INTERNATIONAL STANDARD

ISO 12233

First edition
2000-09-01

Photography — Electronic still-picture cameras — Resolution measurements

Photographie — Appareils de prises de vue électroniques — Mesurages de la résolution

Reference number
ISO 12233:2000(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

International Standard ISO 12233 was prepared by Technical Committee ISO/TC 42, *Photography*.

Annex C forms a normative part of this International Standard. Annexes A, B and D are for information only.

0 Introduction

0.1 Purpose

The spatial resolution capability is an important attribute of an electronic still-picture camera. Resolution measurement standards allow users to compare and verify spatial resolution measurements. This International Standard defines terminology, test charts and test methods for performing resolution measurements for analog and digital electronic still-picture cameras.

0.2 Technical background

One of the most important characteristics of an electronic still-picture camera is the ability of the camera to capture fine detail found in the original scene. This ability to resolve detail is determined by a number of factors, including the performance of the camera lens, the number of addressable photoelements in the optical imaging device, and the electrical circuits in the camera, which may include image compression and gamma correction functions. Different measurement methods can provide different metrics to quantify the resolution of an imaging system, or a component of an imaging system, such as a lens. Resolution measurement metrics include resolving power, limiting resolution (at some specified contrast), spatial frequency response, MTF and OTF.

The first step in measuring resolution is to capture an image of a suitable test-chart with the camera under test. The test chart should include patterns with sufficiently fine detail, such as edges, lines, square waves, or sine wave patterns. The test chart defined in this International Standard has been designed specifically to evaluate electronic still-picture cameras. It has not been designed to evaluate other electronic imaging equipment such as input scanners, CRT displays, hard-copy printers, or electrophotographic copiers, nor individual components of an electronic still-picture camera, such as the lens.

The resolution measurements described in this International Standard are performed in the digital domain, using digital analysis techniques. For electronic still-picture cameras that include only analog outputs, the analog signal needs to be digitized, so that the digital measurement can be performed. The digitizing equipment is characterized, so that the effects of the digitization process can be removed from the measurement results. When this is not possible, the type of digitizing equipment used shall be reported along with the measurement results.

The spatial frequency response (SFR) measurement method described in this International Standard uses a computer algorithm to analyse digital image data from the electronic still-picture camera. Digitized image values near slanted vertical and horizontal black to white edges are digitized and used to compute the SFR values. The use of a slanted edge allows the edge gradient to be measured at many phases relative to the image sensor photoelements, in order to eliminate the effects of aliasing. This technique is mathematically equivalent to performing a moving knife edge measurement.

Photography — Electronic still-picture cameras — Resolution measurements

1 Scope

This International Standard specifies methods for measuring the resolution of electronic still-picture cameras. It is applicable to the measurement of both monochrome and colour cameras which output digital data or analog video signals.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 7589:1984, *Photography — Illuminants for sensitometry — Specifications for daylight and incandescent tungsten.*

ISO 14524:1999, Photography — Electronic still-picture cameras — Methods for measuring opto-electronic conversion functions (OECFs).

3 Terms and definitions

For the purposes of this International Standard, the following terms and definitions apply.

3.1
addressable photoelements
number of active photoelements in an image, which is equal to the number of active lines of photoelements multiplied by the number of active photoelements per line

NOTE It is possible that the number of addressable photoelements may be different for the different colour records of an image. When the signal values of the photoelements are digitized, the digitized code values may be referred to as picture elements, or pixels.

3.2
aliasing
output image artifacts that occur in a sampled imaging system for input images having significant energy at frequencies higher than the Nyquist frequency of the system

3.3
cycles per millimetre
cy/mm
unit used for specifying resolution characteristics in terms of the response of an imaging system to a linear radiance sine wave input, as a function of the frequency of the sine wave

NOTE 1 A range of input sine wave frequencies is obtained in this International Standard through the use of a sharp edged target.

NOTE 2 Most pictorial imaging systems exhibit non-linear behaviour, which may result in the nature of the target affecting the measured resolution characteristics. Distance units other than millimetres may also be used.

3.4
aliasing ratio
value equal to the "maximum minus minimum" modulation divided by the "average" modulation of an electronic still-picture camera when imaging a frequency burst of constant spatial frequency

NOTE The aliasing ratio is described in 6.4.

3.5
edge spread function
ESF
normalized spatial signal distribution in the linearized output of an imaging system resulting from imaging a theoretical infinitely sharp edge

3.6
effectively spectrally neutral
having spectral characteristics which result in a specific imaging system producing the same output as for a spectrally neutral object

3.7
electronic still-picture camera
camera incorporating an image sensor that outputs an analog or digital signal representing a still-picture, or records an analog or digital signal representing a still-picture on a removable media, such as a memory card or magnetic disc

3.8
gamma correction
process that alters the image data in order to modify the tone reproduction

3.9
horizontal resolution
resolution value measured in the longer image dimension, corresponding to the horizontal direction for a "landscape" image orientation, typically using a vertically oriented test-chart feature

3.10
image aspect ratio
ratio of the image width to the image height

3.11
image compression
process that alters the way digital image data is encoded in order to reduce the size of an image file

3.12
image sensor
electronic device that converts an optical image into an electronic signal; for example a charge coupled device (CCD) array

3.13
limiting resolution
value of that portion of a specified resolution test pattern, measured in line widths per picture height, that corresponds to an average modulation value equal to some specified percentage of the modulation value at a specified reference frequency

EXAMPLE The limiting resolution may be the test pattern value, in line widths per picture height (LW/PH), corresponding to a camera output modulation level of 5 % of the camera output modulation level at a reference frequency of 10 LW/PH.

3.14
line pairs per millimetre
lp/mm
metric for specifying resolution in terms of the number of equal width black and white line pairs per millimetre that can be resolved according to some criterion, such as visual resolution or limiting resolution

NOTE distance units other than millimetres may also be used.

3.15
lines per millimetre
lines/mm
metric for specifying resolution in terms of the number of equal-width black and white lines per millimetre that can be resolved according to some criterion, such as visual resolution or limiting resolution

NOTE Distance units other than millimetres may also be used.

3.16
line spread function
LSF
normalized spatial signal distribution in the linearized output of an imaging system resulting from imaging a theoretical infinitely thin line

NOTE If the imaging system is operating in an isoplanatic region and in its linear range, the LSF is equal to the first derivative of the ESF.

