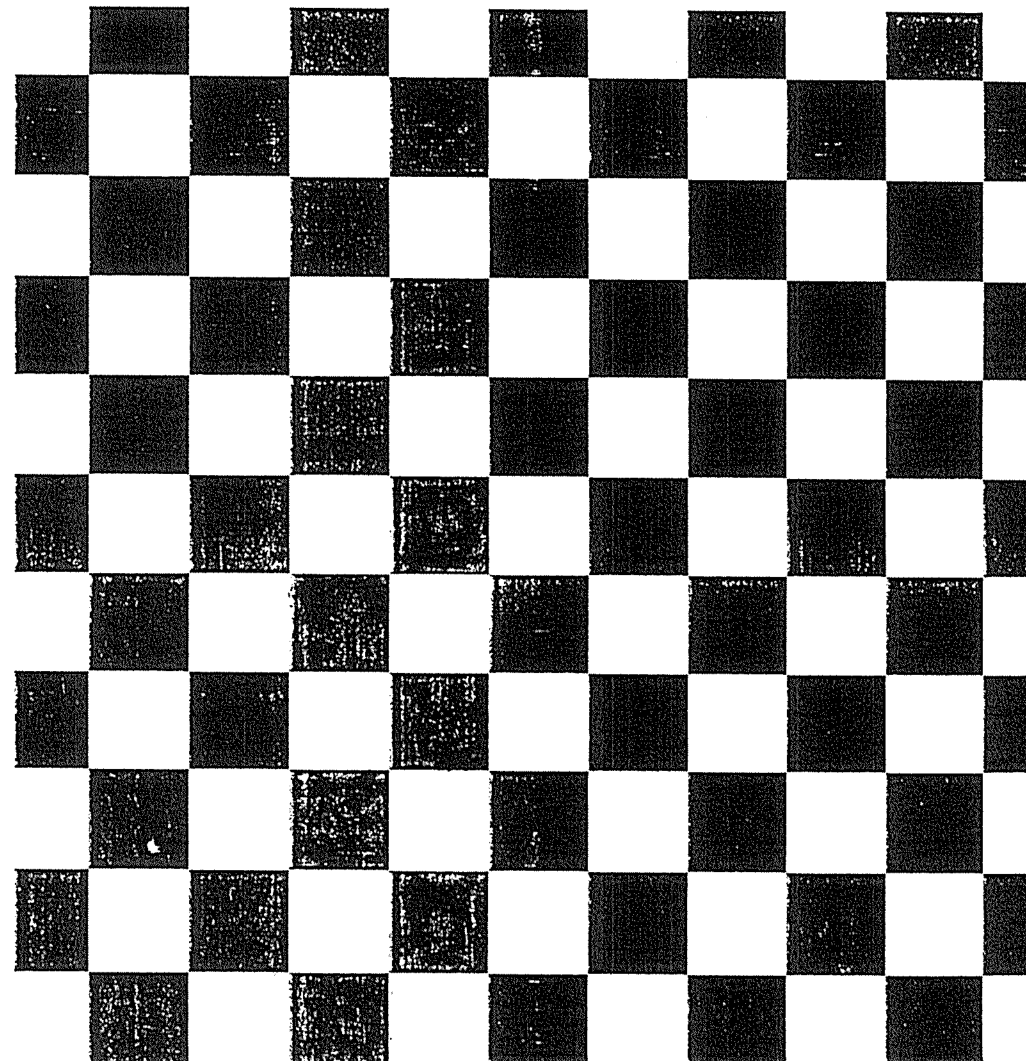
```
    end;

   {for checking}

{   st1 := code(deg_th, trans.x, trans.y);
   str((deg_th+trans.x+trans.y):6:2, st2);
   op_file_name := concat(st1, st2);}

{writing results of translation and rotation to calc_pic_file,
 i.e.     the new intensities.}
   append(calc_pic_file);
   write_line(calc_pic_file);
   write(calc_pic_file, 'rot : ',deg_th:6:2,'  trans x : ',trans.x:6:3,'   y
:',trans.y:6:3);
   write_line(calc_pic_file);
   write(calc_pic_file, 'rotated around : (',spec_pt.x,' , ',spec_pt.y,' ).');
   write_line(calc_pic_file);
     for y:= (spec_pt.y-win_rad) to (spec_pt.y+win_rad) do
       begin
         for x := (spec_pt.x-win_rad) to (spec_pt.x+win_rad) do
           write(calc_pic_file, calc_pic[x,y], '  ');
         write_line(calc_pic_file);
       end;
   close(calc_pic_file);

end;


{***********************MAIN PROGRAMME ******************}


end.                    {main programme}
```

Rotated checker images

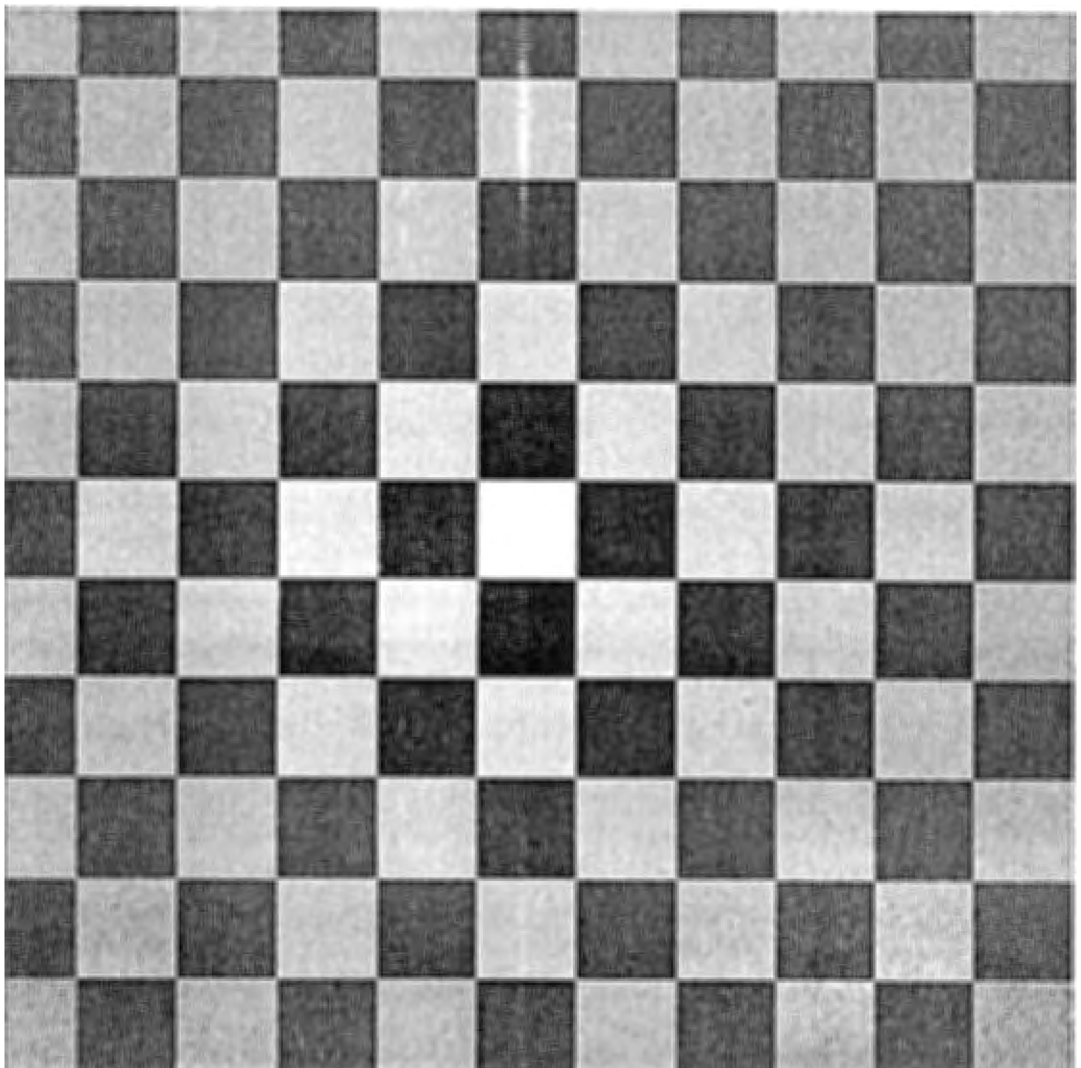
1 by 1 pixel Checker pattern
rotated 1 °



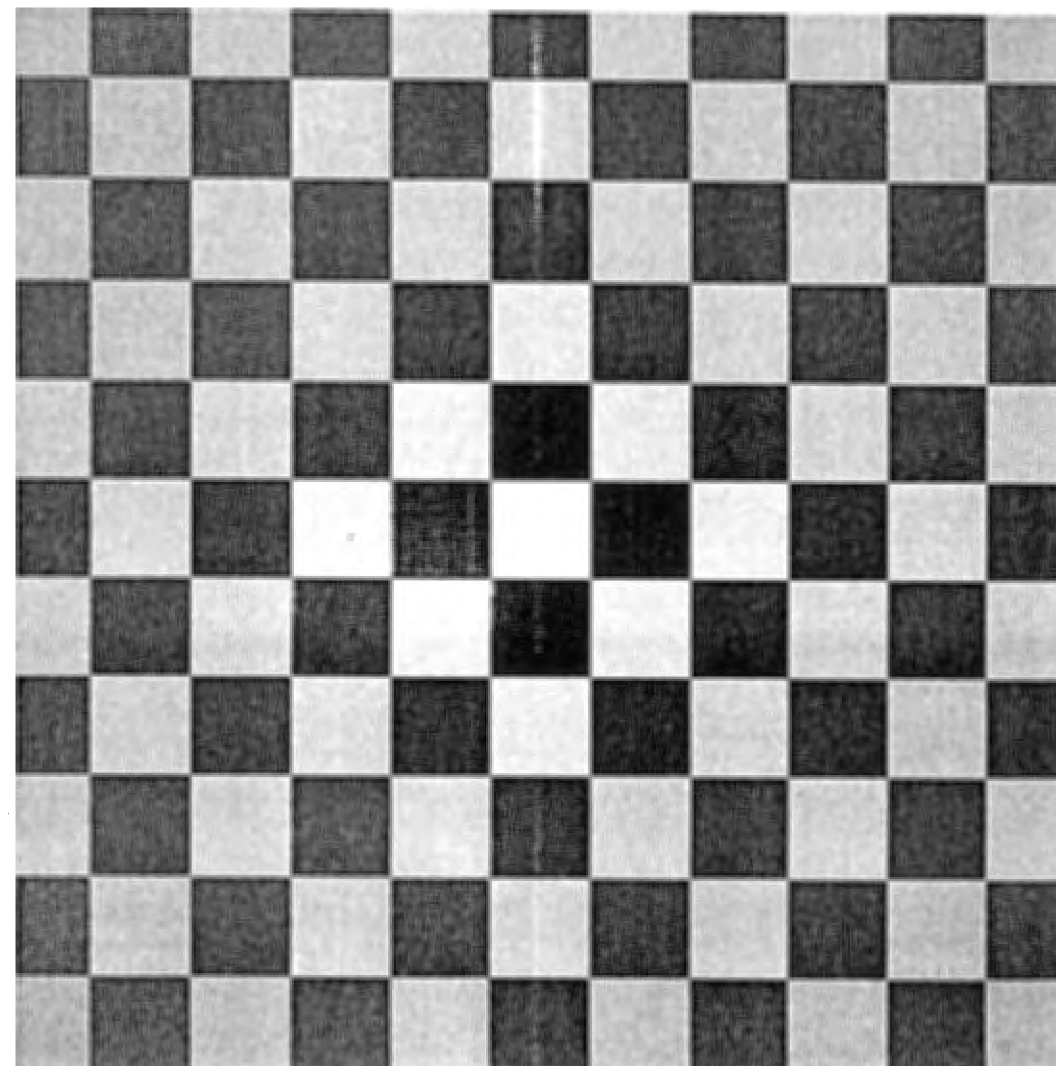
1 by 1 pixel Checker pattern
rotated 0 °



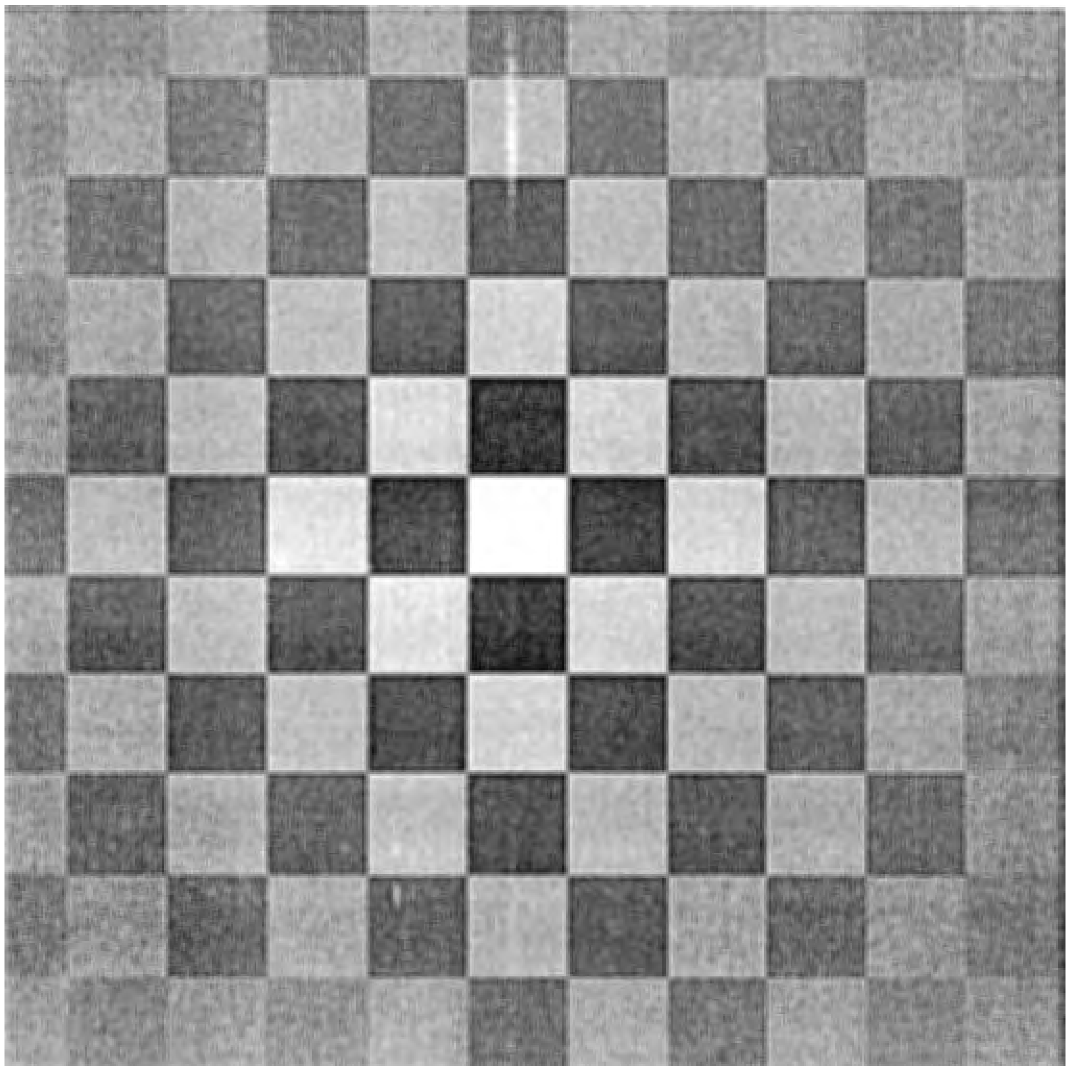
1 by 1 pixel Checker pattern
rotated 3 ^{o}



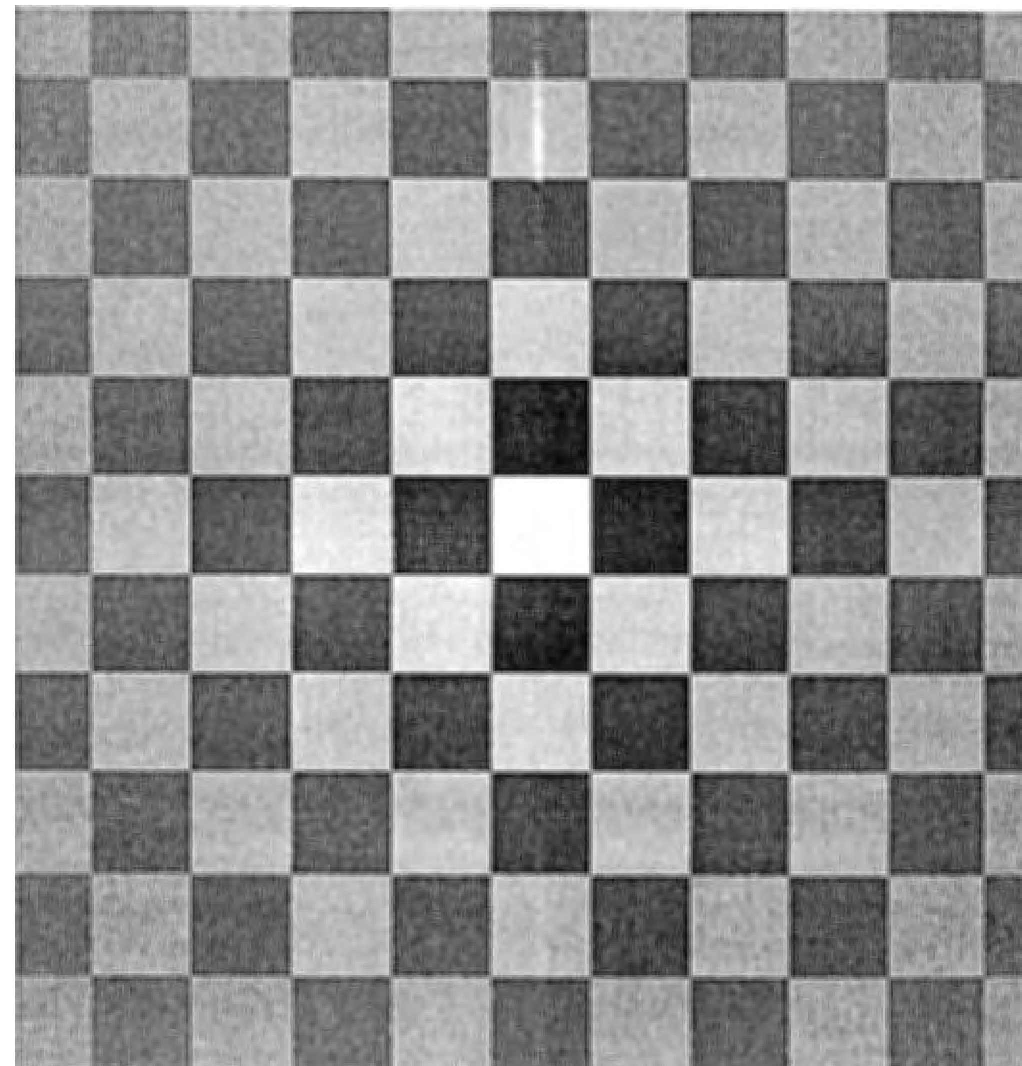
1 by 1 pixel Checker pattern
rotated 2 ^{o}



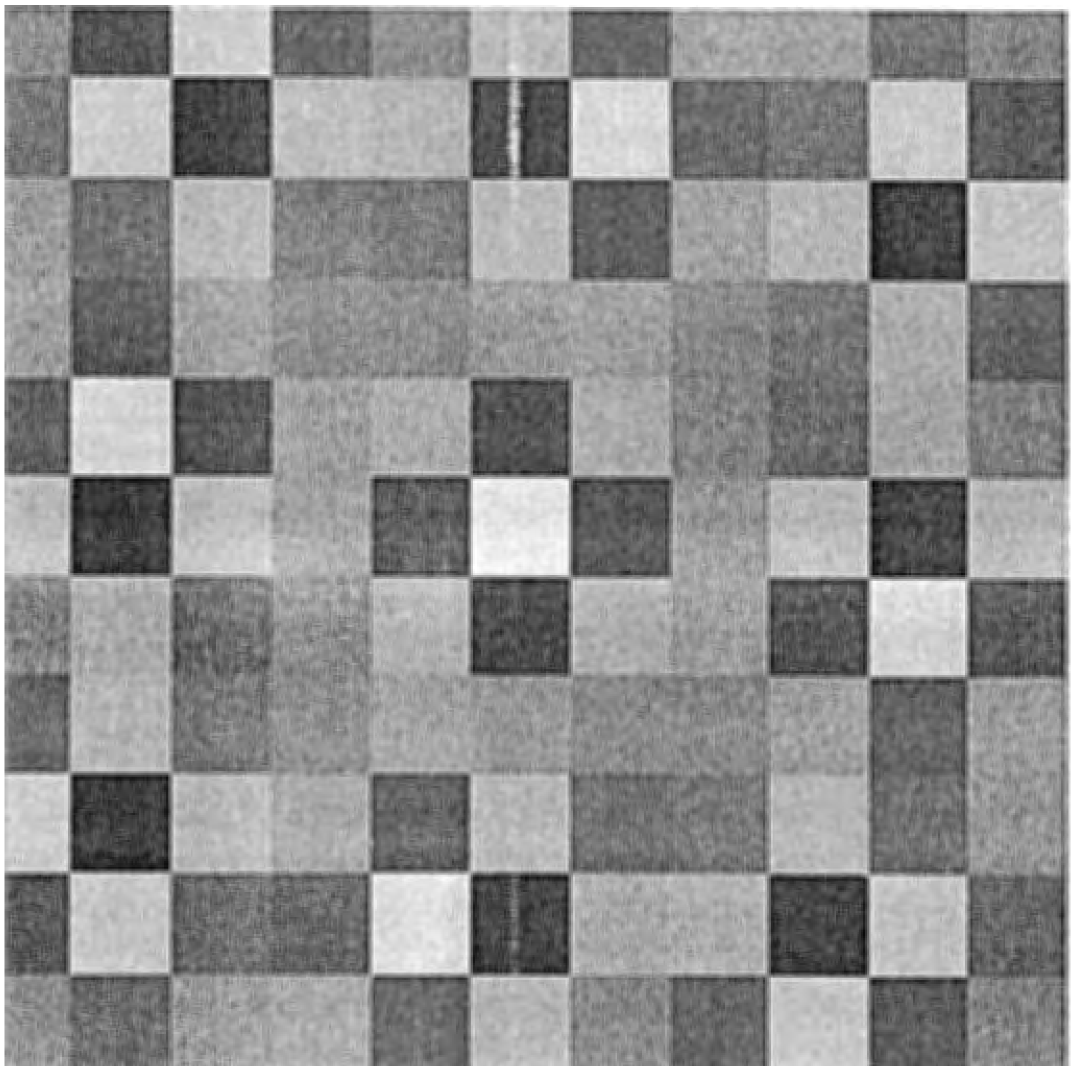
1 by 1 pixel Checker pattern
rotated 5 ^{o}



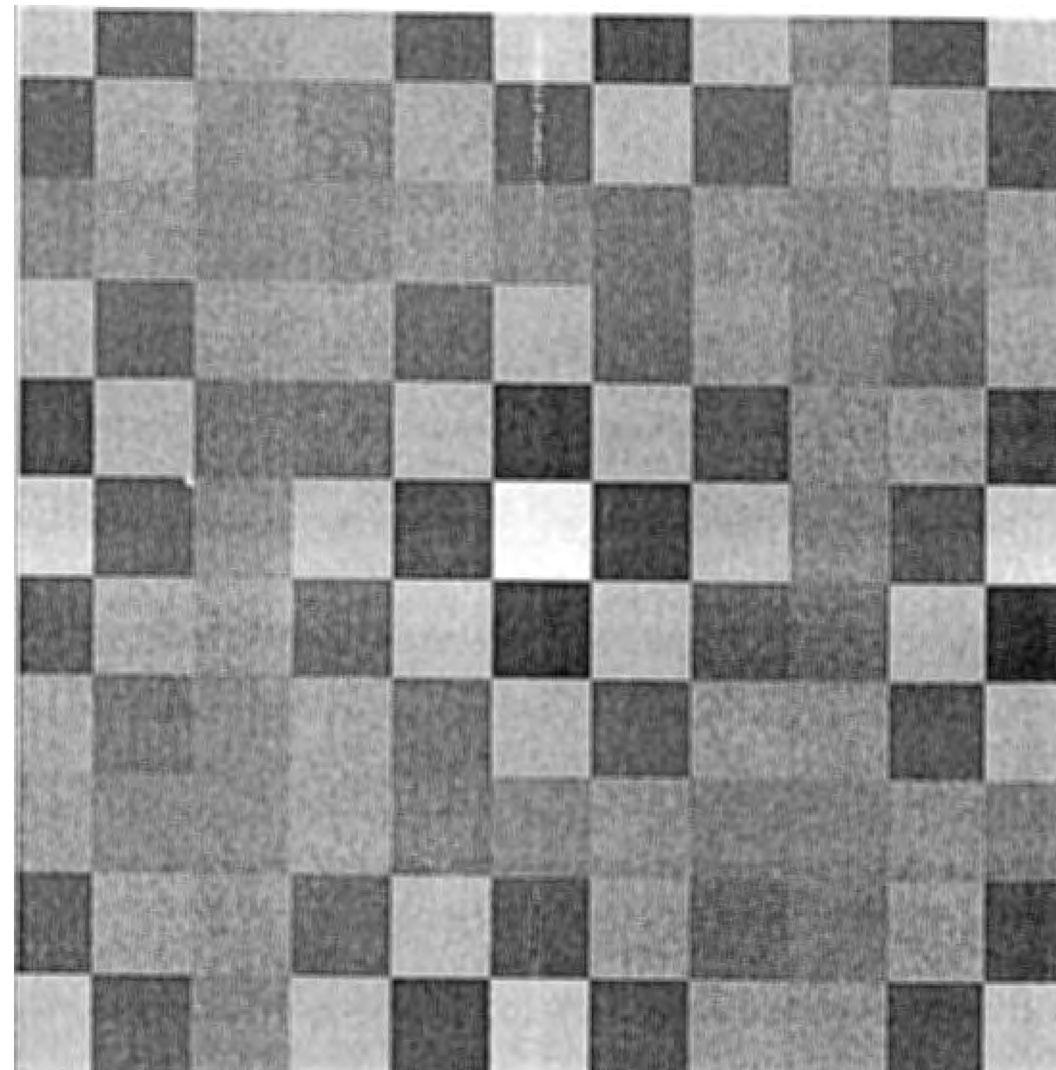
1 by 1 pixel Checker pattern
rotated 4 ^{o}



