ADAPTIVE MACHINE VISION
ANNUAL REPORT
FOR SDI/IST

ONR Contract N00014-86-C-0601

TABLE OF CONTENTS

1.0 Introduction

This report covering the period of Jan 1988 to Dec 1988 contains the results of research on two topics, the adaptive photosensor array model and initiation of target tracking in a high clutter, high target density environment.

The photosensor model was originated by Michael H. Brill as a model for a biological retina. However, the adaptivity of the model retina's spatial and temporal resolution to light level, and the model retina's adaptive contrast sensitivity would be advantageous in SDI sensors, which must maintain function in the face of large, sudden and non-uniform light level changes.

The retinal model is not tied to any particular electronic technology, and could conceivably be implemented in silicon with either bipolar or MOS technology. Our preliminary investigation of implementation issues is presented in Section 2.0.

The problem of tracking multiple targets occurs not only in SDI but also in natural and robotic vision. In psychology the problem is called the "correspondence problem" because in successive looks at a complex moving scene, the visual system must find corresponding objects in each look to perceive the objects as moving. Exactly how the visual system makes this correspondence is unknown. However, any animal that moves solves the problem, because nearby objects have higher apparent velocities than distant objects. For example, birds must solve the correspondence problem in order to land on a tree branch.

When there is a large number of moving targets in a scene, as in SDI midcourse scenarios, solving the correspondence problem is a major computational burden. There are three reasons for the severity of the problem. One is the large number of pixels that must be monitored for targets. Another is the high target density and high density of false targets introduced by background clutter. The third reason is the long period, typically a few seconds or more, between looks at the scene. The large number of pixels that must be monitored implies scanning, as opposed to staring, focal plane arrays. The re-visit time between successive scans by a given target (sometimes called the frame period) is too long for targets A, B, C etc. on one frame to be unequivocally identified on the next frame. The presence of noise events that mimic targets on both frames further confuses the situation. In the sensor industry, this correspondence problem is known as "frame-to-frame association" or as "scan-to-scan correlation". Since the basic problem is solved by natural vision systems, we are applying insights from visual processing to the SDI target tracking problems (Section 3.1).

Performance evaluations are an integral part of any technology development program. We have begun evaluating candidate track initiation algorithms in terms of the ROC methodology. ROC stands for Relative Operating Characteristic; the method was originally introduced to evaluate radar displays; subsequently it has been accepted as a method for evaluating medical diagnostic techniques. This work is presented in Section 3.2. In Section 3.3 we present our work and plans for simulations of tracking algorithms.

2.0 Implementation of An Adaptive Photosensor Array

The adaptive photosensor array model of Brill is illustrated in Figure 1. It consists of a lattice of photosensors which are connected to a resistive network. The resistive network performs spatial averaging. Temporal processing is performed at each photosensor with a push-pull circuit that differences the present and recent past signal levels. The temporal and spatial operations are made adaptive to light level by adaptation of resistor values. See Brill, Bergeron and Stoner, "Retinal Model with adaptive contrast sensitivity and resolution," pp 4993-4998, Applied Optics 26 #23 (1987).

To implement the model as a working circuit, many choices must be made. Is the technology bipolar or MOS? Is the processing distributed at each photosensor, or are the signals multiplexed and processed at a small number of higher speed circuits? Is the processing analog or digital? If analog, is the processing performed in continuous time, or in discrete time steps (sampled analog). Our preliminary answers to the implementation question are provided below in Section 2.1 and 2.2. As a prelude to this discussion we first discuss problems with the conventional approaches to image sensing. In a conventional staring or scanning system, analog signals are fed to an analog/digital converter located very near to the focal plane array. In a staring array, it is necessary to perform parallel-to-serial multiplexing to channel the signals to a common A/D converter. Since detector arrays are imperfect, it is necessary to correct the signals for offset biases and gain variations, either before or after A/D. Typically this is accomplished by recalling from a digital memory the offset bias corrections, and performing subtractions on the serial, analog data stream, and then performing the gain corrections in a similar manner with a Multiplying A/D Converter (MDAC). These high speed, hybrid analog-digital operations must be performed close to the detector

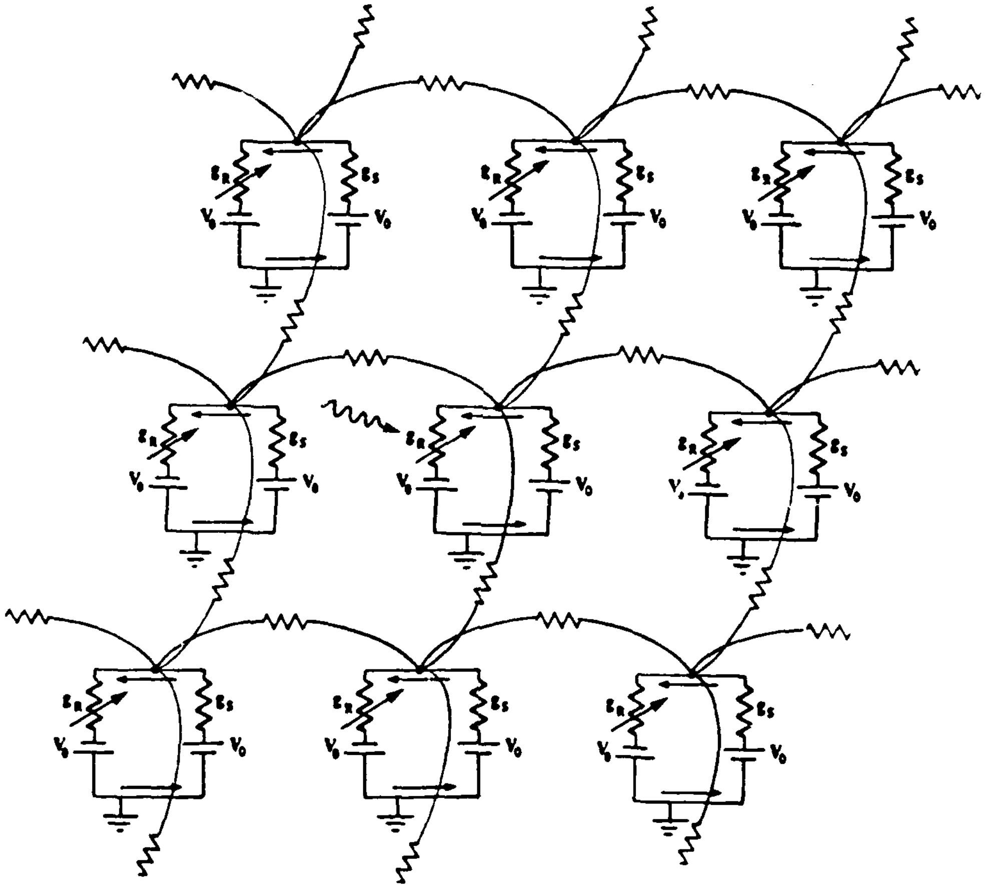


FIGURE 1. A photosensor array with adaptivity to light level. A resistive network provides coupling between photosensors to reduce the impact of photon noise at low light levels. This coupling is reduced at high light levels because the resistivity of each photosensor is reduced as the light level increases. Contrast is maximized at all light levels by the push-pull mechanism of the individual photosensors.

array in order to convert the data to noise immune digital form before it is carried off the sensor gimbal. Otherwise RF pick-up and EMI on the data link would be more likely to corrupt the signal levels. All of this high-speed processing deposits watts of power near the detector array. This is a serious engineering problem for infrared detectors which must be operated within a few tens of degrees Kelvin of absolute zero. Every watt of heat carried away from the focal plane array requires from 100 to 5000 watts of input power to the cooling system. The severity of the problem is indicated by the tens of watts which high-speed A/D converters dissipate. (Note see the Infrared Handbook, Section 15.2 for the ratio of produced cooling power to supplied power for various cooler types. The numbers we quoted above are typical for a Gifford-McMahon cooler operating over a temperature range of 10 to 77 degrees Kelvin.)

The above discussion illustrates the importance for infrared detectors of reducing the bandwidth of the data exported off the chip or off the sensor gimbal. Professor Carver Mead of the California Institute of Technology has identified another reason for performing certain operations on the chip, before A/D conversion. To reduce the bandwidth of the raw image data, delta modulation techniques may be used, which pass along only the changes in the signal values. This is nearly the same as taking the temporal or spatial derivatives of the raw imagery. If the data is available as continuous time signals, a simple RC differentiator circuit or op-amp circuit can perform the differentiation in time. However, if the data is sampled in time steps for A/D conversion, the time step must be small enough to permit accurate numerical differentiation of the digitized samples. The differentiation step is also noise sensitive, because it involves taking the difference of nearly equal quantities. It would therefore be best to perform the differentiation while the signals are in analog form, and then to digitize at a relatively low bandwidth. This appears to be the strategy used by the human eye since graded potentials are involved at the photoreceptor, horizontal cell and amacrine cell levels; only after the differentiation and change detection have been performed does the signal flow reach the retinal ganglion cell, where an action potential (spike) is produced for output along the optic nerve to the higher level processing center in the lateral geniculate nucleus.

We now go on to short discussions of two options for implementation of the photosensor array -- first an unconventional fabrication involving photoconductors, and then an approach using MOS processes.

2.1 Concept for an Adaptive Photosensor Array Circuit Using Photoconductors.

Although standard integrated circuit processes do not incorporate photoconductive materials, we have found that photoconductors provide a simple circuit design. Later we will discuss how the photoconductors might be replaced by MOSFETs biased in the triode region, and how the RC differentiators and integrators in the circuit might be replaced by MOS op-amps. This translation of the photoconductor circuit would then be entirely based upon MOS devices, and should be a practical design. However, the complexity of the MOS circuit obscures the principles of the adaptive photosensor array, which are more clearly visible in the photoconductor version, shown in Figure 2. Since the circuit uses just four types of devices, photoconductors, resistors, conductors and capacitors, it might be feasible to fabricate it with thin-film processes, since photoconductors such as CdS have been used in experimental spatial light modulators, and in commercial photocells for nightlights and cameras.

The circuit is inspired by the Brill retinal model, but uses the principles, rather than the mechanisms of the model. For example in the model retina, the temporal derivative of the light level

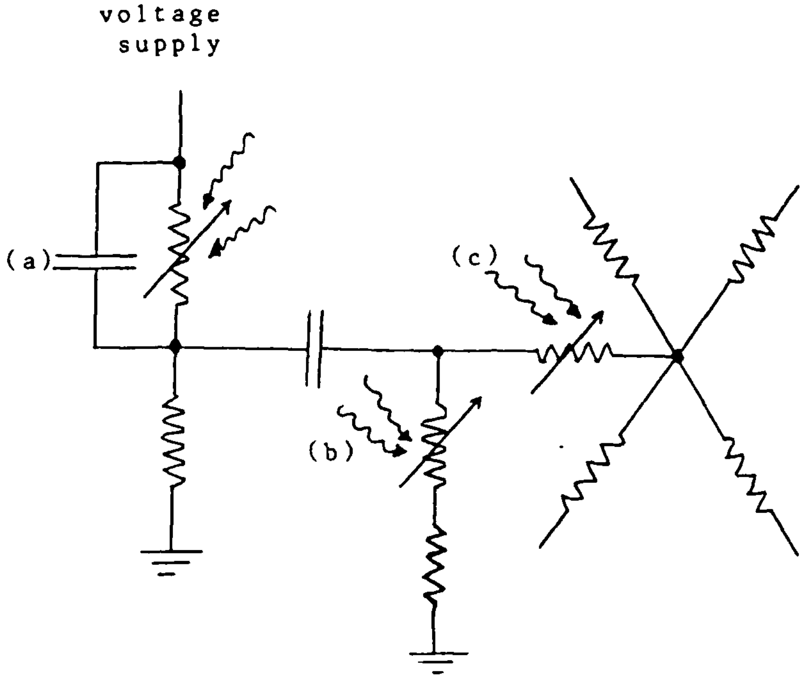


Figure 2. Circuit concept for an adaptive sensor array. One photosensor is shown, the array is coupled spatially through a resistive net that performs spatial averaging at low light levels. The variable resistors might be implemented with photoconductors, or by MOS circuitry. The circuit at (a) provides a voltage division which is ac coupled through the capacitor at (b). Signal integration is provided by the capacitor at (a). The variable resistor at (c) serves to vary the coupling of the photosensor to its neighbors through the resistive network. At low light levels, the network is more conductive than the variable resistor, and so the network provides tight coupling. The opposite is true at high light levels.

is obtained by a push-pull circuit that obeys the chemical kinetics of excited photopigment. From the model, we learn the importance of the temporal derivative, but we obtain the derivative with an RC differentiator in the photoconductor circuit, or an op-amp in the proposed MOS implementation.

The photoconductive circuit concept of Figure 2 requires four devices, which may be fabricated as follows:

1) photoconductors made by vacuum disposition of CdS,

2) resistors made from light-shielded photoconductors to provide a measure of temperature compensation to the design,

3) conductive lines of vacuum deposited of aluminum,

4) capacitors made of vacuum deposited aluminum insulated with silicon dioxide (from the oxidation of polysilicon).

The output of the resistive net is the voltages of each node. Our ultimate goal is to perform local processing on these voltages to extract image features such as local texture, edges, motion, etc. The transmission of image features rather than pixel level values would greatly reduce the bandwidth of the data channel connecting the sensor chip with a higher level image processor. However, at the present time, methods of performing feature detection on the sensor chip are not ready for implementation. We must therefore think in terms of reading out the voltages at each node, and processing for features with a digital or optical computer.

To readout these voltages, we propose either parallel optical readout or serial readout by means of cross-bar x, y addressing of each node. For optical readout, the resistive net would have to be part of a sandwich with the detector array on one side, and the voltage driven spatial light modulator on the other side. For serial readout, the nodes of the resistive net must be switched into contact with a voltage sensitive readout device such as an FET. A crossbar system, such as that shown in Figure 3 might be used to address the MOSFET switches +hat connect each network node to the readout FET. This readout system could be fabricated as an underlying layer of the photoconductor array.

2.2 Implementation of an Adaptive Photosensor Array with MOS Processes

In the previous section, we have suggested a straightforward approach to implementing a resistive network, by using a light shielded photoconductor strip. Other passive materials should work, for example, polysilicon or ion implantation strips. Indeed, these two approaches are used to fabricate resistors on integrated circuit analog-to-digital converters. See for example, Bernard Loriferne, Analog-Digital and Digital-Analog Conversion, pp 165-170 (Heydin and Son, Ltd., 1982). However, the low resistance values obtainable, and the difficulties in obtaining precise resistor ratios has led the analog processing community to seek other techniques.

Since an MOSFET biased in the triode region provides appropriate resistive values, this is an attractive option. The circuit of Figure 2 uses photoconductors for light controlled resistors and fixed resistors. It is therefore possible to use this basic circuit with MOS processes, provided that the fixed resistors are implemented as biased MOSFETS, and the photoconductor resistances are implemented by arranging a light sensitive MOS device to control the bias on a MOSΓET. Figure 4 shows the biasing of a MOSFET to provide a resistor. The nonlinearity of this circuit may be greatly reduced by clever use of symmetry as discussed by Y. P. Tsividis in Design of MOS VLSI Circuits for Telecommunications, Chapter 11, "Continuous-Time Filters," Section 2, "Processor Continuous-Time Filters," (edited by Y. P. Tsividis and P. Antognetti, Prentice-Hall, Inc. Englewood Cliffs, NJ, 1985).

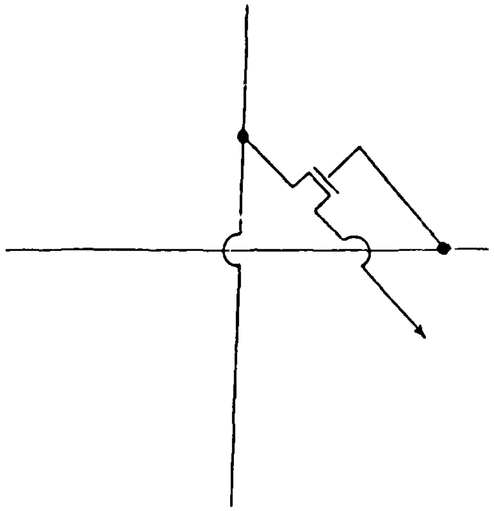


Figure 3. An x-y addressed crossbar arrangement may be used to serially readout the voltage at nodes the resistive net. A high input impedance readout device such as an FET is needed to avoid disturbing the network voltages.

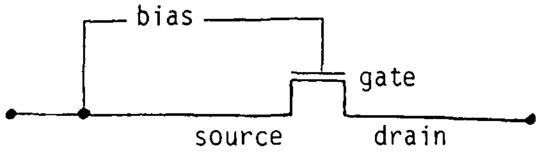


Figure 4. An MOSFET biased in the triode region operates as a resistor.

In the conceptual photoconductor circuit, temporal differentiation and integration is obtained with simple RC circuits. However these circuits are lossy. Since MOS fabrication provides op-amps as a basic building block, these RC filtering principles may be carried over to the MOS *implementation, without incurring* the undesired signal losses of simple RC filters.

In the foregoing discussion of the photosensor array, a square array has been assumed. Section 5, Appendix on Image Sampling Theory for Hexagonal and Square Sampling Arrays, shows that the sampling theorem may be satisfied with approximately 0.866 of the pixels needed with square pixels if hexagonal pixels are used. The resistive network concept carries over to hexagonal arrays, and so the implementation thoughts presented above are applicable to hexagonal photosensor arrays. Not only is the pixel count reduced, but this reduction very likely will result in a reduction in subsequent data processing. Moreover, the acquisition range for detection of a point target should be increased, because of the increased size of hexagonal pixels compared to square pixels (with both the hexagonal and square sampling arrays satisfying Nyquist). The ratio in area is approximately 1.15.

3.0 SDI Tracking

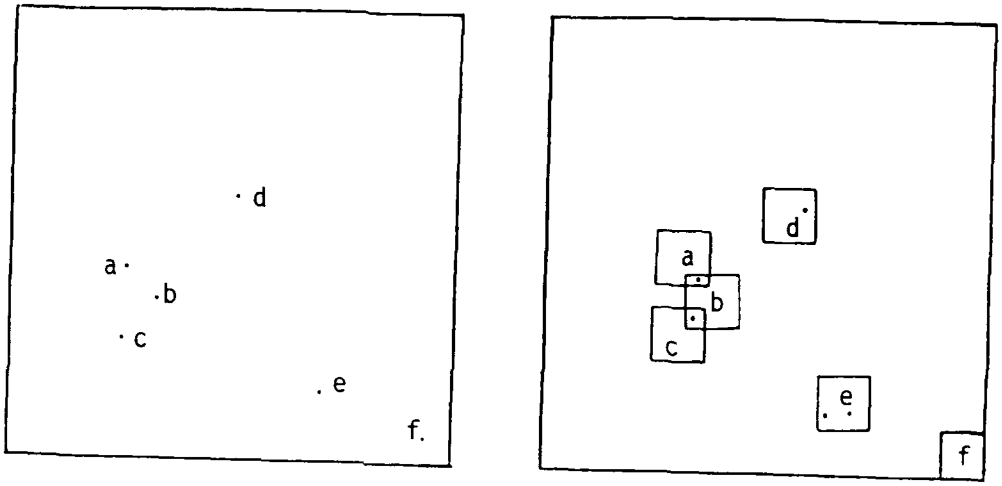
3.1 Application of Early Visual Processing to SDI Target Tracking

Statement of the Problem

Each of the three phases of SDI defense poses a different tracking problem. In boost phase, the targets are bright and easily detected. But the booster plume that is detected is centered 100's of meters away from the hard body we must intercept. The problem is to find and track the hard body. In midcourse, the problem is the sheer number of targets to be tracked. This is the problem we have focused on. The re-entry phase tracking problem is difficult because high altitude nuclear bursts may occur. The ballistic warheads are not disrupted by the glowing and highly structured (striated) disturbed atmosphere created by a high altitude nuclear burst, but the tracking system must operate in the presence of bright background clutter.