3.17
line widths per picture height
LW/PH
metric for specifying the width of a solid line on a test chart, relative to the height of the active area of the chart, which is equal to the height of the active area of the test chart divided by the width of a black line, that is equal to the total number of lines of the same width which can be placed edge to edge within the height of a test target, or within the vertical field of view of a camera

NOTE If the height of the active area of the chart equals 20 cm, a black line of 1 000 LW/PH has a width equal to 20/1 000 cm.

3.18
linearized
digital signal conversion performed to invert the camera opto-electronic conversion function (OECF) so that the resulting signal is approximately linearly proportional to the scene luminance

3.19
modulation
difference between the minimum and maximum signal levels divided by the sum of these levels

3.20
modulation transfer function
MTF
modulus of the optical transfer function

3.21
normalized spatial frequency
unit used for expressing spatial frequency response, where the distance dimension has been removed by multiplying the spatial frequency in cycles per millimetre by the sampling period in millimetres

NOTE Normalized spatial frequency is particularly appropriate for comparing the spatial frequency response of imaging systems where the rendering magnification is unknown, and the total number of samples is equal.

3.22
Nyquist limit
spatial frequency equal to 1/2 times the inverse of the sampling period

NOTE Energy at an input spatial frequency above the Nyquist limit will alias to a spatial frequency below the Nyquist limit in the output image. The Nyquist limit may be different in the two orthogonal directions.

3.23
optical transfer function
OTF
two-dimensional Fourier transform of the imaging system's point spread function

NOTE 1 For the OTF to have significance, it is necessary that the imaging system be operating in an isoplanatic region and in its linear range.

NOTE 2 The OTF is a complex function whose modulus has unity value at zero spatial frequency.

3.24
point spread function
PSF
normalized spatial signal distribution in the linearized output of an imaging system resulting from imaging a theoretical infinitely small point source

3.25
resolution
measure of the ability of a camera system, or a component of a camera system, to depict picture detail

NOTE Resolution measurement metrics include resolving power, limiting resolution, spatial frequency response (SFR), MTF and OTF.

3.26
sampling aspect ratio
ratio of the sample spacing in the two orthogonal sampling directions

NOTE If the sample spacing is equal, the aspect ratio of the sampling grid is 1:1 or "square", so that the sampling aspect ratio provides "square pixels".

3.27
sampled imaging system
imaging system or device which generates an image signal by sampling an image at an array of discrete points, or along a set of discrete lines, rather than a continuum of points

NOTE The sampling at each point is done using a finite size sampling aperture or area.

3.28
sample spacing
physical distance between sampling points or sampling lines

NOTE The sample spacing may be different in the two orthogonal sampling directions.

3.29
spatial frequency response
SFR
measured amplitude response of an imaging system as a function of relative input spatial frequency

NOTE 1 The SFR is normally represented by a curve of the output response to an input signal of unit amplitude, over a range of spatial frequencies.

NOTE 2 The SFR is normalized to yield a value of unity at a spatial frequency of 0.

3.30
spectrally neutral
test chart is spectrally neutral if the relative spectral power distributions of the incident and reflected (or transmitted) light are equal

3.31
test chart
arrangement of test patterns designed to test particular aspects of an imaging system

3.32
test pattern
specified arrangement of spectral reflectance or transmittance characteristics used in measuring an image quality attribute

The test pattern spectral characteristics include the following types:

3.32.1
bi-tonal pattern
pattern that is spectrally neutral or effectively spectrally neutral, and consists exclusively of two reflectance or transmittance values in a prescribed spatial arrangement

NOTE Bi-tonal patterns are typically used to measure resolving power, limiting resolution and SFR.

3.32.2
grey scale pattern
pattern that is spectrally neutral or effectively spectrally neutral, and consists of a large number of different reflectance or transmittance values in a prescribed spatial arrangement

NOTE Grey scale patterns are typically used to measure opto-electronic conversion functions.

3.32.3
spectral pattern
pattern that is specified by the spatial arrangement of features with differing spectral reflectance or transmittance values

NOTE Spectral patterns are typically used to measure colour reproduction.

3.33
vertical resolution
resolution value measured in the shorter image dimension, corresponding to the vertical direction for a "landscape" image orientation, typically using a horizontally oriented test-chart feature

3.34
visual resolution
spatial frequency at which the individual black and white lines of a test pattern reproduced on a display or print can no longer be distinguished by human observers, or are reproduced at a spatial frequency lower than the spatial frequency of the corresponding area of the test chart, as a result of aliasing

4 Test chart

4.1 Introduction

A reproduction of a test chart for measuring the resolution of an electronic still-picture camera is shown in Figure 1. Figure 2 is a diagram showing the locations of particular features of the test chart, which may be either a reflective or transmissive chart. The purpose of each test pattern element is listed in Table 1. A complete spreadsheet specification of the test chart is given for information in annex A.

The measurements described in clause 6 can be performed using test charts other than the chart shown in Figure 1. For example, the test patterns present in the test chart can be separated and rearranged, if required, for specific test objectives. The framing and the reproduction scale of the patterns can also be varied, if required, for specific test objectives.

The chart shown in Figure 1 is designed to measure cameras having a limiting resolution of less than 2 000 LW/PH. Nevertheless, it is possible to use the chart to measure the visual resolution or limiting resolution of an electronic still camera having a limiting resolution greater than 2 000 LW/PH. This is accomplished by adjusting the camera to target distance, or the focal length of the camera lens, so that the test-chart active area fills only a fraction of the vertical image height of the camera. This fraction is then measured in the digital image, by dividing the number of image lines in the camera image by the number of lines in the active chart area. The values of all test-chart features, in LW/PH, printed on the chart or specified in this International Standard, are multiplied by this fraction, to obtain their correct values. For example, if the chart fills 1/4 of the vertical image height of the camera, a feature labelled as 1 000 LW/PH on the chart corresponds to 4 000 LW/PH using this chart framing.

4.2 Material

The test chart may be either a transparency that is rear illuminated, or a reflection test card that is front illuminated. A reflection chart shall have an approximately Lambertian base material. A transparency chart shall be rear illuminated by a diffuse source.