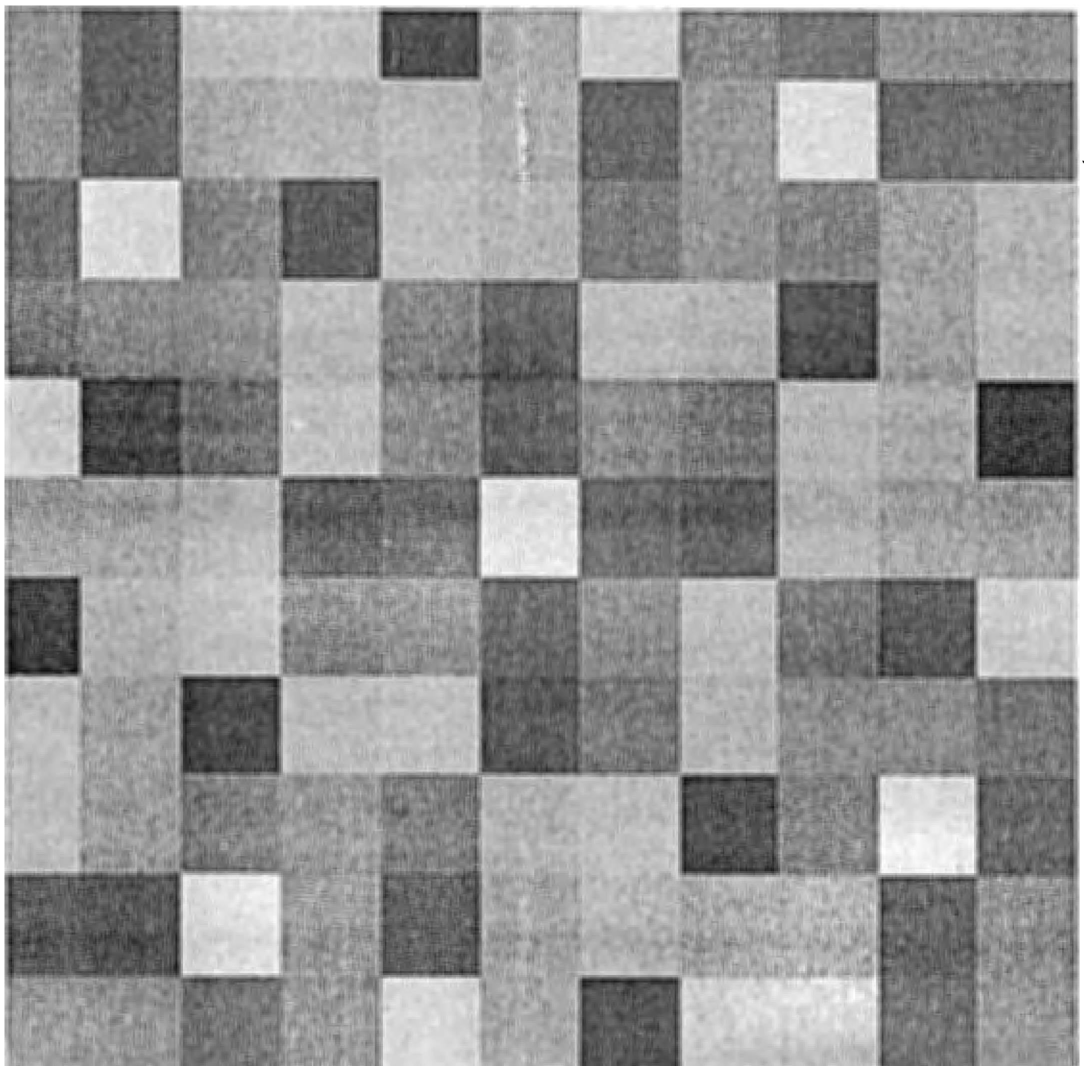
1 by 1 pixel Checker pattern
rotated 15 ^{o}



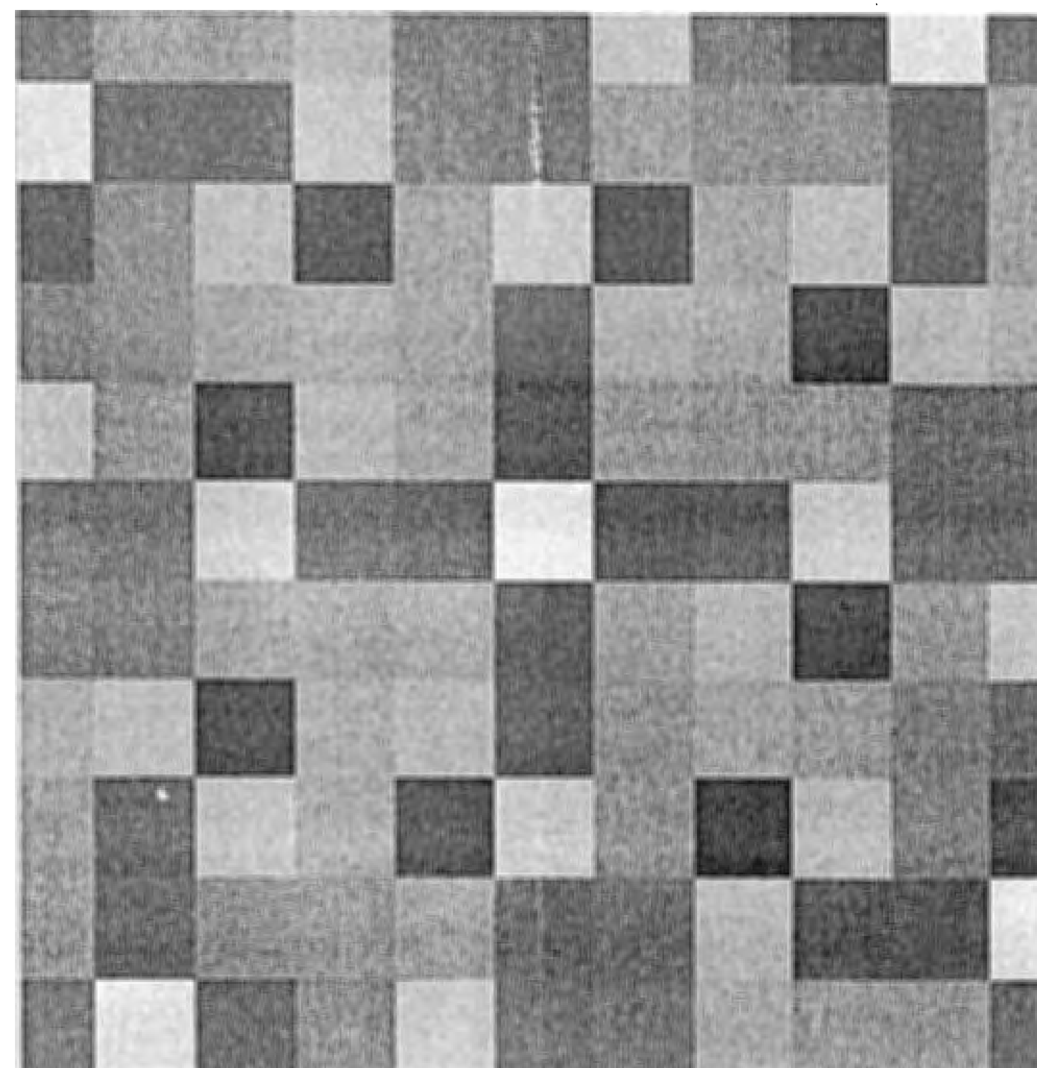
1 by 1 pixel Checker pattern
rotated 10 ^{o}



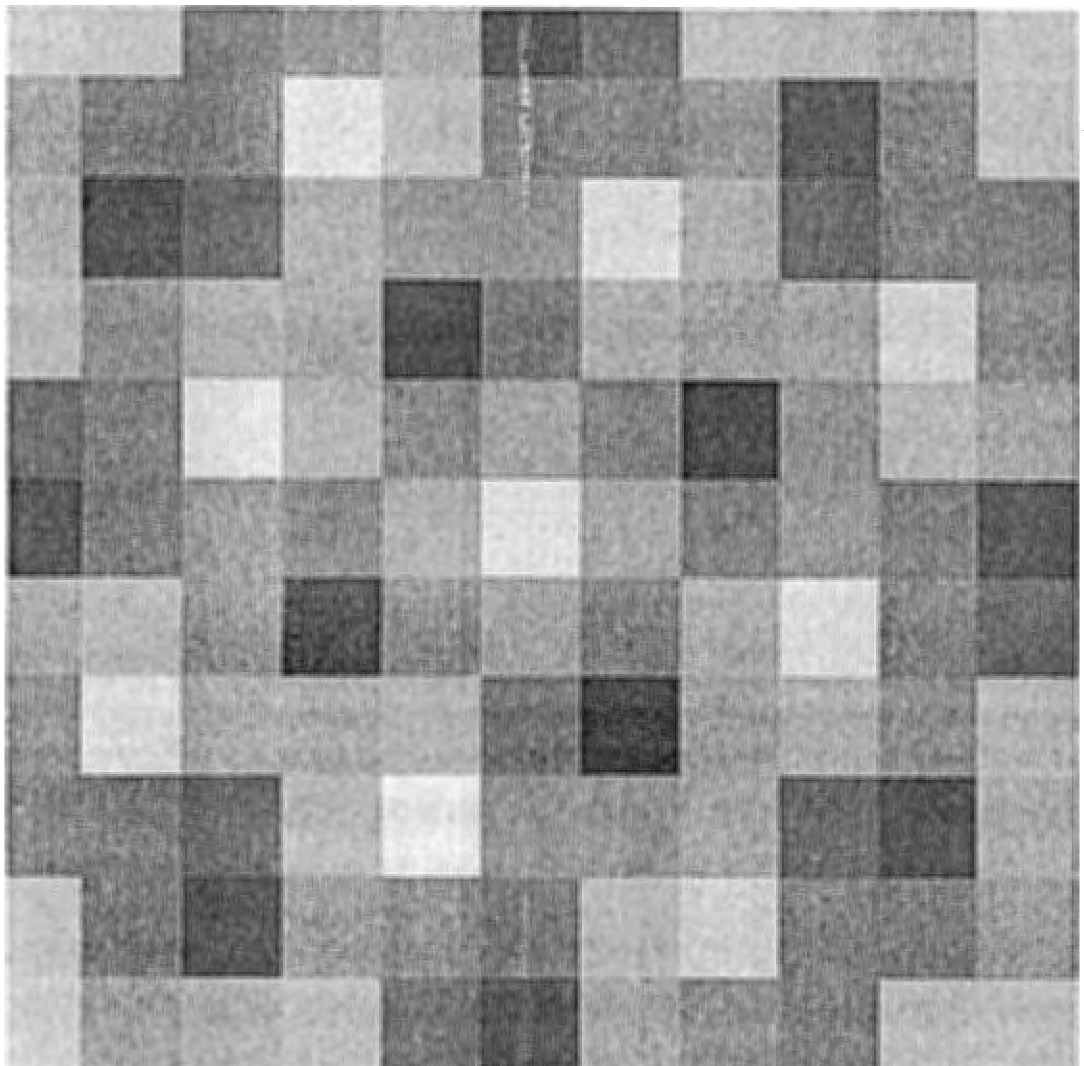
1 by 1 pixel Checker pattern
rotated 25 ^{o}



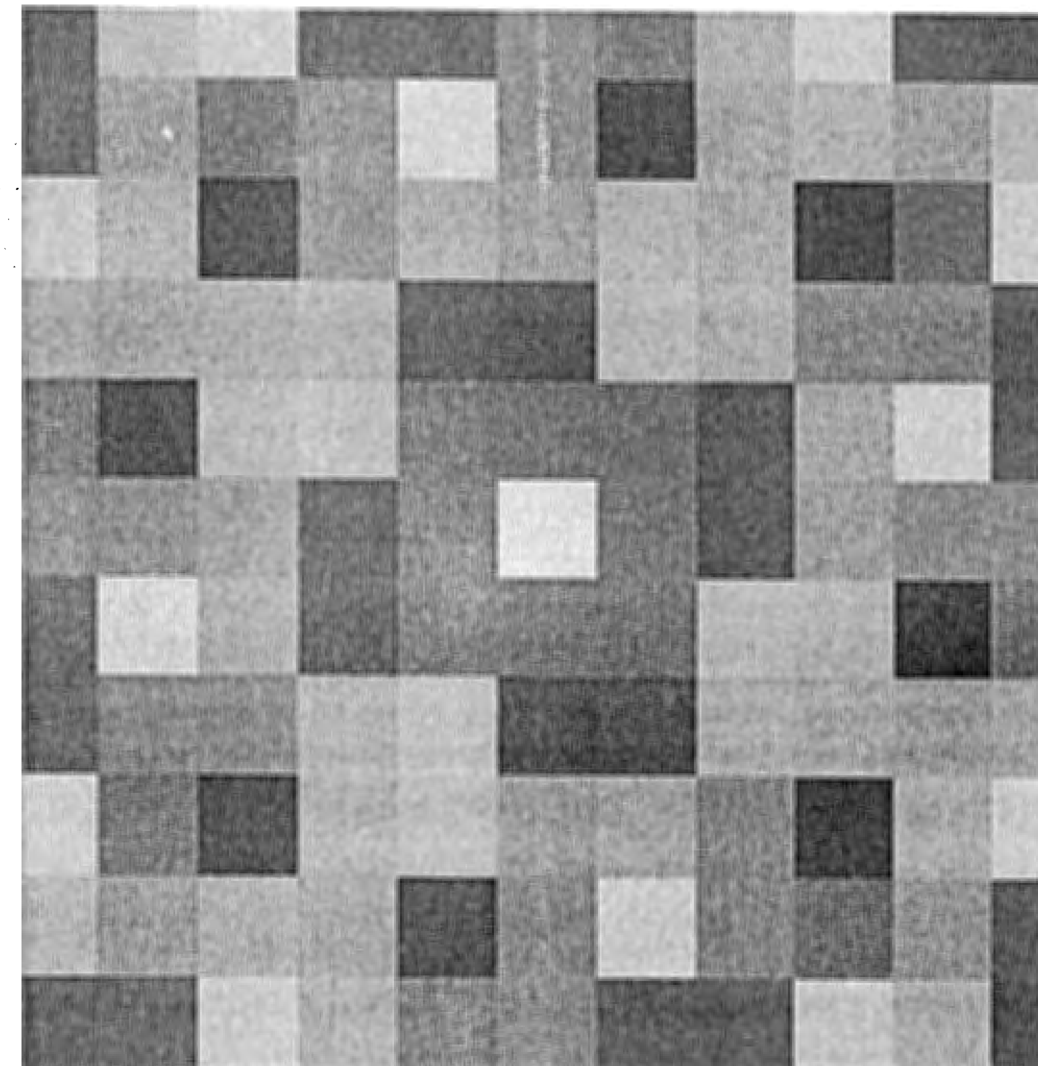
1 by 1 pixel Checker pattern
rotated 20 ^{o}



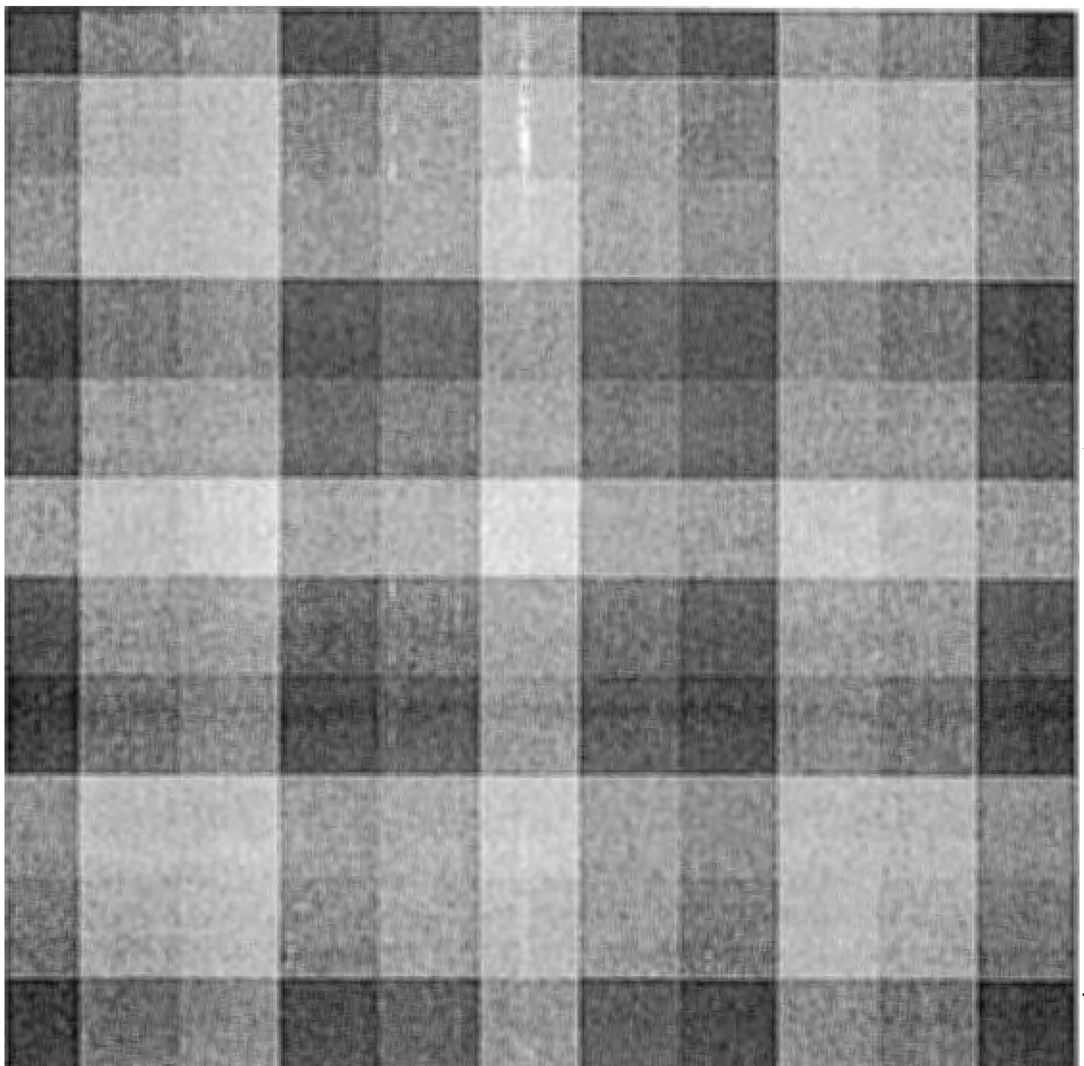
1 by 1 pixel Checker pattern
rotated 35 ^{o}



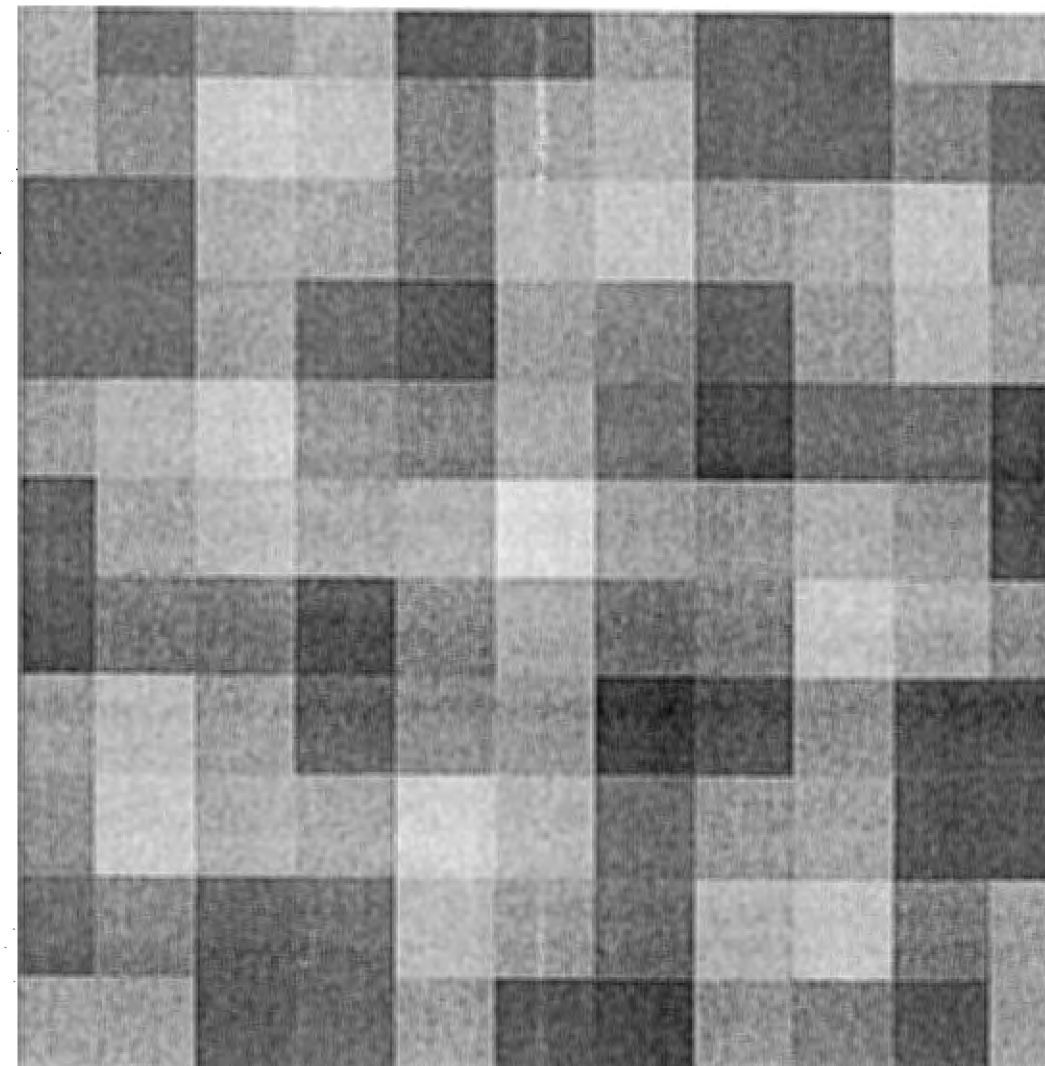
1 by 1 pixel Checker pattern
rotated 30 ^{o}



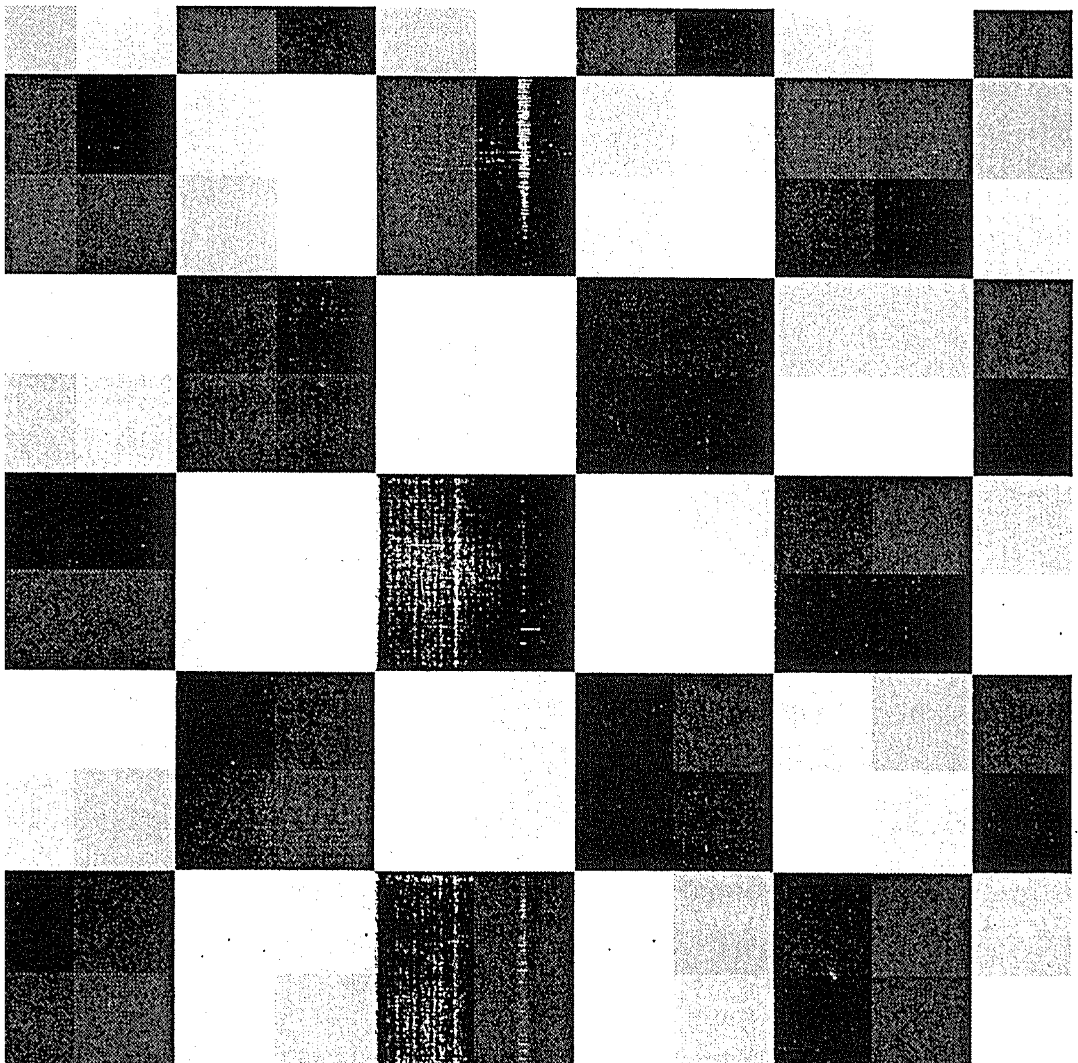
1 by 1 pixel Checker pattern
rotated 45 o