Resolution of the Midcourse Track Indication Problems with Early Visual Processing

A scanning sensor must be used to acquire targets during mid course, because staring sensors are impractical over the large field of regard that must be covered. Since individual targets are not tracked continuously, frame-to-frame correlation is performed to sort out consistent trajectories among all of the possible associations between target detections on successive frames. See Figure 5. The problem is complicated by the presence of false target detections, and missed detections. To remain non-committal about the threshold exceedances used to detect targets, we shall use the word "hit" to refer to a detection. A hit may actually be a false alarm.



frame at t_1 frame at t_2

Figure 5. To find trajectories of targets found on successive image frames, the targets (a through f) detected at t_1, must be associated with the detections at t_2. Since the velocities of targets are unknown initially, large "windows" are placed around their original locations to make the associations. The association of target d can be made, since only one detection occurs inside the window for d. However, targets in overlapping windows (a, b and b, c) cannot be unambiguously assigned with detections at t_1. Other problems are noise events, new targets, and targets moving off the frame.

Analysis shows that the number of candidate tracks to be sorted out in the frame-to-frame correlation process grows as the hit density to the 3rd or higher power. If the hit density is low, three successive frames suffice to identify most of the inconsistent tracks, because the first two frames provide an estimate of the hypothetical target's angular rate, and the third frame will either show a target close enough to the expected position or not. But at higher hit densities, the presence of one or more hits within the boundaries of a consistent track forces the examination of additional image frames before a track is confirmed. If m frames must be jointly processed to confirm tracks, the processing grows as the hit density to the mth power, when the hits are individually processed.

The midcourse targets are not completely independent, however. Each post boost vehicle may release hundred of targets, mostly decoys. This leads to a natural clustering of the targets which can be exploited to reduce the processing load for track initiation. It may even turn out that the decoys fill out gaps between RVs and help to distinguish the target clusters arriving from each post boost vehicle, so that the decoys help, rather than hinder track initiation processing!

In human vision, motion and depth perception is performed at an earlier processing stage than pattern recognition. Segregation of imagery by motion is an every day experience. We can perceive a flock of birds and gauge their aggregate flow without concentrating on individual birds. If two flocks of birds criss-crossed we would have no difficulty following one flock's motion. Yet this is exactly the sort of target tracking situation that is stressing for frame-to-frame correlation algorithms that work at the individual target level.

Using random dot stereo pairs, which offer no monocular cues, Julesz showed that stereopsis can also occur before pattern recognition. We seek to endow SDI trackers with a similar capability.

One can conceive of other vision-like image segregation processes to assist track initiation, for example spectral classification, in analogy to color, or classification by target brightness or polarization. In re-entry phase, spectral pre-processing may be useful for clutter reduction in the presence of a nuclear disturbed atmosphere, for example. We have chosen motion and depth because they fit the midcourse tracking problem well.

The application of vision-like stereo and motion processing to midcourse image frames will benefit the track initiation process in two ways. Stereo processing will allow target complexes to be segregated in depth, so that within a given depth, the target density will be reduced. Subsequent frame-to-frame processing will therefore require less throughput. Motion segregation of target clusters will allow the approximate angular rates of targets within clusters to be estimated before the processing of individual targets is begun. These preliminary target angular rates will be much more accurate than a priori upper bounds. Therefore, the number of chance hits falling within the tolerances about an expected target position on the next frame will be reduced, and so fewer candidate tracks must be followed to identify the consistent tracks.

The proposed track initiation process is thus a combination of early visual processing for depth and motion to find target complexes, followed by conventional target association from frame-to-frame within each target complex. After consistent tracks have been established for targets, precision tracking is performed by a Kalman filter or a Kalman filter operating with the Joint Probabilistic Data Association (JPDA) algorithm. The JPDA algorithm is useful in maintaining track in the presence of criss-crossing tracks. However the computational load required for JPDA prohibits its use for track initiation when there is a high hit density. The JPDA algorithm has the weakness that it does not extract and use the target clustering information provided by motion and depth segregation.

3.2 Performance Evaluation of Candidate Tracking Algorithms

In the following section, ROC calculations are made to evaluate tracking algorithms. The algorithms are reduced to one angular dimension, rather than two (azimuth and elevation), but this should not invalidate the rankings of algorithm performance.

The findings of the evaluations may be generalized by one commonsense conclusion: all of the information should be used. The information available from a scanning sensor is the signal level at each pixel at discrete times t_1, t_2...

Image processing algorithms that search for target streaks in superimposed frame images do not use the time sequence information, and the evaluations quantify the penalty this omission incurs. Another type of information is the signal level. Ideally, the signal strength at each pixel would be used in the track detection processing; however, some algorithms work only with the thresholded frame data, which reduces all candidate target events to a common level.

In the future, we intend to improve the analysis of tracking algorithms by more accurately modelling the image collection characterisics of scanning and TDI (Time Delay and Integration) sensors. Towards this end we have completed an analysis of the operation of a scanning sensor which is designed to satisfy the sampling requirements for a band limited image. This work is presented in Section 4.0, Appendix on Image Collection by a Scanning Sensor.

Performance Comparison of Rival Track Detection Procedures

Any method of validating tracks is vulnerabl to chance "hits" masquerading as a valid track. However, there are wide variations in reliability among rival track validation techniques. At low hit densities, the probability of a chance sequence of hits masquerading as a track will be low, and reliability may not be as important an issue as others, such as ease of implementation. But at high hit densities, reliable track validation is essential.

The simplest track validation procedure is streak detection. Usually this method is applied to an integrating, staring sensor, such as a photographic camera. However, successive image frames from a scanning sensor can be superimposed to approximate the type of imagery provided by a staring sensor with a long integration period. The basis of streak detection is that an actual target will follow a smooth trajectory, and short segments of this trajectory will be nearly linear. Therefore, a line detecting procedure such as the Hough transform technique can be used to detect the track. If *a priori* knowledge of the ballistic trajectory of the targets is available, it should also be possible to offset the individual image frames from a scanning sensor before they are superimposed, so that typical curved tracks are more or less straightened out into linear tracks. The Hough transform procedure integrates image points, i(x,y) that fall along straight lines. These straight lines may be parametrized by their slope, m and y axis intercept, b:

$$H(m,b) = \int dx \int dy \; i(x,y) \; \delta(y-mx-b).$$

The Dirac delta function $\delta(y-mx-b)$ picks out only those points in the image which lie on the line with slope m and y intercept b to form the Hough transform value H(m, b). If there is a streak along this line, the Hough transform value will be above average, so that a threshold, T, may be selected to declare tracks whenever $H(m,b) \geq T$.

The Hough transform is attractive from the standpoint of implementation. For example, a rotating cylindrical lens might be used to compute the line integrals of images displayed by a spatial light modulator, such as a liquid crystal device. Calculations provided below will show, however, that the Hough transform streak detection procedure is not reliable for track validation at high hit densities. The weakness lies partly in the streak data, and partly in the Hough transform. Two dimensional streak imagery does not provide information about the streak development over time. Hence nonsensical hit patterns that by chance fall along lines on the superimposed frame image will be classified as valid tracks. The Hough transform is also a weak criterion for track validation with a framing sensor. Rather than testing for streaks in the superimposed frame data, we should test for regularly spaced, collinear arrays of hits, because targets in ballistic motion should have nearly uniform angular rates over the integration period, and successive frames will catch targets at discrete locations along their tracks.

An algorithm that tests for regularly spaced collinear hits is the generalized Hough transform,

$$H_g(m,b,f,p) = \int dx \int dy \; i(x,y) \; r(x,y,m,f,\theta) \; \delta(y-mx-b),$$

where $r(x,y,m,f,\theta)$ is a masking function like a ronchi, that has "opaque and transparent" bars ($r=0$ and $r=1$). The masking function has a fundamental spatial frequency of f, spatial frequency phase θ, and the bars are oriented perpendicularly to lines of slope m. If a biased cosine function $1/2(1+\cos(2\pi f+\theta))$ is used for the masking function, the transform will be identical to a generalized projection (see references by Stroke, Farhat) and may be implemented optically. The masking of the Hough transform line integrals to search for regularly spaced hits, rather than randomly spaced hits greatly increases the reliability of track validation with superimposed frame data.

However, false tracks will still be declared whenever random hits on the individual frames happen to line up with a regularly spacing when the frames are superimposed. To eliminate chance hits that masquerade as a regularly spaced linear track on the superimposed frames, but do not follow a consistent trajectory in time, we must preserve the track history. This may be done by stacking the successive frames along the third dimension to represent the time sequencing of the data. We now introduce a track detection procedure which processes such 3-D data.

Figure 1 provides an illustration of the 3-D track detection approach. Successive frames from the scanning sensor are stacked along a time axis to provide a 3-D record of the data. Over a short enough data collection period, ballistic targets will form nearly collinear tracks through the 3-D record volume. These tracks may be detected by a 3-D generalization of the Hough transform. See Figure 1. Since a shadowcasting implementation of the 3-D Hough transform may be easily devised, as shown in Figure 2, other, more precise optical implementations should be possible. A compromise 2-D implementation shown in Figure 3, uses a series of isometric projections of successive frame data onto 2-D. As shown in the figure, the stacking of

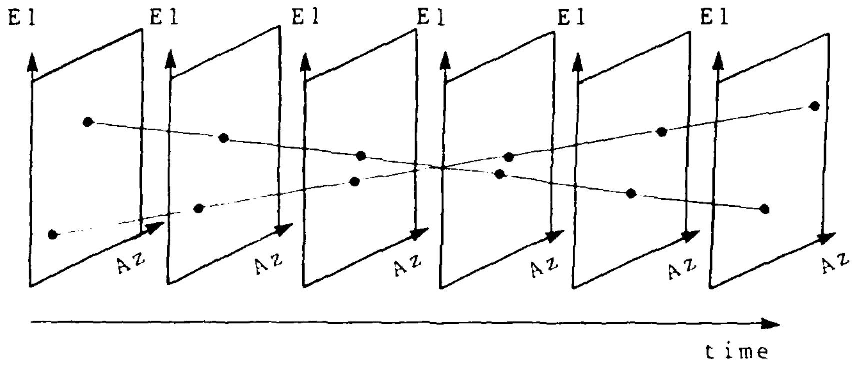


Figure 1. A series of frame images provides a record in space and time of the target motion. This record may be displayed by stacking the image frames along a time axis as shown above. Over short periods, ballistic targets will exhibit nearly collinear tracks through this 3-D volume. A 3-D version of the Hough transform may be used to detect the tracks.

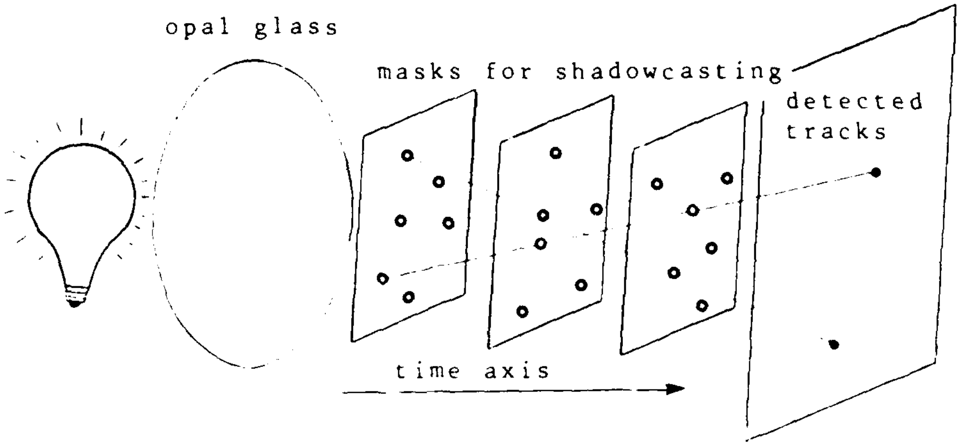


Figure 2. A shadowcasting arrangement to identify collinear tracks in space and time. The sequential image frame data is stacked along a time axis and displayed as opaque masks except for transparent holes marking each detection. This scheme is severely limited by diffraction, but it may be improved by replacing every other mask by a holomask, with hololenses marking the locations of hits.

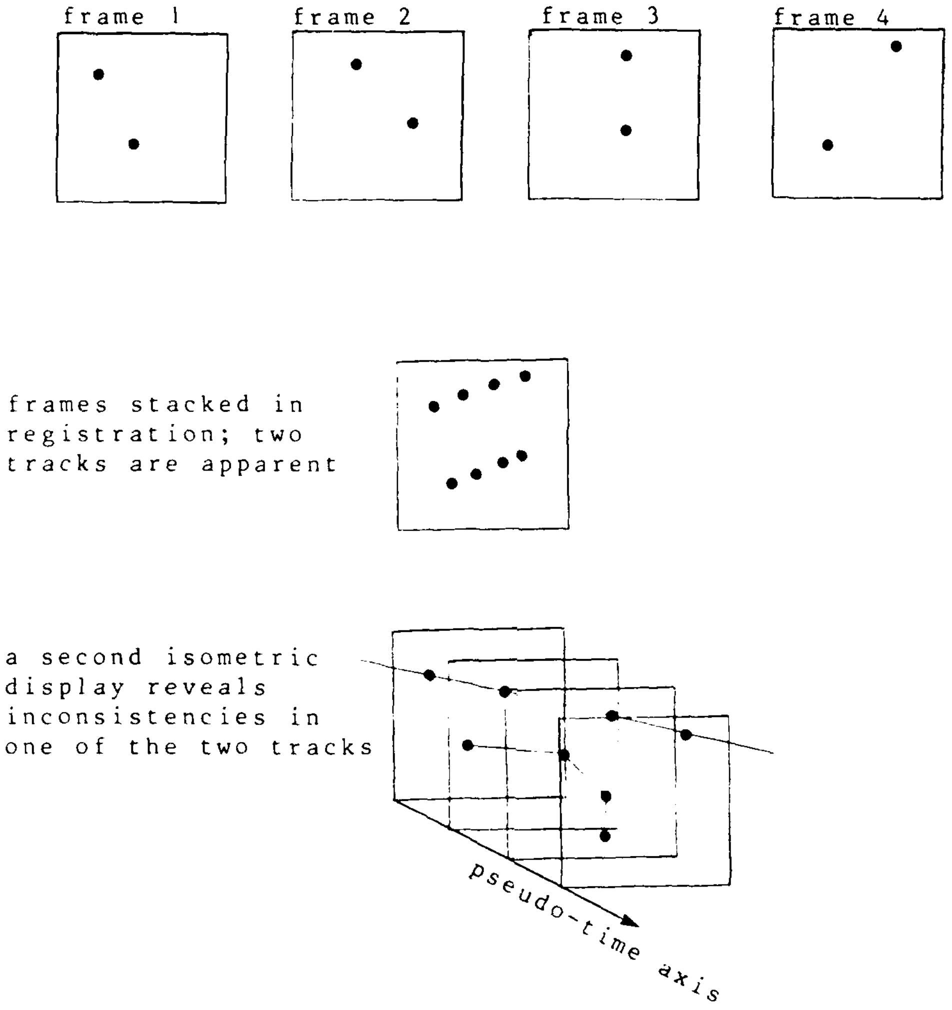

frame 1
frame 2
frame 3
frame 4
frames stacked in registration; two tracks are apparent
a second isometric display reveals inconsistencies in one of the two tracks
pseudo-time axis

Figure 3. The 3-D space and time track record may be displayed as an isometric projection in 2-D. On any single display, an inconsistent pattern of hits may lie on a collinear track, but this false track falls apart as the isometric display is changed.

successive frames for subsequent Hough or generalized Hough transformation can be performed with perfect registration, or with a frame-to-frame offset that progressively shifts later frames along a pseudo-time axis, as in an isometric display. On any single isometric display, extraneous hits may fall by chance along a line, masquerading as a track. But this chance track falls apart as the isometric display is rotated, by stretching or shrinking the offsets between frames that introduce the pseudo-time axis. If we require that a track persist on two or more settings of the isometric display, most of the capability of the 3-D track procedure to screen out extraneous hits is achieved, without the need to process data in 3-D.

ROC Curve Computations for Track Validation

The rival track detection procedures,

- streak detection by Hough transform
- generalized Hough
- 3-D track detection
- track detection with multiple isometric displays

differ in their capability to screen out false tracks. Criss-crossing trajectories and noise contribute hits that by chance may line up and masquerade as a valid track. With a given procedure, a trade-off must be made between detecting valid tracks and rejecting false tracks, and this tradeoff will be more favorable for one procedure than for the others.

The Relative Operating Characteristic or ROC curve provides a quantitative way to represent the tradeoff. The ROC curve plots the probability of detection against the probability of false alarm. To illustrate the idea, Figure 4 gives

Fig. 1 (top). The ROC graph, in which the true-positive proportion is plotted against the false-positive proportion for various possible settings of the decision criterion. The idealized curves shown correspond to the indicated values of the accuracy measure *A*. **Fig. 2 (bottom).** Example of empirical ROCs, showing standard and enhanced interpretations of mammograms.

TAKEN FROM:
John A. Swets,
Measuring the
Accuracy of
Diagnostic
Systems,
Science, p1287,
3 June 1988.

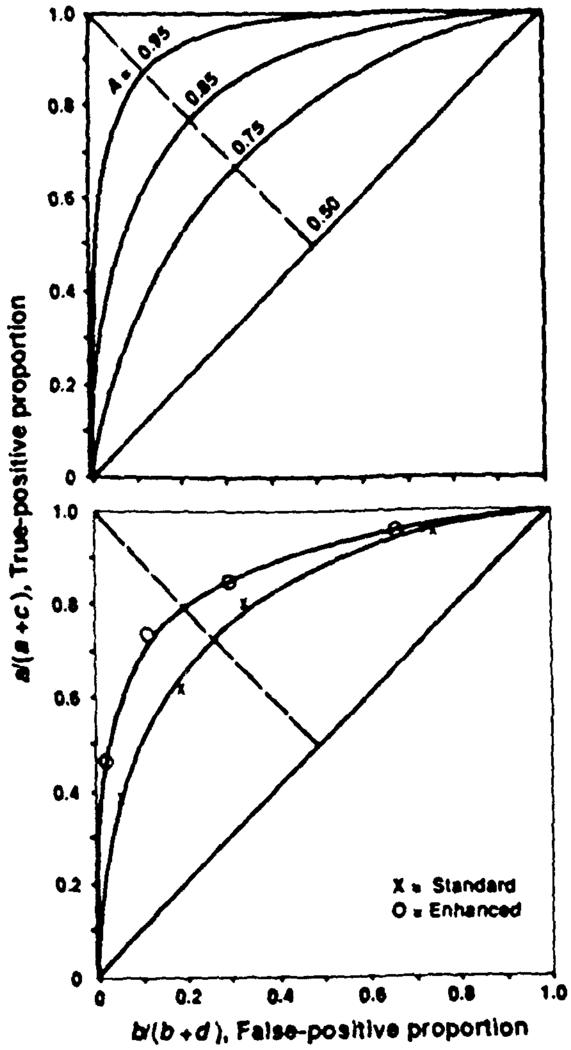


Figure 4. Representative ROC curves.

typical ROC curves from a review article. (See "Measuring the Accuracy of Diagnostic Systems, by John A. Swets, Science, p1285, 3 June 1988.) Ideally, the detection procedure will attain a probability of detection of nearly 1.0 while incurring a very low probability of false alarm. In such a case, the ROC curve will closely hug the upper left boundary of the ROC plot. On the other hand, a detection procedure which poorly discriminates against false alarms will have an ROC curve that falls closer to the diagonal line. The diagonal corresponds to a totally useless procedure, such as flipping a coin to decide if a track is valid.

The probability of detecting a target on any single frame is less than 1.0, even for targets within the nominal acquisition range. For the purpose of calculating ROC curves, we may select a reasonable value for α, the probability of detecting a target on a single frame. Then the probability of detecting a valid track may be assessed as a function of the threshold number of counts required to declare a track. For example, the following list of binominal probabilities shows how the probability of detection for a valid track in the absence of background events varies with the number of hits required out of m successive frames:

m out of m, $P_d = \alpha^m$

(m-1)or more, $P_d = m\,\alpha^{m-1}(1-\alpha) + \alpha^m$

(m-2) or more, $P_d = m(m-1)/2\ \alpha^{m-2}\ (1-\alpha)^2 +$

$m\,\alpha^{m-1}\ (1-\alpha) + \alpha^m,$

and so on.