4.3 Size

The active height of reflection test charts should be not less than 20 cm. The active height of transparencies shall be not less than 10 cm. The chart should have a 16:9 image aspect ratio, with indicators for 1:1, 4:3 and 3:2 image aspect ratios.

4.4 Test patterns

The test chart shall have bi-tonal patterns and should be spectrally neutral.

NOTE Use of bi-tonal test charts provides the sharpest possible features and minimizes the cost of producing the chart.

4.5 Test-pattern modulation

For reflectance charts, the ratio of the maximum chart reflectance R_{max} to the minimum chart reflectance R_{min} for large test-pattern areas should be not less than 40:1 and not greater than 80:1, and shall be reported if it is outside this range. For transmissive charts, the ratio of the maximum chart transmittance T_{max} to the minimum chart transmittance T_{min} for a large test pattern should be not less than 40:1 and not greater than 80:1, and shall be reported if it is outside this range. Modulation ratios for the finer test-chart features, relative to the ratio for large test-pattern areas, should preferably be reported by the chart manufacturer as described in annex B, so that these values may be used to correct the SFR values measured using the chart.

4.6 Units

All test-chart features are specified in units of line widths per picture height (LW/PH), where the height is the active image distance in the shorter test-chart dimension.

NOTE This allows measurements to be reported using units that are independent of the sample spacing and the image aspect ratio.

4.7 Features

The test chart should include horizontal, vertical and diagonally oriented hyperbolic wedges, sweeps and tilted bursts. It may also include a circle and long, slightly slanted lines to measure geometric linearity (distortion).

NOTE The finest features are 2 000 LW/PH, which is equivalent to 1 000 line pairs per picture height.

4.8 Positional tolerance

The position of any test-chart feature shall be proportional to the values given in normative annex C and shall be reproduced with a tolerance of ± 1/1 000 picture heights (equivalent to ± 1/10 % of the active test-chart height). In addition, the width and duty cycle ratio of each feature (white or black line) of J, JS, K, KS, O and P in Figure 2 shall be reproduced with a tolerance of ± 5 % of the feature width.

This tolerance requires all test-chart features to be accurately located to with ± 2/10 mm for a test-chart height of 20 cm. In addition, the width of the white or black lines comprising the 2 000 LW/PH portion of K1 are required to fall within the range of 1/2 000 ± 5 % of the actual test-chart height, equal to 95/1 000 mm to 105/1 000 mm for a test-chart height of 20 cm.