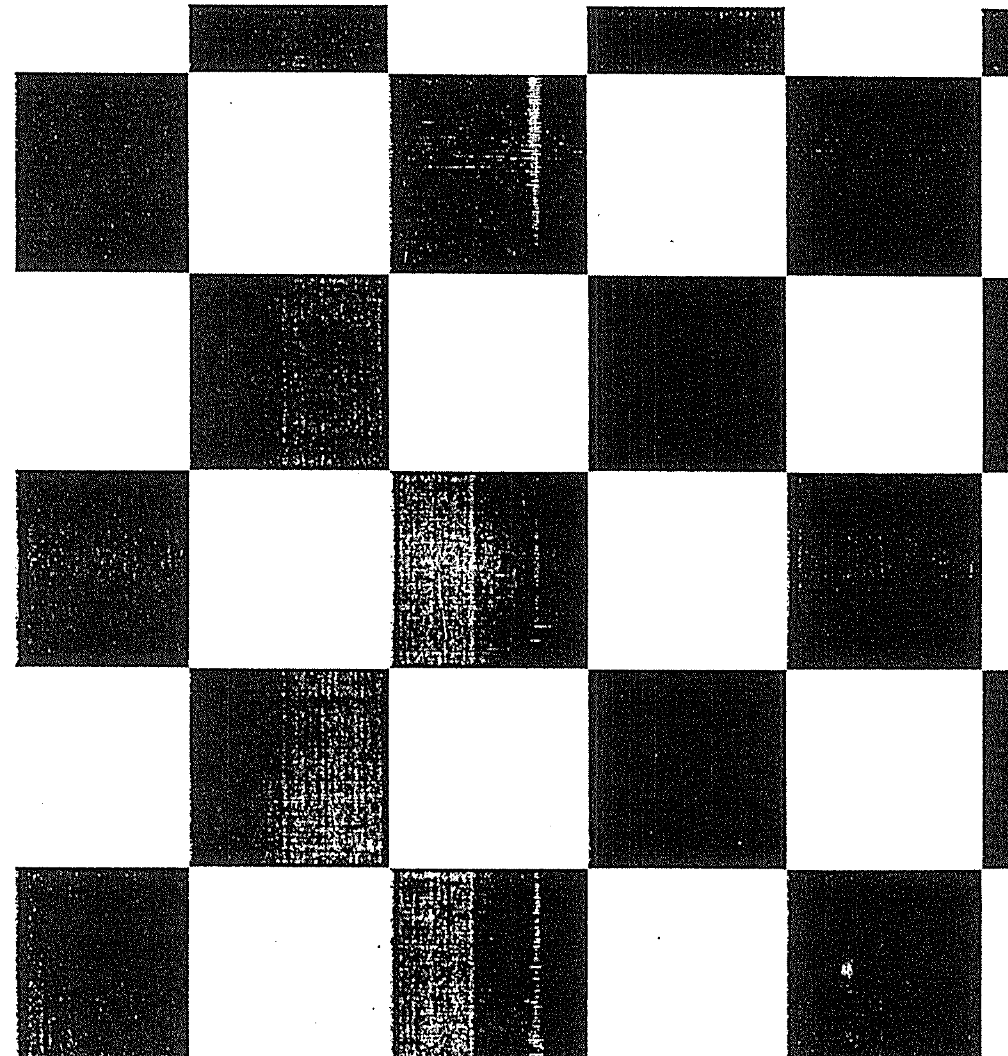
1 by 1 pixel Checker pattern
rotated 40 o



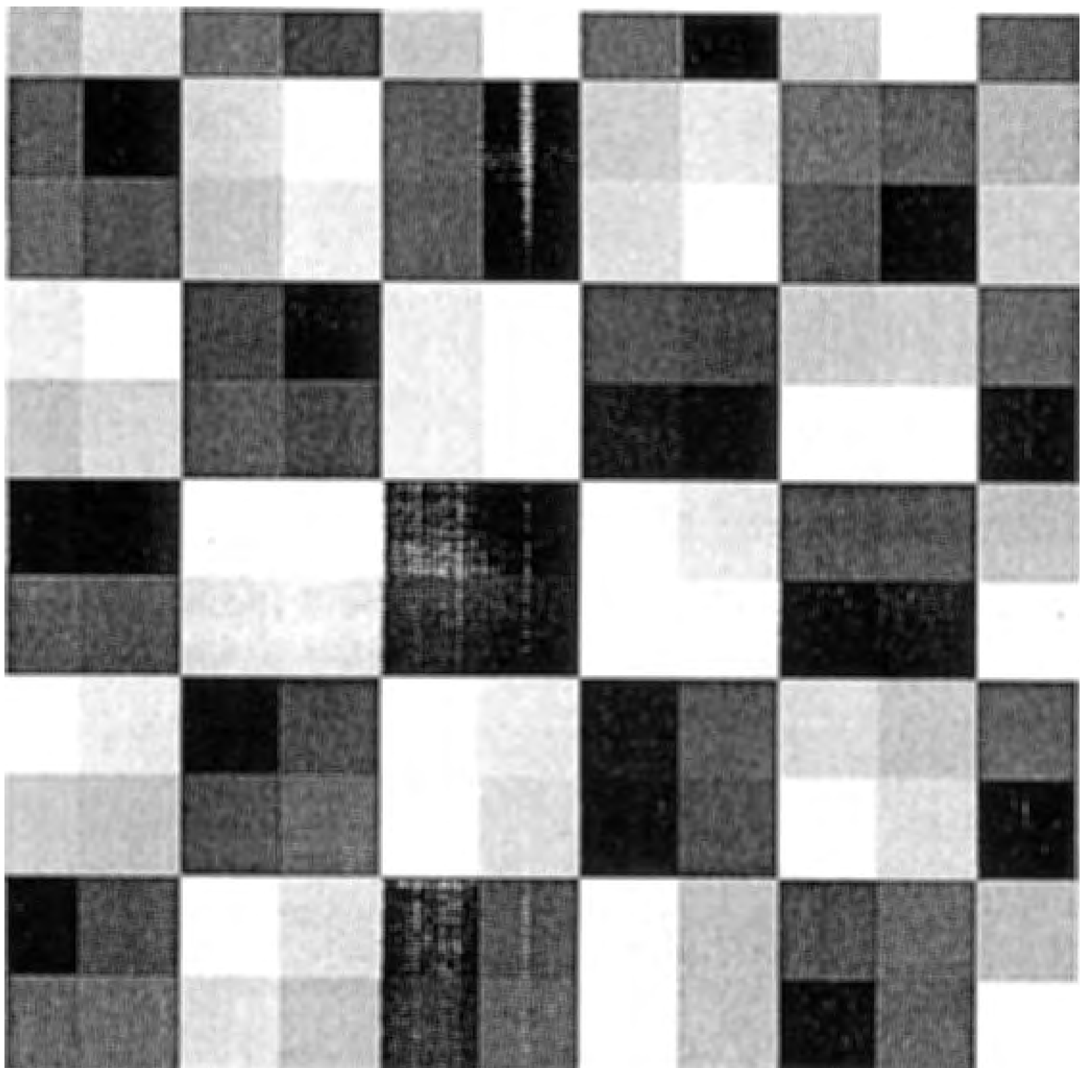
2 by 2 pixel Checker pattern
rotated 1°



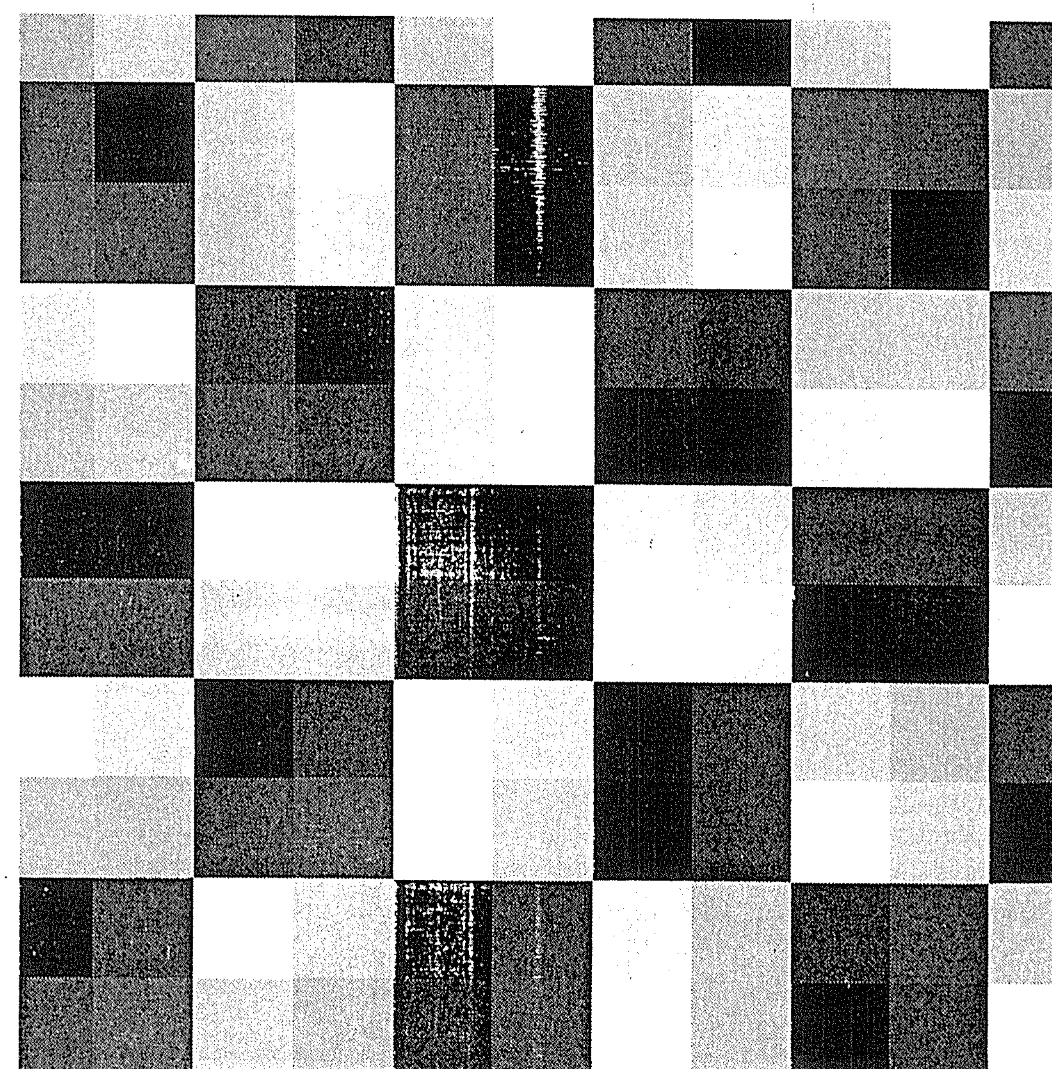
2 by 2 pixel Checker pattern
rotated 0°



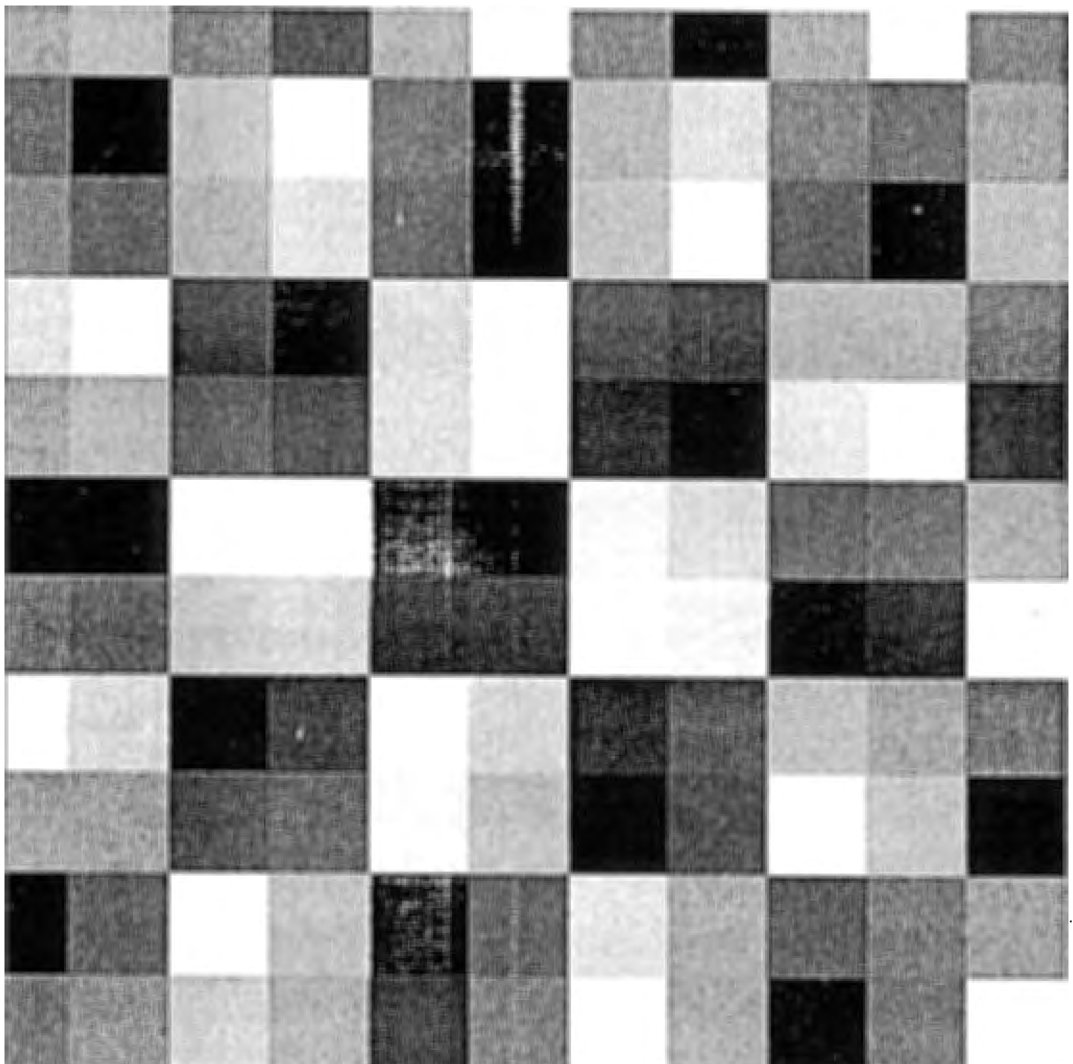
2 by 2 pixel Checker pattern
rotated 3 ^{o}



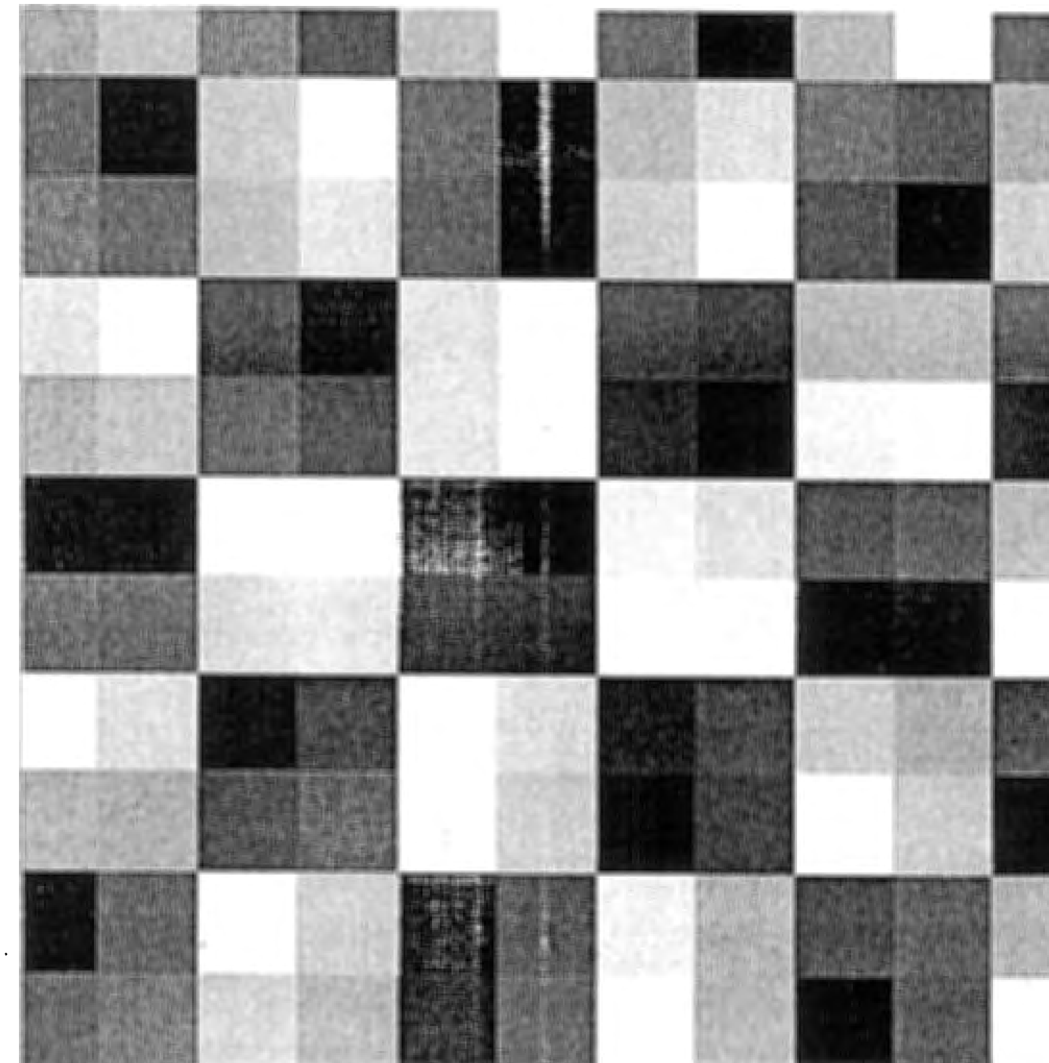
2 by 2 pixel Checker pattern
rotated 2 ^{o}



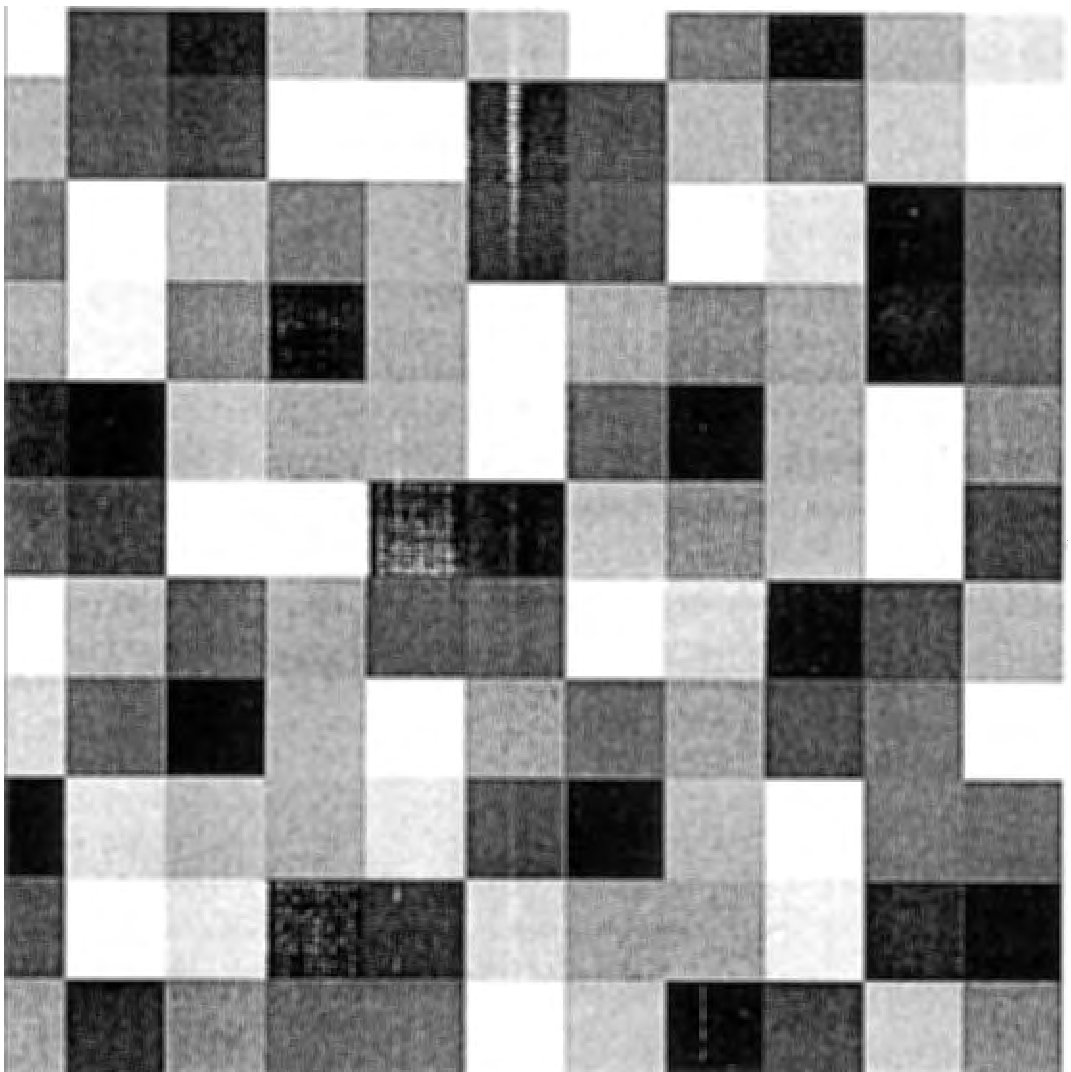
2 by 2 pixel Checker pattern rotated 5 ^{o}



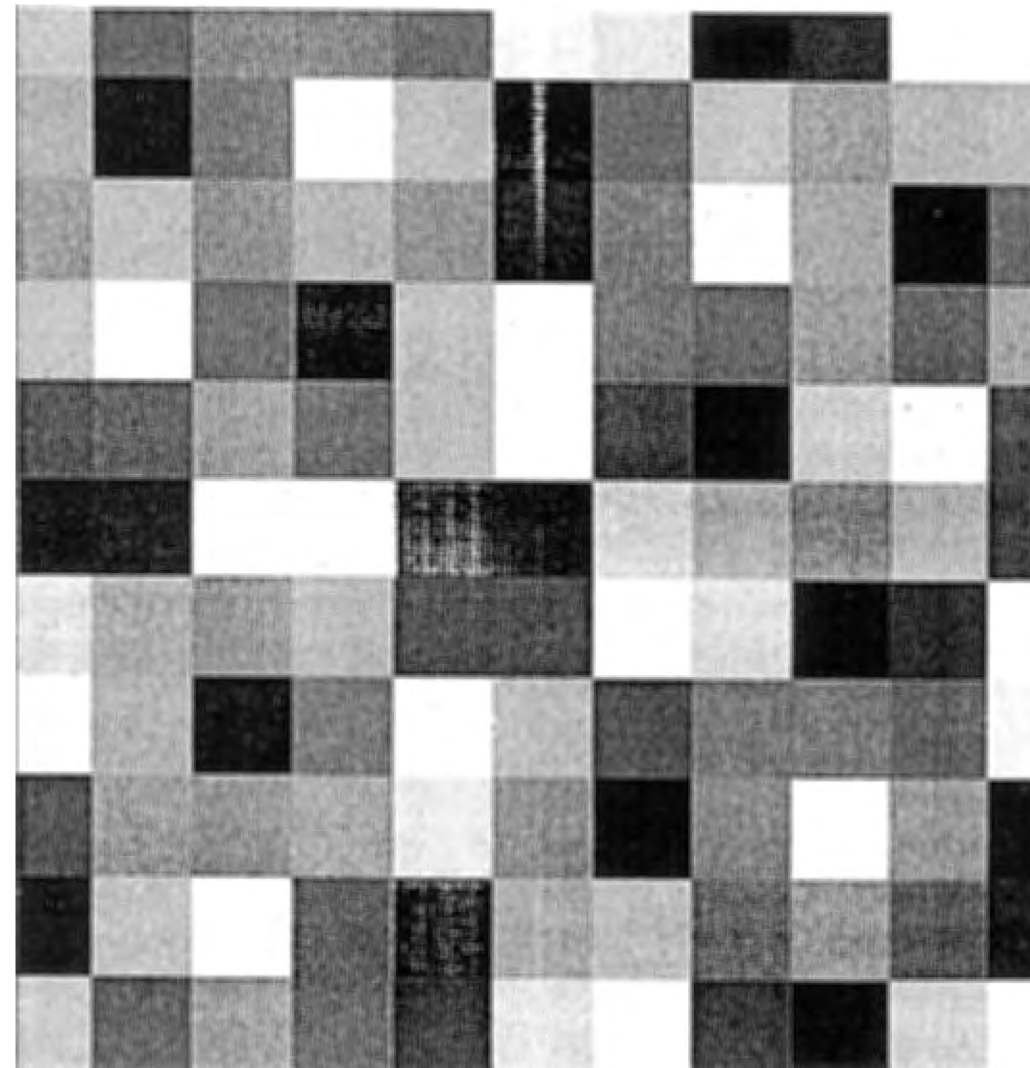
2 by 2 pixel Checker pattern rotated 4 ^{o}



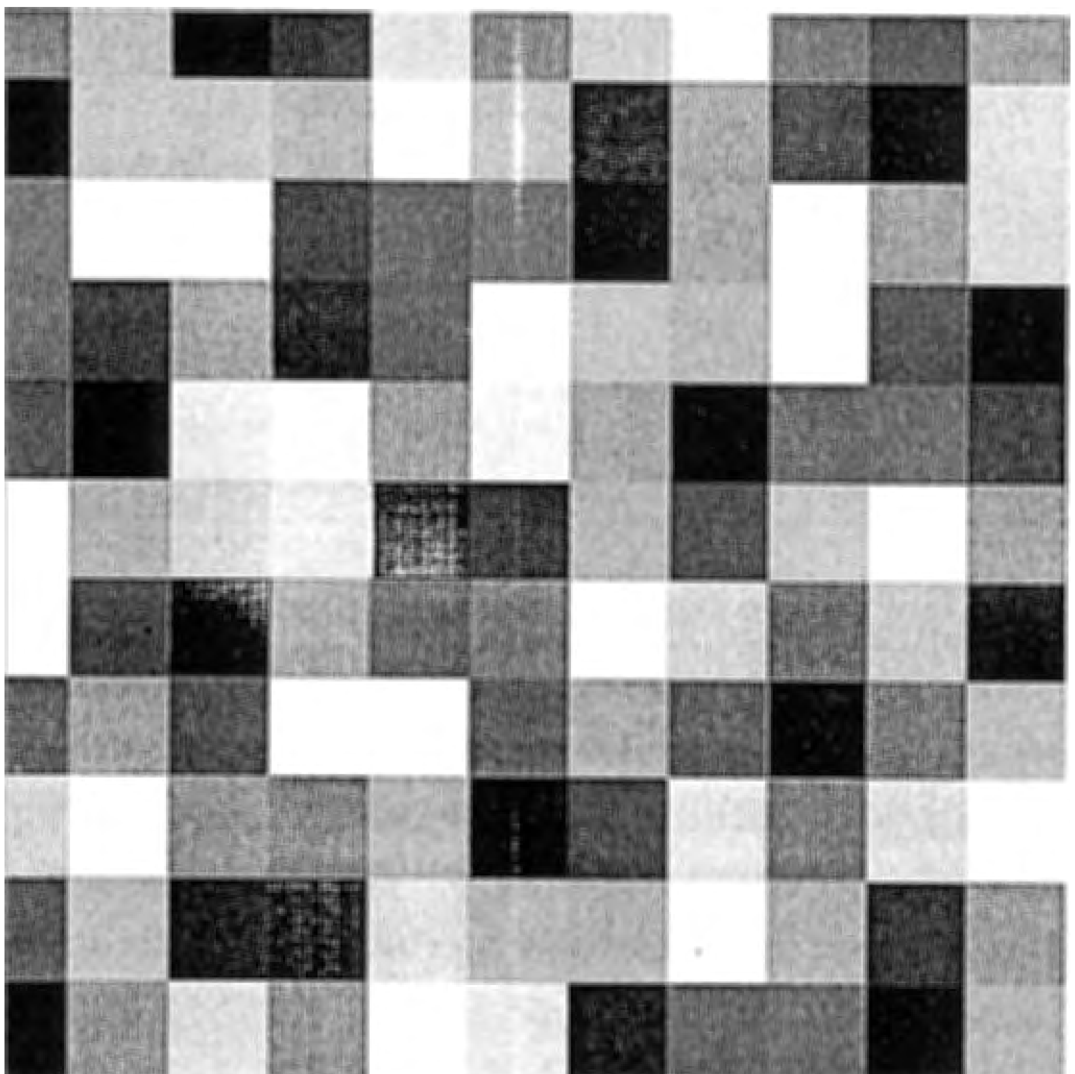
2 by 2 pixel Checker pattern
rotated 15 o