For each of the above detection criteria (m hits out of m frames, m-1 hits out of m frames, etc) a given track detection algorithm will have a probability of accepting chance alignments of hits as a valid track. These probabilities are specific to each of the rival procedures, and will be calculated below for each procedure in turn. Then the probability of detection versus probability of false alarm or ROC curves are computed. A figure of merit for the entire ROC curve is the area under the curve. This figure of merit provides a way to rank rival detection procedures. (See Measuring the Accuracy of Diagnosis Systems, by John A. Swets, Science, p 1285, 3 June 1988).

With this single number, the area under a ROC curve as a measure of a track detection procedure's effectiveness, we can study how the procedure behaves as the hit density increases, by plotting the ROC curve area versus hit density. Similarly, the sensitivity of a procedure to other important parameters, such as the number of jointly processed frames, or the probability of target detection on a single frame may be investigated by plotting the ROC curve area versus the parameter of interest.

Preview of the Scene and Sensor Parameters

All of the track detection procedures will be analyzed on a common footing. The following parameters will be used:

- d = the background hit density (randomly distributed) per steradian
- m = number of frames jointly processed to detect tracks
- x = sensor resolution (radians)

- y = upper bound on track length (radians) over m frame period
- α = probability of target detection on a single frame in the absence of background events

When necessary for the purposes of illustration, the above parameters will be fixed at the following values:

$d = 10^5$ hits per steradian

$m = 10$ frames

$y = 10^{-1}$ radian

$x = 10^{-4}$ radian

$\alpha = 0.95$ probability of target detection on a single frame in the absence of background events

For example, with $\alpha = 0.95$, and $m=10$, the following values are obtained for track detection probability (in the absence of background hits) using the binomial expressions given above:

P_d (m out of m) = 0.5987
P_d (m-1 out of m) = 0.9138
P_d (m-2 out of m) = 0.9885
P_d (m-3 out of m) = 0.9989
P_d (m-4 out of m) = 0.9999

In addition to these parameters, we must make some simplifying assumptions about the data provided by the sensor. Although these assumptions may appear very restrictive, our evaluations of rival track procedures should not be invalidated if we make the same assumptions when comparing the different procedures. Our approach is to make the analysis as independent of sensor parameters and detailed operation as possible, so that we can concentrate on the evaluation of procedures for detecting tracks.

In this spirit, we begin our analysis with the simplifying assumption that a target image always falls inside a single image pixel, and that the trajectory images fall on a 1-D strip on the f҅cal plane, rather than over a 2-D image area. These assumptions would be unacceptable if our aim were to establish the absolute performance of a given track detection procedure; however, the relative performance of track detection procedures should not be affected.

As a rule, target detections will be performed as a pre-processing step on individual image frames before successive frames of data are jointly processed to detect tracks. After target detection, a given image pixel will either register a target detection or not; the original signal strength is unavailable for subsequent processing. It is therefore inconsistent to model the track detection procedures by a statistical procedure that gives a probability for a given pixel on an individual frame to register more than a single detection. However, this minor inconsistency should not invalidate our results when the target density is very low. We strive to gain insight into the track detection problem, and to achieve this we begin with simple analyses, and then refine the analyses step by step to eliminate inconsistencies and improve our approximations.

Our analysis will also be non-committal about sensor parameters and operations, such as the detector spacing, readout rate, and charge integration or Time Delay and

Integration (TDI). A single parameter d is used for the number of random detections per steradian on an individual image frame. In this way, we avoid such details as the selection of a target detection threshold, or the operation of spike adaptive TDI (to combat background noise from high energy radiation). Moreover, our parametric analysis applies equally well to diverse types of backgrounds, such as high energy radiation, thermal noise, clutter from a nuclear disturbed atmosphere, or a high density of actual targets. Since a pixel on an individual frame may declare a detection from any one of a number of background or target input events, we use the term "hit" rather that "detection" to remain non-committal about the actual presence of a target.

Analytical treatments of the ROC curves for each of the rival track detection procedures follow. Accompanying each analytical discussion is an actual calculation of the ROC curve made with the MathSoft Inc. program MathCAD on an IBM PCAT. As nearly as possible, the same notation is used in the discussion and the MathCAD calculation for each ROC.

Note on Mathematical Conventions

In some of the sums given below, the probability of a negative number of hits is indicated. We set all such probabilities to zero. For example, in the convolution

$$\sum_{S\geq T}^{m^2-m} \sum_{S'=0}^{m} P_a(S')\; P_b(S-S'),$$

if S' is greater than T, then S-S' is negative for S=T. We do not want to include the case of S' positive counts and (S-S') negative counts.

MathCAD does not allow sums over an index S satisfying a condition such as $S \geq T$. However, we can compute the same sum with MathCAD if we use a fixed range for S from 1 to m^2-m, and set to zero all terms that violate the condition $S \geq T$. MathCAD provides the Heaviside function for this purpose.

Another technique necessary with MathCAD is the recursive computation of binomial probability expressions. This avoids overflow which otherwise occurs in computing the factorials of large numbers.

ROC Curve Calculation for Streak Detection by Hough Transform

We first provide an analysis of the probability of false alarm and the probability of track detection using the Poisson distribution along with some simplifying assumptions. At the completion of the analysis, we provide a critique and then launch into a refined analysis (ROC Curve Calculations for Hough Transform with Target Detection on Each Frame).

The Hough transform counts the hits along a track in the superimposed data from m successive frames. For a track of length y radians and width x radians (the resolution of the sensor), the average hit count from random hits will be xymd, if m frames are superimposed, and the actual number of counts will follow a Poisson distribution. So the probability of obtaining n' hits from the background is given by:

$$\mathrm{Pbkg}(n') = \frac{(xymd)^{n'}}{n'!} \exp(-xymd)$$

In presence of a target track, there will also be hits from the target. The probability of getting n hits along the track from some combination of the background and the target is given by the convolution of the probability distributions of counts from the background and from the target. Over m superimposed frames, the probability of getting (n-n') hits from the target is given by the binomial distribution function

$$\mathrm{Ptarget}(n-n') = \frac{m!}{(m-n+n')!(n-n')!} (\alpha)^{n-n'} (1-\alpha)^{m-n+n'}$$

where α is the probability of receiving a hit on a single image frame in the absence of a background. Hence the convolution is given by

$$\text{Ptotal}(n) \quad = \sum_{n'=n-m}^{n} \quad \text{Pbkg}(n') \quad \text{Ptarget}(n-n')$$

For a track declaration, the integrated hits along the track must equal or exceed a threshold value T. Therefore, the probability of declaring a track in the presence of a target is given by

$$\text{Pstreak}(T) = \sum_{n=T}^{my/x} \text{Ptotal}(n) = \sum_{n=T}^{my/x} \quad \sum_{n'=n-m}^{n} \text{Pbkg}(n')\text{Ptarget}(n-n')$$

(The maximum possible number of hits is (my/x) because on any one of m frames, there are y/x pixels along the track.) This expression is used to compute the probability of track detection for a given threshold count.

The probability of meeting or exceeding the threshold with hits from the background alone is:

$$\text{Pfalse}(T) = \sum_{n'=T}^{my/x} \text{Pbkg}(n') \quad \doteq \quad 1 - \sum_{n'=0}^{T-1} \text{Pbkg}(n')$$

$$\doteq 1 - \exp(-xymd) \quad \sum_{n'=0}^{T-1} \frac{(xymd)^{n'}}{n'!}$$

(The approximation is valid for dx^2<<1: if this is not satisfied the threshold for detecting targets on a single frame is too low!)

The above expression gives the probability of declaring a track from chance hits in the absence of a target; i.e., the probability of false alarm. It is a function of the random hit density d, the threshold for track declaration, and the number of jointly processed frames.

Critique: Strictly speaking, our expression for Pbkg(n') does not apply if "hit" or "no hit" decisions are applied at each pixel before the image frames are jointly processed to detect tracks. Our expression gives a small, non-zero probability for n'>(my/x), and yet for tracks of y/x pixels in length, at most y/x hits will register, and so over m integrated frames the maximum possible value for n' is (my/x). Another shortcoming is our treatment of background and target hits as independent in the *expression for* Ptotal(n). This is consistent with our treatment, since allows more than one "hit" to a pixel; but after we refine our treatment to designate a "hit" or "no hit" at individual pixels, we should include the effects of the background (if present) when calculating the probability of a hit from the target.

MathCAD Computation

of ROC for Track Detection with the Hough Transform

<u>before</u> Pixel Quantization

First cut ROC curve Computation for Hough Transform Procedure

$d := 100000$ d is the density of hits per steradian

$m := 10$ m is the number of jointly processed frames

$x := 100 \cdot 10^{-6}$ x is the angular resolution (radians) of a pixel

$y := 0.1$ y is the maximum track length during m consecutive frames

$\alpha := 0.95$ α is the probability of detection on a single frame in the absence of a background

$b := x \cdot y \cdot m \cdot d$
$b = 10$

$g := 1 \,..\, \mathrm{ceil}(20\ b)$
$g := \mathrm{ceil}(10\ b)$

To reduce the needless computation of insignificant probabilities, we seek a maximum count, Cmax, above which terms may be safely dropped.

$$\mathrm{Cmax} := m + \mathrm{ceil}\left[\mathrm{root}\left[\left[\frac{b^{g}}{(\mathrm{ceil}(g))!}\ \exp(-b)\right], g\right]\right]$$

$\mathrm{Cmax} = 111$

$n := 0 \,..\, \mathrm{Cmax}$

$$\mathrm{Poisson}(n) := \frac{(x\ y\ m\ d)^{n}}{n!}\ \exp(-x\ y\ m\ d)$$

$\mathrm{Pbkg}(n) := \mathrm{if}(n \ge 0, \mathrm{Poisson}(n), 0)$

$l := 0 \,..\, m$

$$\mathrm{binary}(l) := \frac{m!}{(m-l)! \cdot l!} \cdot \alpha^{l}\ (1-\alpha)^{m-l}$$

$\mathrm{Ptarget}(l) := \mathrm{if}(l \ge 0, \mathrm{binary}(l), 0)$

$$i := 1 .. \mathrm{Cmax}$$

$$\mathrm{Ptotal}(i) := \sum_{l} \mathrm{Pbkg}(i - l) \cdot \mathrm{Ptarget}(l)$$

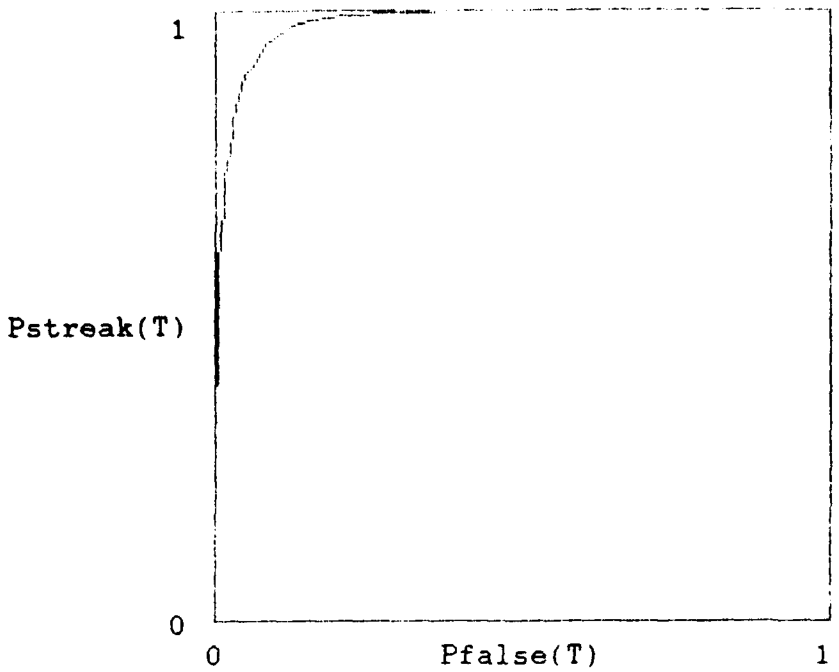
$$T := 1 .. 3 \cdot m$$

$$\mathrm{Pstreak}(T) := \sum_{i} (\mathrm{Ptotal}(i)) \cdot (\Phi(i - T))$$

$$\mathrm{Pfalse}(T) := \sum_{i} (\mathrm{Pbkg}(i)) \cdot (\Phi(i - T))$$

Pstreak(T)
1
1
1
1
1
1
1
1
1
1
0.999
0.998
0.992
0.978
0.949
0.898
0.822
0.722
0.605
0.481
0.362
0.258
0.174
0.111
0.068
0.039
0.021
0.011
0.006
0.003

Pfalse(T)
1
1
0.997
0.99
0.971
0.933
0.87
0.78
0.667
0.542
0.417
0.303
0.208
0.136
0.083
0.049
0.027
0.014
0.007
0.003
0.002
$6.997\ 10^{-4}$
$2.957\ 10^{-4}$
$1.201\ 10^{-4}$
$4.695\ 10^{-5}$
$1.768\ 10^{-5}$
$6.423\ 10^{-6}$
$2.254\ 10^{-6}$
$7.645\ 10^{-7}$
$2.51\ 10^{-7}$



$$j := 0 \,..\, (3\ m - 1)$$

$$\text{midarea} := \sum_{j} \left[\frac{(\text{Pstreak}(j + 1)) + (\text{Pstreak}(j))}{2}\right] ((\text{Pfalse}(j)) - (\text{Pfalse}(j + 1$$

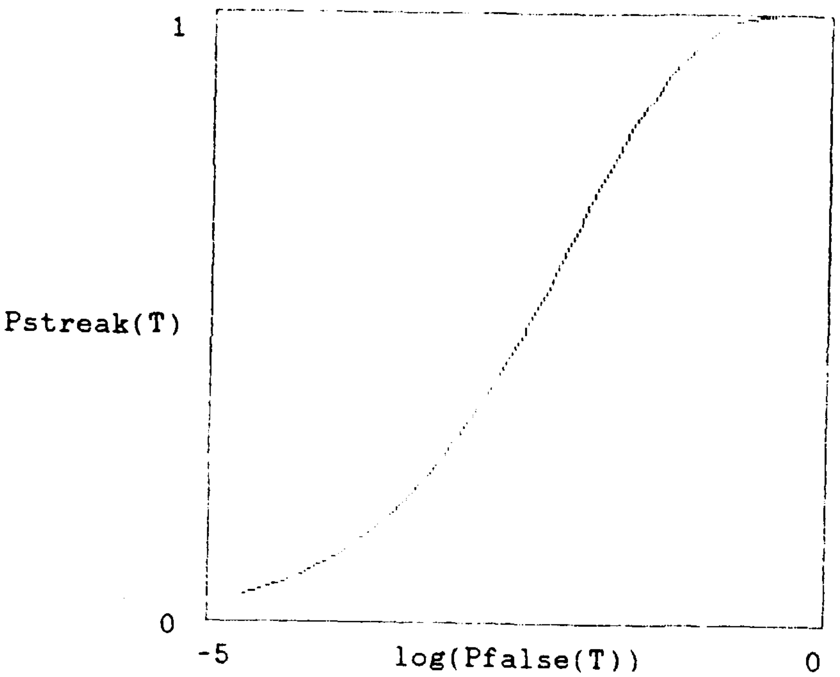
$$\text{leftend} := \left[\frac{\text{Pstreak}(3\ m) \cdot \text{Pfalse}(3\ m)}{2}\right]$$

$$\text{rightend} := \left[\frac{(1 + \text{Pstreak}(0)) \cdot (1 - \text{Pfalse}(0))}{2}\right]$$

$$\text{ROCarea} := (\text{leftend} + \text{midarea} + \text{rightend}) - \left[\frac{1}{2}\right]$$

$$\text{ROCarea} = 0.481$$

Replot with horizontal axis on a log scale to show behavior at low values of Pfalse.



ROC Curve Calculation for Hough Transform with Target Detection on Each Frame

Assume m frames are separately thresholded for target detection (hit vs. no hit) and then integrated for track detection ($S \geq T$ hits along the track).

For a track length of (y/x) pixels, there are (my/x) locations for hits. Since there are only two possibilities at each location, hit or no hit, the number of ways a total of S hits can be arranged over the (my/x) locations is:

$$\frac{(my/x)!}{(my/x-S)!\ S!}$$

In the following, we will sometimes use M as a shorthand for (my/x). Each pixel covers a solid angle of x^2 steradian, and so the expected number of hits per pixel is x^2d, for a random hit density of d per steradian. By the Poisson distribution, the probability of a pixel receiving no hit is

$$P(\text{no hit})_{\text{bkg}} = \exp(-x^2d).$$

Since the detection process is binary (hit vs no hit) the probability of the pixel declaring a hit from a random background event is:

$$P(\text{hit})_{\text{bkg}} = [1-\exp(-x^2d)].$$

Combining the results so far, the probability of a chance occurrence of S hits from the background along the integrated track length of y/x radian is

$$\text{PMbkg}(S) = \frac{(my/x)!}{(my/x-S)!\ S!}[1-\exp(-x^2d)]^{S}\ [\exp(-x^2d)]^{my/x-S}.$$

The M in PMbkg is a reminder for (my/x). The probability of declaring a false track for a threshold count of T is:

$$\mathrm{Pfalse}(T) = \sum_{S \geq T}^{my/x} \mathrm{PMbkg}(S)$$

In the absence of a background, the probability of detecting a target on a single frame is taken as α. In the presence of both a target and background, a hit may be declared from the background alone, from the target alone or from the target and background. The probability of not declaring a hit is the product of the probabilities for not detecting the target or the background:

$$\mathrm{Pnohit} = [\exp(-x^2 d)][1-\alpha]$$

A given target will show up on only m locations in the superimposition of m frames. Therefore in m locations there is a target and a background, while in (my/x-m) locations there is only the background. The probability of obtaining a count of S' over the m locations where the target is present is:

$$\mathrm{Ptarget}(S') = \frac{m!}{(m-S')!\ S'!} [\mathrm{Phit}]^{S'} [\mathrm{Pnohit}]^{m-S'}$$

Over the (my/x-m) locations where the target is absent, the probability of obtaining a count of (S-S') is:

$$\mathrm{PMmbkg}(S-S') = \frac{(my/x-m)!}{(my/x-m-S+S')!(S-S')!} [\exp(-x^2 d]^{(my/x-m-S+S')} [1-\exp(-x^2 d)]^{(S-S')}$$

Where the Mm in PMmbkg is a reminder for (my/x-m). The probability of obtaining a count of S hits in the presence of a target is therefore:

$$\mathrm{Ptb}(S) = \sum_{S'=0}^{S} \mathrm{Ptarget}(S')\ \mathrm{PMmbkg}(S-S')$$

and so the probability of declaring a track in the presence of the target is

$$\mathrm{Ptrack}(T) = \sum_{S \geq T}^{my/x} \mathrm{Ptb}(S)\ .$$

MathCAD Computation of ROC for Track Detection with Hough Transform *after* Pixel Quantization to One Bit

ROC curve Computation for Hough Transform Procedure with "hit" or "no hit" Decisions at Individual Pixels

$d := 100000$ d is the density of hits per steradian

$m := 10$ m is the number of jointly processed frames

$x := 100 \cdot 10^{-6}$ x is the angular resolution (radians) of a pixel

$y := 0.1$ y is the maximum track length during m consecutive frames

$\alpha := 0.95$ α is the probability of detection on a single frame in the absence of a background

$u := x^{2} \cdot d$

$u = 1 \cdot 10^{-3}$

$ww := (\exp(u) - 1)$

$ww = 0.001$

$M := m \cdot \frac{y}{x}$

$M = 1 \cdot 10^{4}$

$Cmax := 44$ Theoretically, the maximum hit count is M, but we need not carry the summations to more than Cmax since the probability of getting more hits is so low.