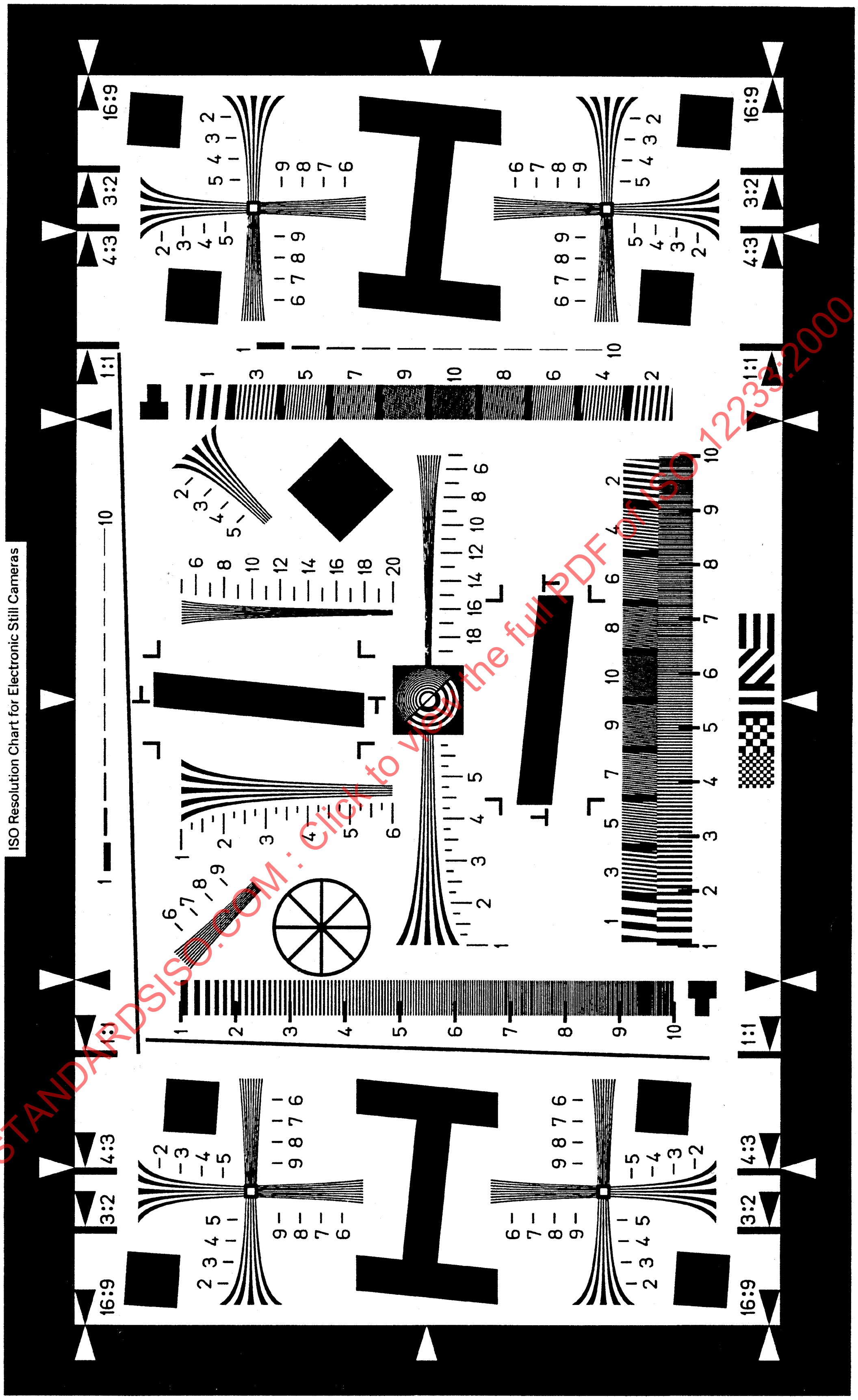


Figure 1 — Resolution test chart

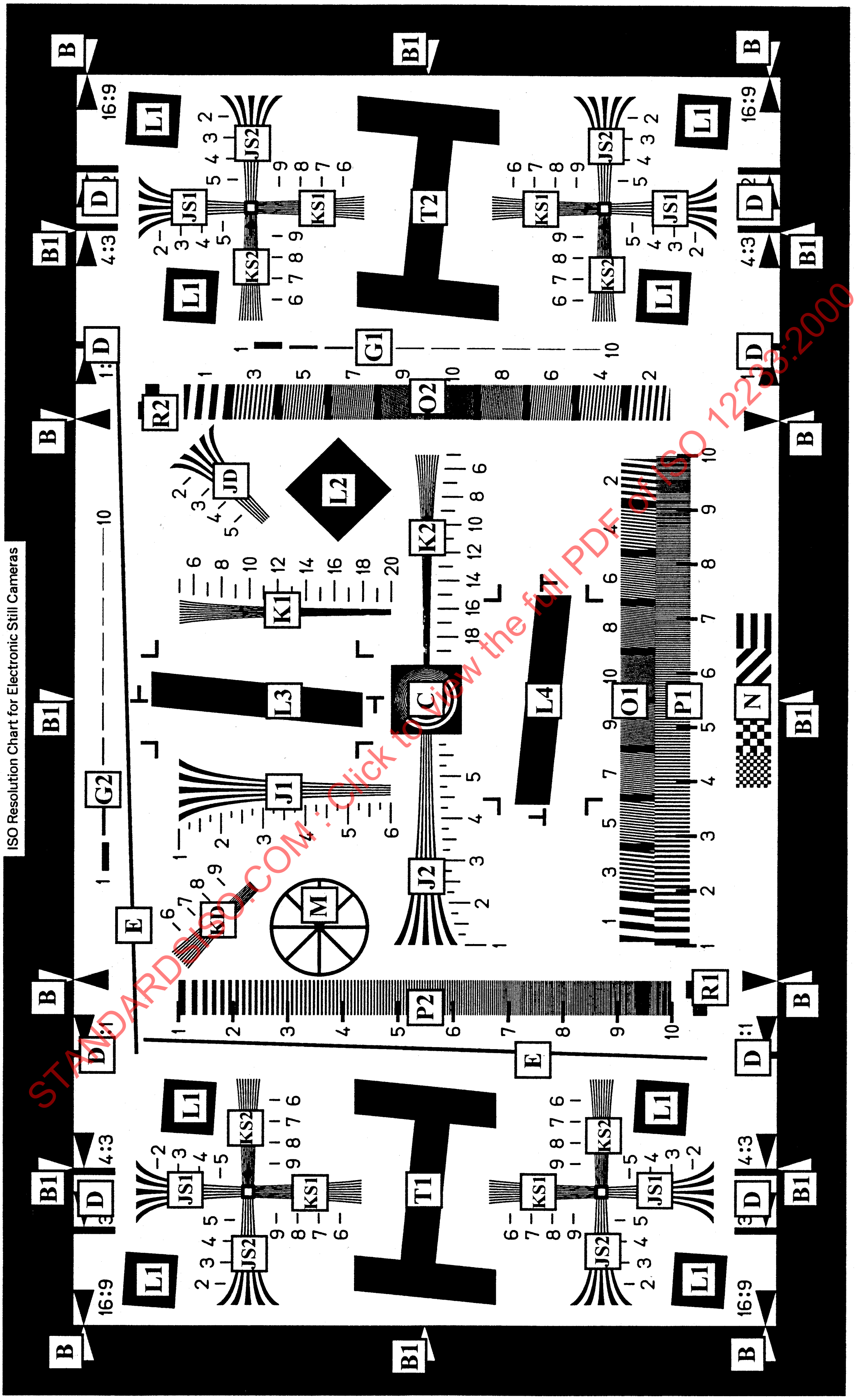


Figure 2 — Test-chart features

Table 1 — Test-chart elements

Element	Purpose
A	Black border with inner edge which defines active target area
B [a]	Black and white framing arrows used to frame target vertically Used for horizontal framing only for 16:9 aspect ratio image formats
B1	White framing arrows used to assist in framing target
C [a]	Centre dual-frequency zone plate inside black square used to set focus
D [a]	Framing lines and arrows that define 1:1, 4:3 and 3:2 aspect ratios
E [a]	Slightly slanted lines used to check scan linearity and "stair stepping"
G1 [a]	100 LW/PH to 1 000 LW/PH black bars to measure horizontal pulse response
G2 [a]	100 LW/PH to 1 000 LW/PH black bars to measure vertical pulse response
J1	100 LW/PH to 600 LW/PH hyperbolic zone plate used to measure centre horizontal visual resolution
J2	100 LW/PH to 600 LW/PH hyperbolic zone plate used to measure centre vertical visual resolution
JS1 [a]	100 LW/PH to 600 line width hyperbolic zone plate used to measure corner horizontal visual resolution
JS2 [a]	100 LW/PH to 600 line width hyperbolic zone plate used to measure corner vertical visual resolution
K1	500 LW/PH to 2 000 LW/PH hyperbolic zone plate used to measure centre horizontal visual resolution
K2	500 LW/PH to 2 000 LW/PH hyperbolic zone plate used to measure centre vertical visual resolution
KS1 [a]	500 LW/PH to 1 000 line width hyperbolic zone plate used to measure corner horizontal visual resolution
KS2 [a]	500 LW/PH to 1 000 line width hyperbolic zone plate used to measure corner vertical visual resolution
L1 [a]	Slightly slanted (approx. 5°) small black squares used to measure vertical and horizontal SFR at extreme corners of image
L2 [a]	45° diagonal black square used to measure diagonal SFR
L3	Slightly slanted (approx. 5°) black bar used to measure centre horizontal SFR
L4	Slightly slanted (approx. 5°) black bar used to measure centre vertical SFR
M [a]	Circle with cross and × used to observe scanning nonlinearities
N [a]	Checkerboard patterns used to observe image compression artifacts
O1	Tilted (approx. 5°) square wave bursts used to measure horizontal aliasing ratio
O2	Tilted (approx. 5°) square wave bursts used to measure vertical aliasing ratio
P1 [a]	100 to 1 000 line square wave sweep
P2 [a]	100 to 1 000 line square wave sweep
R [a]	Indicators that can be used for automatic target alignment
T1, T2 [a]	Slanted (approx. 5°) H-shaped bars used to measure SFR at far sides of image

[a] Indicates optional element.