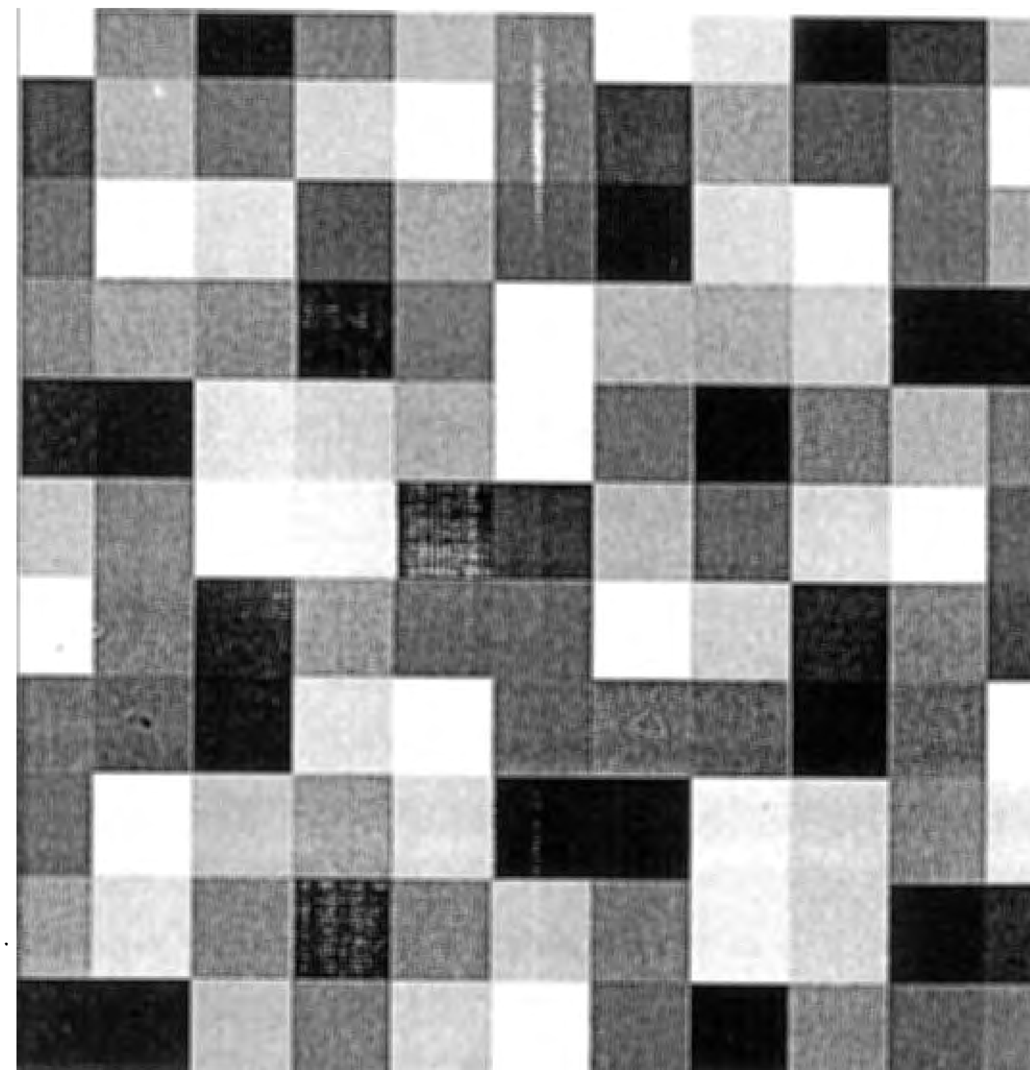
2 by 2 pixel Checker pattern
rotated 10 o



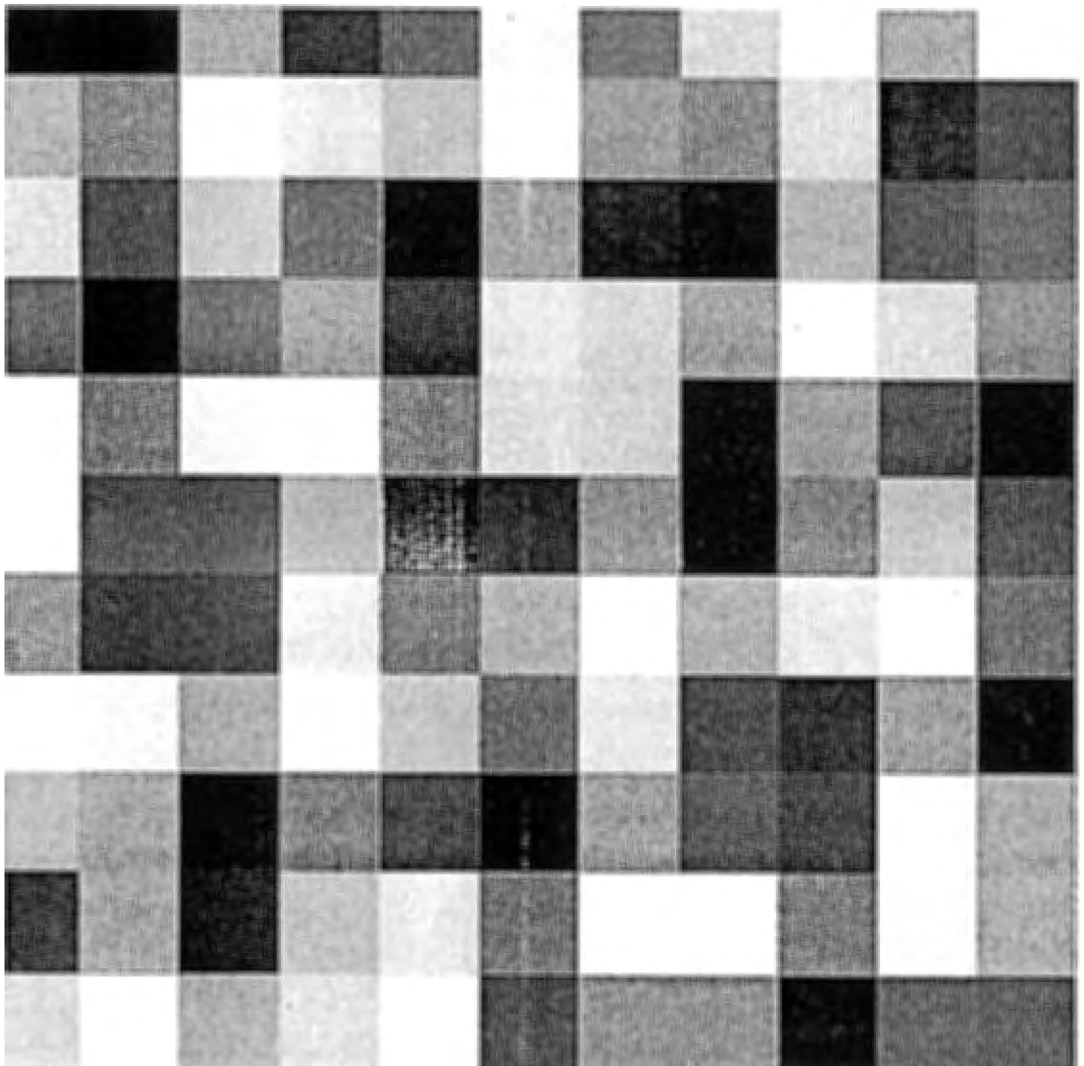
2 by 2 pixel Checker pattern
rotated 25 °



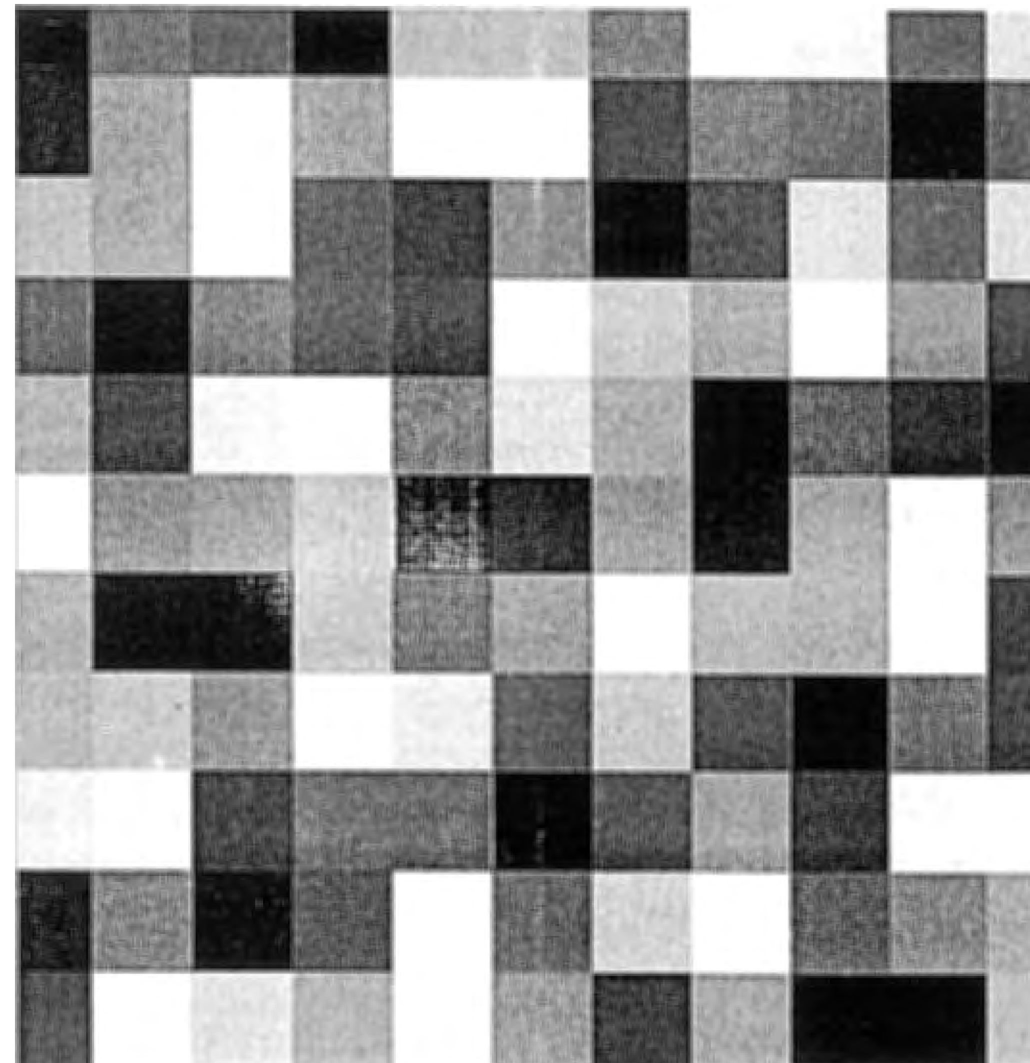
2 by 2 pixel Checker pattern
rotated 20 °



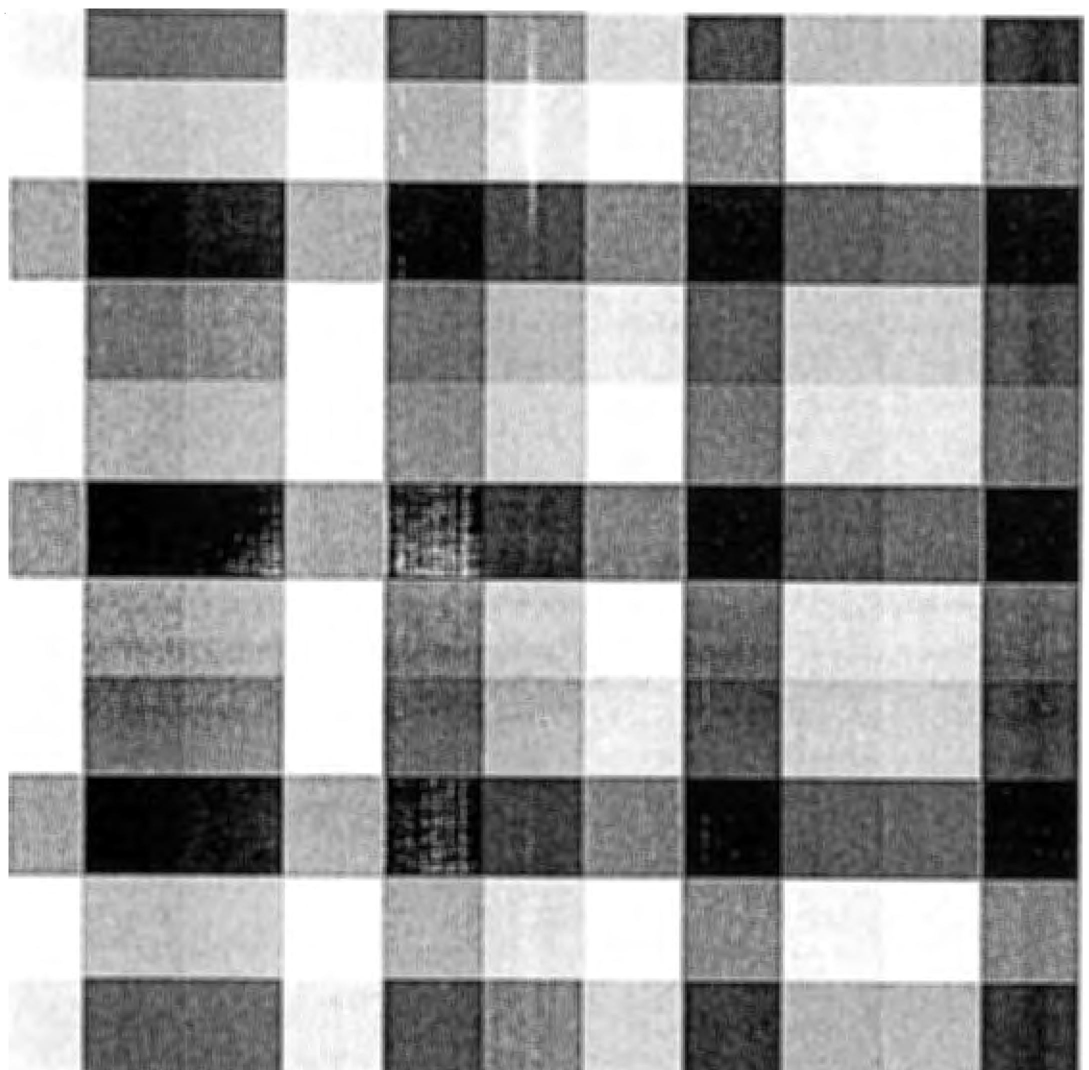
2 by 2 pixel Checker pattern
rotated 35 °



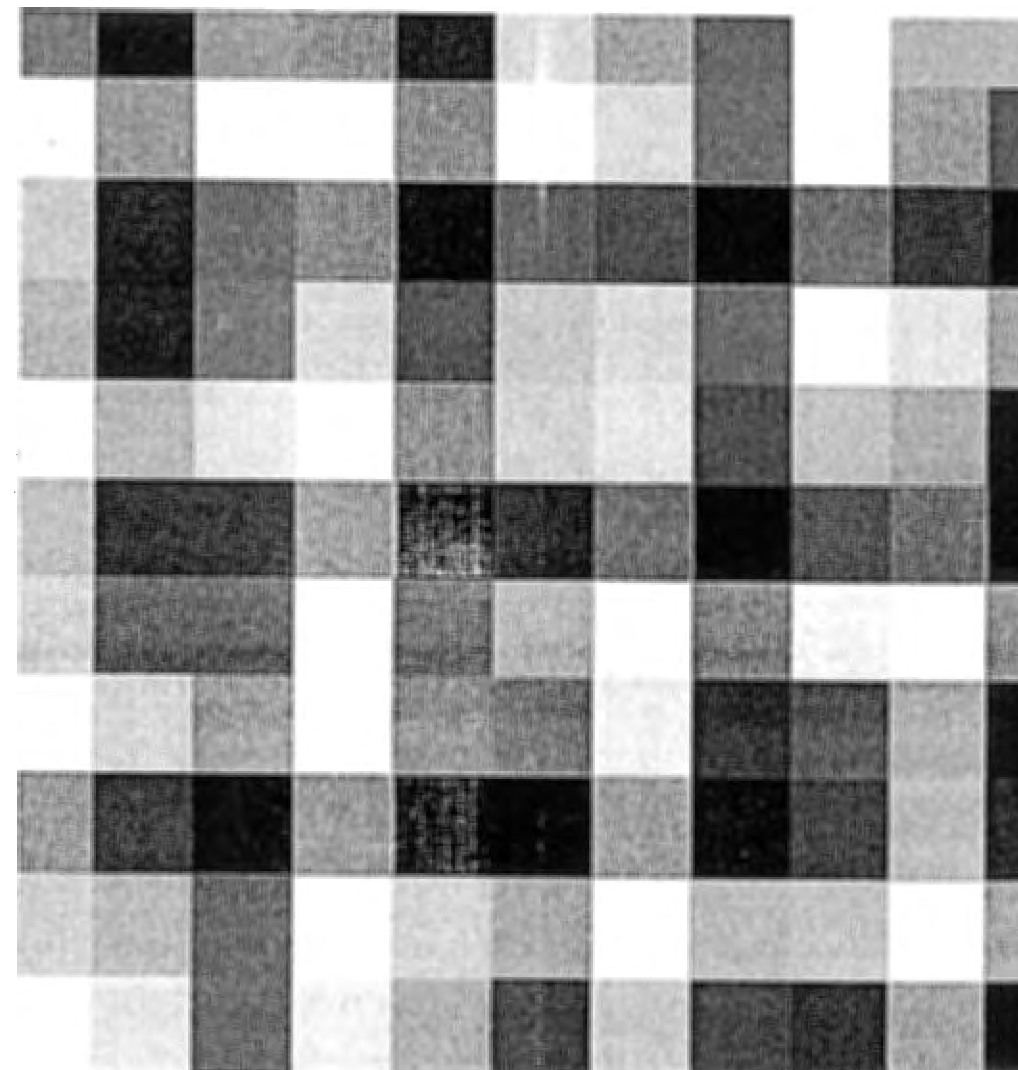
2 by 2 pixel Checker pattern
rotated 30 °



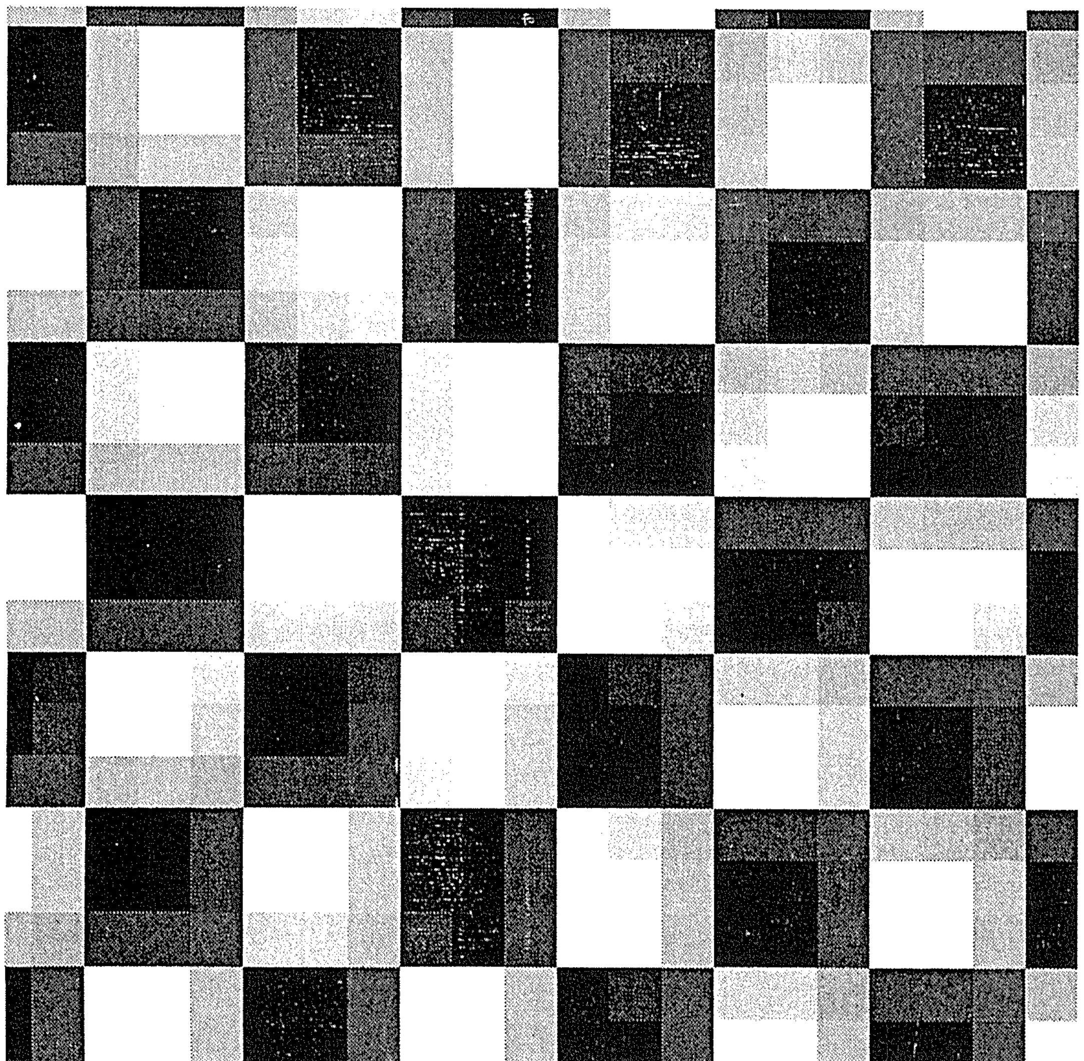
2 by 2 pixel Checker pattern
rotated 45 o



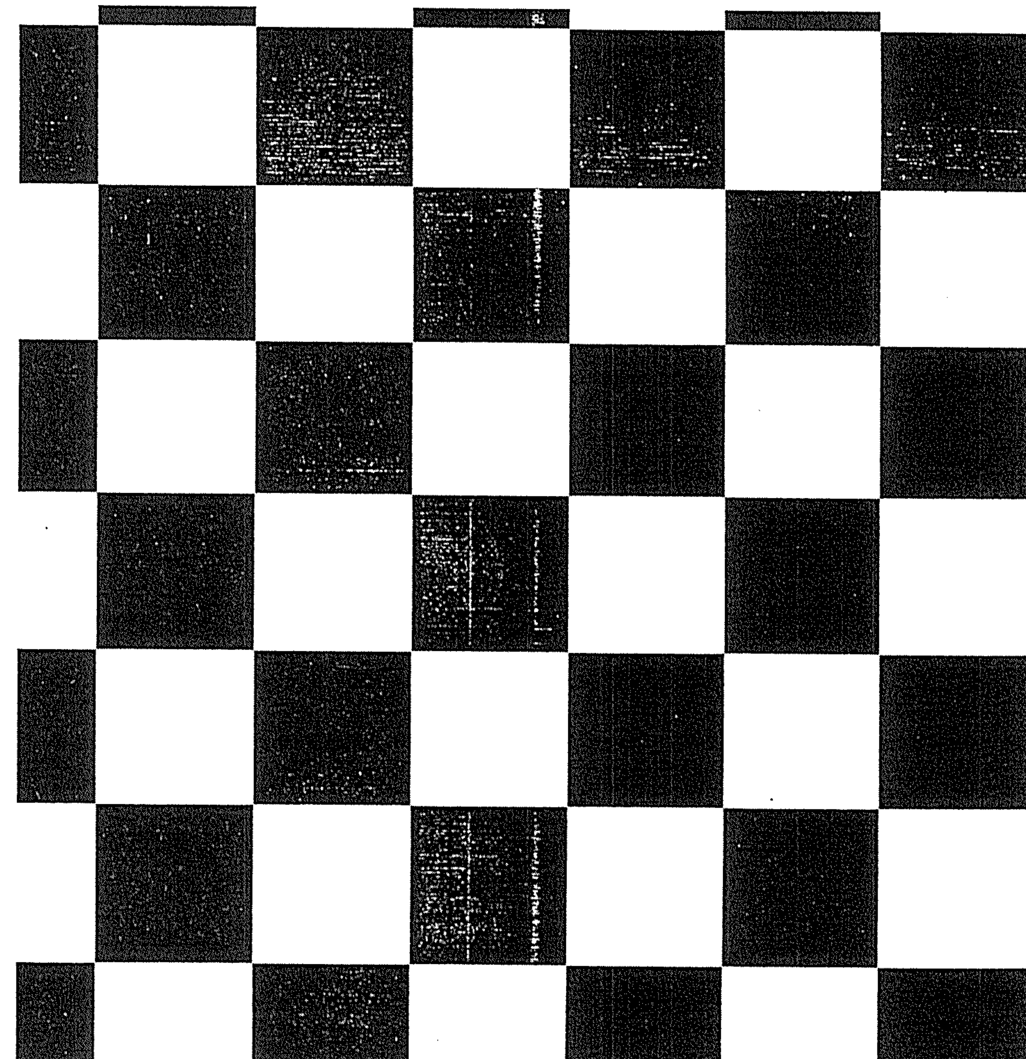
2 by 2 pixel Checker pattern
rotated 40 o