$N := (Cmax + m)$

$S := 0 .. N$

$PMbkg_{0} := \exp(-u \cdot M)$

$$\mathrm{PMbkg}_{S+1} := \mathrm{PMbkg}_{S} \cdot \frac{(M - S) \cdot ww}{S + 1}$$

$$\mathrm{PMbkg}_{Cmax} = 1.588 \cdot 10^{-15}$$

$$\sum_{S} \mathrm{PMbkg}_{S} = 1$$

$$Mm := M - m$$

$$\mathrm{PMmbg}_{0} := \exp(-u \cdot Mm)$$

$$\mathrm{PMmbg}_{S+1} := \mathrm{PMmbg}_{S} \cdot \frac{(Mm - S) \cdot (ww)}{S + 1}$$

$$k := -m \,..\, (Cmax + m)$$

$$\mathrm{PMmbkg}(k) := \mathrm{if}\left[k \ge 0, \mathrm{PMmbg}_{k}, 0\right]$$

$$\mathrm{PMmbkg}(Cmax) = 1.535 \cdot 10^{-15}$$

$$\mathrm{PMmbkg}(0) = 4.586 \cdot 10^{-5}$$

$$\sum_{S} \mathrm{PMmbkg}(S) = 1$$

$$\mathrm{Pnohit} := (\exp(-u)) \cdot (1 - \alpha)$$

$\mathrm{Pnohit} = 0.05$

$\mathrm{Phit} := 1 - \mathrm{Pnohit}$

$\mathrm{Phit} = 0.95$

$l := 0\,..m$

$$\mathrm{Ptarget}(l) := \frac{m!}{(m-l)!\cdot l!}\cdot \mathrm{Phit}^{l}\cdot(\mathrm{Pnohit})^{m-l}$$

$i := 1\,..(\mathrm{Cmax} + m)$

$$\mathrm{Ptb}(i) := \sum_{l} \mathrm{PMmbkg}(i - l)\cdot \mathrm{Ptarget}(l)$$

$\mathrm{Ptb}(\mathrm{Cmax} + 1) = 2.843\cdot 10^{-10}$

$T := 0\,..(\mathrm{ceil}(4.5\cdot m))$

$$\mathrm{Ptrack}(T) := \sum_{i} (\mathrm{Ptb}(i))\cdot(\Phi(i - T))$$

$$\mathrm{Pfalse}(T) := \sum_{i} \overrightarrow{\left[\mathrm{PMbkg}_{i}\right]}\cdot(\Phi(i - T))$$

Ptrack(T)
1
1
1
1
1
1

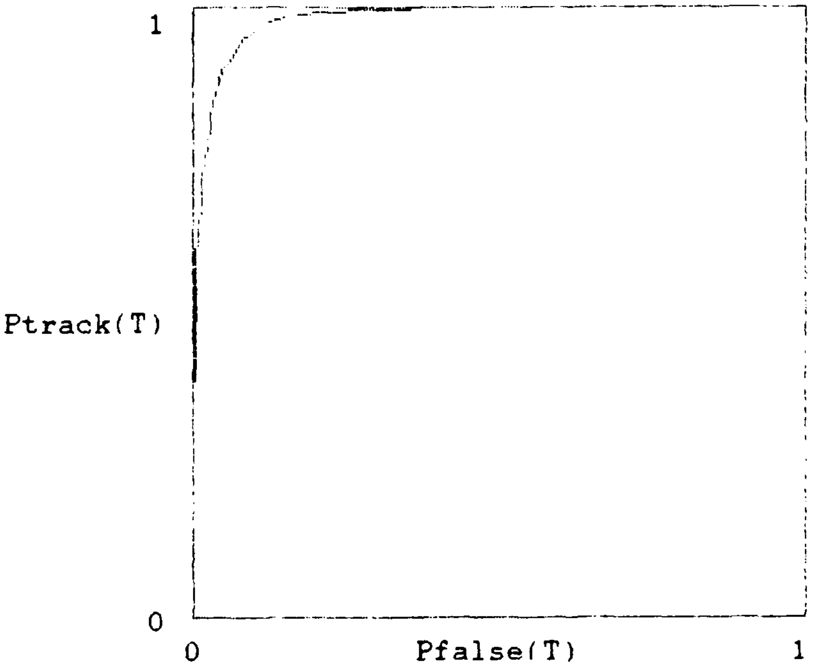
Pfalse(T)
1
1
1
0.997
0.99
0.971

1
1
1
1
1
0.999
0.998
0.992
0.978
0.949
0.898
0.821
0.721
0.603
0.479
0.361
0.257
0.173
0.11
0.067
0.038
0.021
0.011
0.005
0.003
0.001
$5.08\ 10^{-4}$
$2.124\ 10^{-4}$
$8.535\ 10^{-5}$
$3.302\ 10^{-5}$
$1.231\ 10^{-5}$
$4.432\ 10^{-6}$
$1.541\ 10^{-6}$
$5.181\ 10^{-7}$
$1.686\ 10^{-7}$
$5.319\ 10^{-8}$
$1.627\ 10^{-8}$
$4.832\ 10^{-9}$

0.933
0.87
0.779
0.667
0.542
0.416
0.303
0.208
0.135
0.083
0.048
0.027
0.014
0.007
0.003
0.002
$6.904\ 10^{-4}$
$2.913\ 10^{-4}$
$1.181\ 10^{-4}$
$4.608\ 10^{-5}$
$1.732\ 10^{-5}$
$6.278\ 10^{-6}$
$2.198\ 10^{-6}$
$7.439\ 10^{-7}$
$2.436\ 10^{-7}$
$7.731\ 10^{-8}$
$2.378\ 10^{-8}$
$7.101\ 10^{-9}$
$2.059\ 10^{-9}$
$5.804\ 10^{-10}$
$1.591\ 10^{-10}$
$4.248\ 10^{-11}$
$1.104\ 10^{-11}$

$1.394\ 10^{-9}$
$3.91\ 10^{-10}$

$2.799\ 10^{-12}$
$6.92\ 10^{-13}$
$1.669\ 10^{-13}$
$3.932\ 10^{-14}$
$9.05\ 10^{-15}$
$2.036\ 10^{-15}$
0



$$j := 0 \,..\, (3 \cdot m - 1)$$

$$\mathrm{midarea} := \sum_{j} \frac{\mathrm{Ptrack}(j+1) + \mathrm{Ptrack}(j)}{2} \cdot (\mathrm{Pfalse}(j) - \mathrm{Pfalse}(j+1))$$

$$\mathrm{leftend} := \frac{\mathrm{Ptrack}(3 \cdot m) \cdot \mathrm{Pfalse}(3 \cdot m)}{2}$$

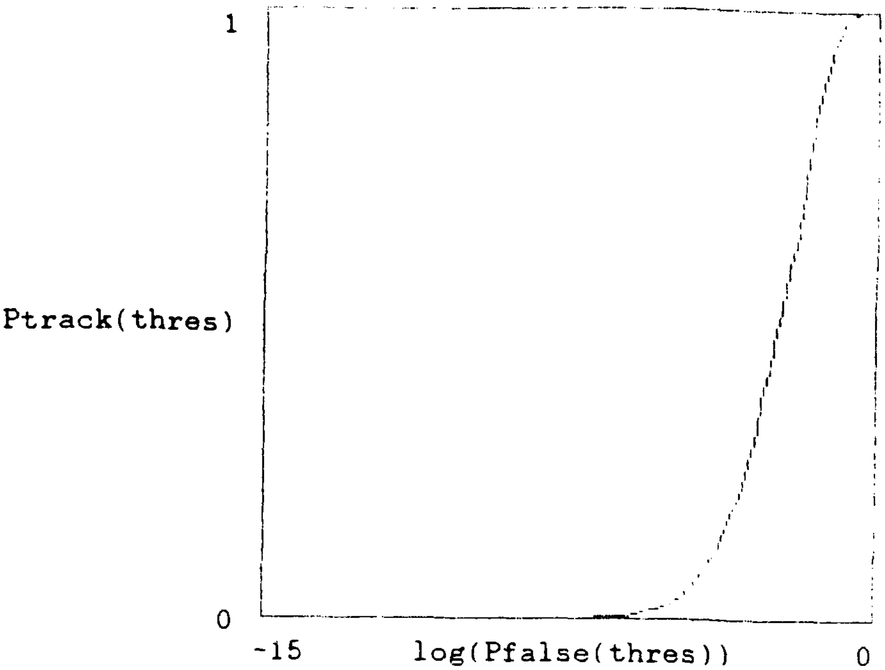
$$\mathrm{rightend} := \frac{(1 + \mathrm{Ptrack}(0)) \cdot (1 - \mathrm{Pfalse}(0))}{2}$$

$$\mathrm{ROCarea} := (\mathrm{leftend} + \mathrm{midarea} + \mathrm{rightend}) - \left[\frac{1}{2}\right]$$

$$\mathrm{ROCarea} = 0.481$$

thres := 0 ..45

To better visualize the behavior of Ptrack vs Pfalse, we re-plot the ROC using a log scale on the horizontal axis.



ROC Curves for Generalized Hough Transforms with Target Detection on Individual Frames

A specific target trajectory will be sampled on each of m successive frames at m regularly spaced pixels. Rather than integrating "hits" over the maximal track length of y/x pixels, the generalized Hough transform integrates hits at the m regularly spaced locations corresponding to a particular consistent track. At each of these m locations, a total of m frames are superimposed, so the hit count originates over a total of m^2 pixels.

For a background hit density of d hits/steradian, and a pixel solid angle of x^2, the expected number of hits per pixel is (x^2d). By the Poisson probability distribution, the probability of obtaining no hit is:

$$P_{bkg}(\text{no hit}) = \exp(-x^2d)$$

Therefore, the probability of obtaining a hit from chance alone is:

$$P_{bkg}(\text{hit}) = [1-\exp(-x^2d)]$$

The probability of a specific arrangement of S hits and (m^2-S) non-hits is therefore

$$[1-\exp(-x^2d)]^{S}\ [\exp(-x^2d)]^{m^2-S}$$

and the number of ways of arranging S hits over m^2 pixels is:

$$\frac{(m^2)!}{(m^2-S)!\ S!}$$

Therefore, the probability of obtaining a count of S hits from chance hits alone with the generalized Hough transform is:

$$\text{Pbkg}(S) = \frac{(m^2)!}{(m^2-S)!\ S!}\,[1-\exp(-x^2d)]^{S}\ [\exp(-x^2d)]^{m^2-S}$$

For a threshold of T, the probability of declaring a false track is:

$$\text{Pfalse}(T) = \sum_{S\geq T}^{m^2} \text{Pbkg}(S)$$

In the absence of a background, the probability of detecting a target on a single frame is taken as α. In the presence of both a target and background, a hit may be declared from the background alone, from the target alone or from the target and background. The probability of not declaring a hit is the product of the probabilities for not detecting the target or the background:

$$\text{Pnohit} = [\exp(-x^2d)][1-\alpha].$$

The probability of declaring a hit in the presence of both the target and the background is therefore:

$$\text{Phit} = [1-\text{Pnohit}] = 1-[\exp(-x^2d)]\ [1-\alpha]$$

A given target enters only m of the m^2 pixels integrated by the generalized Hough transform. Therefore the probability of obtaining a hit count of S from the generalized Hough transform is the convolution of the probabilities of obtaining S' hits over the m pixels containing the target, and (S-S') hits over the (m^2-m) pixels not containing the target. These separate probabilities are as follows:

$$\text{Ptarget}(S') = \frac{m!}{(m-S')!S'!}[\text{Phit}]^{S'}[\text{Pnohit}]^{m-S'}$$

and,

$$\text{Pmmbkg}(S-S') = \frac{(m^2-m)!}{(m^2-m-S+S')!(S-S')!}[\exp(-x^2d)]^{(m^2-m-S+S')}[1-\exp(-x^2d)]^{(S-S')} .$$

Where the mm in Pmmbkg is a reminder for (m^2-m). The probability of obtaining S hits between these two distributions is:

$$\text{Ptb}(S) = \sum_{S'=0}^{m} \text{Ptarget}(S')\ \text{Pmmbkg}(S-S').$$

Therefore the probability of declaring a track in the presence of a target with the generalized Hough procedure is:

$$\text{Ptrack}(T) = \sum_{S\geq T}^{m^2} \text{Ptb}(S).$$

MathCAD Computation of ROC for Track Detection with the Generalized Hough Transform

ROC Curve Computation for Generalized Hough Transform Procedure with "hit" or "no hit" Decisions at Individual Pixels

$d := 100000$ d is the density of hits per steradian

$m := 10$ m is the number of jointly processed frames

$x := 100\ 10^{-6}$ x is the angular resolution (radians) of a pixel

$y := 0.1$ y is the maximum track length during m consecutive frames

$\alpha := 0.95$ α is the probability of detection on a single frame in the absence of a background

$$u := x^{2} \cdot d$$

$$u = 1 \cdot 10^{-3}$$

$$ww := (\exp(u) - 1)$$

$$ww = 0.001$$

$Cmax := \mathrm{ceil}\left[0.05 \cdot m^{2}\right]$ Theoretically, the maximum hit count is m^2, but for counts greater than roughly five percent of this the probabililites turn out so low that we need not compute them

$$N := (Cmax + m)$$

$$S := 0 \,..N$$

$$Pbkg_{0} := \exp\left[-u \cdot m^{2}\right]$$

$$Pbkg_{S+1} := Pbkg_{S} \frac{\left[m^{2} - S\right] \cdot ww}{S + 1}$$

$$\mathrm{Pbkg}_{\mathrm{Cmax}} = 6.829 \cdot 10^{-8}$$

$$\sum_{S} \mathrm{Pbkg}_{S} = 1$$

$$\mathrm{mm} := m^{2} - m$$

$$\mathrm{Pmmbg}_{0} := \exp(-u \cdot \mathrm{mm})$$

$$\mathrm{Pmmbg}_{S+1} := \mathrm{Pmmbg}_{S} \cdot \frac{(\mathrm{mm} - S) \cdot (\mathrm{ww})}{S + 1}$$

$$k := -m \,..\, (\mathrm{Cmax} + m)$$

$$\mathrm{Pmmbkg}(k) := \mathrm{if}\left[k \ge 0, \mathrm{Pmmbg}_{k}, 0\right]$$

$$\mathrm{Pmmbkg}(\mathrm{Cmax}) = 4.027 \cdot 10^{-8}$$

$$\mathrm{Pmmbkg}(0) = 0.914$$

$$\sum_{S} \mathrm{Pmmbkg}(S) = 1$$

$$\mathrm{Pnohit} := (\exp(-u)) \cdot (1 - \alpha)$$

$$\mathrm{Pnohit} = 0.05$$

$$\mathrm{Phit} := 1 - \mathrm{Pnohit}$$

$$\mathrm{Phit} = 0.95$$

$$l := 0 \,..\, m$$

$$\mathrm{Ptarget}(l) := \frac{m!}{(m-l)!\cdot l!}\cdot \mathrm{Phit}^{l}\cdot(\mathrm{Pnohit})^{m-l}$$

$$i := 1 \,..\, (\mathrm{Cmax} + m)$$

$$\mathrm{Ptb}(i) := \sum_{l} \mathrm{Pmmbkg}(i-l)\cdot \mathrm{Ptarget}(l)$$

$$\mathrm{Ptb}(\mathrm{Cmax} + 1) = 8.835\cdot 10^{-4}$$

$$T := 0 \,..\, (m + \mathrm{Cmax})$$

Theoretically, the maximum hit count is m^2, but even on a track it rarely exceeds (m + Cmax).

$$\mathrm{Ptrack}(T) := \sum_{i} (\mathrm{Ptb}(i))\,(\Phi(i - T))$$

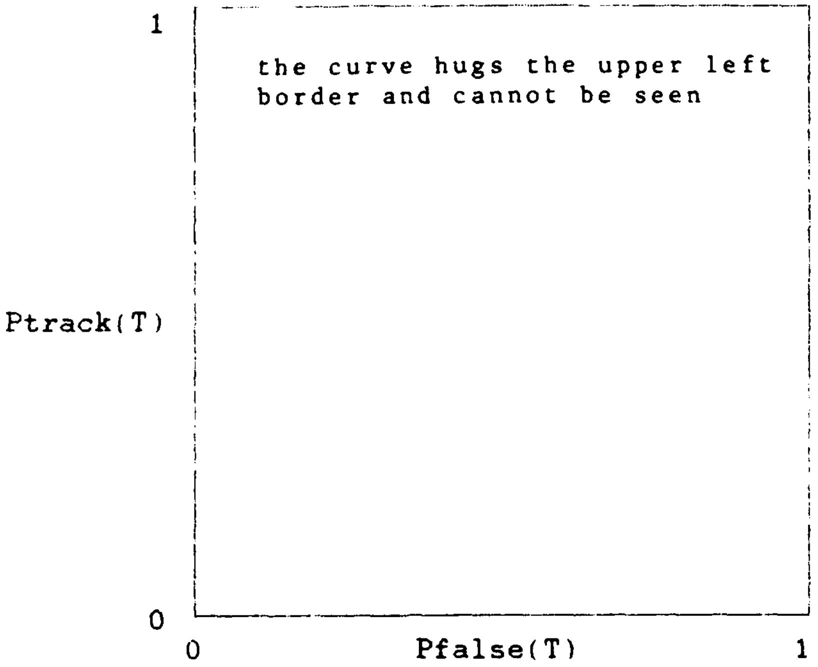
$$\mathrm{Pfalse}(T) := \sum_{i} \left[\mathrm{Pbkg}_{i}\right]\,(\Phi(i - T))$$

Ptrack(T)
1
1
1
1
1
1
1
0.999

Pfalse(T)
0.095
0.095
0.005
$1.502\cdot 10^{-4}$
$3.625\cdot 10^{-6}$
-8

0.989
0.92
0.626
0.053
0.002
$6.659 \cdot 10^{-5}$
$1.439 \cdot 10^{-6}$
$2.43 \cdot 10^{-8}$

$6.939 \cdot 10$
$1.097 \cdot 10^{-9}$
$1.471 \cdot 10^{-11}$
$1.708 \cdot 10^{-13}$
$1.745 \cdot 10^{-15}$
0
0
0
0
0
0



the curve hugs the upper left border and cannot be seen

$j := 0 \,..(m + Cmax - 1)$

$$\mathrm{midarea} := \sum_{j} \frac{\mathrm{Ptrack}(j + 1) + \mathrm{Ptrack}(j)}{2} (\mathrm{Pfalse}(j) - \mathrm{Pfalse}(j + 1))$$

$$\mathrm{leftend} := \frac{\mathrm{Ptrack}(m + \mathrm{Cmax}) \cdot \mathrm{Pfalse}(m + \mathrm{Cmax})}{2}$$

$$\mathrm{rightend} := \frac{(1 + \mathrm{Ptrack}(0)) \cdot (1 - \mathrm{Pfalse}(0))}{2}$$

$$\mathrm{ROCarea} := (\mathrm{leftend} + \mathrm{midarea} + \mathrm{rightend}) - \left[\frac{1}{2}\right]$$

$$\mathrm{ROCarea} = 0.5$$

To compare the false track probability of streak detection with m superimposed frames to the false track probability of the generalized Hough procedure analyzed above, we must take into account the fact that streak detection procedure simultaneously tests for a large number of tracks that must be tested for individually if the generalized Hough procedure is used. To make comparison, we must add up the probabilities for all combinations of false tracks that are distinguished by the generalized Hough procedure that are lumped onto a single streak if successive frames are superimposed. If the generalized Hough procedure distinguishes R loci that all fall on the same streak in the superimposed frame image, then this sum of probabilities can be shown to equal the lefthand expression and approximately equal to the righthand expression for large R:

$$1-(1-\mathrm{Pfalse})^{R} = (1-\exp(-R*\mathrm{Pfalse}))$$

A count of all the loci distinguished by the generalized Hough procedure that fall on a given streak yields for R the expression:

$$R := \left[\frac{1}{2}\right]\left[\left[\frac{\left[\frac{y}{x}\right]^{2}}{m - 1}\right] - \left[\frac{y}{x}\right] + \left[\frac{m - 2}{m - 1}\right]\right]$$

$R = 5.506 \cdot 10^{4}$

$t := 0 \,..\, (m + Cmax)$

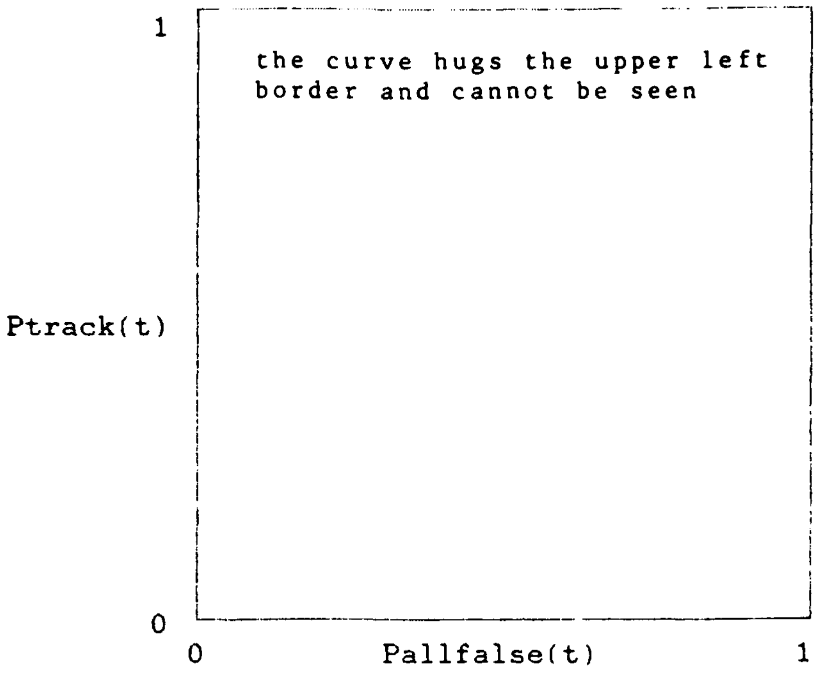
Pfalseall(t) := (1 - exp(-R·Pfalse(t)))

Pallfalse(t) := if(t > 0,Pfalseall(t),1)

This step to to overcome numerical problems when Pfalse is 1.