5 Test conditions

5.1 Test-chart illumination

The luminance of the test chart shall be sufficient to provide an acceptable camera output signal level. The test chart shall be uniformly illuminated as shown in Figure 3, so that the luminance of any white area of the chart is within ± 10 % of the average luminance near the centre of the chart. The illumination sources should be baffled to prevent direct illumination of the camera lens by the illumination sources. The area surrounding the test chart should be of low reflectance, to minimize flare light. The chart should be shielded from any reflected light. The illuminated test chart shall be effectively spectrally neutral with respect to either the daylight or tungsten illuminants given in ISO 7589.

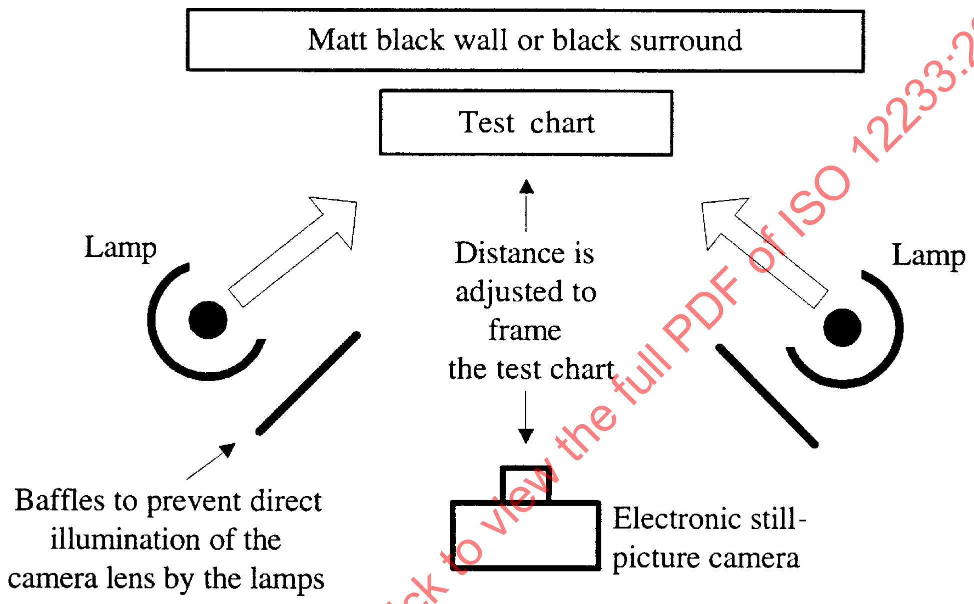


Figure 3 — Test-chart illumination method

5.2 Camera framing and lens focal-length setting

The camera shall be positioned to properly frame the test target. The vertical framing arrows are used to adjust the magnification and the horizontal arrows are used to centre the target horizontally. The tips of the centre vertical black framing arrows should be fully visible and the tips of the centre white framing arrows should not be visible. The target shall be oriented so that the horizontal edge of the chart is approximately parallel to the horizontal camera frame line. The approximate distance between the camera and the test chart should be reported along with the measurement results.

5.3 Camera focusing

The camera focus should be set by performing a series of image captures at varying focus settings, and selecting the focus setting that provides the highest average modulation level at a spatial frequency of about 1/4 the camera Nyquist frequency. Alternately, the camera focus may be set so that the zone plate in the centre of the chart exhibits the maximum aliasing possible.

5.4 Camera settings

The camera lens aperture (if adjustable) and the exposure time should be adjusted to provide a near maximum signal level from the white test target areas. The settings shall not result in signal clipping in either the white or black areas of the test chart, or regions of edge transitions.

Electronic still-picture cameras may include image compression, to reduce the size of the image files and allow more images to be stored. The use of image compression can significantly affect resolution measurements. Some cameras have switches that allow the camera to operate in various compression or resolution modes. The values of all camera settings that may affect the results of the measurement, including lens focal length, aperture and resolution or compression mode (if adjustable), shall be reported along with the measurement results.

Multiple SFR measurements may be reported for different camera settings, including a setting that uses the maximum lens aperture size (minimum f-number) and maximum camera gain.

5.5 White balance

For a colour camera, the camera white balance should be adjusted, if possible, to provide proper white balance [equal red, green and blue (RGB) signal levels] for the illumination light source, as specified in ISO 14524.

5.6 Luminance and colour measurements

Resolution measurements are normally performed on the camera luminance signal. For colour cameras that do not provide a luminance output signal, a luminance signal should be formed from an appropriate combination of the colour records, rather than from a single channel such as green.

5.7 Gamma correction

The signal representing the image from an electronic still-picture camera will probably be a non-linear function of the scene luminance values. Since the SFR measurement is defined on a linearized output signal, and such a non-linear response will affect SFR values, the signal shall be linearized before the data analysis is performed. Linearization is accomplished by applying the inverse of the camera OECF to the output signal via a lookup table or appropriate equation. The measurement of the OECF shall be as specified in ISO 14524, using the standard reflection camera OECF test chart.

6 Test measurements

6.1 Visual resolution

The visual resolution is the lowest value of the test pattern, in LW/PH, where the individual black and white lines can no longer be distinguished, or are reproduced at a spatial frequency lower than the spatial frequency of the corresponding area of the test chart, as a result of aliasing. The test chart includes vertical, horizontal, and diagonal hyperbolic wedges near the centre and corners of the target, labelled as K1, K2, KS1, KS2, KD, J1, J2, JS1, JS2 and JD in Figure 2, which are used to perform this test.

The centre horizontal visual resolution is determined by observing the vertically oriented hyperbolic wedges labelled J1 and K1 in Figure 2, and the centre vertical visual resolution is determined by observing the horizontally oriented hyperbolic wedges labelled J2 and K2. The 45° diagonal visual resolution is determined using the diagonally oriented hyperbolic wedges labelled KD and JD. The four corner horizontal and vertical visual-resolution values may be determined using the appropriate corner wedges, labelled JS1, JS2, KS1 and KS2.

To determine the visual resolution, the image of the test target is reproduced on a monitor or hard-copy print, and the visual resolution is subjectively judged. Observers should be well acquainted with the appearance of aliasing, so that they do not seriously misjudge the visual resolution of the camera. The visual-resolution value shall not exceed the Nyquist limit.