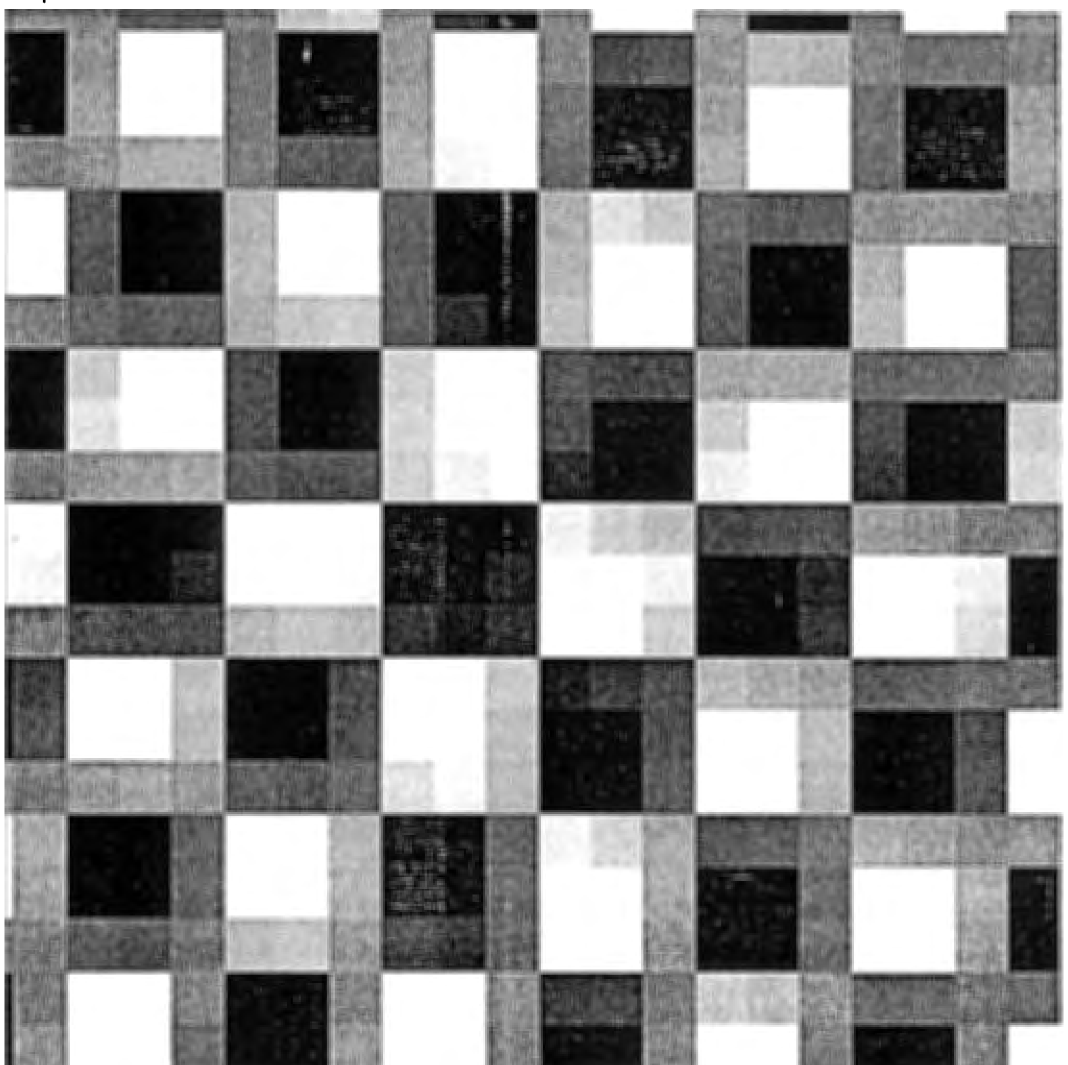
3 by 3 pixel Checker pattern
rotated 1 o



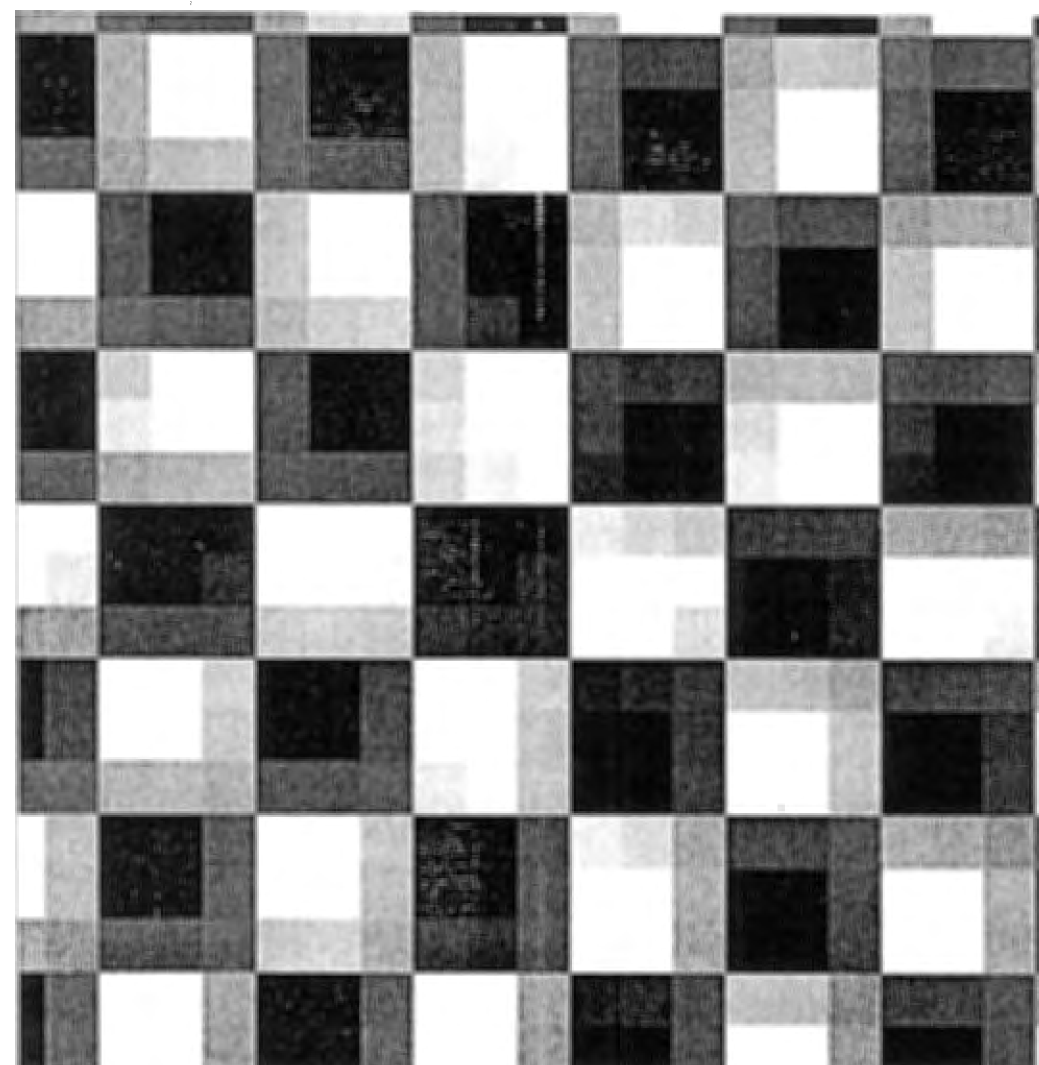
3 by 3 pixel Checker pattern
rotated 0 o



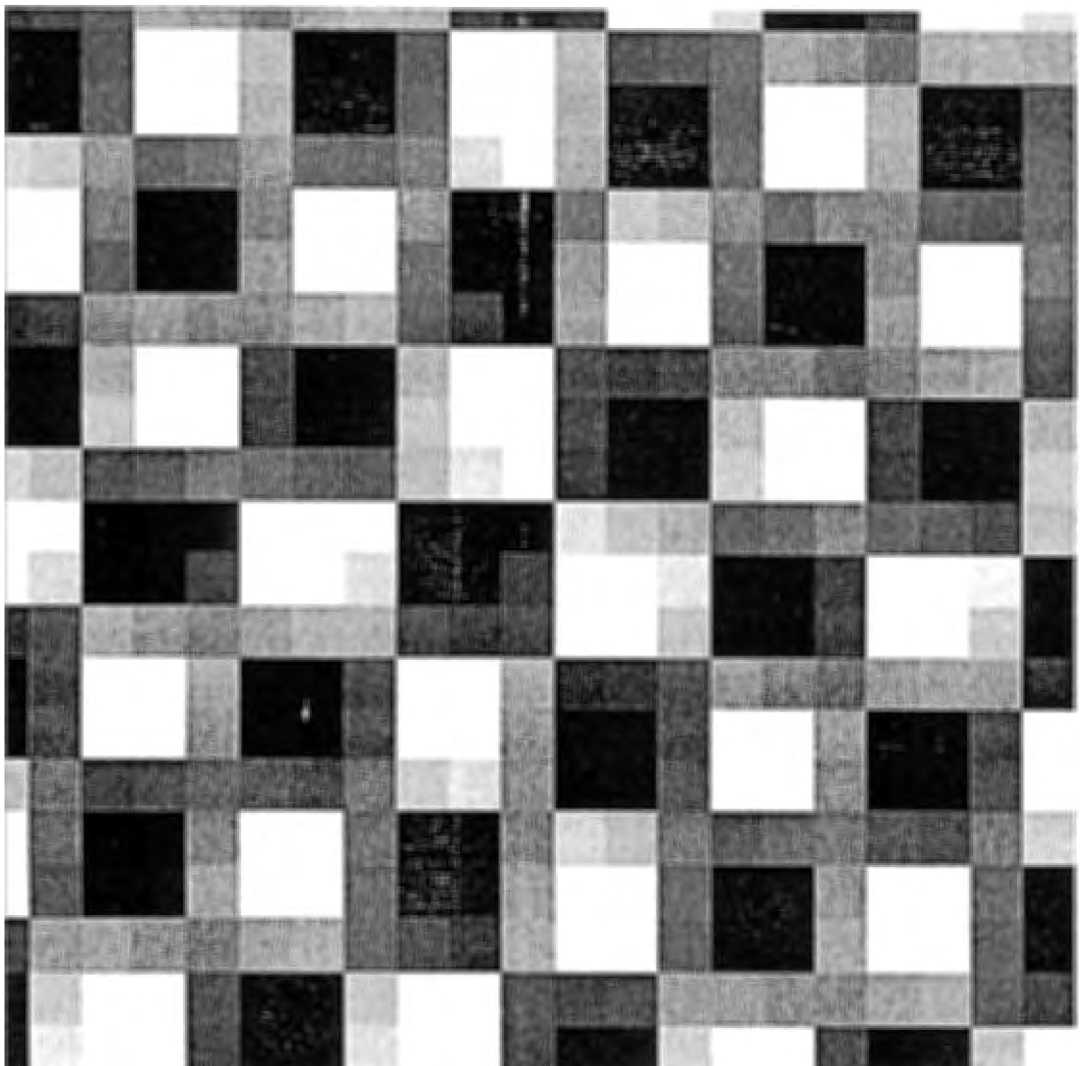
3 by 3 pixel Checker pattern
rotated 3 ^{o}



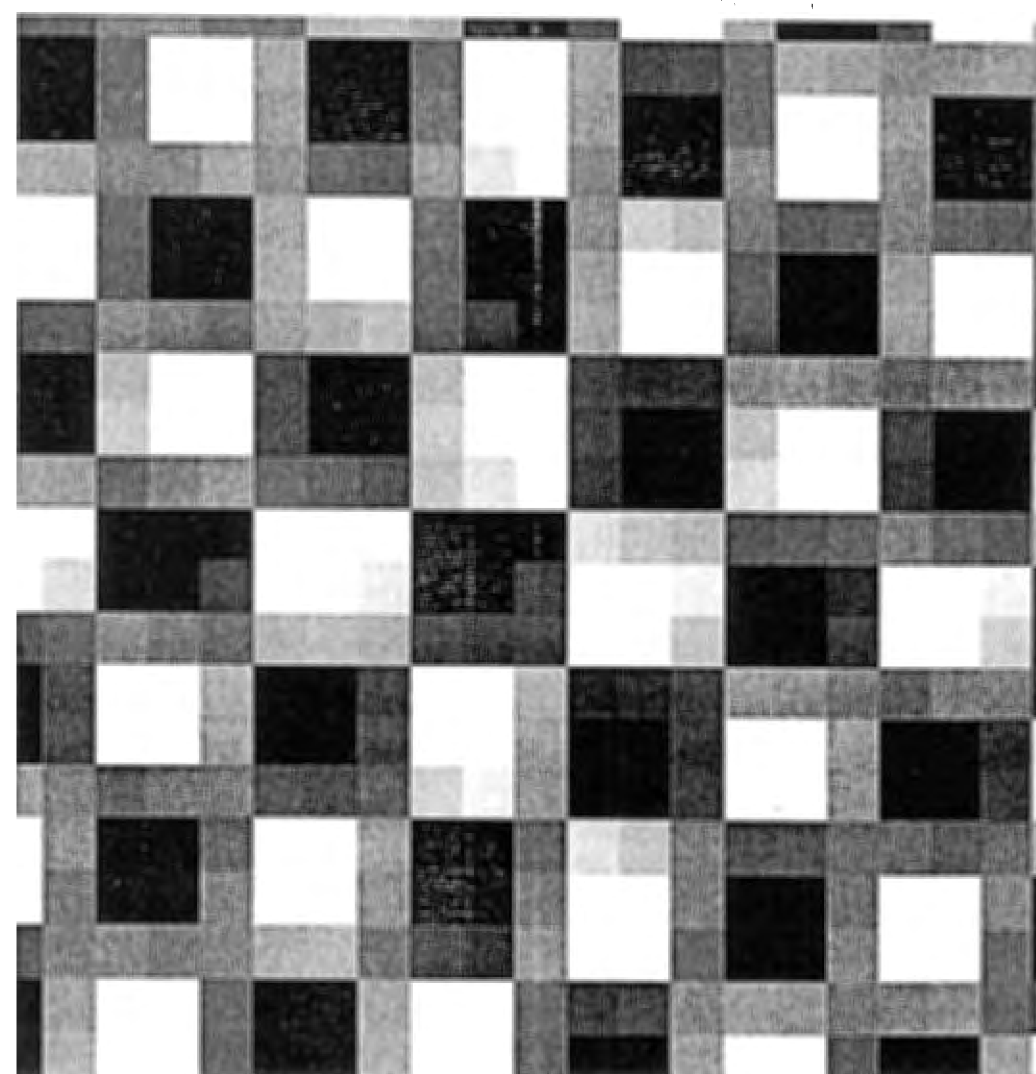
3 by 3 pixel Checker pattern
rotated 2 ^{o}



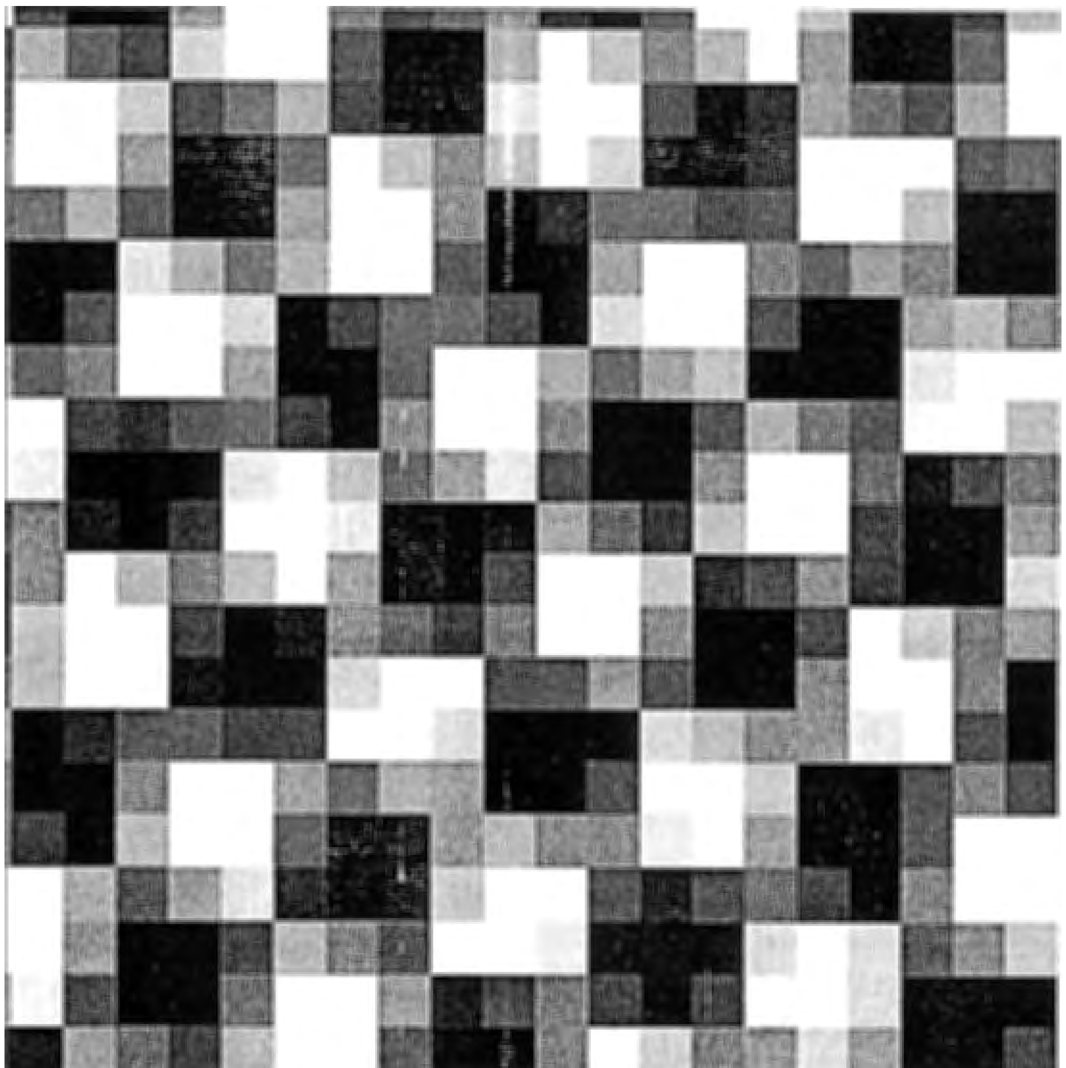
3 by 3 pixel Checker pattern
rotated 5 ^{o}



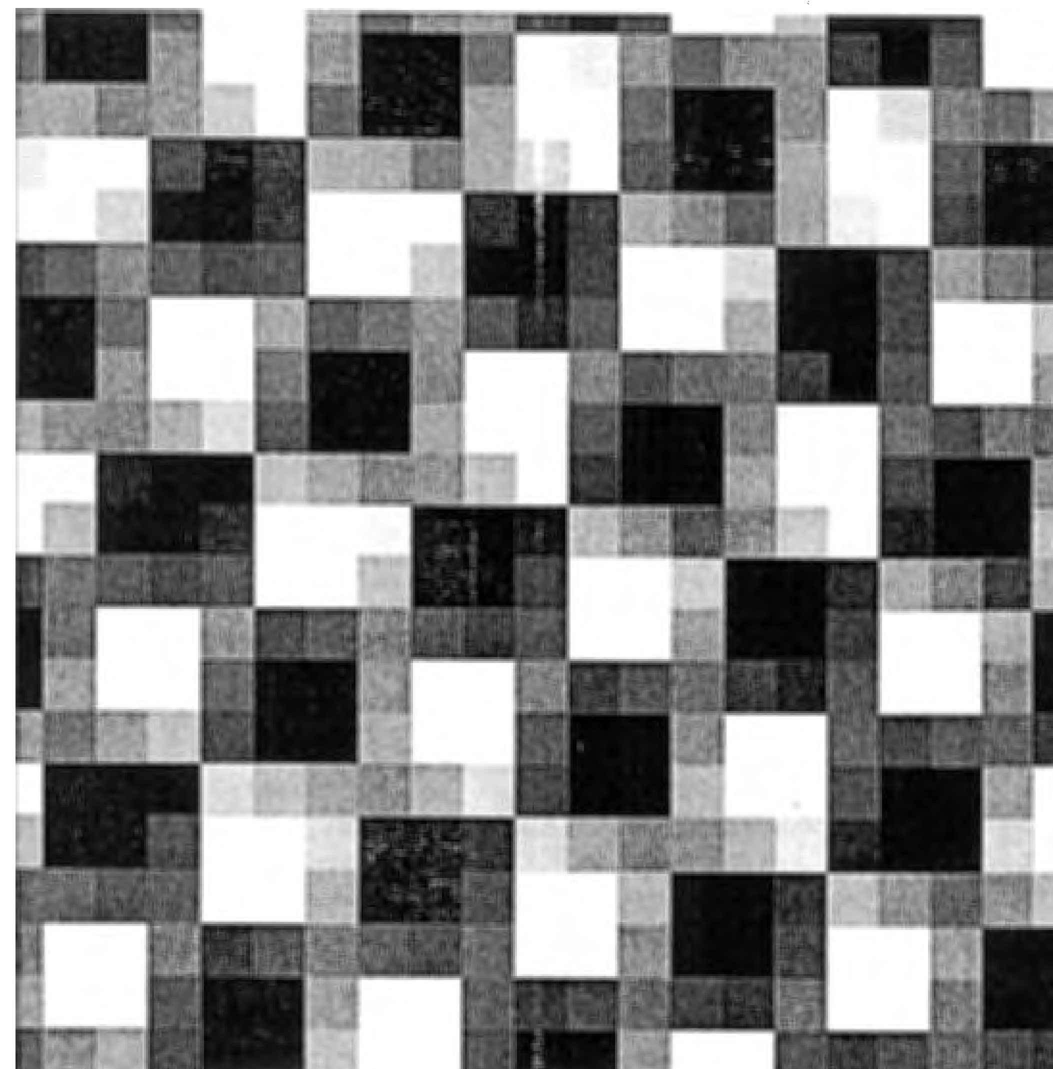
3 by 3 pixel Checker pattern
rotated 4 ^{o}



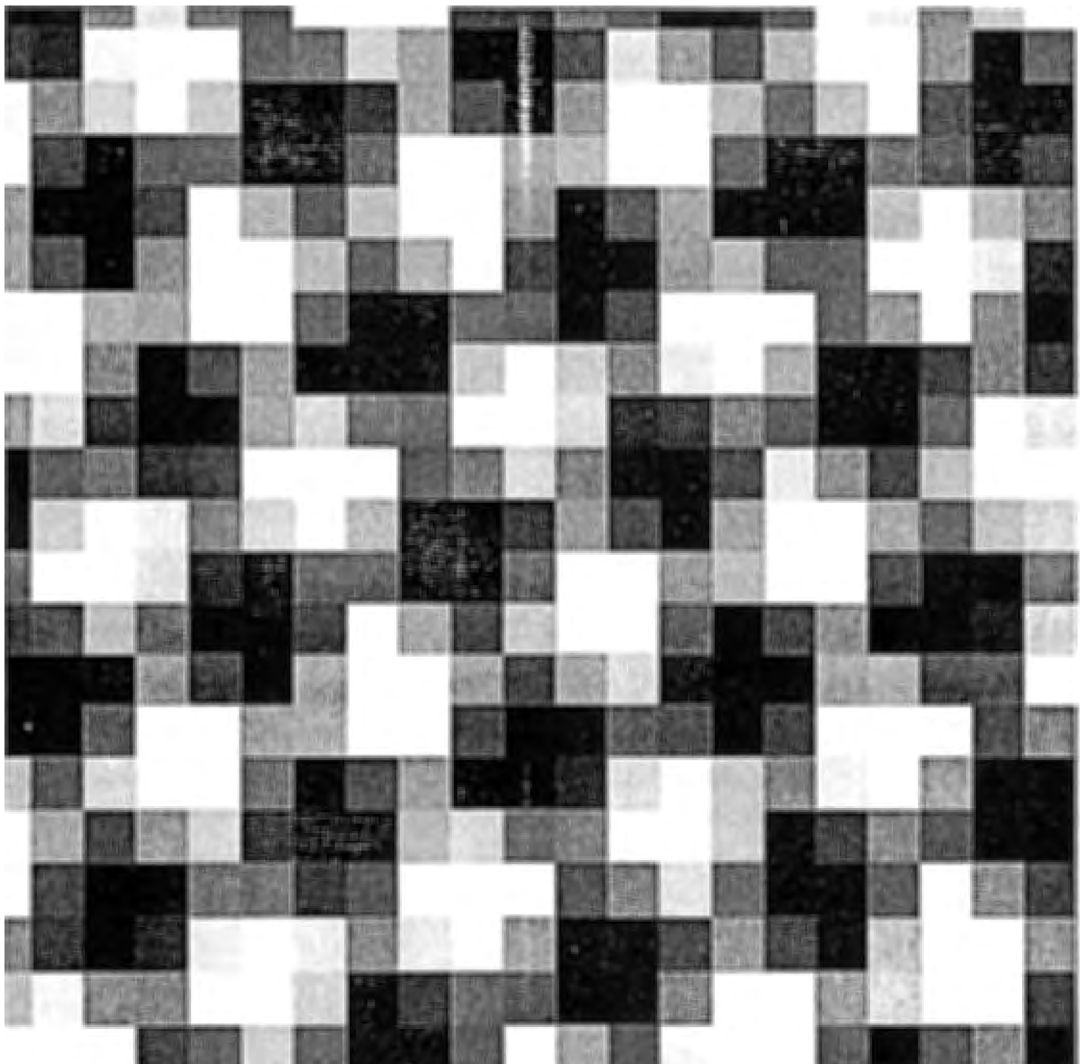
3 by 3 pixel Checker pattern
rotated 15 ^{o}