Ptrack(t)
1
1
1
1
1
1
1
0.999
0.989
0.92
0.626
0.053
0.002
$6.659 \cdot 10^{-5}$
$1.439 \cdot 10^{-6}$
$2.43 \cdot 10^{-8}$

Pallfalse(t)
1
1
1
1
0.181
0.004
$6.037 \cdot 10^{-5}$
$8.096 \cdot 10^{-7}$
$9.404 \cdot 10^{-9}$
$9.606 \cdot 10^{-11}$
$8.739 \cdot 10^{-13}$
$7.105 \cdot 10^{-15}$
0
0
0
0



$$j := 0 \,..\, (m - 1)$$

$$\text{midarea} := \sum_{j} \frac{\text{Ptrack}(j + 1) + \text{Ptrack}(j)}{2} (\text{Pallfalse}(j) - \text{Pallfalse}(j + 1))$$

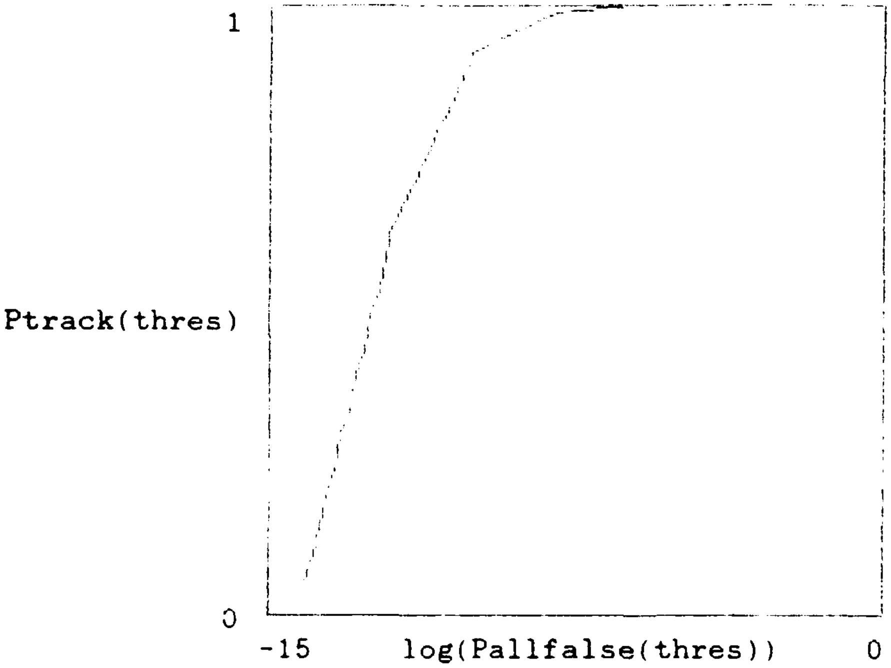
$$\text{leftend} := \frac{\text{Ptrack}(m)\ \text{Pallfalse}(m)}{2}$$

$$\text{rightend} := \frac{(1 + \text{Ptrack}(0))\,(1 - \text{Pallfalse}(0))}{2}$$

$$\text{ROCarea} := (\text{leftend}) + (\text{midarea}) + (\text{rightend}) - \left[\frac{1}{2}\right]$$

ROCarea = 0.5

thres := 0 ..(m + Cmax - 4)



ROC Curve Calculation for Track Detection in 3-D

A track will be designated by the 3-D method whenever a threshold number T, of collinear hits, occur on m consecutive image frames stacked in 3-D along the time axis. If the number of chance hits occurring in m collinear positions in this 3-D volume equals or exceeds T, a false track will be designated.

For a sensor of angular resolution x (radians), each pixel will occupy x^2 = xx steradians. With a random hit density of d hits per steradian, xxd hits are expected over the pixel solid angle of xx steradians. By Poisson statistics, the probability of a pixel not having a hit is:

$$\underset{xxd}{P(0)} = \frac{(x^2d)^0}{0!} \exp(-x^2d) = \exp(-x^2d).$$

Therefore the probability of getting 1 or more chance events leading to a hit is:

$$\left[\underset{xxd}{P(1)} + \underset{xxd}{P(2)} + \ldots + \underset{xxd}{P(n)} + \ldots\right] = 1-\exp(-x^2d)$$

The probability of S hits and m-S non-hits from the background alone is given by:

$$Pbkg(S) = \frac{m!}{S!(m-S)!} [1-\exp(-x^2d)]^S [\exp(-x^2d)]^{m-S}$$

The sum of this expression for $S \geq T$ to $S=m$ gives Pfalse(T), the probability of getting a false track along one particular direction in the 3-D volume.

The 3-D track detection procedure distinguishes tracks that are lumped as a single track by the streak detection approach. To make a correspondence between the probability of false alarms for these two procedures, we must first make a count of the number of 3 D tracks that correspond to a single streak. Consider Figure 5, which shows (y/x) distinct tracks in 3-D originating at a common pixel on the first frame. All of these tracks fall on a single streak in the 2-D projection of the frame data used for streak detection.

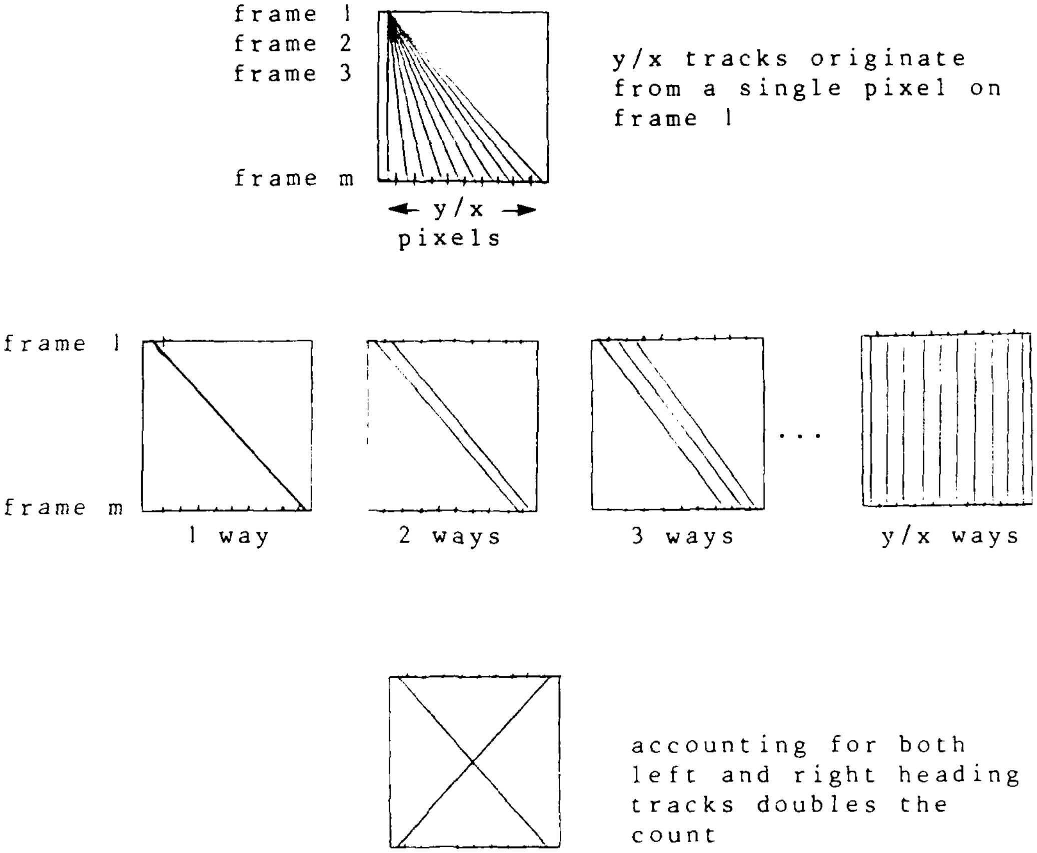
Moreover, although the longest track fits only 1 way, the next longest track (y/x-1 in length) fits on a streak of y/x pixels 2 ways. Counting up all the possible shifts of a 3-D track along the 2-D streak leads to the following arithmetic sum for the number of 3-D tracks which fall on the same 2-D streak:

$$(1+2+3+...+ y/x) = 1/2(y/x)(y/x +1).$$

As shown in the figure, this count must be multiplied by 2 to account for both rightward and leftward running tracks, so the total number of distinct tracks in 3-D that are lumped together as one streak in 2-D is:

$$(y/x)(y/x +1).$$

Critique: in making the above derivation, we have not excluded from the count those 3-D tracks which straddle more than a single pixel on a single image frame. As a rule, pixel straddling will occur unless the target image is very small compared to the pixels. But to keep our preliminary analysis as simple as possible, we have been assuming that the target image always falls into just one pixel on each



y/x tracks originate from a single pixel on frame 1

accounting for both left and right heading tracks doubles the count

Figure 5. Top: y/x tracks are shown which all fall on the same streak. Middle: the longest track fits only 1 way, the next to longest 2 ways, etc. Bottom: the track count is doubled when both leftward and rightward tracks are accounted for.

frame. To remain consistent, we must revise the track count given above by counting only those 3-D tracks that progress an integer number of pixels from frame to frame. Let q be this integer. Over (m-1) frames the target will cover its track length in increments of q. Generally, (m-1)q will be less than y/x, so we have

$$(m-1)q + s = y/x$$

where s is the number of ways that the array of m hits may be shifted along a maximum track length of y/x pixels. The case of q=0 corresponds to a motionless target, q=1 to a target that shifts just 1 pixel between frames, and so forth. The largest permissible value of q corresponds to a track that fits just 1 way into the maximum track length, or

$$(m-1)q_{max} + 1 = y/x; \qquad q_{max} = (y/x - 1)/(m-1).$$

So the revised 3-D track count is given by the sum of the 2s(q) from q=0 to q_{max}, or from q=1 to q_{max}, depending on whether the motionless target case is to be included or not. The factor of 2 is to include both rightward moving and leftward moving tracks.

$$\sum_{q=1}^{q_{max}} 2s(q) = \sum_{q=1}^{q_{max}} 2[(y/x)-(m-1)q] =$$

$$(y/x)^2/(m-1) - (y/x) + (m-2)/(m-1).$$

If we include the q=0 case, and set m=2, the resultant sum is

$$(y/x)(y/x + 1),$$

which is the same result we previously derived when we included tracks which straddled pixels on frames in-between the first and the last. For m=2, there are no such frames, and that's why the results are the same in this instance.

Which 3-D track count is correct? Since our ROC analysis so far has considered only the non-straddling tracks, and since we have neglected the motionless target case, the relevant track count is given by the sum of 2s(q) from q=1 to q_{max}. However, our ROC analysis would be more realistic if we revised it to include pixel straddling, since in an actual system, the target image will not be small compared to the pixels. So the relevant 3-D track count depends on the fidelity of of ROC analysis. Until we revise it to include pixel straddling, the relevant count of 3-D tracks that correspond to a single streak is given by

$$R = \left[(y/x)^2/(m-1) - (y/x) + (m-2)/(m-1)\right].$$

To compare the false track probability for streak detection with the 3-D track detection false track probability derived above, we must take into account the fact that a false track encountered by streak detection may correspond to 1 or more distinct false tracks in 3-D. In other words, the false track probabilities we have derived for the streak detection procedure and the 3-D track detection procedures are not comparable. To make a correspondence, we must add up the probabilities for all possible combinations of false tracks in 3-D that are lumped onto a single streak in 2-D. We can write down a relatively compact expression for this probability by recognizing that if the probability of an event e is P(e), the probability of a non-occurrence of e is 1-P(e). Rather than forming the sum of probabilities for all of the possible ways 1 or more false 3-D tracks may occur, we form the probability for the non-occurrence of all relevant false tracks.

We begin by expressing the probability that a particular 3-D locus has less than the threshold T of hits needed to designate a track in terms of (1- the probability that the locus has T or more hits). Let S be the number of hits for one particular locus. Then there is a probability of

$$\mathrm{Pbkg}(S) = \frac{m!}{S!(m-S)!}[1-\exp(-x^2 d)]^{S}\ [\exp(-x^2 d)]^{m-S}$$

that S hits are received along the locus, and that there are no hits in the remaining m-S pixels along the locus. Since the loci will satisfy the track criterion for $T \leq S \leq m$, the probability of getting a false track for a particular locus is:

$$\mathrm{Pfalse}(T) = \sum_{S=T}^{m} \frac{m!}{S!(m-S)!}[1-\exp(-x^2 d)]^{S}\ [\exp(-x^2 d)]^{m-S} .$$

Therefore, the probability of not getting a false track along a particular locus is:

$$1 - \mathrm{Pfalse}(T) =$$

$$1- \sum_{S=T}^{m} \frac{m!}{S!(m-S)!}[1-\exp(-x^2 d)]^{S}\ [\exp(-x^2 d)]^{m-S} .$$

The probability of this occurring for all of the relevant 3-D loci falling on a single 2-D streak is given below with R representing the number of relevant 3-D loci.

$$[1 - \text{Pfalse}(T)]^{R} =$$

$$\left[1- \sum_{S=T}^{m} \frac{m!}{S!(m-S)!} 1-\exp(-x^{2}d)^{S} \exp(-x^{2}d)^{m-S}\right]^{R}.$$

Therefore, the probability of having one or more false tracks on the R relevant loci is

$$1 - [1 - \text{Pfalse}(T)]^{R} =$$

$$1- \left[1- \sum_{S=T}^{m} \frac{m!}{S!(m-S)!} [1-\exp(-x^{2}d)]^{S} [\exp(-x^{2}d)]^{m-S}\right]^{R}.$$

This probability sums up all of the 3-D false track probabilities which correspond to a false track designation by the streak detection procedures.

If we let T=m, the expression reduces to:

$$1- \left[1- [1-\exp(-x^{2}d)]^{m}\right]^{R}.$$

By the binomial expansion theorem, we may expand this expression as a series. Then we obtain:

$$1- \left[1- \frac{R}{1!}[1-\exp(-x^{2}d)]^{m} + \frac{R(R-1)}{2!}[1-\exp(-x^{2}d)]^{2m} - \ldots\right]$$

$$= \left[R\ [1-\exp(-x^{2}d)]^{m} - \frac{R(R-1)}{2!}\ [1-\exp(-x^{2}d)]^{2m} + \ldots\right].$$

We now have the "machinery" to make comparisons of the false alarm probabilities for any two track detection procedures A and B. If procedure B distinguishes R tracks that are not distinguished by procedure A, we expect that procedure A will have a higher false track probability than B. To make a comparison that takes into account the R times we would have to apply procedure B to find a track that was found in one application of procedure A, we compute

$$1-(1-P_B(\text{false}))^R.$$

This expression corresponds to the probability of finding one or more false tracks using procedure B over the loci that are indistinguishable by procedure A. If procedure B is better than procedure A over this class of loci, we will have

$$P_A(\text{false}) > 1-(1-P_B(\text{false}))^R.$$

The direct comparison of P_A(false) with P_B(false) fails to take into account that procedure A is capable of finding a track along one or more of the R relevant loci in a single test, while procedure B tests each locus individually.

To compare streak detection with the generalized Hough procedure, the number of relevant loci is

$$R' = \sum_{q=1}^{q_{max}} s(q).$$

This is a factor of 2 less than the relevant track count for the 3-D track procedure vs. streak detection, because the generalized Hough procedure does not distinguish rightward moving tracks from leftward moving tracks. Hence we must compare

$$P(\text{false})_{\text{streak}} \quad \text{with} \quad 1-(1-P(\text{false})_{\text{gH}})^{R'}.$$

To compare the 3-D track procedure with the generalized Hough procedure, the relevant track count is R"=2, because the 3-D procedure distinguishes rightward and leftward tracks, while the generalized Hough does not. Therefore we compare

$$P(\text{false})_{\text{gH}} \quad \text{with} \quad 1-(1-P(\text{false})_{\text{3-D}})^{2}.$$

MathCAD Computation of ROC for Track Detection Using the Full 3-D Track History in Space and Time

ROC Curve Computation for 3D Track Detection Procedure with "hit" or "no hit" Decisions at Individual Pixels