To perform the test properly, the monitor or hard-copy print shall have sufficient resolution so as not to affect the camera visual-resolution measurement. A preferred method for reducing the effect of the output device is to digitally magnify the portions of the image representing the hyperbolic wedges by replicating lines and pixels, so that the number of addressable pixels used to display this portion of the image is an integer multiple of the number of addressable pixels of digital data provided from the electronic still-picture camera.

6.2 Limiting resolution

The limiting resolution is the value, in LW/PH, of that portion of a black and white resolution wedge where the resolution response (average depth of modulation value) equals 5 % of the reference response. The test chart includes vertical and horizontal square wave sweeps, labelled as P1 and P2 in Figure 2, which are used to perform this measurement. For P1, the reference response is defined as the difference between the signal values from the slanted black bar L4 and the white region just below the L4 bar. For P2, the reference response is defined as the difference between the signal values from the right vertical slanted black bar of T1 and the white region just to the right of this bar. The sweeps include fiducial marks labelled a1 to 10, which correspond to 100 LW/PH to 1 000 LW/PH. Care should be taken in determining the 5 % level to avoid the influence of noise. If necessary, multiple images should be averaged to reduce the noise level.

6.3 Spatial frequency response

6.3.1 Introduction

The spatial frequency response (SFR) of an electronic still-picture camera is measured by analysing the camera data near a slanted black to white edge. For the target shown in Figure 2, the black L3 bar shall be used to measure the horizontal SFR and the black L4 bar shall be used to measure the vertical SFR, in the centre of the image. The diagonal black square L2 shall be used to measure the diagonal SFR near the centre of the image. The L1 and L4 test patterns may be used to measure the SFR at other locations in the image.

The SFR measurement can be performed automatically by image-processing software, as part of an easy to use image processing or analysis software package[1)]. To perform the measurement, the digital camera output data in the region of specified black to white and white to black edges on the test chart are analysed by a defined numerical algorithm. If the camera provides only analog output signals, the signals shall be digitized by a suitable analog to digital converter and stored in a suitable memory to allow the data to be analysed by the algorithm.

6.3.2 SFR algorithm

The SFR algorithm is given in normative annex C and shown in flow-chart form in Figure 4. A diagram depicting the key steps of the SFR algorithm is shown in Figure 5. A sample C-code is given for information in annex D.

The algorithm can automatically compute the SFR, using image data from a user-defined rectangular region of the image which represents a vertically oriented black to white or white to black slightly slanted edge, depicting a "horizontal" transition. To measure the vertical SFR, a horizontally oriented edge is used, and the digital image data is rotated 90° before performing the calculation. The user then selects the region containing the slightly slanted edge. If the image is coloured, a luminance record is created before the SFR calculation is performed. The image code values shall then be linearized by inverting the opto-electronic conversion function (OECF) of the camera. The opto-electronic conversion function shall be measured as specified in ISO 14524.

Next, for each line of pixels perpendicular to the edge, the derivative of the linearized image data is computed using a "1/2, + 1/2" finite impulse response (FIR) filter, meaning that the derivative value for pixel "X" is equal to –1/2 times the value of the pixel immediately to the left, plus 1/2 times the value of the pixel to the right. The centroid of this derivative is calculated to determine the position of the edge on each line. A best-line fit to the centroid locations is then calculated. Error messages shall be reported if any centroid is within 2 pixels of either side of the input image edges, or if the edge does not contain at least 20 % modulation. The number of lines used in the analysis shall be truncated to provide an equal number of lines at each phase of the edge position relative to the horizontal centre of the pixel. This may be done by keeping the largest integer number of phase rotations within the block, and deleting any remaining rows at the bottom of the block. A one-dimensional supersampled line spread function shall be formed using the derivatives of the truncated two-dimensional image data. Using the first line as reference points, the data points from all the other lines shall be placed into one of four "bins" between these

1) For example, as part of the development of this standard, the SFR algorithm, written in the C programming language, has been incorporated in a plug-in for Adobe Photoshop. Photoshop is the trade-name of a product supplied by Adobe Systems Incorporated. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by ISO of the product named. Equivalent products may be used if they can be shown to lead to the same results.

reference points, according to the distance from the edge for that particular line. This creates a single supersampled "composite" line spread function, having four times as many points along the line as the original image data. The line spread function shall be multiplied by a Hamming window, to reduce the effects of noise by reducing the influence of pixels at the extremes of the window, which have response due to noise but little response due to the image edge located at the centre of the window. The discrete Fourier transform (DFT) of the windowed line spread function shall be calculated. The SFR is the normalized modulus of the DFT of the line spread function. The SFR shall be reported as defined in 7.1.