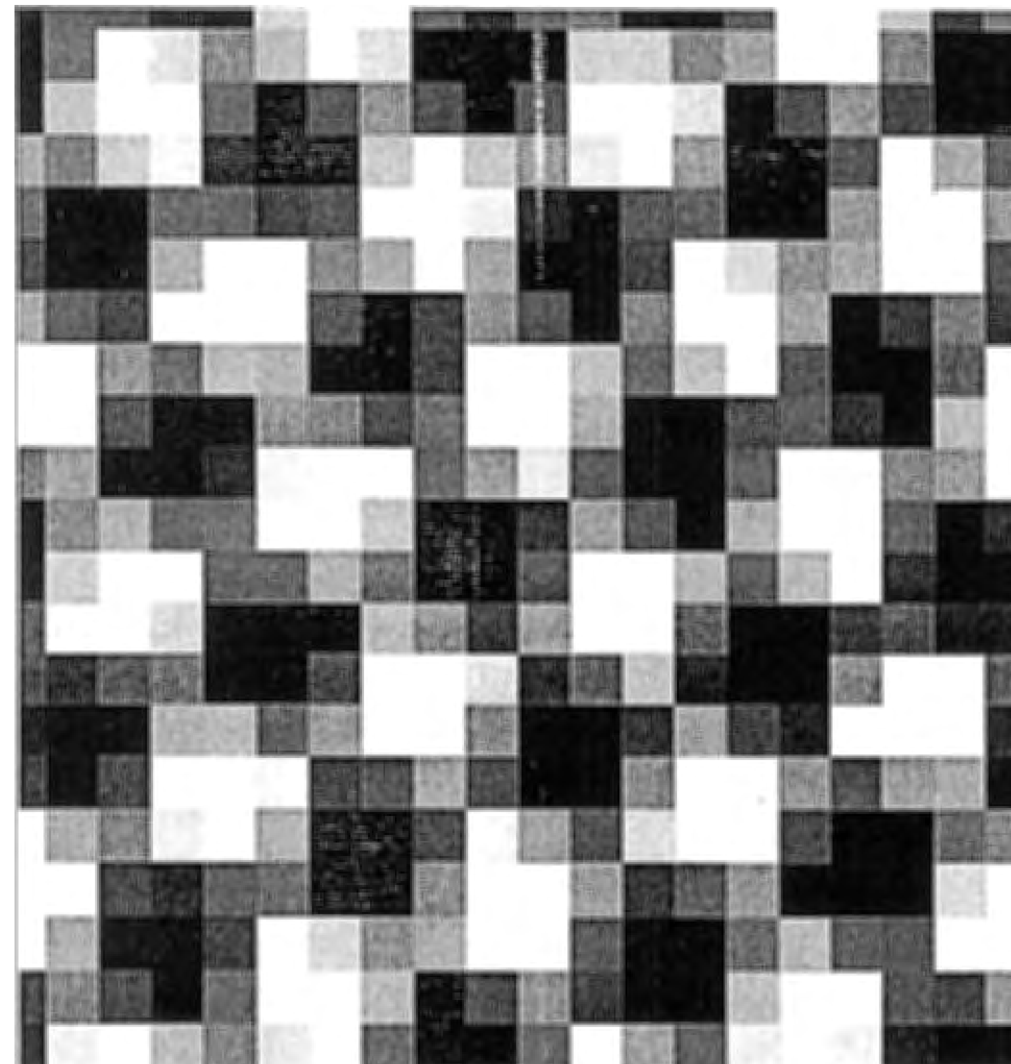
3 by 3 pixel Checker pattern
rotated 10 ^{o}



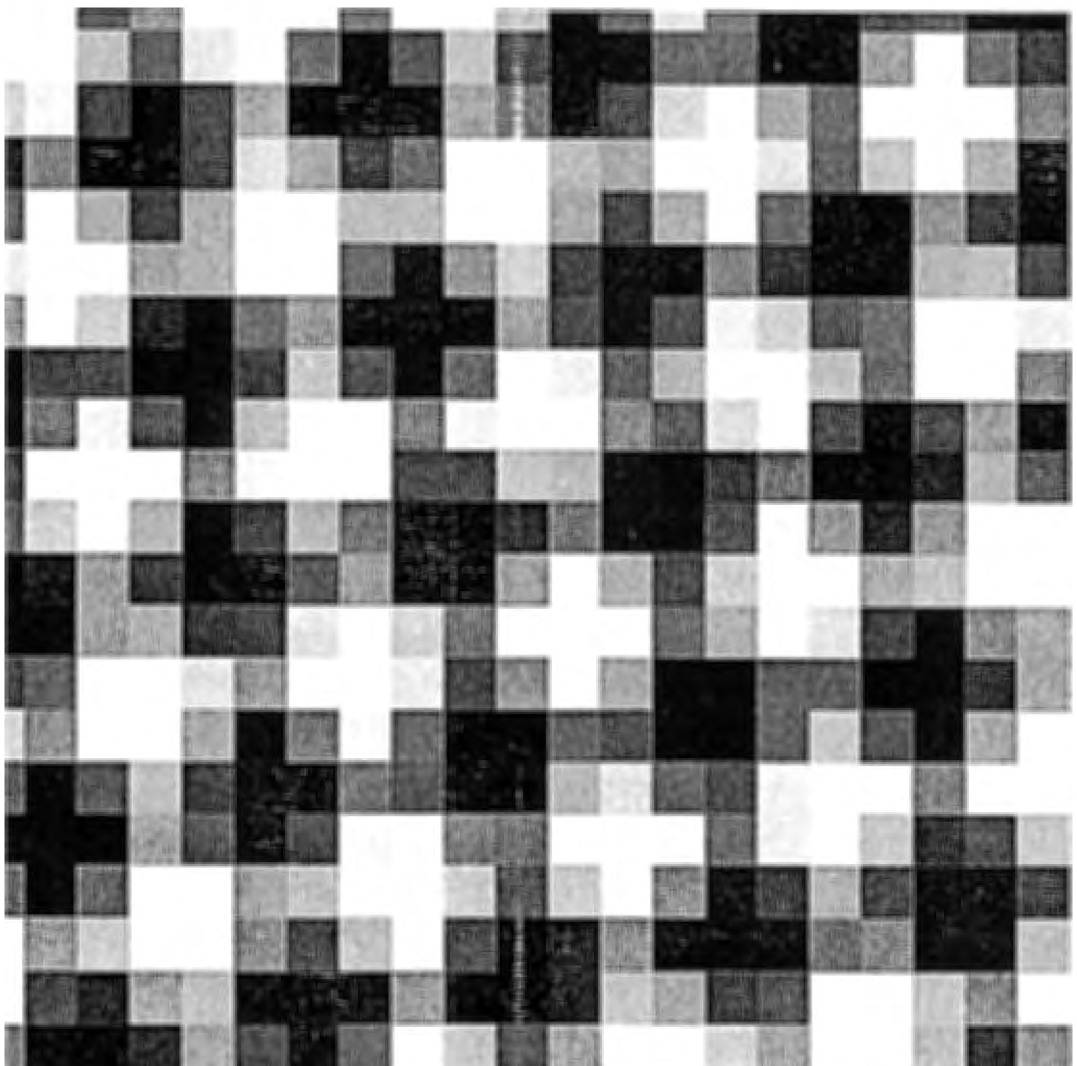
3 by 3 pixel Checker pattern
rotated 25 ^{o}



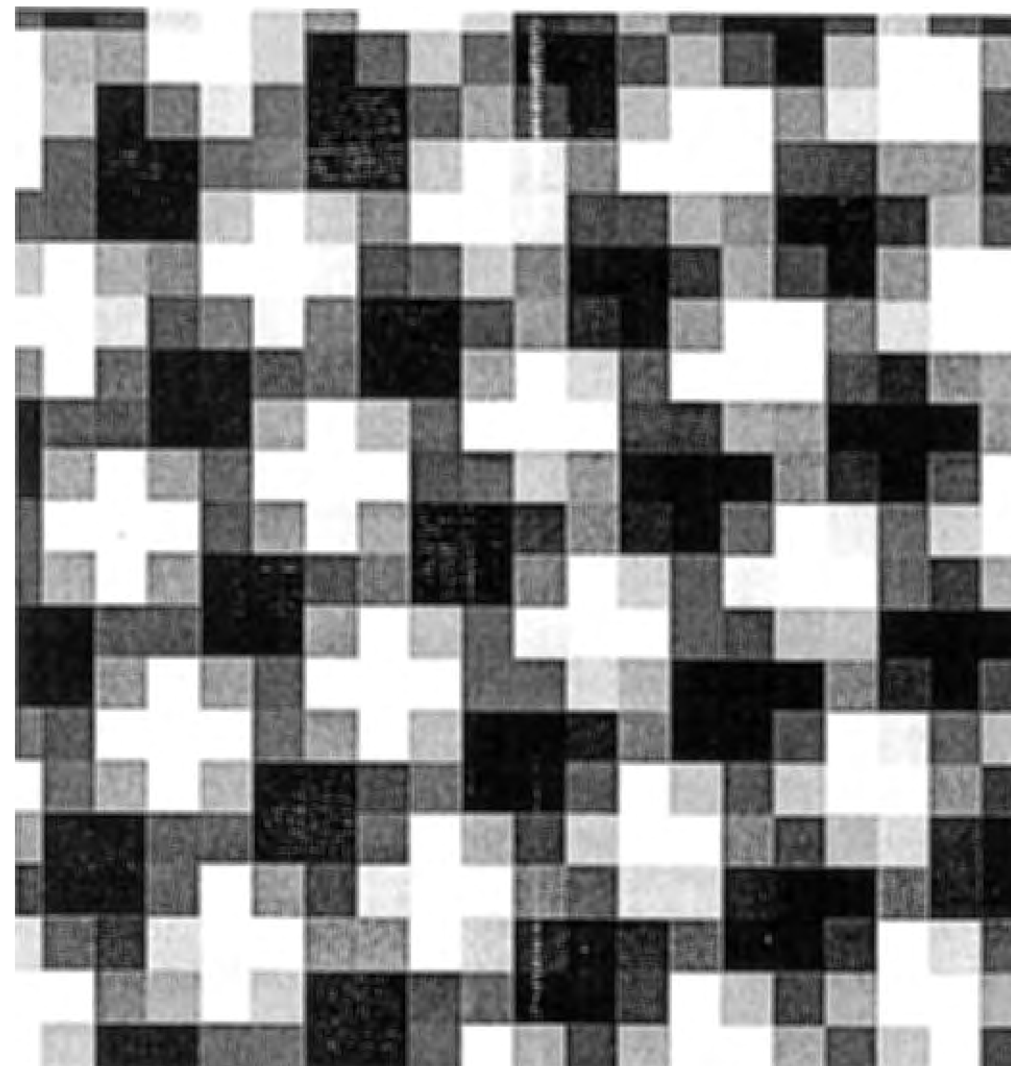
3 by 3 pixel Checker pattern
rotated 20 ^{o}



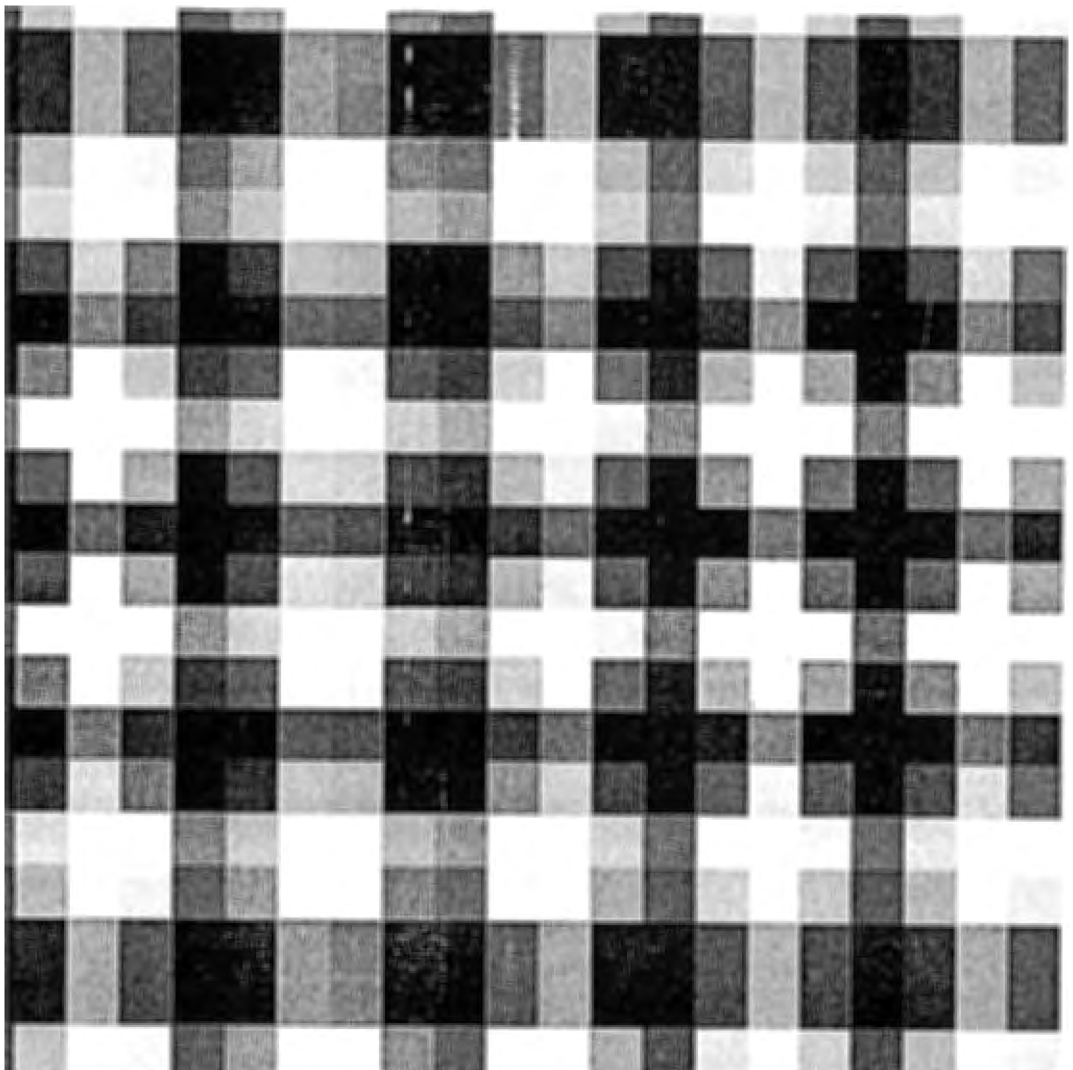
3 by 3 pixel Checker pattern
rotated 35 °



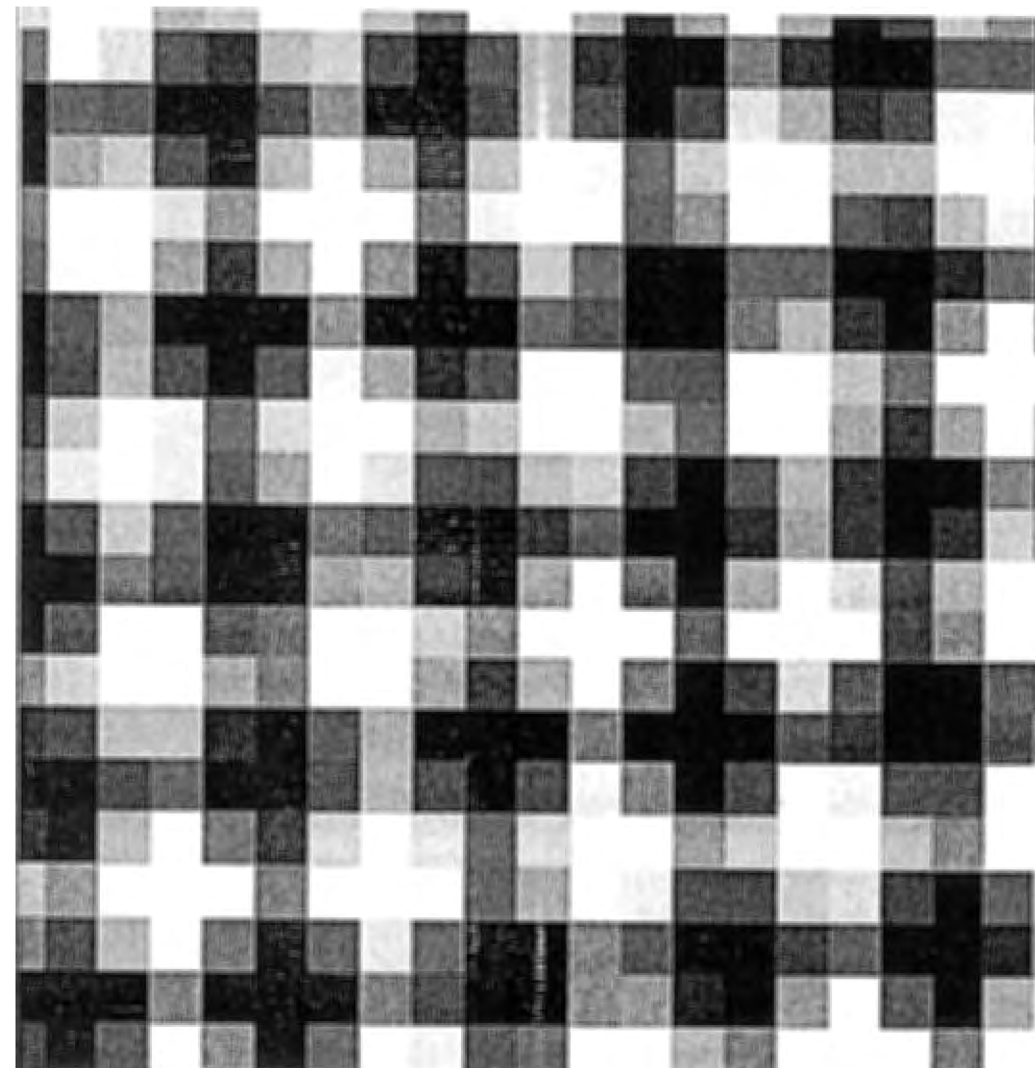
3 by 3 pixel Checker pattern
rotated 30 °



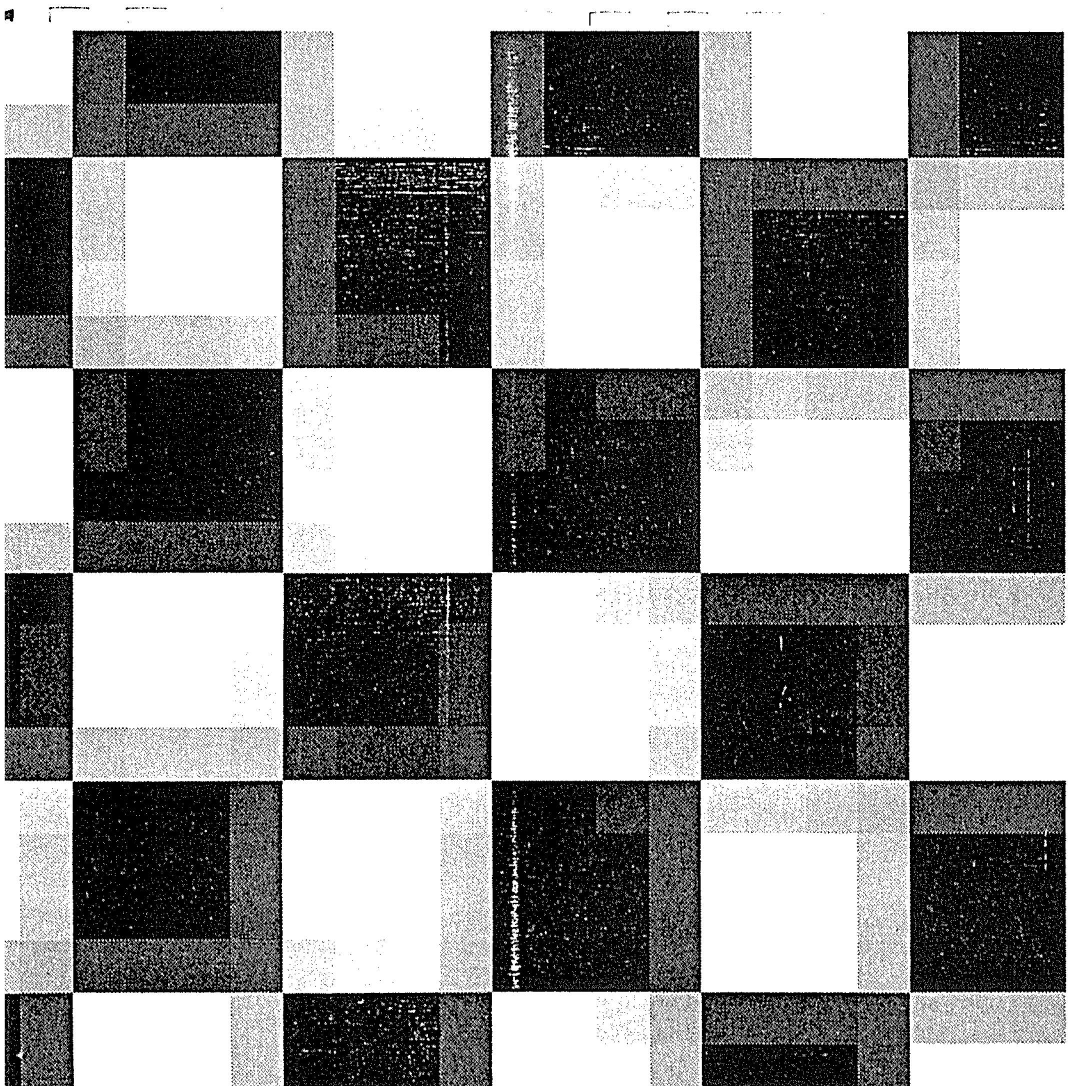
3 by 3 pixel Checker pattern
rotated 45 ^{o}



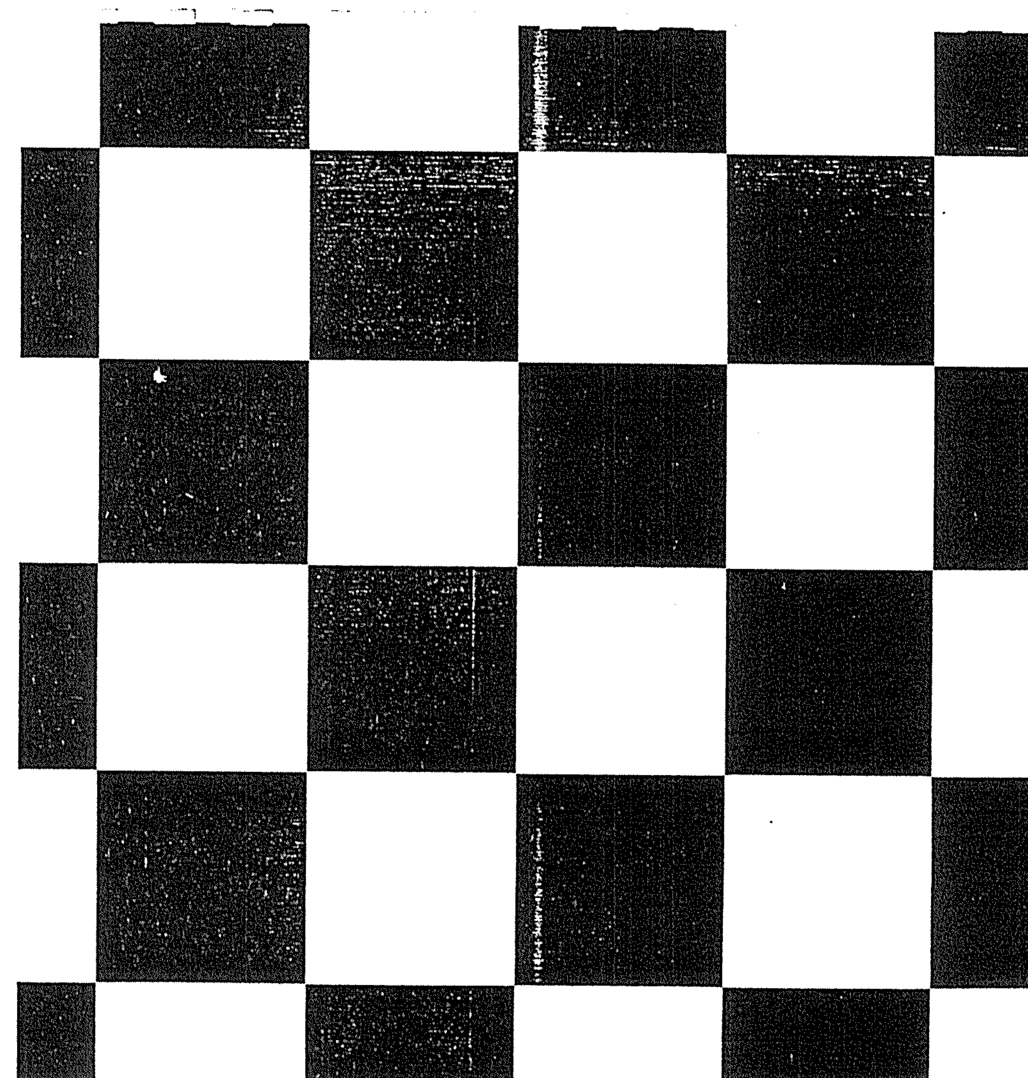
3 by 3 pixel Checker pattern
rotated 40 ^{o}