$d := 100000$ d is the density of hits per steradian

$m := 10$ m is the number of jointly processed frames

$x := 100 \cdot 10^{-6}$ x is the angular resolution (radians) of a pixel

$y := 0.1$ y is the maximum track length during m consecutive frames

$\alpha := 0.95$ α is the probability of detection on a single frame in the absence of a background

$$u := x^{2} \cdot d$$

$$u = 1 \cdot 10^{-3}$$

$$ww := (\exp(u) - 1)$$

$$ww = 0.001$$

$$S := 0 \,..\, m$$

$$Pbkg_{0} := \exp(-u \cdot m)$$

$$Pbkg_{S+1} := Pbkg_{S} \cdot \frac{(m - S) \cdot ww}{S + 1}$$

$$Pbkg_{m} = 0$$

$$\sum_{S} Pbkg_{S} = 1$$

$$\mathrm{Pnohit} := (\exp(-u))\,(1 - \alpha)$$

$$\mathrm{Pnohit} = 0.05$$

$$\mathrm{Phit} := 1 - \mathrm{Pnohit}$$

$$\mathrm{Phit} = 0.95$$

$$\mathrm{Ptb}(S) := \frac{m!}{(m - S)!\,S!}\,\mathrm{Phit}^{S}\,(\mathrm{Pnohit})^{m-S}$$

$$\mathrm{Ptb}(m) = 0.599$$

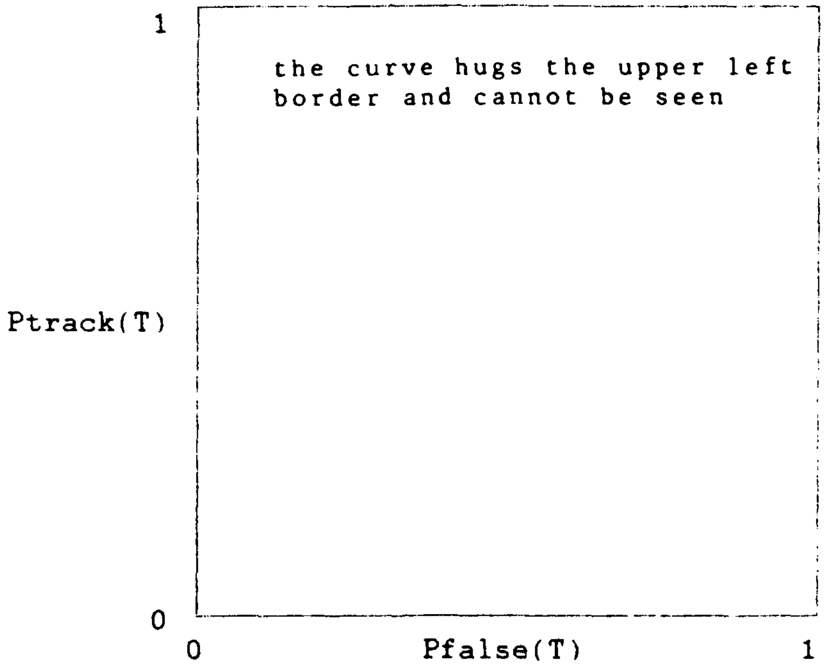
$$T := 0\,..m$$

$$\mathrm{Ptrack}(T) := \sum_{S} (\mathrm{Ptb}(S))\,(\Phi(S - T))$$

$$\mathrm{Pfalse}(T) := \sum_{S} \left[\mathrm{Pbkg}_{S}\right]\,(\Phi(S - T))$$

Ptrack(T)
1
1
1
1
1
1
1
0.999
0.989
0.914
0.599

Pfalse(T)
1
0.01
$4.472 \cdot 10^{-5}$
$1.192 \cdot 10^{-7}$
$2.086 \cdot 10^{-10}$
$2.503 \cdot 10^{-13}$
0
0
0
0
0



the curve hugs the upper left border and cannot be seen

j := 0 ..(m - 1)

$$\text{midarea} := \sum_{j} \frac{\text{Ptrack}(j + 1) + \text{Ptrack}(j)}{2} \cdot (\text{Pfalse}(j) - \text{Pfalse}(j + 1))$$

$$\text{leftend} := \frac{\text{Ptrack}(m) \cdot \text{Pfalse}(m)}{2}$$

$$\text{rightend} := \frac{(1 + \text{Ptrack}(0)) \cdot (1 - \text{Pfalse}(0))}{2}$$

$$\text{ROCarea} := (\text{leftend} + \text{midarea} + \text{rightend}) - \left[\frac{1}{2}\right]$$

$$\text{ROCarea} = 0.5$$

To compare the false track probability of streak detection with m superimposed frames to the false track probability of the 3D procedure analyzed above, we must take into account the fact that streak detection procedure simultaneously tests for a large number of tracks that must be tested for individually if the 3D procedure is used. To make a comparison, we must add up the probabilities for all combinations of false tracks in 3D that are lumped onto a single streak if successive frames are superimposed. If there are R loci in 3D (where we refer to 2D images stacked along a time axis as "3D"), then this sum of probabilities can be shown to equal the lefthand expression and approximately equal to the righthand expression for large R:

$$(1-(1-\text{Pfalse})^{R} = (1-\exp(-R*\text{Pfalse}))$$

A count of all the loci in 3D that fall on a given streak yields for R the expression:

$$R := \frac{\left[\frac{y}{x}\right]^{2}}{m - 1} + \left[\frac{y}{x}\right] + \frac{m - 2}{m - 1}$$

$R = 1.121\ 10^{5}$

$t := 0\ ..m$

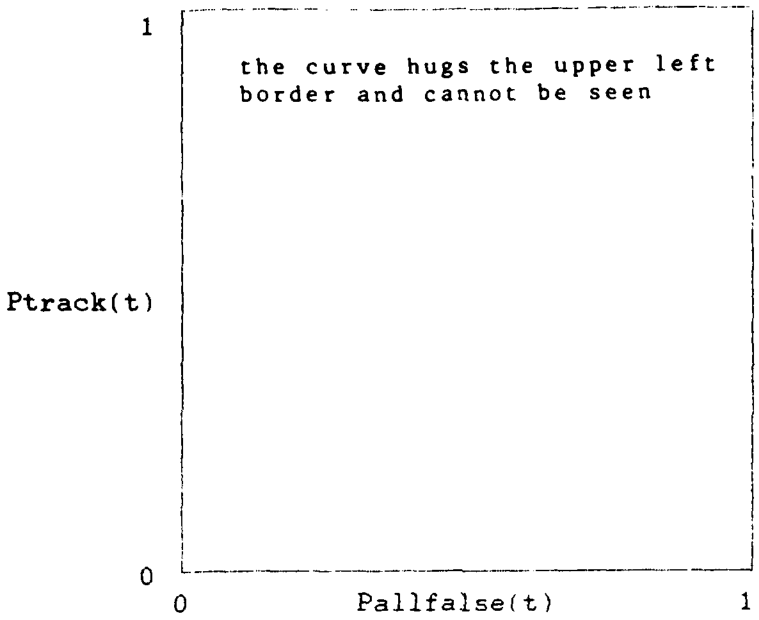
Pfalseall(t) := (1 - exp(-R Pfalse(t)))

Pallfalse(t) := if(t > 0,Pfalseall(t),1)

This step to to overcome numerical problems when Pfalse is 1.

Ptrack(t)
1
1
1
1
1
1
1
0.999
0.989
0.914
0.599

Pallfalse(t)
1
1
0.993
0.013
$2.338\ 10^{-5}$
$2.806\ 10^{-8}$
$2.339\ 10^{-11}$
$1.332\ 10^{-14}$
0
0
0



$$j := 0 \,..(m - 1)$$

$$\mathrm{midarea} := \sum_{j} \frac{\mathrm{Ptrack}(j + 1) + \mathrm{Ptrack}(j)}{2} (\mathrm{Pallfalse}(j) - \mathrm{Pallfalse}(j + 1))$$

$$\mathrm{leftend} := \frac{\mathrm{Ptrack}(m)\ \mathrm{Pallfalse}(m)}{2}$$

$$\mathrm{rightend} := \frac{(1 + \mathrm{Ptrack}(0))\ (1 - \mathrm{Pallfalse}(0))}{2}$$

$$\text{ROCarea} := (\text{leftend}) + (\text{midarea}) + (\text{rightend}) - \left[\frac{1}{2}\right]$$

$$\text{ROCarea} = 0.5$$

ROC Calculations for Track Detection on Isometric Displays

An isometric display of the 3-D track record may be generated for any desired 3-D perspective by offsetting each frame in the superimposed stack by a spatial offset that corresponds to the time of the frame. By varying the scale factor of these offsets, the isometric display may be changed to generate different perspectives of the 3-D track record. Each perspective may be individually processed for tracks using the streak or generalized Hough procedure.

Consider the effect of a perspective change on the background hits that enter into the processing. With the generalized Hough procedure, m pixels on each of m frames contribute to the hit count. Only m of these m^2 pixels belong to a specific 3-D locus. The other (m^2-m) pixels contribute background hits that reduce the reliability of the "track" or "no track" decisions. After the isometric display is changed by increasing or decreasing the scale factor, the generalized Hough procedure may be repeated; m of the pixels -- those belonging to the 3-D locus of interest -- will remain in the set of pixels that contribute to the hit count. The other (m^2-m) pixels contributing to the previous hit count will be replaced with a fresh set of (m^2-m) pixels. If all m^2 of the pixels were freshly selected with each change of the isometric display, the probability of obtaining k false alarms out of k different isometric displays would be

$$P^{k}_{bkg}(\text{false}),$$

and similarly, the probability of obtaining k track detections out of k different displays would be

$$P^{k}_{t+b}(\text{track}).$$

However, the m pixels belonging to the 3-D locus of interest will enter into the generalized Hough procedure for each of the k isometric displays. The probability of getting S hits from these m pixels will not be independent from display to display. In the absence of a target, this probability, whether on 1 display or 100 displays is given by

$$\text{Pmbkg}(S) = \frac{m!}{(m-S)!(S)!}\left(\exp(-x^{2}d)\right)^{m-S}\left(1-\exp(-x^{2}d)\right)^{S},$$

where the m in Pmbkg is a reminder for m in the righthand side of the equation. The probability of obtaining (S'-S) hits over the remaining (m^2-m) pixels is given by

$$\text{Pmmbkg}(S'-S) = \frac{(m^{2}-m)!}{(m^{2}-m-S'+S)!(S'-S)!}\left(\exp(-x^{2}d)\right)^{(m^{2}-m-S'+S)}\left(1-\exp(-x^{2}d)\right)^{(S'-S)},$$

where the mm in Pmmbkg is a reminder for (m^2-m). Suppose S hits do occur by chance over the m pixels comprising a certain 3-D locus. Then a track will be detected provided that $S'\geq T$ hits occur on the (m^2-m) background pixels that enter into the generalized Hough count. The probability of obtaining T or more hits over the $(m + m^2 - m)$ pixels entering into the total generalized Hough count is

$$\sum_{S'\geq T}^{m^{2}-m}\ \sum_{S=0}^{m} \text{Pmbkg}(S)\text{Pmmbkg}(S'-S).$$

Each time the isometric display is changed, a fresh set of (m^2-m) pixels enter into the count, so the probability of obtaining T or more hits on 2 different isometric displays is

$$Pf(T) = \sum_{S''\geq T}^{m^2-m} \sum_{S'\geq T}^{m^2-m} \sum_{S=0}^{m} Pmbkg(S)Pmmbkg(S'-S)Pmmbkg(S''-S).$$

Where Pf(T) denotes the probability of declaring a false track with the isometric procedure. The generalization to k>2 should be clear from this example. A similar argument to that given above for Pf(T) shows that the probability of detecting a track in the presence of both the target and the background on 2 different isometric displays is

$$Pt(T) = \sum_{S''\geq T}^{m^2-m} \sum_{S'\geq T}^{m^2-m} \sum_{S=0}^{m} Ptb(S)Pmmbkg(S'-S)Pmmbkg(S''-S).$$

The generalization of Pt(T) for k>2 should be clear from the k=2 case shown above.

An expression for Ptb(S) where the tb stands for the presence of both target and background events is derived as follows. The probability of a pixel receiving no hit is given by the product of the probabilities of receiving no hit from the target and the no hit from the background

$$Pnohit = (1-\alpha)(exp-x^2d).$$

The probability of receiving a hit is then

$$Phit = [1 - Pnohit].$$

Thus, the probability of obtaining a hit count for S over the m pixels on the 3-D track is

$$\mathrm{Ptb}(S) = \frac{m!}{(m-S)!(S)!} \ \mathrm{Phit}^{S} \ \mathrm{Pnohit}^{m-S} \ .$$

MathCAD Computation of ROC for Track Detection using Multiple Isometric Displays of Sequential Image Frames

ROC Curve Computation for Isometric Procedure used with the Generalized Hough Transform Procedure with "hit" or "no hit" Decisions at Individual Pixels

$d := 100000$ d is the density of hits per steradian

$m := 10$ m is the number of jointly processed frames

$x := 100 \cdot 10^{-6}$ x is the angular resolution (radians) of a pixel

$y := 0.1$ y is the maximum track length during m consecutive frames

$\alpha := 0.95$ α is the probability of detection on a single frame in the absence of a background

$u := x^{2} \cdot d$

$u = 1 \cdot 10^{-3}$

$ww := (\exp(u) - 1)$

$ww = 0.001$

$Cmax := \mathrm{ceil}\left[0.07 \cdot m^{2}\right]$ Theoretically, the maximum hit count is m^2, but for counts greater than a few percent of this the probabilities turn out so low that we need not comput them

$S := 0 \,..m$

$Pmbg_{0} := \exp(-u \cdot m)$

$$Pmbg_{S+1} := Pmbg_{S} \cdot \frac{(m - S) \cdot ww}{S + 1}$$

$Pmbg_{m} = 0$

$$\sum_{S} Pmbg_{S} = 1$$

$j := 0 ..(Cmax + m)$

$mm := m^{2} - m$

$Pmmbkg_{0} := \exp(-u \cdot mm)$

$$Pmmbkg_{j+1} := Pmmbkg_{j} \cdot \frac{(mm - j) \cdot (ww)}{j + 1}$$

$kk := -m ..(Cmax + m)$

$Pmmbg(kk) := if\left[kk \ge 0, Pmmbkg_{kk}, 0\right]$

$Pmmbg(Cmax) = 0$

$Pmmbg(0) = 0.914$

$$\sum_{j} Pmmbg(j) = 1$$

$Pnohit := (\exp(-u)) \cdot (1 - \alpha)$

$Pnohit = 0.05$

$Phit := 1 - Pnohit$

$Phit = 0.95$

$l := 0 \,..\, m$

$$Ptb(l) := \frac{m!}{(m-l)!\cdot l!}\cdot Phit^{l}\cdot (Pnohit)^{m-l}$$

$T := 0 \,..\, Cmax$

$S1 := 0 \,..\, (Cmax + m)$

$S2 := 0 \,..\, (Cmax + m)$

Theoretically, the maximum hit count is m^2, but even on a track it rarely exceeds (m + Cmax).

We let Pf(T) be the probability of getting a false track from chance events alone when the track must appear on two isometric views:

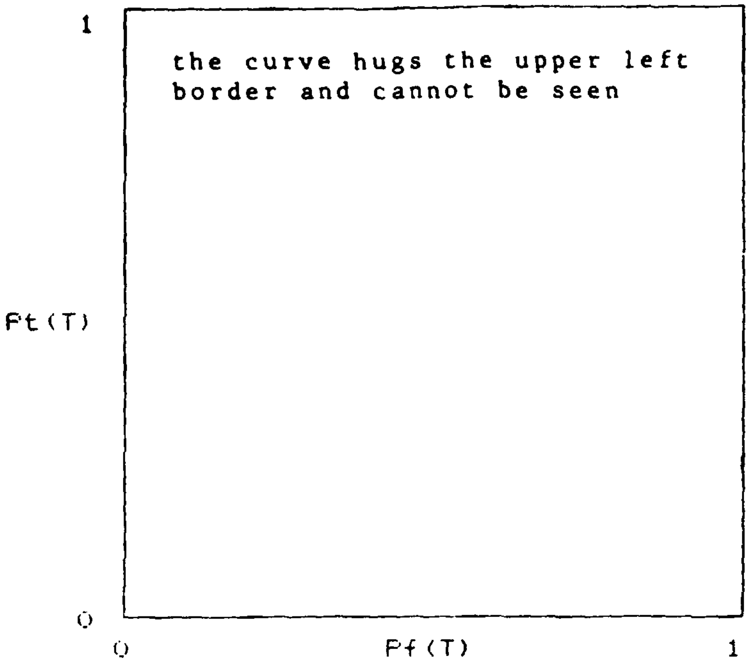
$$Pf(T) := \sum_{S2}\sum_{S1}\sum_{S} Pmbg_{S}\cdot Pmmbg(S1 - S)\cdot Pmmbg(S2 - S)\cdot \Phi(S1 - T)\cdot \Phi(S2 - T)$$

We let Pt(T) be the probability of declaring a track when a target is present and the track must show up on two isometric views;

$$Pt(T) := \sum_{S2}\sum_{S1}\sum_{S} Ptb(S)\cdot Pmmbg(S1 - S)\cdot Pmmbg(S2 - S)\cdot \Phi(S1 - T)\cdot \Phi(S2 - T$$

Pt(T)
1
1
1
1
1
1
1
0.999
0.989
0.915
0.601

Pf(T)
1
0.017
$1.322\cdot 10^{-4}$
$6.026\cdot 10^{-7}$
$1.85\cdot 10^{-9}$
$4.085\cdot 10^{-12}$
$6.735\cdot 10^{-15}$
0
0
0
0



$$j := 0 \,..(Cmax - 1)$$

$$midarea := \sum_{j} \frac{Pt(j+1) + Pt(j)}{2}\cdot(Pf(j) - Pf(j+1))$$

$$leftend := \frac{Pt(m + Cmax)\ Pf(m + Cmax)}{2}$$

$$rightend := \frac{(1 + Pt(0))\cdot(1 - Pf(0))}{2}$$

$$ROCarea := (leftend + midarea + rightend) - \left[\frac{1}{2}\right]$$

$$ROCarea = 0.5$$

To compare the false track probability of streak detection with m superimpose frames to the false track probability of the isometric display, generalized Hough procedure analyzed above, we must take into account that the streak detection procedure simultaneously tests for a large number of tracks that must be tested for individually if the isometric display procedure is used. To make a comparison, we must add up the probabilities for all combinations o false tracks that are distinguished by the isometric display procedure that are lumped onto a single streak if successive frames are superimposed. If th isometric display procedure distinguishes R loci that all fall on the same streak in the superimposed frame image, then this sum of probabilities can be shown to equal the lefthand expression and approximately equal to the righthand expression for large R:

$$1-(1-Pf(T))^{R} = (1-\exp(-R*Pf(T))).$$

A count of all the loci distinguished by the isometric display procedure that fall on a given streak yields the expression:

$$R := \left[\left[\frac{\left[\frac{y}{x}\right]^{2}}{m-1}\right] - \left[\frac{y}{x}\right] + \left[\frac{m-2}{m-1}\right]\right]$$

$$R = 1.101 \cdot 10^{5}$$

$$Pfalseall(T) := (1 - \exp(-R \cdot Pf(T)))$$

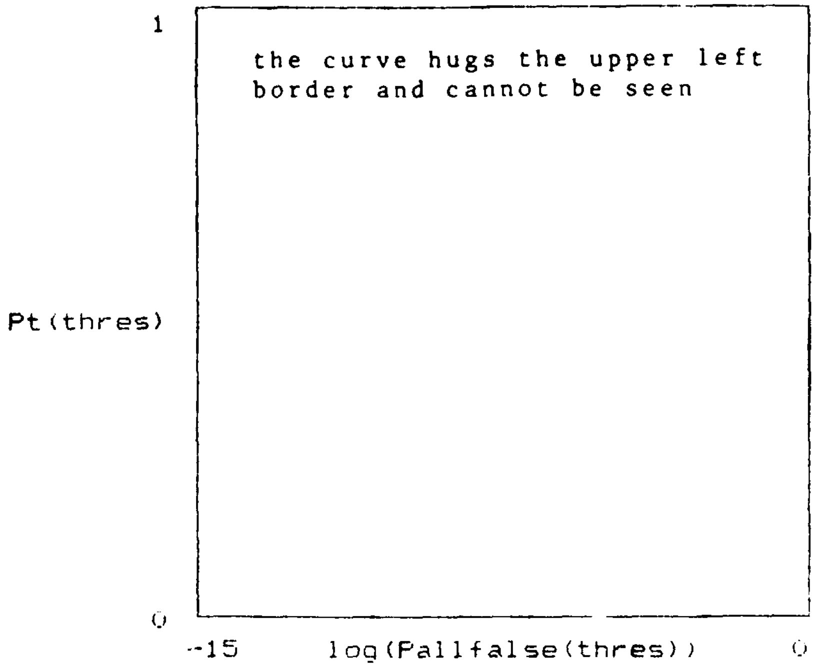
$Pallfalse(T) := if(T > 0, Pfalseall(T), 1)$ This step is intended to overcome numerical problems when Pf is 1.

Pt(T)
1
1
1
1
1
1
1
0.999
0.989
0.915
0.601

Pallfalse(T)
1
1
1
0.064
$2.037 \cdot 10^{-4}$
$4.498 \cdot 10^{-7}$
$7.417 \cdot 10^{-10}$
$9.354 \cdot 10^{-13}$
0

thres := 0 ..6

0
0



the curve hugs the upper left border and cannot be seen

References

Generalized Projection References:

Chi K. Chan and Nabil H. Farhat, "Frequency Swept Tomographic Imaging of Three-Dimensional Perfectly Conducting Objects," IEEE Trans. Antennas and Propagation, AP-29 p312 (1981).

G. W. Stroke and M. Halioua, "Three-dimensional reconstruction in X-ray crystallography and electron microscopy by reduction to two-dimensional holographic implementations," Trans. Amer. Crystallograph. Assoc., 12, p27 (1976).

Relative Operating Characteristic (ROC) Reference:

John A. Swets, "Measuring the Accuracy of Diagnostic Systems," Science, 240, pp1285-1293 (1988).