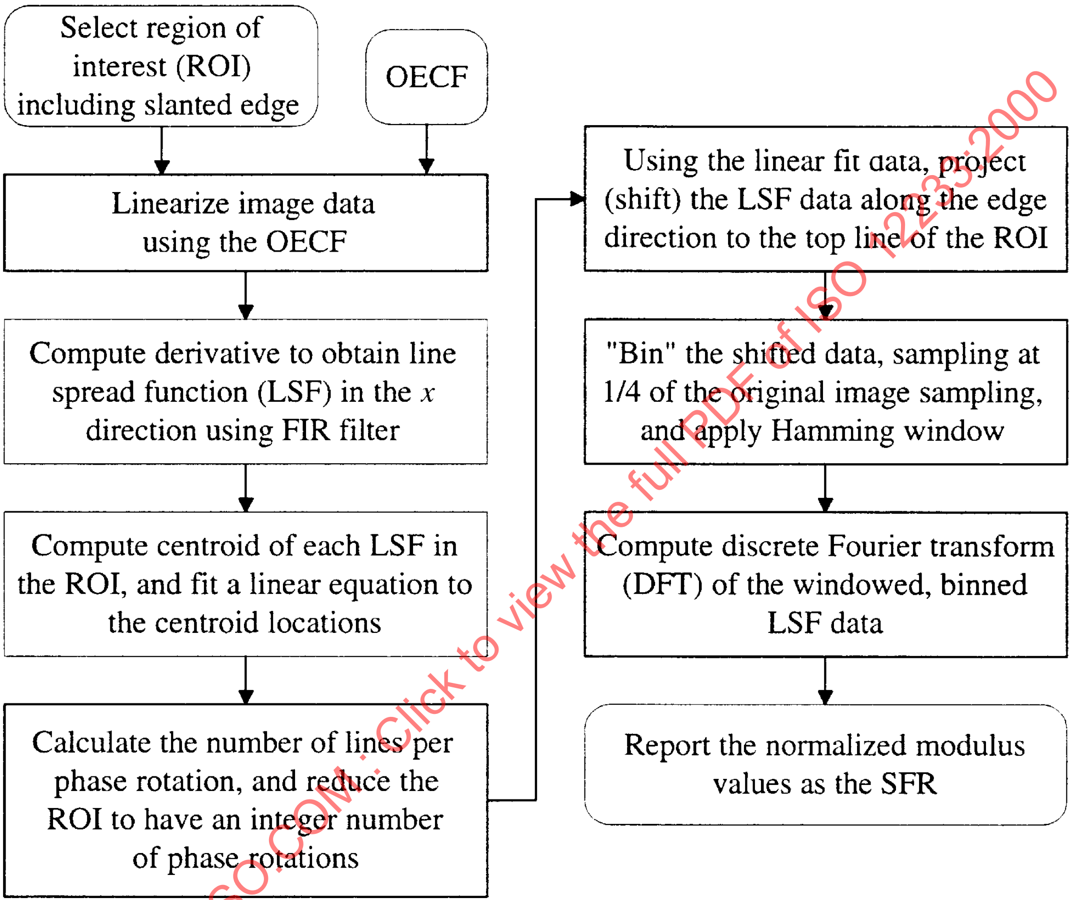


Figure 4 — Flow-chart of SFR measurement algorithm

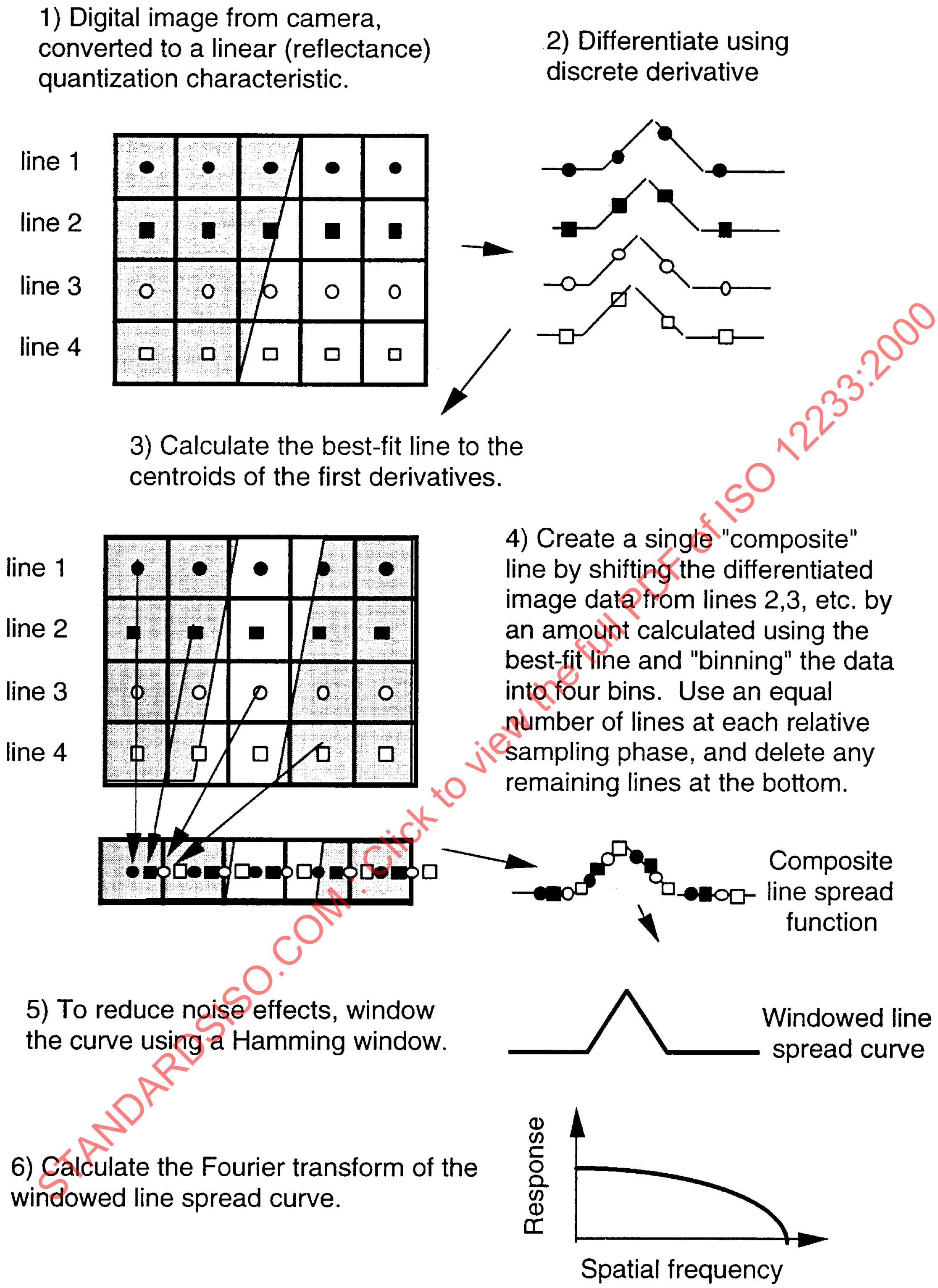


Figure 5 — Key steps of SFR measurement algorithm

6.4 Aliasing ratio

The aliasing level is measured using the horizontal and vertical 100 LW/PH to 1 000 LW/PH slanted burst patterns labelled O1 and O2 (see Figure 2).

In the absence of aliasing, the signal level from each black bar, or each white bar, should be identical for each bar in the burst. Because of aliasing, however, the signal responses of the camera to the white bars of a particular burst may not be identical. Instead, the response may have a maximum signal value for some white bars, and a minimum signal value for other white bars.

The ratio of the "maximum minus the minimum response" for the white bars within a burst to the "average modulation level" (equal to the average white-bar signal minus the average black-bar signal) within the burst provides the aliasing ratio for that particular spatial frequency burst.