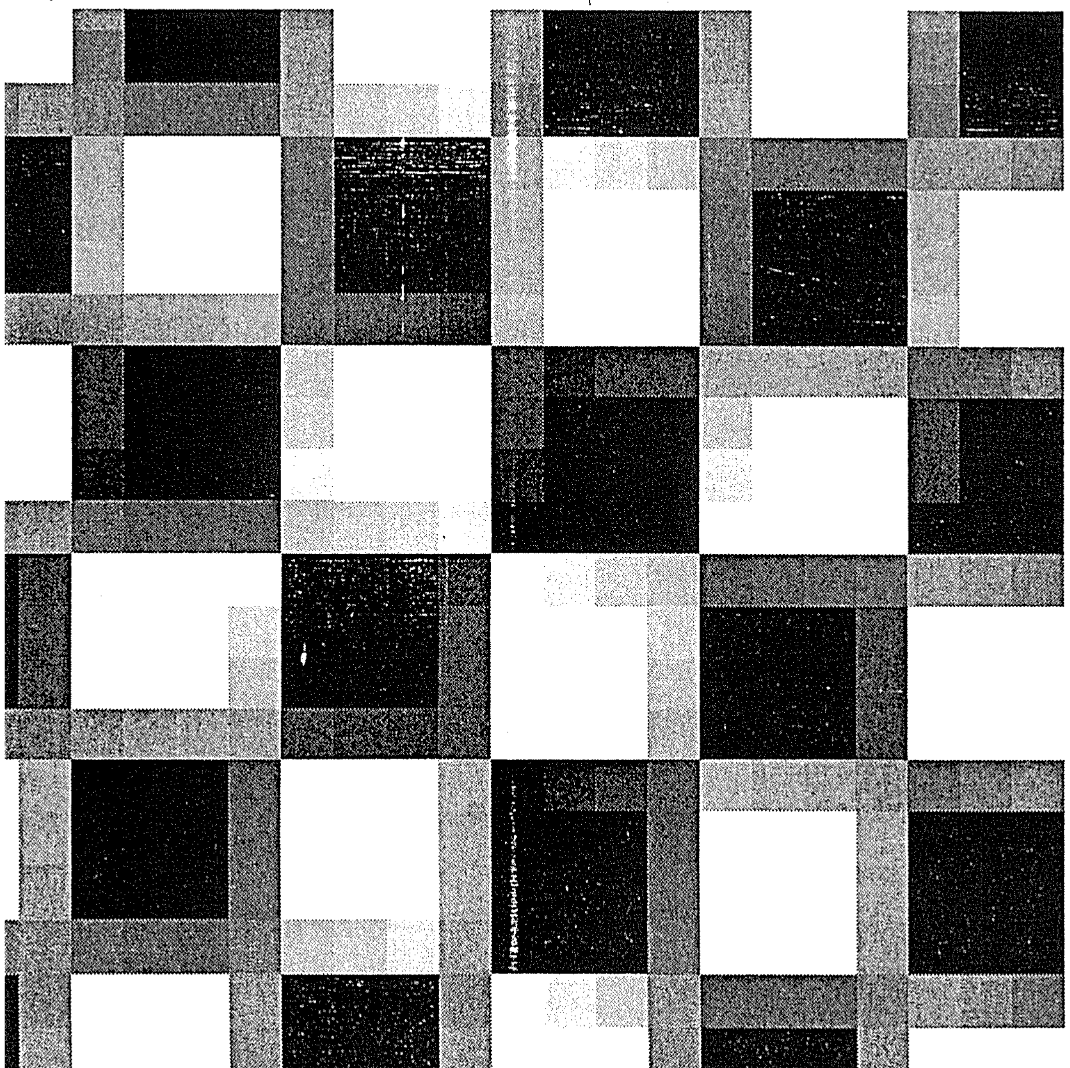
4 by 4 pixel Checker pattern
rotated 1 o



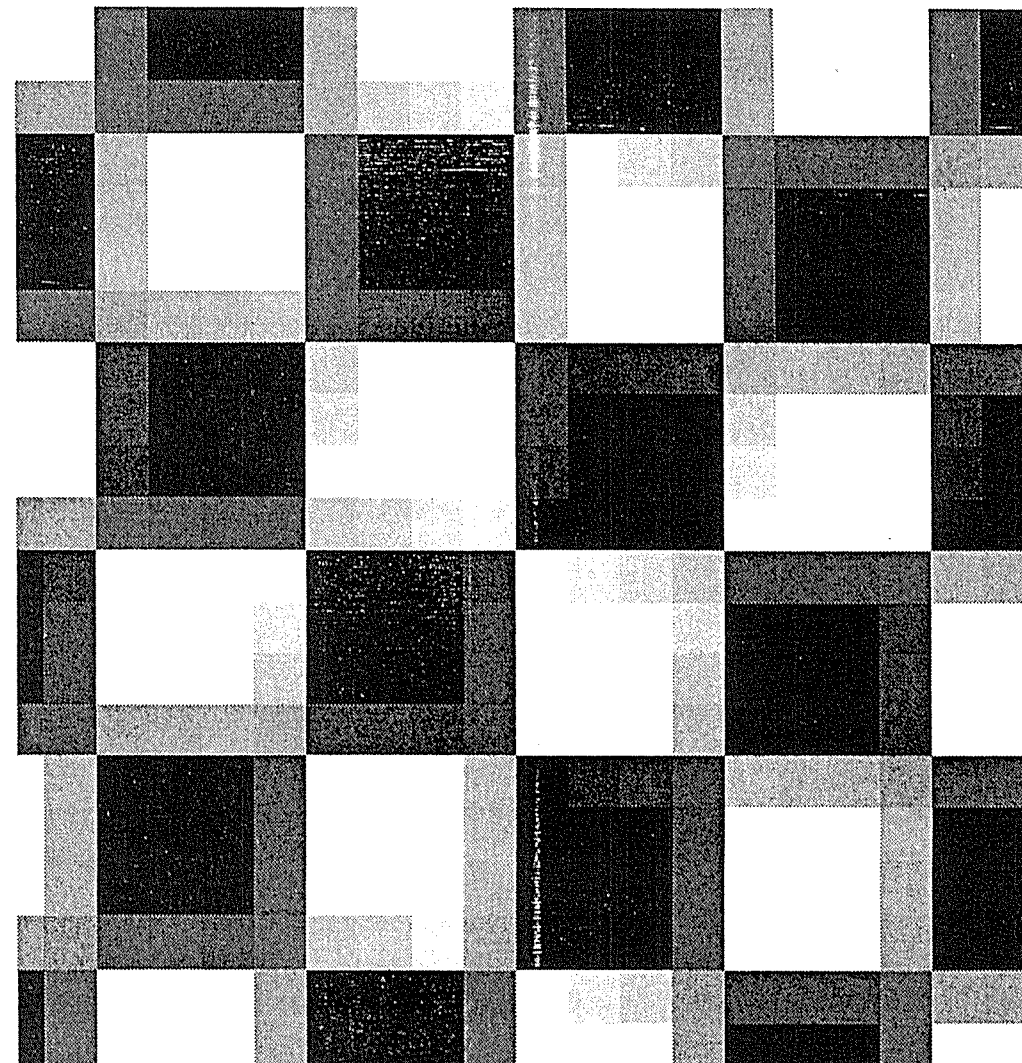
4 by 4 pixel Checker pattern
rotated 0 o



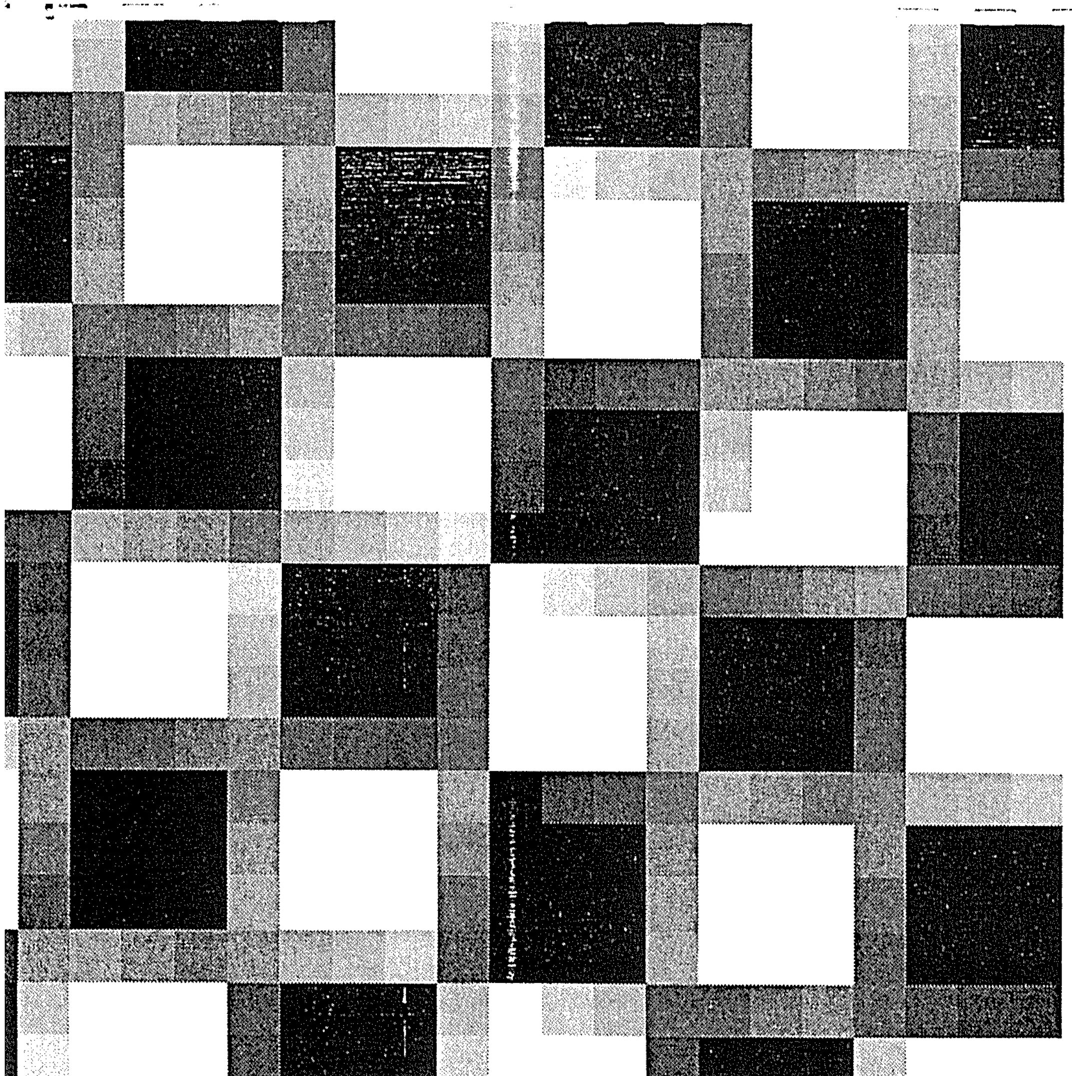
4 by 4 pixel Checker pattern
rotated 3 ^{o}



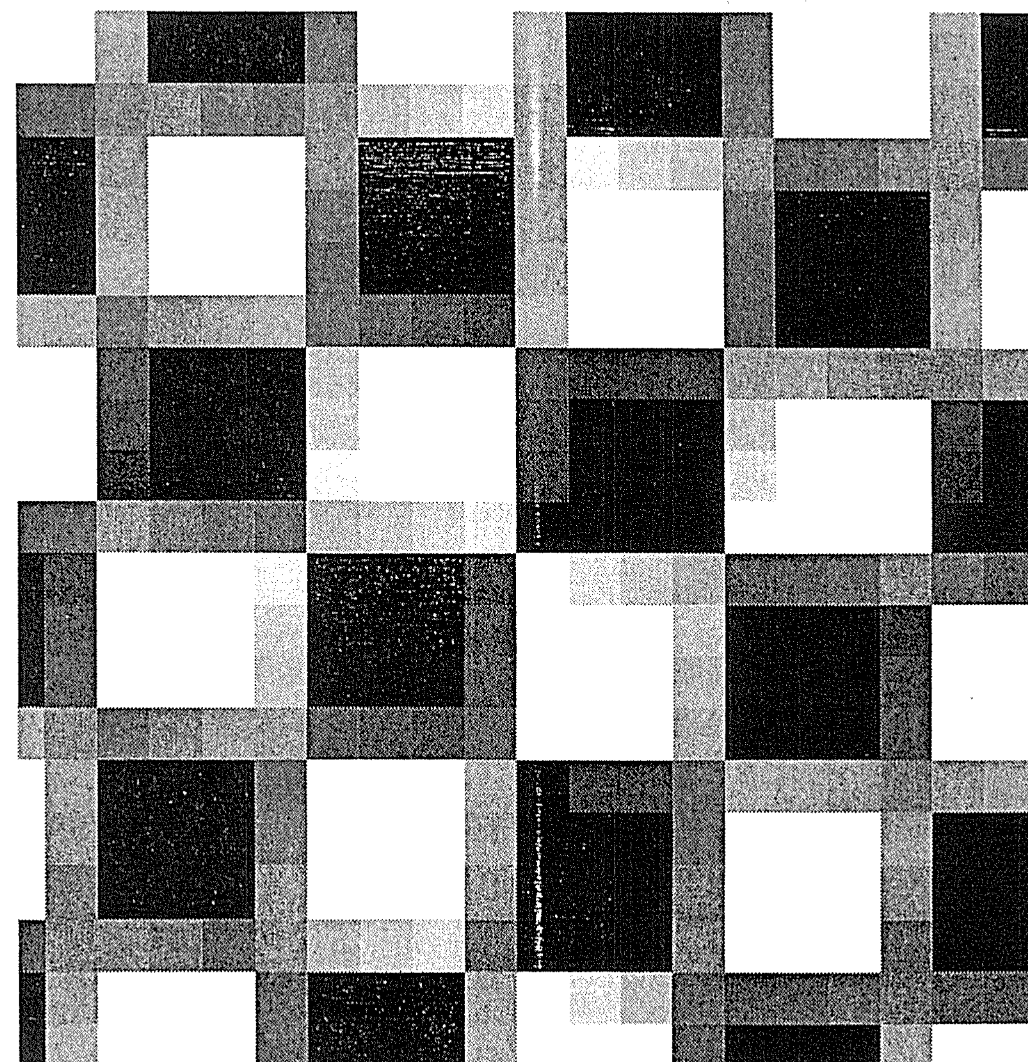
4 by 4 pixel Checker pattern
rotated 2 ^{o}



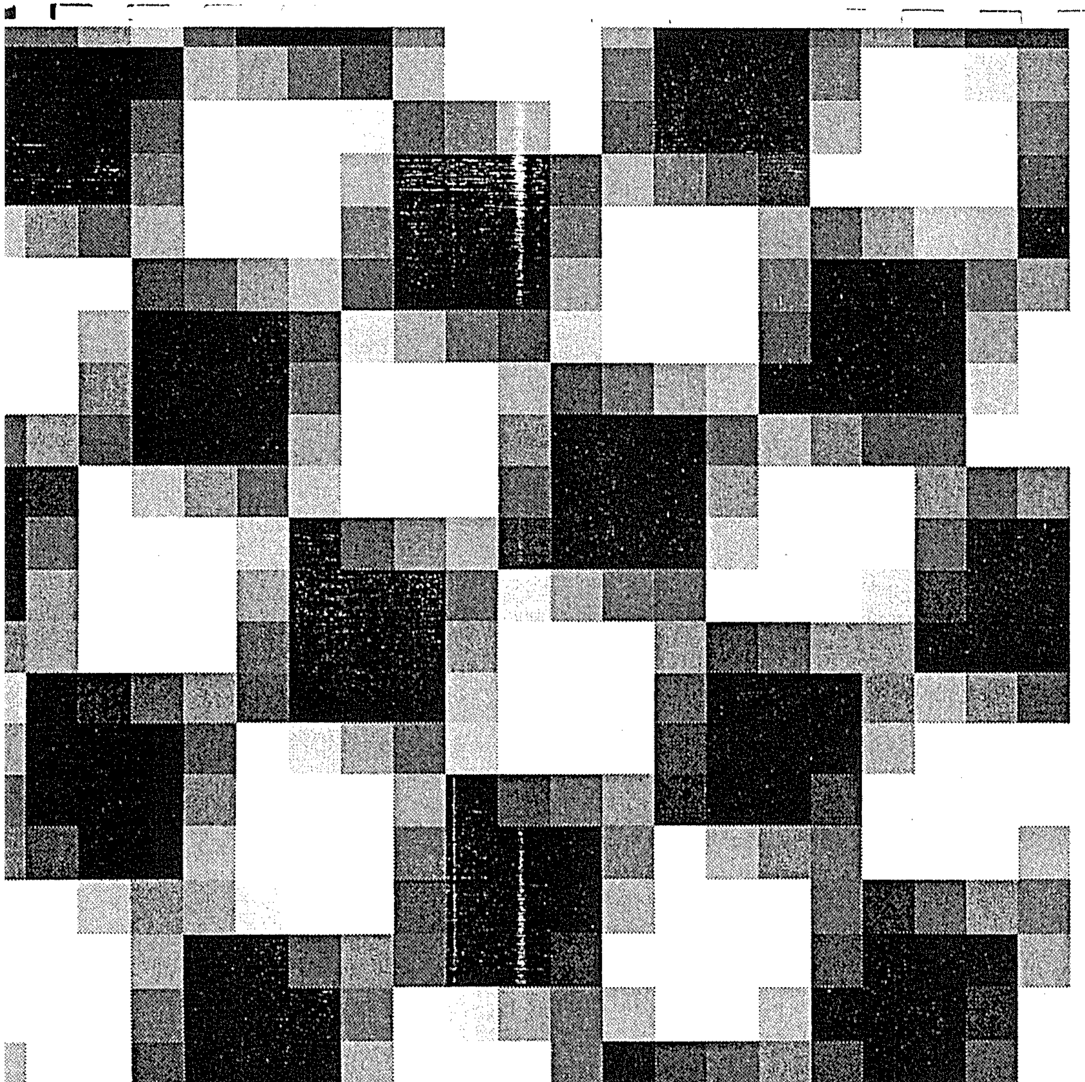
4 by 4 pixel Checker pattern
rotated 5 ^{o}



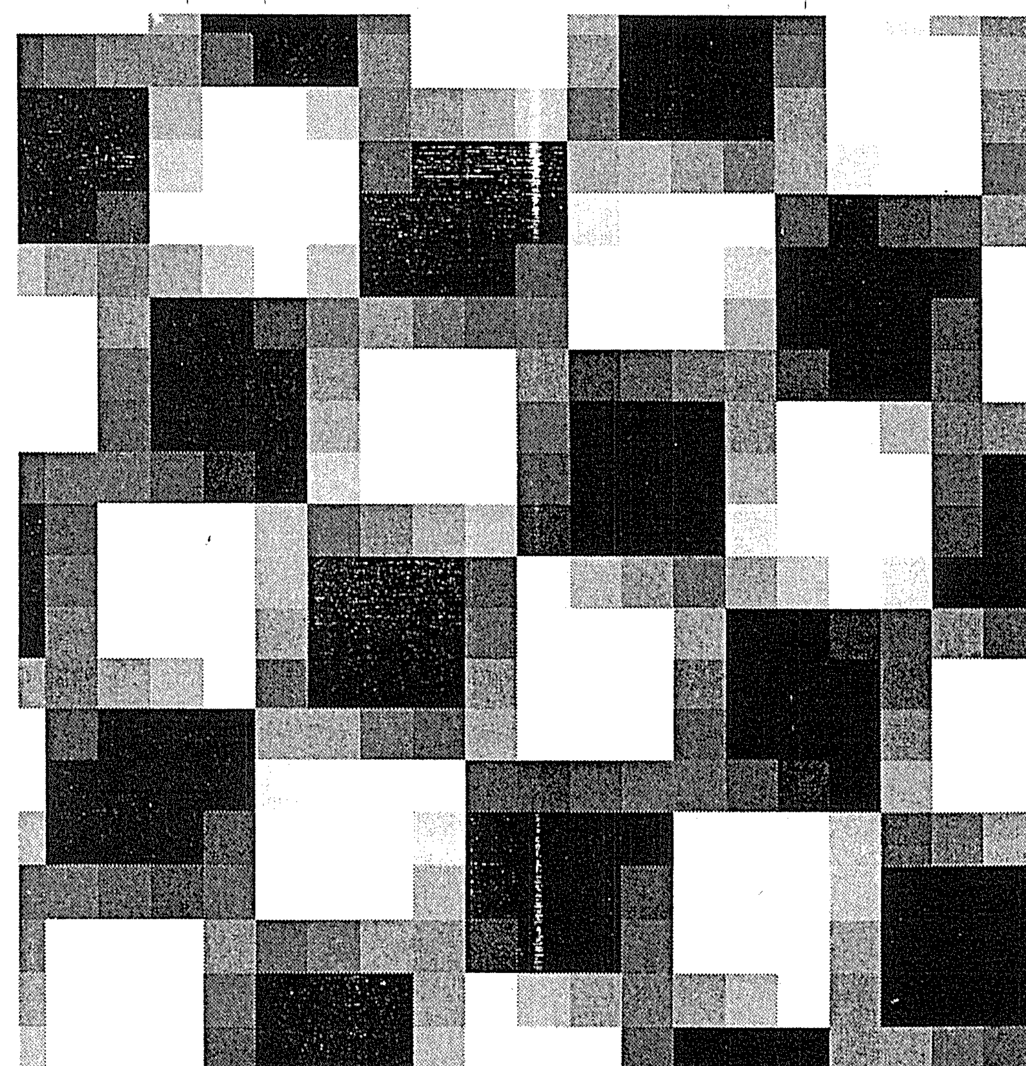
4 by 4 pixel Checker pattern
rotated 4 ^{o}



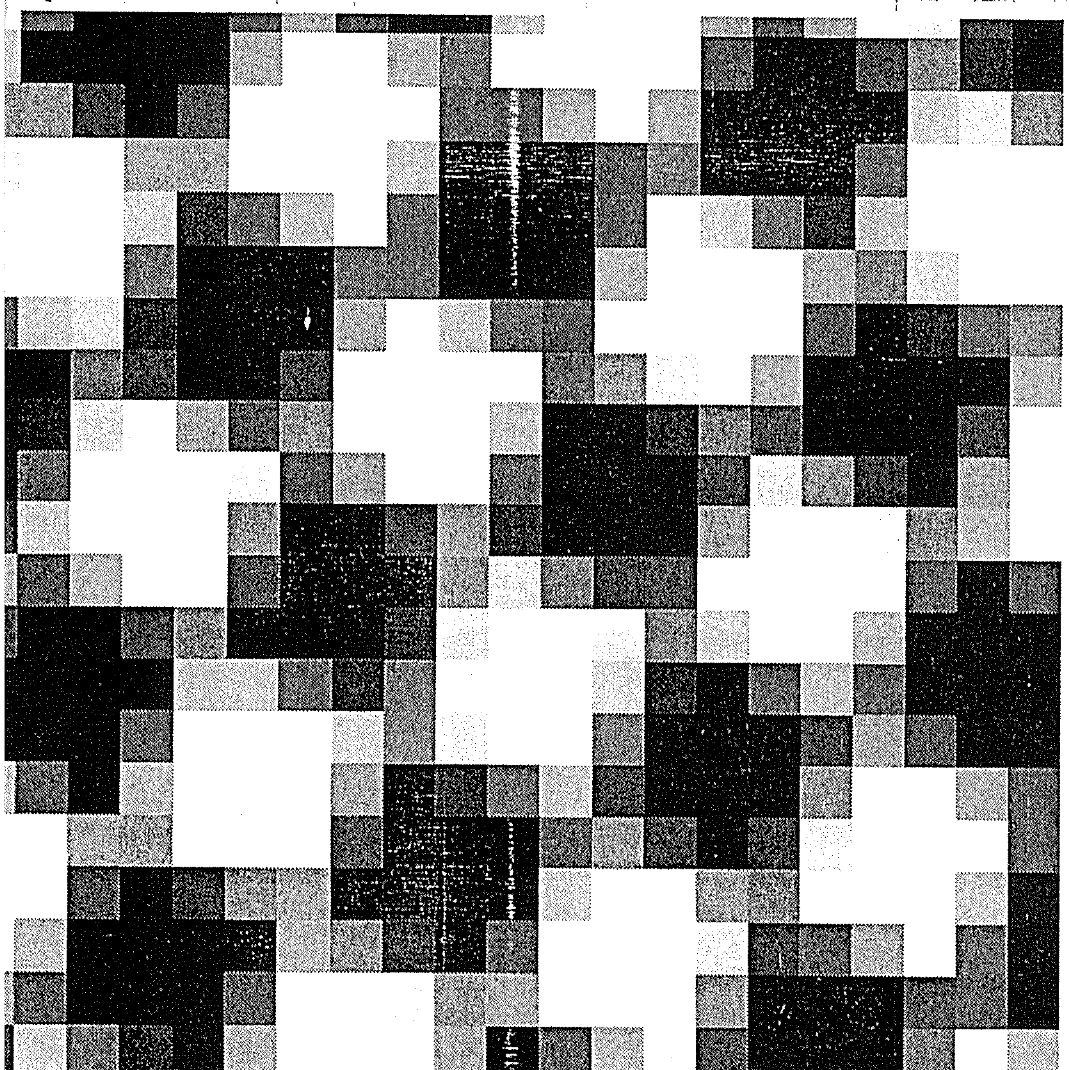
4 by 4 pixel Checker pattern
rotated 15 ^{o}



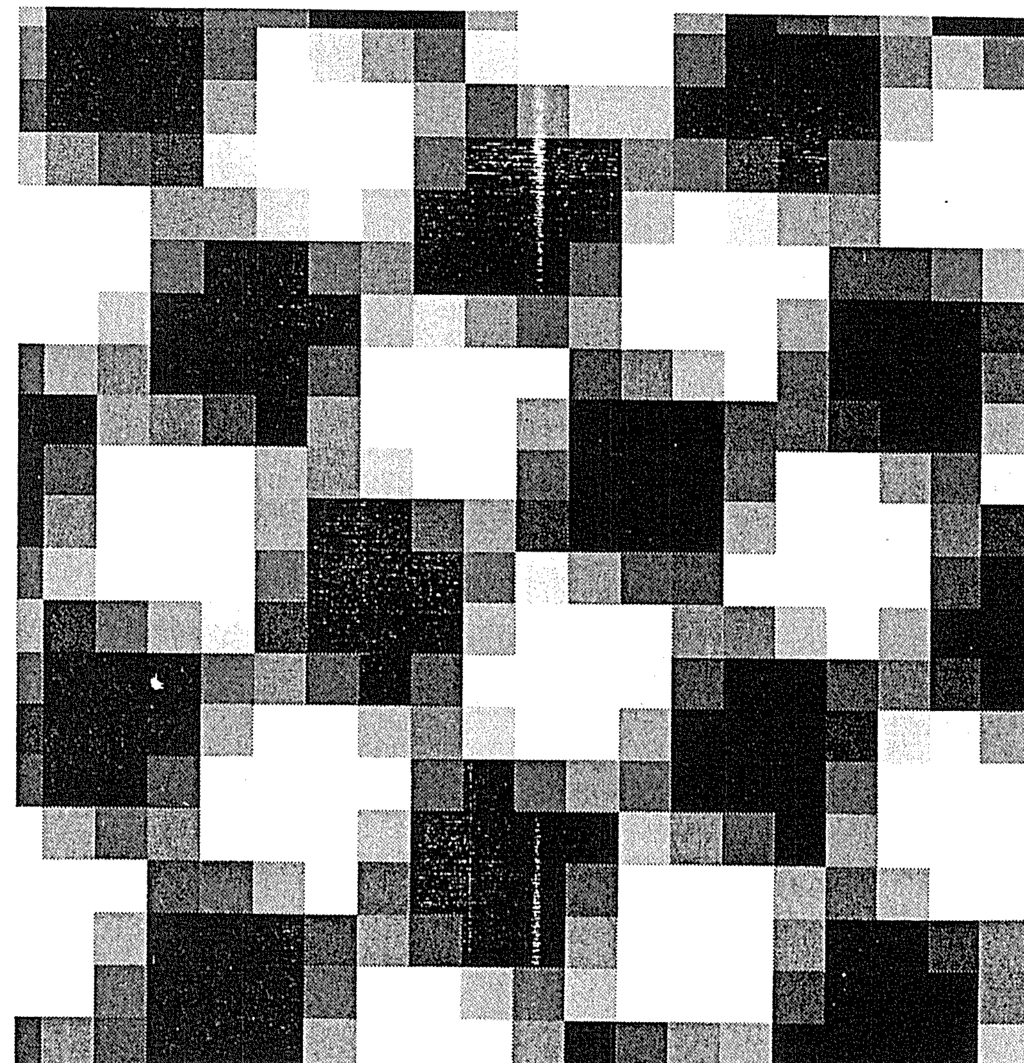
4 by 4 pixel Checker pattern
rotated 10 ^{o}