3.3 Selection of Overlapping Receptive Fields for Uniform Weighting

Many image processing procedures require local processing, rather than global processing, for localized image features. Examples are texture, local power spectra, and, in stereo pairs, stereo disparity. Similar local processing is provided by the receptive fields of a neural network. Although the processing is local, the entire image must be covered, and this is accomplished by overlapping the processing regions (or overlapping the receptive fields of the neural network). It is desirable to overlap roughly 50%, and at the same time weight each pixel in the original image evenly, no matter what its registration is to the boundaries of the local processing regions. In the case of 1-D input data triangularly tapered receptive fields may be overlapped to achieve a uniform weighting of all input samples; see Figure 6. Since our data is 2-D, we need the 2-D analogy to triangularly tapered receptive fields. Ideally, the receptive field taper would be linear, or if not linear, an easily calculated linear function. A step-function or "pill-box" shape might also be satisfactory. The use of binary "on" and "off" weighting eases computational costs, but might reduce performance. For example, tapering is needed to minimize artifacts caused by finite data sample lengths in power spectrum estimation.

Fourier analysis may be used to verify the uniform weighting of an overlapped receptive field function, g(x,y). The overlapping corresponds to a convolution of g(x,y) with a discrete sampling lattice. We require that this convolution be flat, so it can only contain one Fourier component, dc. let the sampling lattice that is used with g(x,y) be l(x,y). Then

$$F(l \otimes g) = F(l) \cdot F(g) = L(f_x,f_y) \cdot G(f_x,f_y)$$

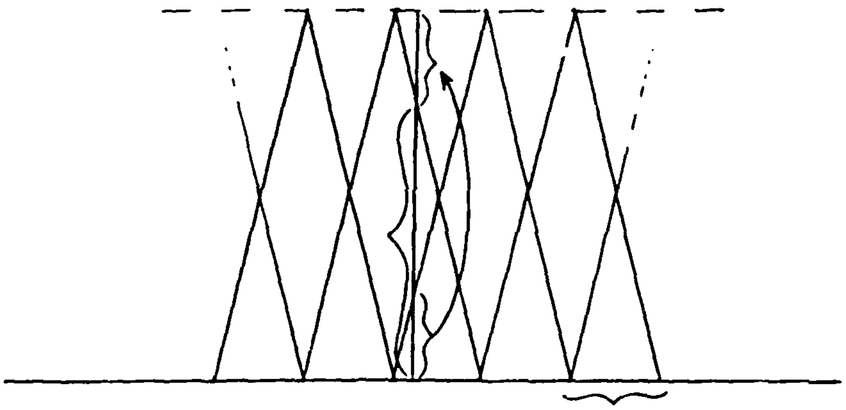


Figure 6. In one dimension, uniform weighting of an input layer is achieved by triangularly tapered receptive fields which are overlapped 50%. The sum of the overlapping responses is a constant, as demonstrated by the sum of the brackets in the figure.

is zero except for $f_x = f_y = 0$. This is possible because the Fourier transform of a sampling lattice is the "reciprocal lattice" and hence is zero except at discrete frequencies. If $G(f_x, f_y)$ is zero at all of these discrete frequencies (except $f_x = f_y = 0$), the uniform weighting condition is satisfied.

We may use this result to search for receptive fields that provide uniform weighting with a given sampling lattice. As illustrated in Figure 7 we need to find functions which are non-zero at one sampling point of the reciprocal lattice, but zero at all other lattice points. In 1-D, the functions that have this property are interpolation functions of the form

$$[\mathrm{sinc}(f_x)]^I = \frac{\sin(\pi f_x)}{\pi f_x}^{\,I}$$

where I is a non-zero integer power. In 2-D, we may construct suitable interpolation functions as separable functions in f_x and f_y, viz,

$$[\mathrm{sinc}(f_x)]^I\,[\mathrm{sinc}(f_y)]^J,$$

where I and J are non-zero integer powers. The inverse Fourier transform of these interpolation functions provides a set of suitable receptive field functions for uniform sampling over a square grid (or rectangular grid, if suitably scaled).

Hexagonal grids may be investigated similarly. In 2-D, an interpolation function is required for the reciprocal hexagonal lattice. Suitable interpolation functions may be constructed as products:

$$[\mathrm{sinc}(f_y)]^I\,[\mathrm{sinc}\,(-(\sqrt{3}/2)f_x+(1/2)f_y)]^J,$$

where I and J are non-zero integers.

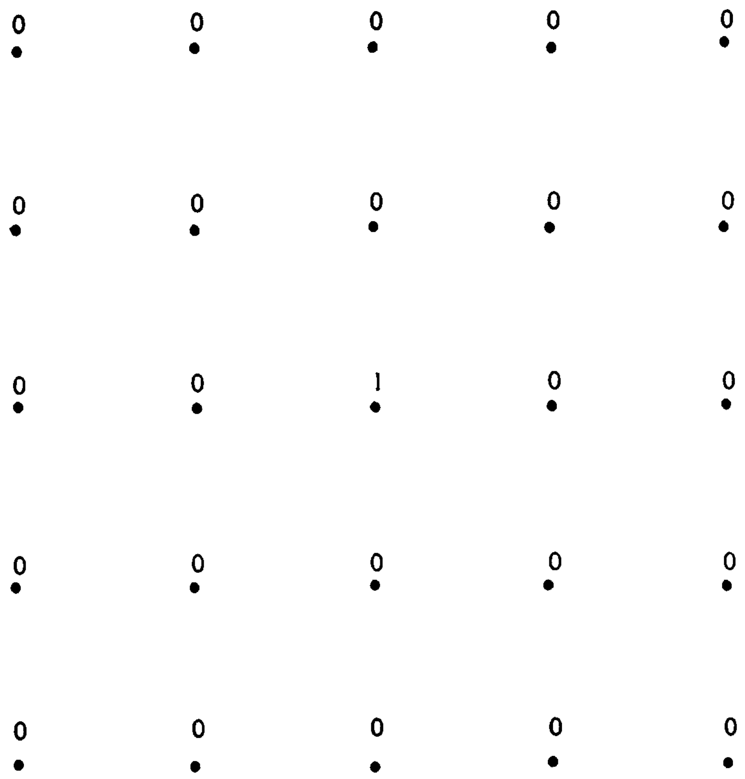


Figure 7. $G(f_x, f_y)$ must be zero at all points on the reciprocal lattice except $f_x = f_y = 0$. Therefore, $G(f_x, f_y)$ is an interpolation function suitable for extending a function sampled at the lattice nodes to continuous values of f_x and f_y.

The receptive field solutions provided above for square and hexagonal sampling lattices do not provide all of the desirable properties illustrated by the 1-D solution shown in Figure 6. Simple to calculate "pill-box" receptive fields obtained with I=J=1 do not provide tapering; however for I and J > 1 the taper is nonlinear. We have discovered that linear tapers can be used while satisfying the condition of uniform weighting provided that two different receptive field functions are allowed, and that non Euclidean distance metrics are used to compute the taper. The non-Euclidean metrics we use are the so called city-block or taxi-cab metric, and the absolute value metric

$$d(1,2) = |x_1-x_2| + |y_1-y_2| \ .$$

An example is provided in Figure 8, which shows equi-level contours for overlapping receptive fields; the receptive fields are either diamond shaped or square shaped. This solution was generated by first overlapping the diamond shaped receptive fields, and after the overlapping was found to be non-uniform, a second set of square receptive fields were added to attain uniformity.

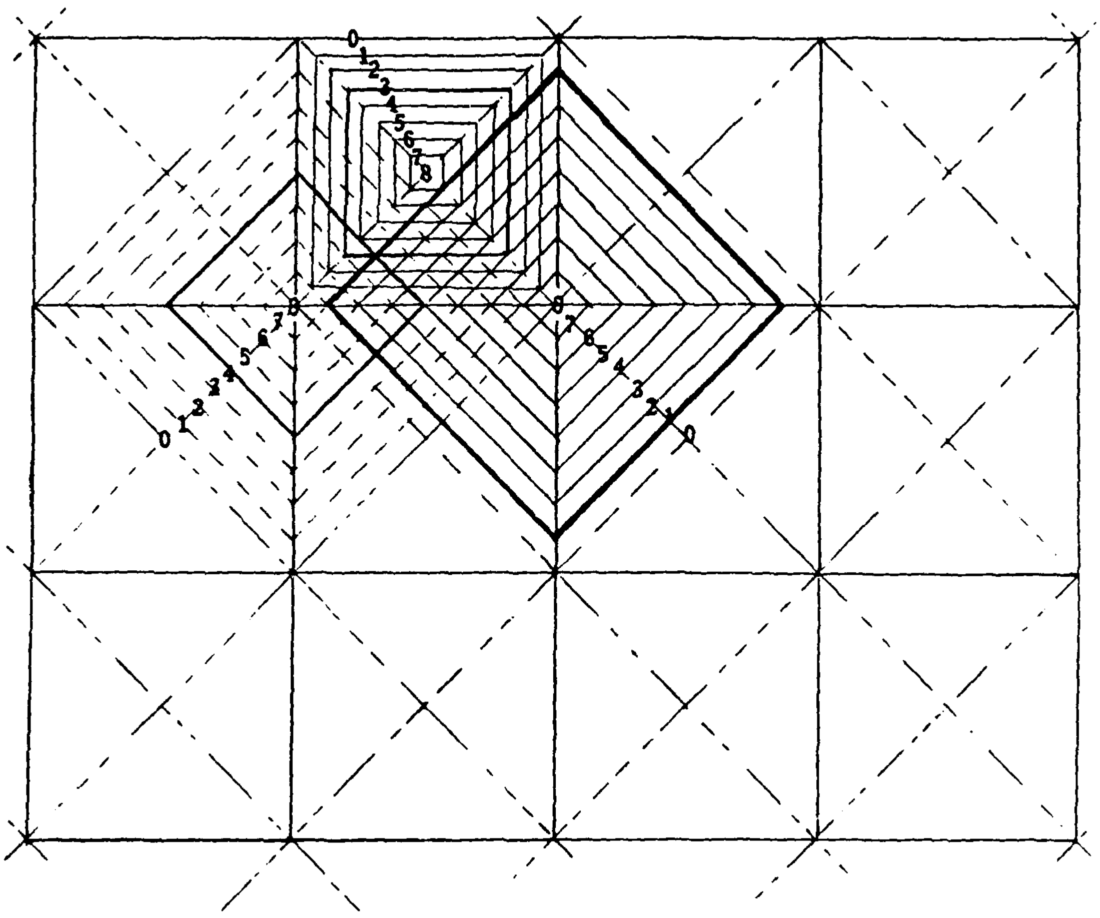


Figure 8. Uniform weighting of an input layer is achieved in 2-D with a system of two different receptive fields, each tapered linearly. Overlapping diamond receptive fields satisfying an absolute value metric, and abutting square receptive fields satisfying the city-block metric provide the desired uniform weighting. Darker lines indicating weights of 4, 3 and 1 illustrate the sum of all fields sums to 8 at a representative point .

4.0 Appendix on Image Collection by a Scanning Sensor

The detected and sampled imagery output by a scanning sensor is subject to three transfer functions. These transfer functions correspond to:

1) image formation by the optical system
2) spatial integration over individual detector elements
3) temporal integration during the scanning process as a result of charge build-up between detector readout samples.

The transfer function of the optical system is determined by actual testing in the optics shop, or in lieu of tests, by wave optic or ray tracing simulations of the sensor performance. By one of these methods, the point spread function of the image formation optics is determined. This point spread function generally varies in size and shape over the focal surface. Over the useful field of view we may use the average point spread function to characterize the optical system. The Fourier transfer function of the average point spread function may be used to characterize the transfer function of the image formation system.

For the purposes of this discussion we assume diffraction-limited optics with the Optical Transfer Function (OTF) cutting off at a spatial frequency of $(D/\lambda F)$ cycles per unit length across the focal surface. (D = entrance pupil diameter, λ= optical wavelength, and F = focal distance). This cut-off permits the optical image to be reconstructed from a finite number of discrete image samples. Even if the optics is not diffraction limited, the imagery will be bandlimited to spatial frequencies of less than $(D/\lambda F)$. As a practical matter, however, the empirical point spread function may impose an effective bandlimit that is less than $(D/\lambda F)$. So let the bandlimit be given by f_c (for cut-off frequency).

For a incoherent imaging system $f_c = D/\lambda F$.

By the Nyquist sampling theorem, the sampling frequency must be $\geq 2 f_c$ to reconstruct the optical image from discrete samples. Assuming the cut-off frequency is invariant with orientation, this sampling criterion must be applied to both the horizontal and the vertical image sampling. For example, in a staring focal plane array, the center-to-center distances between neighboring detectors elements should be less than or equal tc $1/2f_c$. This implies that the detectors elements are dimensioned no larger than $1/2f_c \times 1/2f_c$.

The same principles apply to a scanning sensor, but in this case the scan and cross-scan directions are distinguished by the scanning process. In the cross-scan direction the image sampling is performed by the detector elements, so the center-to-center spacing must be $1/2f_c$. As shown in Figure 9 however, this may be achieved either by sizing the detector elements $1/2f_c$ or less in the cross-scan direction, or by staggering the detector array so that the sampling interval of $1/2f_c$ is satisfied with a larger cross-scan detector width.

In the scan direction, the sampling is performed by electronic readout out time intervals of tau. For a focal surface scan rate of v unit lengths per unit time the Nyquist criterion becomes $v(\text{tau}) \leq 1/2f_c$.

To improve system performance in the presence of photon or readout noise, it is desirable to provide as much spatial integration on the detector elements as possible. In other words, the detectors should be as large as possible.

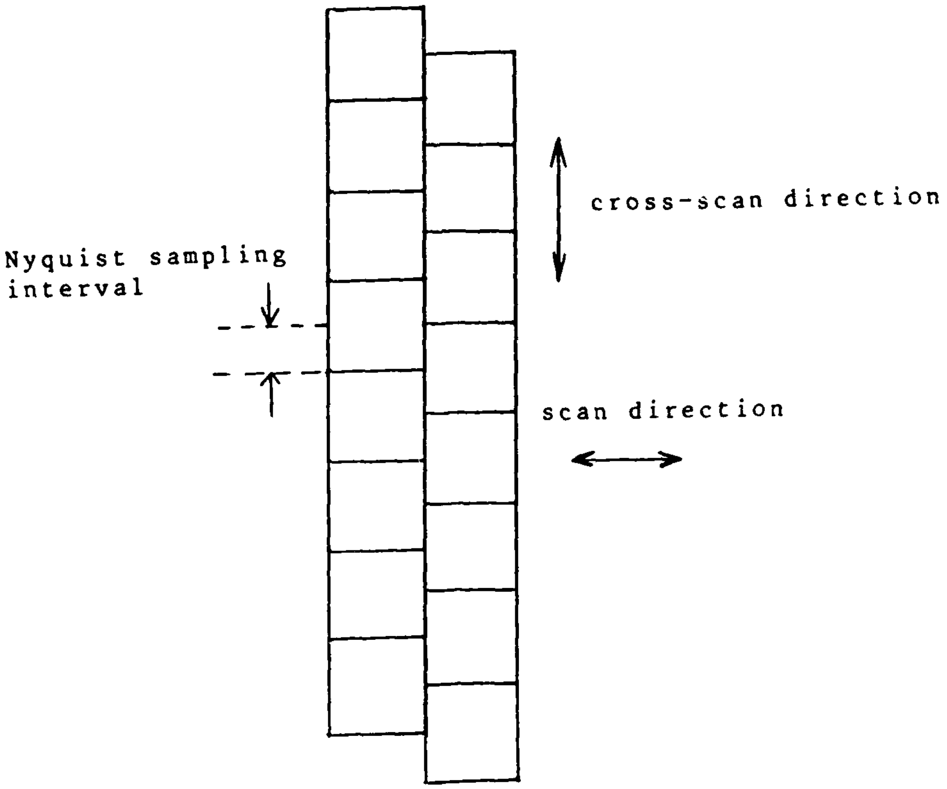


Figure 9. A scanning focal plane array which has a detector separation along the cross-scan direction satisfying the Nyquist criterion can nonetheless have individual detectors dimensioned twice the Nyquist interval. Along the scan direction the dectector readout rate (not the detector width) must satisfy Nyquist.

In the staring sensor case, the detector size is fixed at $1/2f_c$ x $1/2f_c$, and this small size is compensated for by temporal integration. In the scanning sensor case the temporal integration is limited by $v(tau) \leq 1/2f_c$. However, spatial integration may be increased in the cross-scan direction by using a staggered array as indicated in Figure 9. It is also possible to use a technique known as TDI (for Time Delay and Integration) in the scan direction.

In the case of a scanning sensor, a tradeoff must be made between image resolution and spatial integration. As the detector element size is increased, more spatial integration, and hence larger readout charge levels are obtained. However, even if the detector elements are staggered to maintain the sampling interval at $1/2f_c$, the output imagery is degraded by the blurring introduced with larger detector elements. This situation is illustrated in Figure 10 which shows the transfer function resulting from convoluting the optical image with detector elements dimensioned "a" along the scan direction and "b" along the cross-scan direction. The spatial integration (si) transfer function is:

$$T_{si}(f_{scan}, f_{cross}) = \frac{\sin(\pi a f_{scan}) \sin(\pi b f_{cross})}{(\pi^2 \text{ ab } f_{scan} f_{cross})}$$

The first zeros of T_{si} occur for $f_{scan} = 1/a$ and $f_{cross} = 1/b$ respectively.

Since image information is lost at those spatial frequencies where the transfer function is very small or goes to zero, we must satisfy

$$1/a \geq f_c \text{ and } 1/b \geq f_c$$

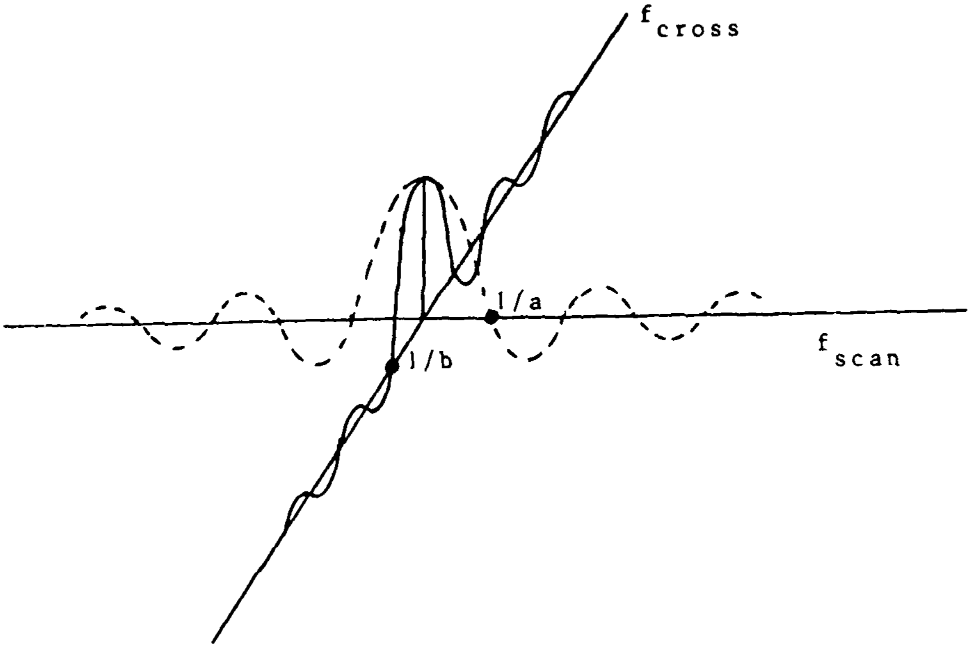


Figure 10. Spatial integration transfer function $T_{si}(f_{scan}, f_{cross})$ for a scanning detector array with detectors dimensioned b along the cross-scan direction and a along the scan direction. The first zero of T_{si} occurs for $f_{scan} = (1/a)$ and $f_{cross} = (1/b)$.

This condition limits the size of the detector elements. In the cross-scan direction, the Nyquist condition on detector-to-detector spacing derived above is $1/2f_c$. Therefore the 2:1 interleaving shown in Figure 9 is acceptable, but further interleaving of 3:1 is prohibited since it introduces image blurring that cannot be compensated for after detection.