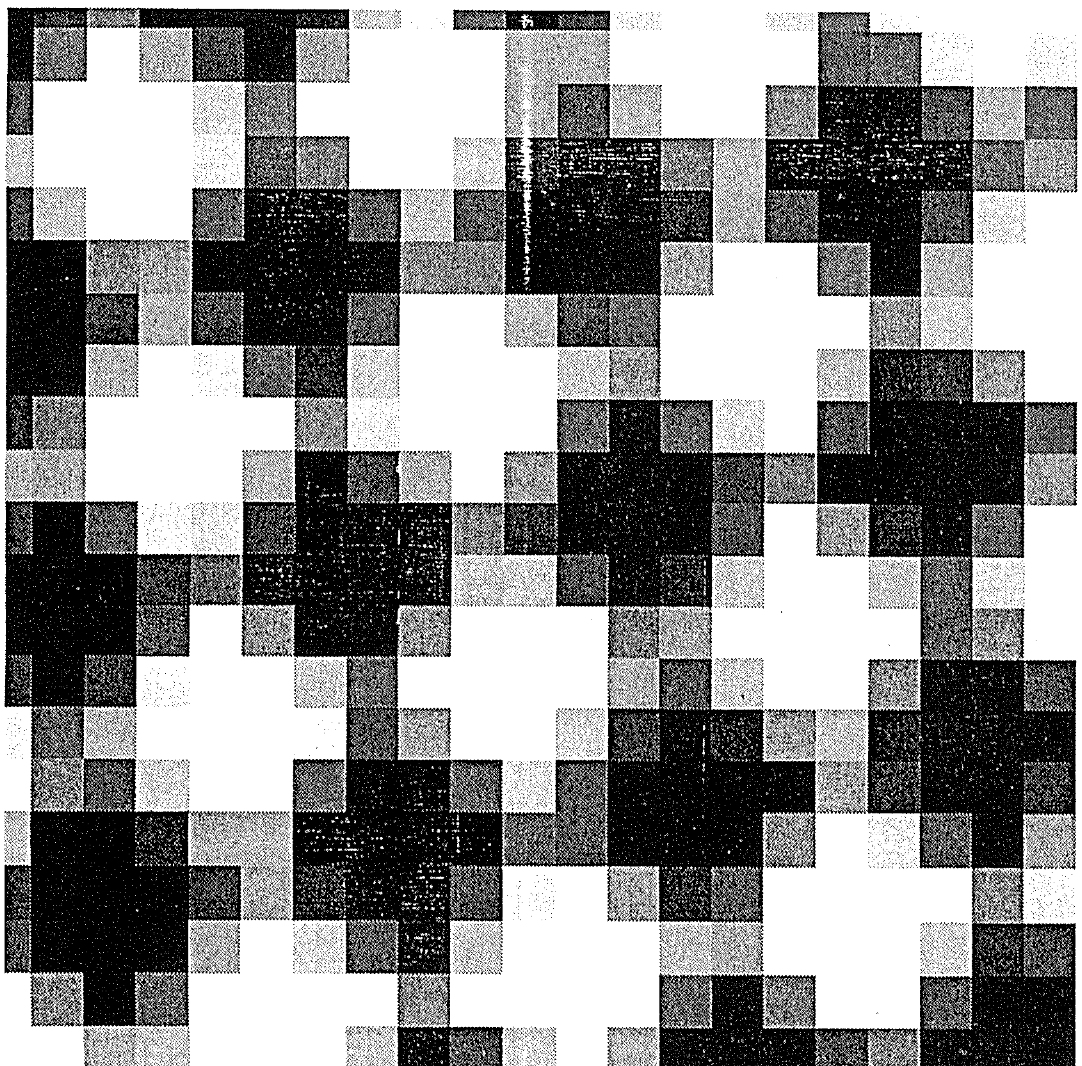
4 by 4 pixel Checker pattern
rotated 25 ^{o}



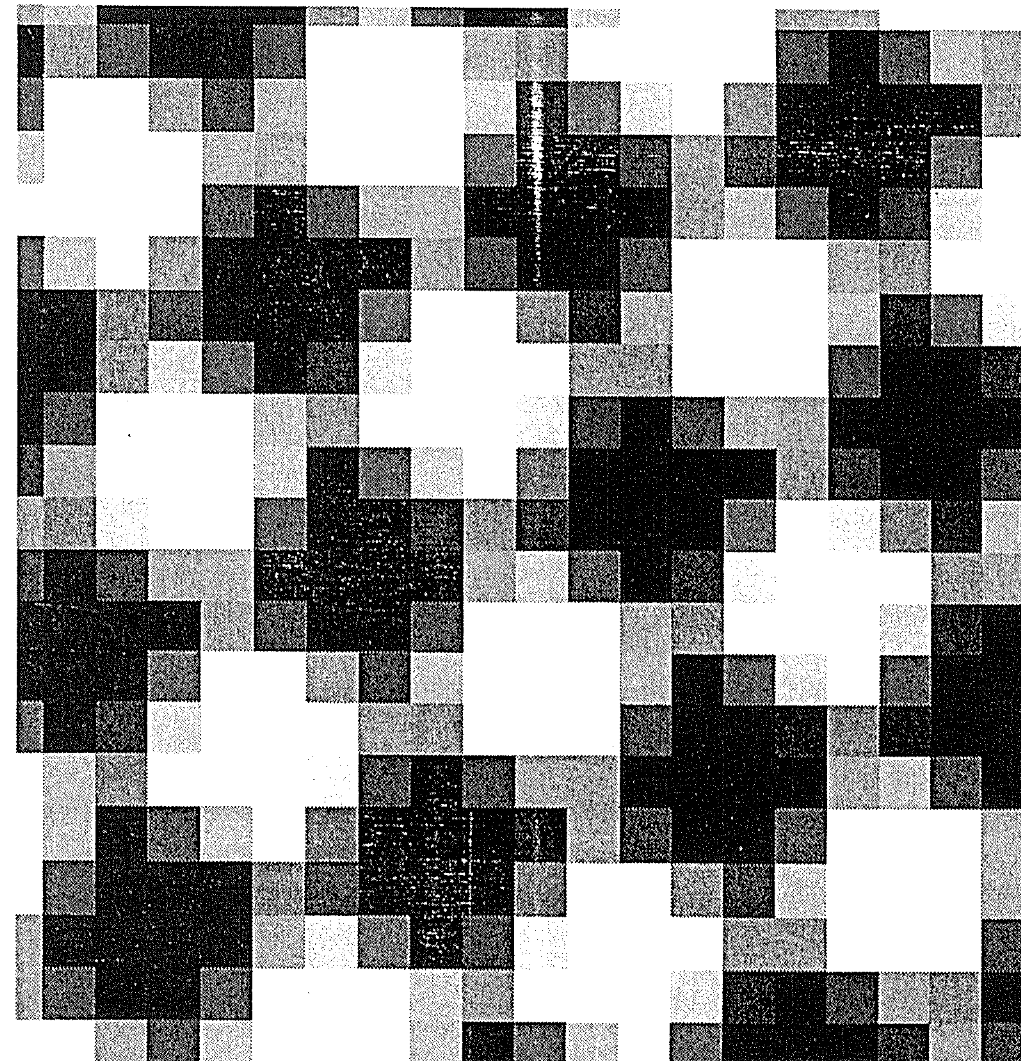
4 by 4 pixel Checker pattern
rotated 20 ^{o}



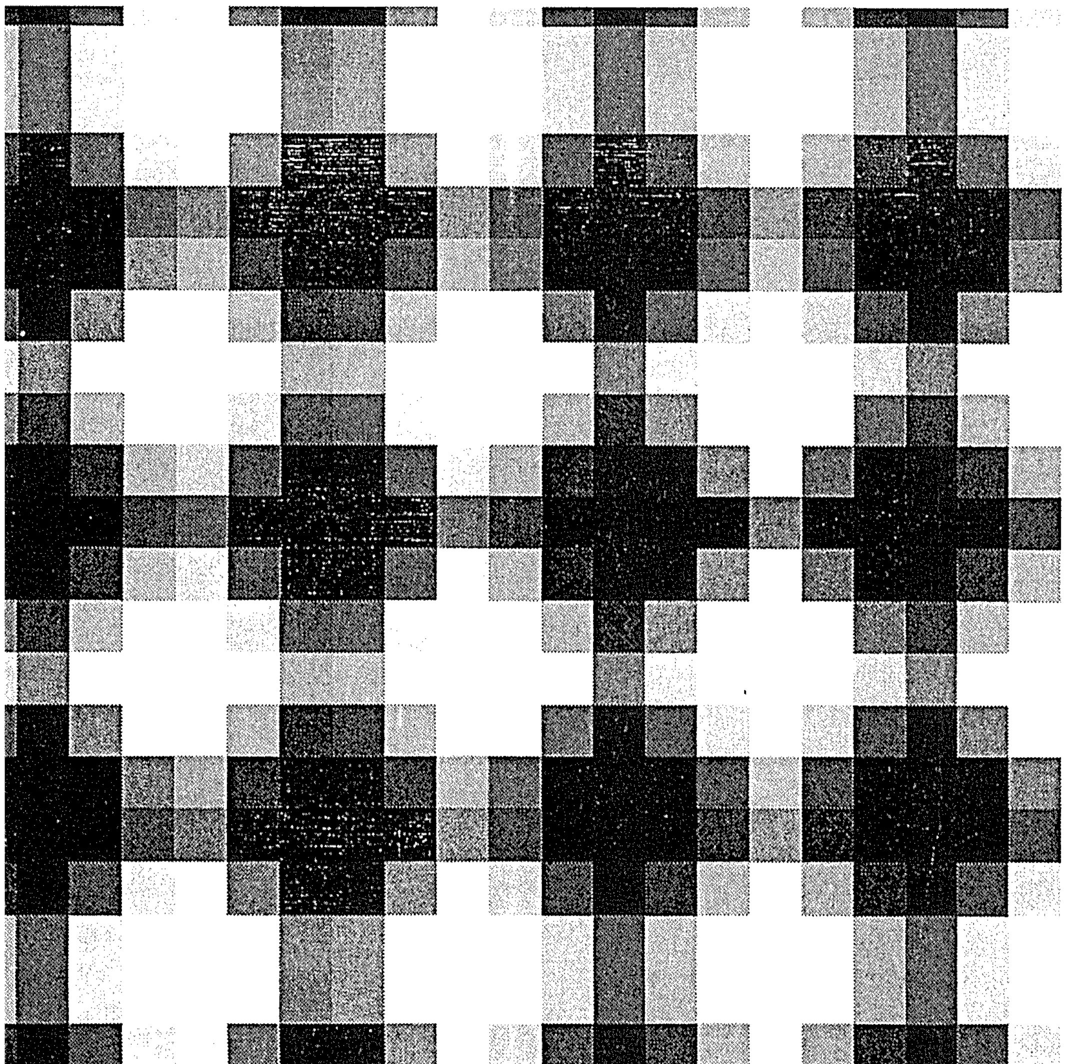
4 by 4 pixel Checker pattern
rotated 35 ^{o}



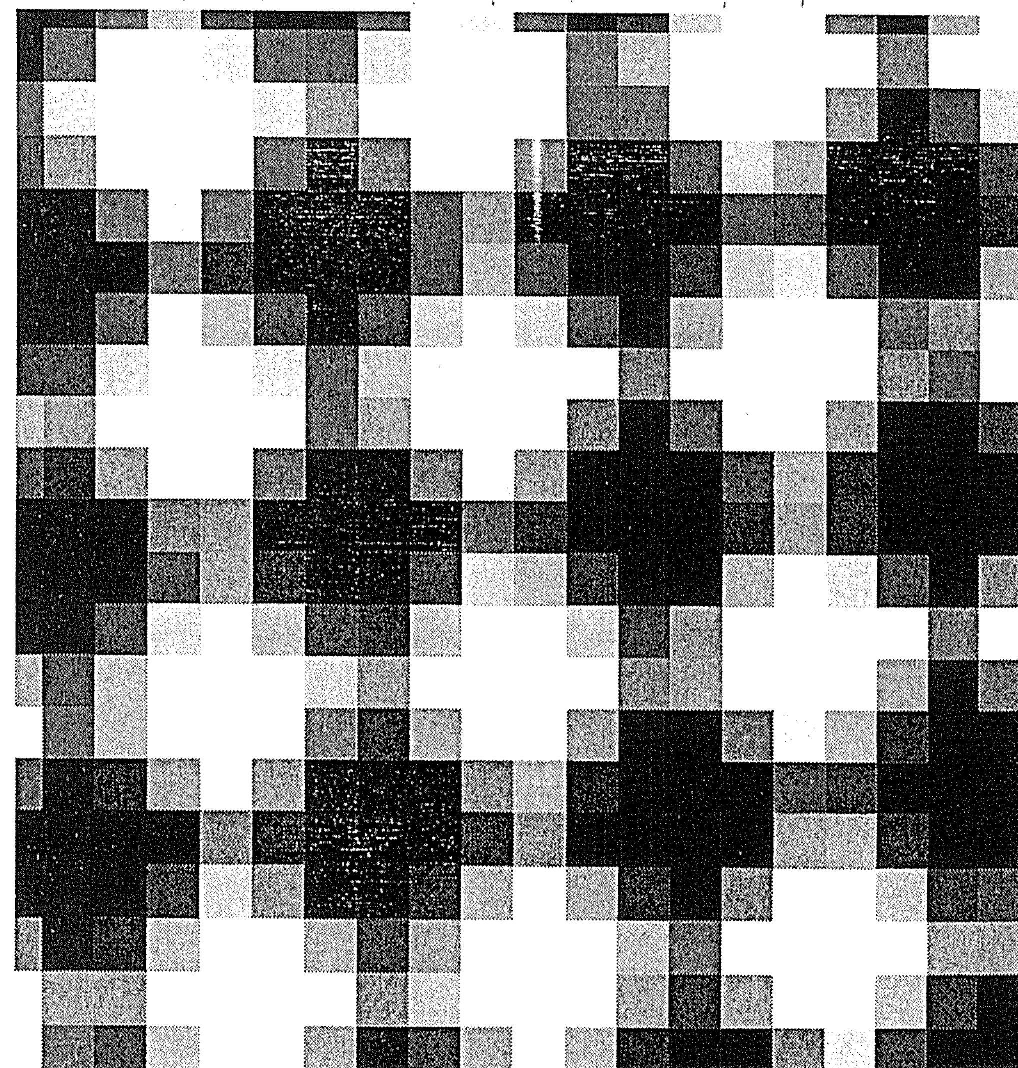
4 by 4 pixel Checker pattern
rotated 30 ^{o}



4 by 4 pixel Checker pattern
rotated 45 °



4 by 4 pixel Checker pattern
rotated 40 °



Appendix C
Derivation of simulation of sub-pixel effect

A frame of a video is made up of a grid of square pixels where a pixel is an area of uniform intensity. A pixels intensity value is between 0 (black) and 255 (white). A pixel within a frame may be indexed by an x and y coordinate. The pixel at the coordinate (0,0) is in the top left hand corner of the frame. Using this notation an equation can be written for the new intensity values of a pixel after a translation and therefore the sum of absolute difference statistic value for a window that has been translated.

For x translation only :

If tr_x is translation undergone between 0 and 1 pixels in the horizontal direction, w is the width of a pixel and $Iold_{(x,y)}$ is the old intensity of a pixel at the position (x,y) then the new intensity, $Inew_{(x,y)}$, of the pixel at location (x,y) is given by:

$$Inew_{(x,y)} = \frac{Iold_{(x-1,y)} * tr_x * w + Iold_{(x,y)} * w * (w - tr_x)}{w^2}$$

The change in intensity of the pixel at (x,y) is:

$$\Delta I_{(x,y)} = Iold_{(x,y)} - Inew_{(x,y)}$$

$$= Iold_{(x,y)} - \left(\frac{Iold_{(x-1,y)} * tr_x * w + Iold_{(x,y)} * w * (w - tr_x)}{w^2}\right)$$

$$= Iold_{(x,y)} * \left(1 - \frac{w * (w - tr_x)}{w^2}\right) - \frac{Iold_{(x-1,y)} * tr_x * w}{w^2}$$

$$= \frac{tr_x}{w} * (Iold_{(x,y)} - Iold_{(x-1,y)})$$

The sum of absolute difference statistic is given by:

$$Statistic = \sum_{x,y}^{window} |Ia_{(x,y)} - Ib_{(x,y)}|$$

where $Ia_{(x,y)}$ is the intensity of the pixel at (x,y) in window a and $Ib_{(x,y)}$ is the intensity of the pixel at (x,y) in window b. When simulating the effect of sub-pixel translation window a is from the frame after sub-pixel translation and window b is from the same frame before sub-pixel translation. So for a window size of n the statistic value for an x translation of tr_x is given by :

$$Statistic = \frac{tr_x}{w} * \sum_{x=-n}^{n} \sum_{y=-n}^{n} |Iold_{(x,y)} - Iold_{(x-1,y)}|$$

For y translation only :

If tr_y is translation undergone between 0 and 1 pixels in the vertical direction the new intensity, $Inew_{(x,y)}$, of the pixel at location (x,y) is given by:

$$Inew_{(x,y)} = \frac{Iold_{(x,y-1)} * tr_y * w + Iold_{(x,y)} * w * (w - tr_y)}{w^2}$$

The change in intensity of the pixel at (x,y) is:

The statistic value for a y translation of tr_y is given by :

$$Statistic = \frac{tr_y}{w} * \sum_{x=-n}^{n} \sum_{y=-n}^{n} |Iold_{(x,y)} - Iold_{(x,y-1)}|$$

For translation in x and y direction

If tr_x is the translation in the x direction and tr_y is translation in the y direction then the new pixel intensity can be found from a combination of four pixel intensities in the untranslated frame. If the coordinate of the pixel whose intensity is being calculated is (x,y) then the coordinates of the four pixels whose intensities are being used in the combination are (x1,y1), (x2,y1), (x2,y2), and (x1,y2) (from top left to bottom left) and are given by the following simple rules:

if $tr_x > 0$ $\quad x1 = x - \text{round}(tr_x + 0.499)$
$x2 = x1 + 1$

if $tr_x = 0$ $\quad x1 = x2 = 0$

if $tr_x < 0$ $\quad x2 = x - \text{round}(tr_x - 0.499)$
$x1 = x2 - 1$

if $tr_y > 0$ $\quad y1 = y - \text{round}(tr_y + 0.499)$
$y2 = y1 + 1$

if $tr_y = 0$ $\quad y1 = y2 = 0$

if $tr_y < 0$ $\quad y2 = y - \text{round}(tr_y - 0.499)$
$y1 = y2 - 1$

and if ftr_x is the fractional part of the translation in the x direction and ftr_y is the fractional part of the translation in the y direction the multipliers $M_{(1,1)}$, $M_{(2,1)}$, $M_{(2,2)}$ and $M_{(1,2)}$ (top left to bottom left) are given by :

if $tr_x > 0$ and $tr_y > 0$ then
$M_{(1,1)} = ftr_x * ftr_y$
$M_{(2,1)} = (w_x - ftr_x) * tr_y$
$M_{(2,2)} = (w_x - ftr_x)*(w_y - ftr_y)$
$M_{(1,2)} = ftr_x * (w_y - ftr_y)$

if $tr_x > 0$ and $tr_y < 0$ then
$M_{(1,1)} = ftr_x * (w_y + ftr_y)$
$M_{(2,1)} = (w_x - ftr_x)*(w_y + ftr_y)$
$M_{(2,2)} = (w_x - ftr_x) * -ftr_y$
$M_{(1,2)} = ftr_x * -ftr_y$

if $tr_x < 0$ and $tr_y > 0$ then
$M_{(1,1)} = (w_x + ftr_x) * ftr_y$
$M_{(2,1)} = -ftr_x * ftr_y$
$M_{(2,2)} = -ftr_x * (w_y - ftr_y)$
$M_{(1,2)} = (w_x + ftr_x)*(w_y - ftr_y)$

if $tr_x < 0$ and $tr_y < 0$ then $M_{(1,1)} = (w_x+ftr_x)*(w_y+ftr_y)$
$M_{(2,1)} = -ftr_x * (w_y+ftr_y)$
$M_{(2,2)} = ftr_x * ftr_y$
$M_{(1,2)} = (w_x+ftr_x) * -ftr_y$

where w_x is the width of the pixel in the x direction and w_y is the width of the pixel in the y direction.

The new intensity of a pixel $Inew_{(x,y)}$ at the location (x,y) is given by :

$$Inew_{(x,y)} = Iold_{(x1,y1)} * M_{(1,1)} + Iold_{(x2,y1)} * M_{(2,1)} + Iold_{(x2,y2)} * M_{(2,2)} + Iold_{(x1,y2)} * M_{(1,2)}$$

Appendix D
Limited testing conditions

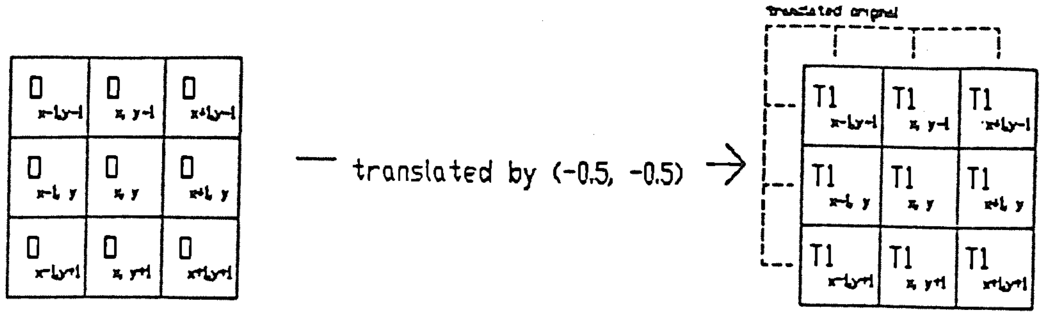
For testing algorithms the ideal situation would be to synthesise image sequences by a repeatable method so all algorithms could be tested under controlled various conditions. Examples of factors that are desirable to vary by controlled known amounts between runs are contrast levels, noise levels, rotation and translation.

For the results in this report translation was synthesised by computer program but the same program was then used to track making the synthesis less independent of the algorithm results. This is an important point when the phenomena of translating or rotating a frame twice is considered (see section below). An alternative method that was considered was to take a very high resolution picture and create a series of frames by extracting parts of that picture and converting it to a lower resolution. This synthesis would not be ideal as in the real world translation could be infinitesimally small where this simulation method results in the smallest translation of about 0.125 pixels. Rotation was also synthesised by software. A further limit of both methods is movement in the z direction can not be synthesised and it is also difficult to simulate independent vertebra translation and rotation. Mechanical translation is possible but the equipment is not presently available at Massey University and is very expensive. The best way to control contrast levels and general image quality of test sequences may be software window filters.

The final conclusions made in this report were based on tracking carried out on an unsynthesised video. The correct locations were calculated by superimposing an outline of the vertebra over each frame and rotating and translating the computer estimate to a best match by eye. To enhance results the moused frame with the fitted outline was also on each screen for an immediate comparison.

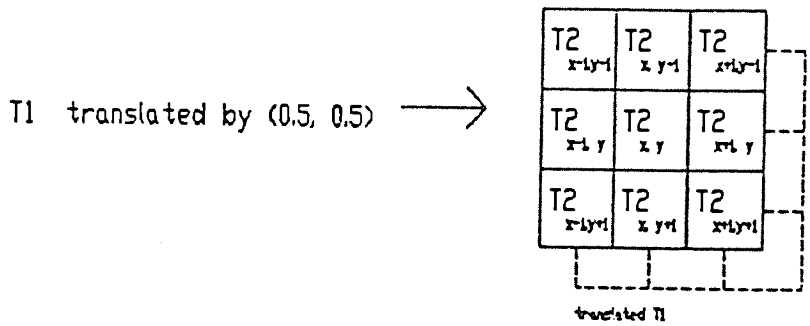
Affect of two way translation

When a frame is translated by a specific vector, using a computer program, then the calculated frame is translated by the reverse of that vector, by the same program, then if a sub-pixel vector component was involved, the final calculated frame is not identical to the initial frame. The reason for this is that after a translation the new pixel value is a combination of two or four pixel values from the original frame. Which pixels and whether it is two or four depends on the translation vector. When this translated picture undergoes the reverse translation the new pixel values are a combination of two or four of the translated pictures pixels so instead of reverting back to the original it is a combination of a number of the originals pixels. An example of this is shown in Figure D-1.



$$T1_{x,y} = 0.25*(O_{x,y} + O_{x+1,y} + O_{x,y+1} + O_{x+1,y+1})$$

Now if this calculated T1 picture undergoes the reverse translation :



$$\text{Then} \quad T2_{x,y} = 0.25*(T1_{x-1,y-1} + T1_{x,y-1} + T1_{x-1,y} + T1_{x,y})$$

$$= 0.25*[\, 0.25*(O_{x-1,y-1} + O_{x,y-1} + O_{x-1,y} + O_{x,y}) + 0.25*(O_{x,y-1} + O_{x+1,y-1} + O_{x,y} + O_{x+1,y}) + 0.25*(O_{x-1,y} + O_{x,y} + O_{x-1,y+1} + O_{x,y+1}) + 0.25*(O_{x,y} + O_{x+1,y} + O_{x,y+1} + O_{x+1,y+1}) \,]$$

$$= 0.0625*(O_{x-1,y-1} + 2O_{x,y-1} + 2O_{x-1,y} + 4O_{x,y} + O_{x+1,y-1} + 2O_{x+1,y} + O_{x-1,y+1} + 2O_{x,y+1} + O_{x+1,y+1})$$

So instead of reverting back to $O_{x,y}$ as expected instead it is a combination of $O_{x,y}$ and the surrounding pixels.

Figure D-1 An example of the affect of two way translation