The selection of dimension "a", the detector length along the scan direction is complicated by the additional image blurring along the scan direction which is introduced by temporal integration on the detector elements during the time interval between charge readouts. For a temporal integration period of tau, the image is blurred by an additional convolution function, as shown in Figure 11. The spatial integration discussed above results in convolution by a rect function of width "a" in the scan direction. Then temporal integration results in an additional convolution by a rect function of width v(tau).

The transfer functions along the scan frequency direction for spatial integration over the detector width of "a", and temporal integration (ti) over a period of tau are given by

$$T_{si}\ (f_{scan}) = \frac{\sin(\pi a f_{scan})}{\pi a f_{scan}}, \text{ and}$$

$$T_{ti}\ (f_{scan}) = \frac{\sin(\pi v(tau) f_{scan})}{\pi v(tau) f_{scan}},$$

The product of these transfer functions give the transfer function for the scanning process along the scan direction. By the Nyquist criterion, $v(tau) \leq 1/2f_c$. Therefore $T_{ti}(f_c) = 2/\pi \doteq 0.637$. Since the image has no content at this spatial frequency, the degradation introduced by $T_{ti}(f_{scan})$ is almost negligible.

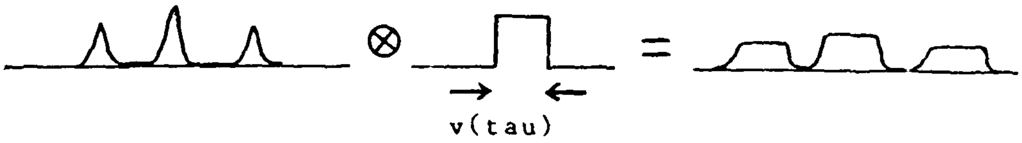


Figure 11. If a scanning detector moves at a speed v over the image, and samples at intervals of tau, the image is smeared by a rectangular convolution function of width v(tau).

Therefore we may use the same criterion for the degradation from spatial integration along f_{scan} as we chose along f_{cross}, i.e., a = $1/f_c$. Thus, a reasonable set of parameters is a = b = $1/f_c$, v(tau) = $1/2f_c$ and interleaved detectors along the cross-scan direction to achieve a detector-to-detector center spacing of $1/2f_c$. This selection of parameters will satisfy the Nyquist criterion and avoid any zeros in the spatial integration and temporal integration transfer functions within the bandlimit imposed by the imaging optics. However, there will be some degradation in the resolution along the scanning direction because of the additional transfer function in this direction.

5.0 Appendix on Image Sampling Theory for Hexagonal and Square Sampling Arrays

We show that hexagonal image sampling provides a reduction of 15% in the number of samples needed to reconstruct a band limited image without error, when compared with sampling with square arrays. We note parenthetically that all optical images are band limited.

The sampling operation can be thought of as multiplication of the image with a 2-D lattice of sampling points. Since multiplication in the spatial domain corresponds to convolution in the Fourier domain, it is possible to analyze the result of sampling by examining the convolution between the image Fourier transform and the sampling lattice Fourier transform. This correspondence is shown in Figure 12.

If the image has a cut-off frequency of f_c, and the Fourier transform of the sampling lattice a spacing of $2f_c$ in Fourier space, the convolution just satisfies the criterion that the image transform is not overlapped (aliased). See Figure 13.

So if any of the step functions shown in Figure 14 multiply the Fourier transform shown in Figure 13, nothing is lost from the original band limited function.

By the Fourier convolution theorem, multiplication in the Fourier domain corresponds to convolution in the spatial domain. So we conclude that if the square sampling lattice has a spacing of $(1/2f_c)$ or less, an interpolation function given by the transform of step functions like those shown in Figure 14 may be used to perfectly interpolate the original image from the discrete lattice of samples.

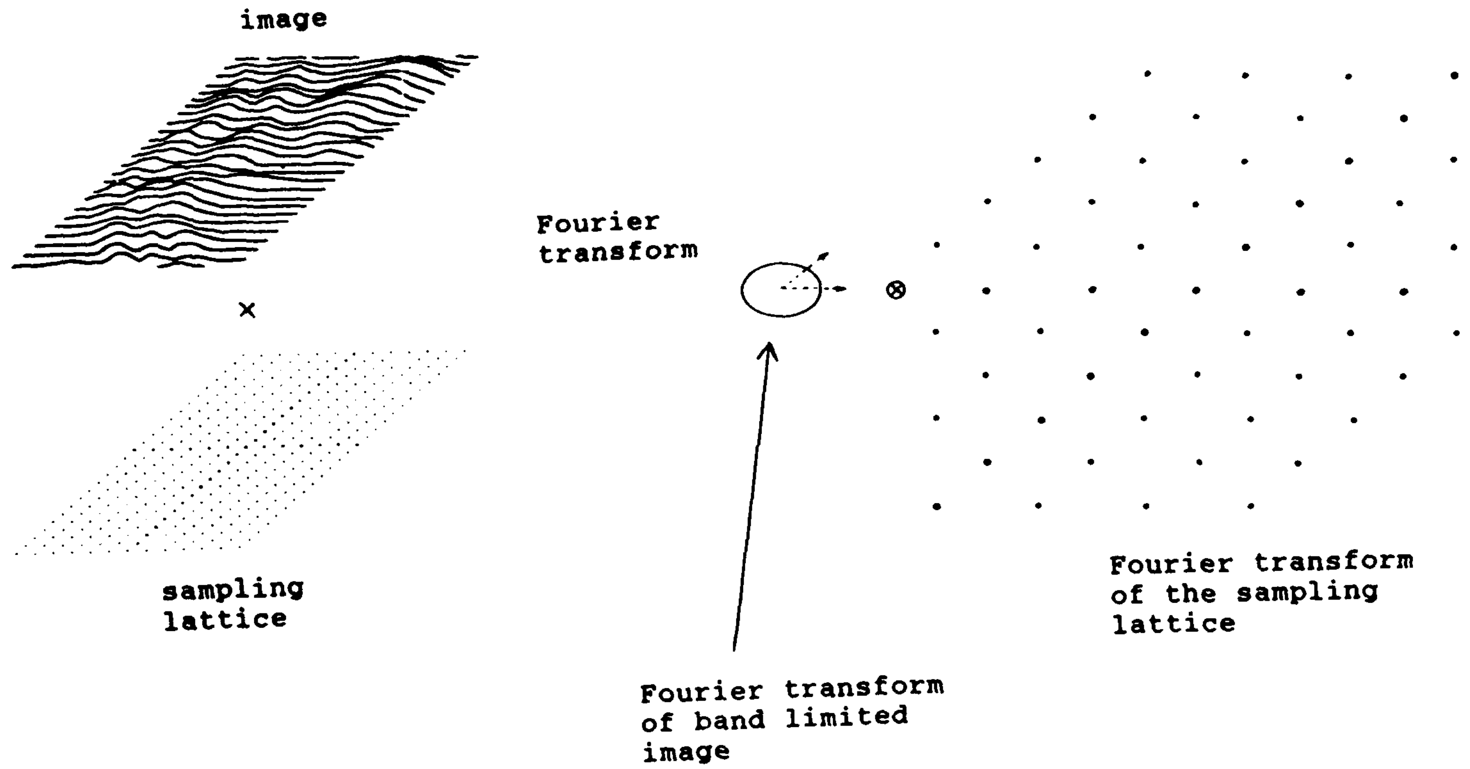


Figure 12. In image space, sampling is equivalent to multiplication by a bed of nails. In Fourier space, sampling is equivalent to convolution by the reciprocal bed of nails.

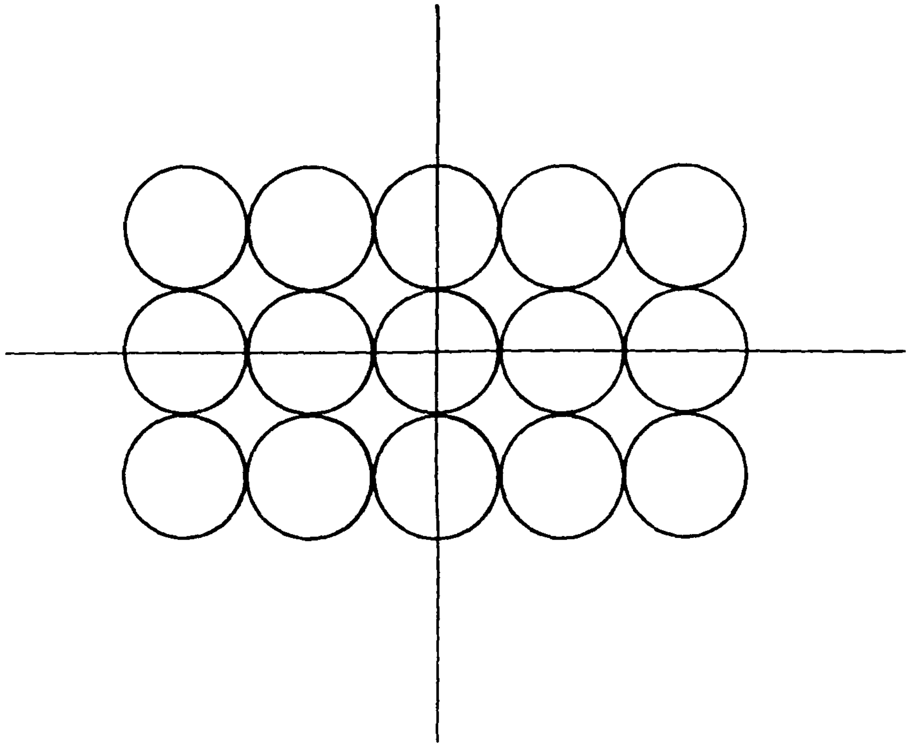


Figure 13. The sampled image has a transform given by the convolution of the original image transform by the reciprocal sampling lattice. The original image transform is recoveraible provided there is no overlap in the resultant transform.

To avoid overlap, the sampling lattice must be spaced closely enough that the reciprocal lattice is spaced widely enough to avoid overlap. This is called the Nyquist criterion, and the resultant sampled image transform is illustrated above.

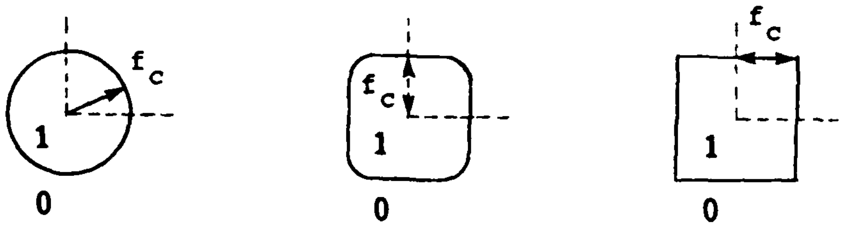


Figure 14. Outlines of step functions that are equal to 1.0 over the non-zero portion of the image transform.

The analogous sampling theorem for an hexagonal sampling array requires that the Fourier transform of the sampling lattice also avoid overlap of the image transform when the two functions are convolved. See Figure 15.

By transforming the frequency domain hexagonal lattice of nearest neighbor spacing $2f_c$, we obtain the maximally spaced hexagonal sampling array in the spatial domain. In the spatial domain, the nearest neighbor spacing is $(1/\sqrt{3})(1/f_c)$.

We can obtain this Fourier transform pair by thinking of the hexagonal array of nearest neighbor spacing $2f_c$ as the convolution of two line arrays of delta functions. The line arrays are oriented at 60 degrees at each other, and the spacing of delta functions along each array is $2f_c$. See Figure 16.

Then the Fourier convolution theorem may be used to express the transform of the Fourier domain hexagonal array as the product of two knife-edge gratings in the spatial domain, as shown in Figure 17.

The nearest neighbor spacing in the spatial domain lattice turns out to be $(1/\sqrt{3})(1/f_c)$, as shown in Figure 18.

The area of the hexagonally-shaped pixels that fill out this sampling array is $(1/2\sqrt{3})\ 1/(f_c^{\ 2})$. Compare this with the $(1/2\ f_c)^2 = 1/4(1/f_c)^2$ area of the square pixels filling out the maximally separated square array. The ratio is $(2/\sqrt{3}) = 1.1547$.

For detection of point targets, this means that in the hexagonal array the most centrally located pixel will collect roughly 15% more energy from the point image than the most centrally located pixel in a square array. Consequently, the signal-to-noise ratio for point target detection will be more favorable if hexagonal

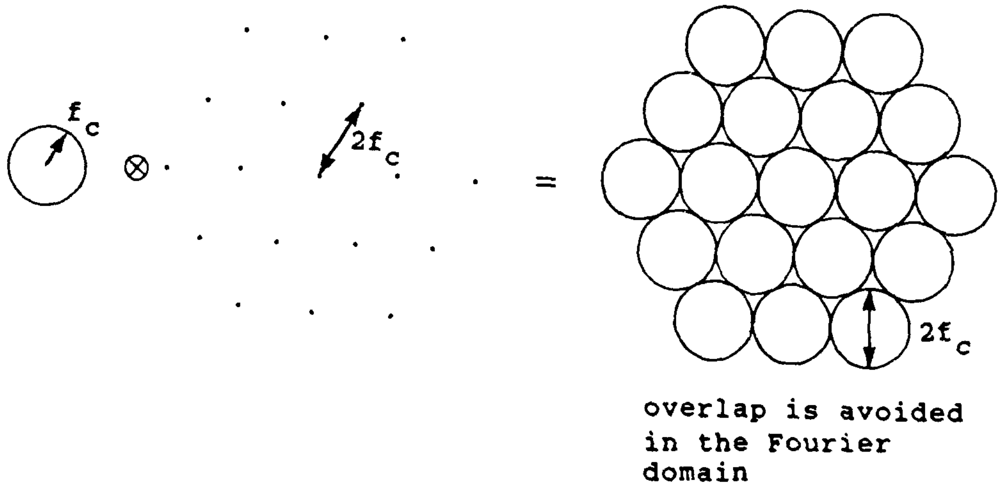


Figure 15. If the image is sampled by an hexagonal array, the reciprocal lattice is also hexagonal. If the sampling satifies Nyquist, overlap is avoided in the Fourier domain.

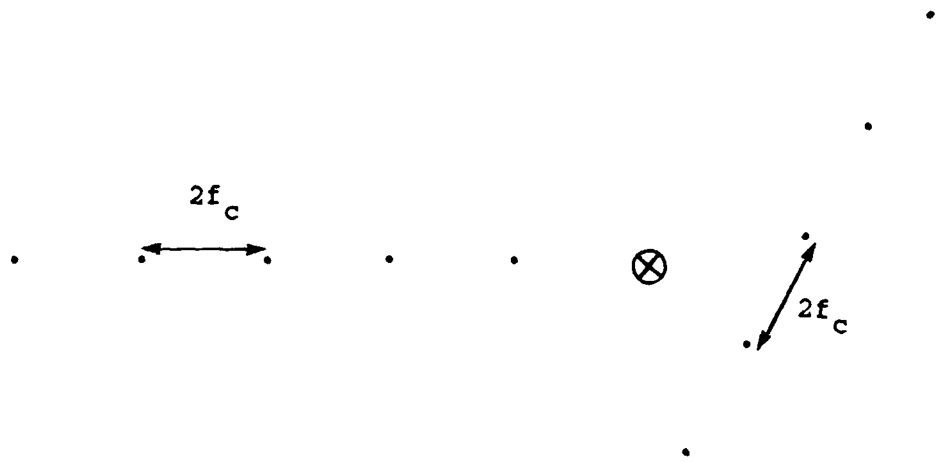


Figure 16. The convolution of two line arrays of delta functions oriented at 60 degrees to each other produces an hexagonal array of delta functions. This representation is handy for finding the Fourier transform of an hexagonal array.

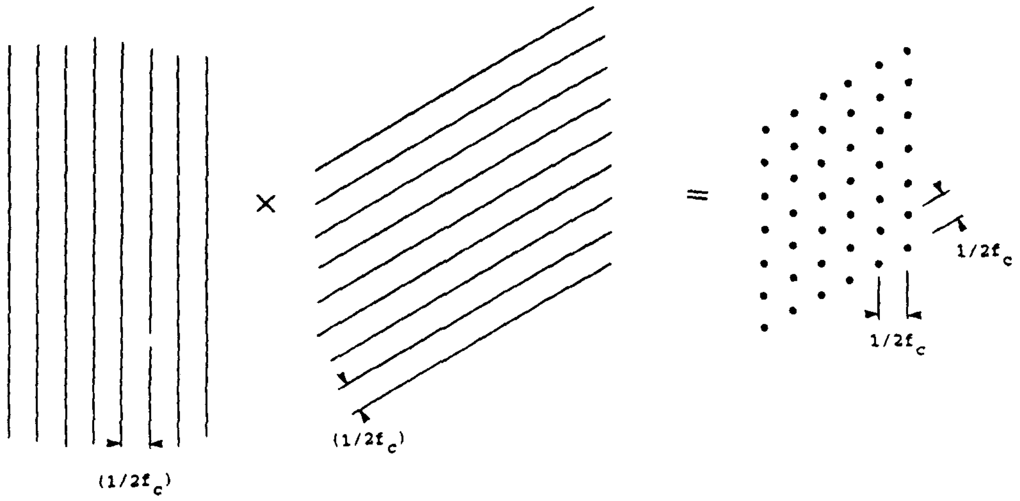


Figure 17. The product of two knife edge arrays, oriented at 60 degrees to each other produces an hexagonal array of delta functions.

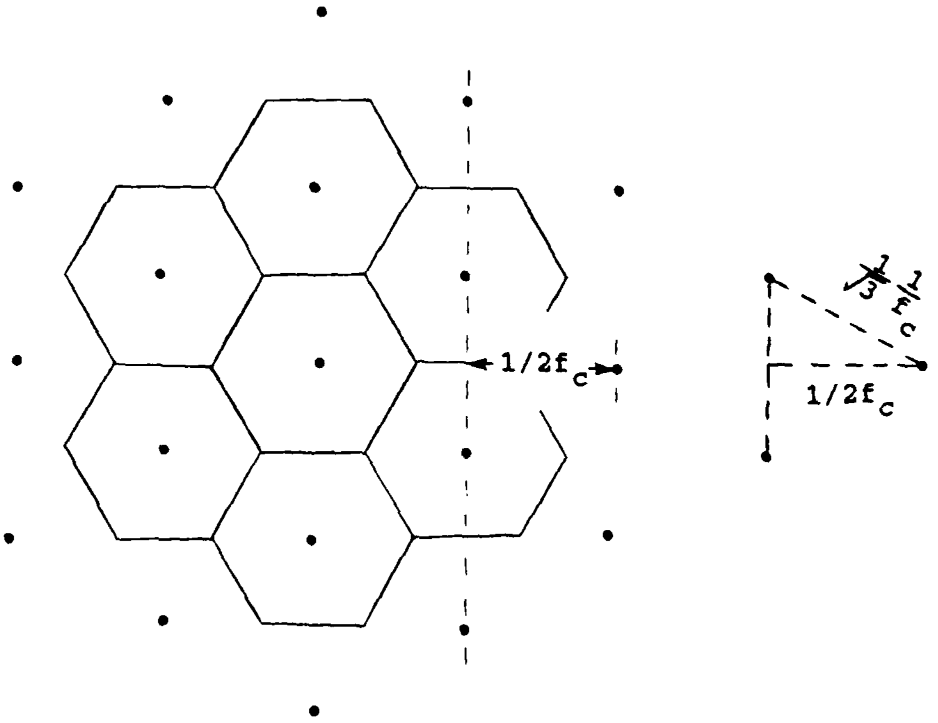


Figure 18. Calculation of the nearest neighbor spacing for the hexagonal sampling lattice which satifies Nyquist. The nearest neighbor spacing is $1/\sqrt{3}f_c$. The area of the hexagons that fill out the array is given by $(1/2\sqrt{3})(1/(f_c^2))$.

sampling is used rather than square sampling. There will also be only 0.866 as many pixels to read out of the detector array, and so less power is dissipated near the focal plane by A/D conversion. Finally, we have every reason to suspect that the subsequent higher level processing will also be reduced, since fewer pixels are needed to cover a given image area.