7 Presentation of results

7.1 Introduction

The results of the resolution measurements shall be reported as described below. The following information shall be reported along with the measurement results.

a) The values of all camera settings that may affect the results of the measurement, including lens focal length and aperture, and resolution or compression mode (if adjustable).

b) Whether or not dark-field and flat-field correction was used.

c) The illuminating source colour temperature and illumination level.

d) For cameras equipped with interchangeable lenses, the type and characteristics of the lens used in the tests.

If possible, the effect of the lens and the test target should be removed mathematically, and the calculated camera measurements without a lens should also be reported.

7.2 Visual resolution

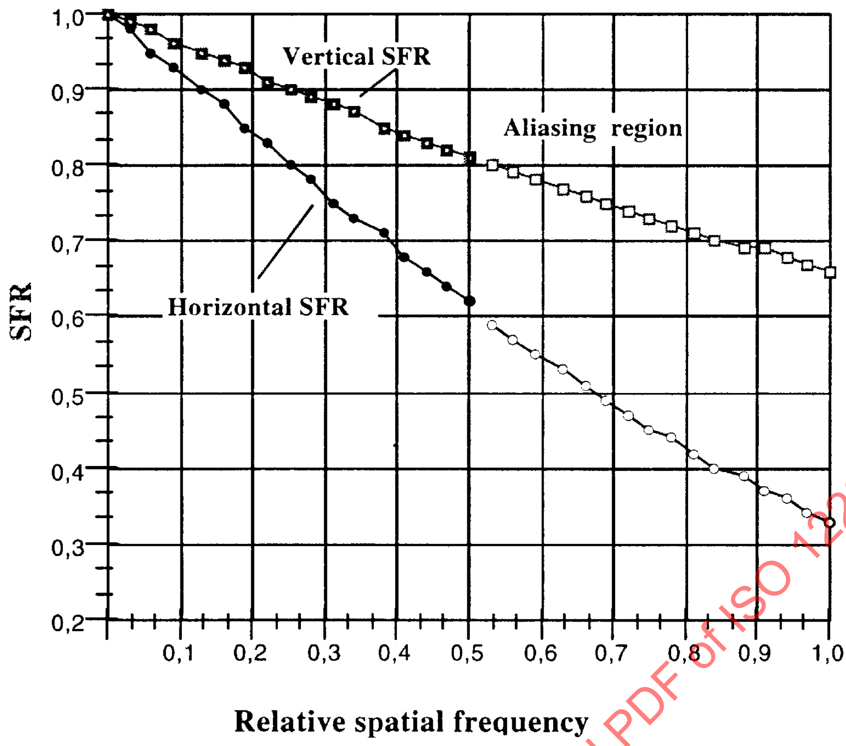
The visual-resolution values shall be reported as spatial frequency values, in LW/PH, for the horizontal, vertical and diagonal directions.

7.3 Limiting resolution

The limiting resolution values shall be reported as spatial frequency values, in LW/PH, for the horizontal and diagonal directions.

7.4 Spatial frequency response (SFR)

The SFR results shall be reported using a graph plotting the modulation level (having a value of 1 at 0 spatial frequency) versus spatial frequency, or in a list of SFR values versus spatial frequency. The SFR values shall be reported separately for the horizontal and vertical directions. The values shall be the average of four SFR measurements of a black to white edge, and four SFR measurements of a white to black edge. The measurement location is the middle of the vertical "L4" bar, for the horizontal SFR, and the middle of the horizontal "L4" bar, for the vertical SFR. The spatial frequency axis should preferably be labelled with three units: frequency relative to the sensor sampling frequency, line widths per picture height (LW/PH), and cycles/mm on the sensor, or with equations representing the relationship between these units. There shall be a minimum of 32 equally spaced measurement values for spatial frequencies between 0 and the sensor sampling frequency. The camera Nyquist frequency shall be reported. Values between 1/2 and 1 times the sensor sampling frequency shall be marked so as to indicate that these spatial frequencies lead to aliasing. Figure 6 demonstrates one suitable method of reporting SFR values.



Measurement conditions:

Lens focal length = 55 mm

Lens aperture = *f*/4

Camera compression = off

White balance setting = daylight

Dark-field and flat-field correction were not used

2 000 lx daylight illumination

NOTE A relative spatial frequency of 1 corresponds to 2 048 LW/PH and 110 cycles/mm on the image-sensor camera settings.

Figure 6 — SFR measurement report

7.5 Aliasing ratio

The aliasing-ratio values, if measured, should be reported in a list that indicates the frequency in LW/PH, from 100 LW/PH up to the camera Nyquist frequency, and the aliasing ratios in both the horizontal and vertical directions for each frequency. An example is shown in Table 2.

Table 2 — Aliasing ratios

Spatial frequency LW/PH	**Horizontal aliasing ratio**	**Vertical aliasing ratio**
100	1/100	1/100
200	2/100	3/100
300	4/100	6/100
400	8/100	10/100
500	14/100	18/100

Annex A
(informative)

SFR measurement algorithm C-code

```
/***************************************************************************/
// sfr_computation.c
// Copyright (c) PIMA 1998     All Rights Reserved.
//
// Requests for permission to duplicate, distribute, publish, or otherwise copy this code should be directed to PIMA
/***************************************************************************/
#include "plugin.h"
#include "sfr_computation.h"
#include "io_routines.h"
#include "UserInterface.h"
#include "radiometric_transform.h"
/***************************************************************************/
// macros
static double dmaxarg1,dmaxarg2;
#define DMAX(a,b) (dmaxarg1=(a),dmaxarg2=(b),(dmaxarg1) > (dmaxarg2) ?\
        (dmaxarg1): (dmaxarg2))
static double dminarg1,dminarg2;
#define DMIN(a,b) (dminarg1=(a),dminarg2=(b),(dminarg1) < (dminarg2) ?\
        (dminarg1): (dminarg2))
static double sqrarg;
#define SQR(a) ((sqrarg=(a)) == 0.0 ? 0.0: sqrarg*sqrarg)
/***************************************************************************/
short sfrProc ( GHdl globals, dblHandle FreqHdl, dblHandle dispString)
{
    unsigned short      i, j, nSamplesPerPixel, tmpalpha, err = 0;
    long                pcnt, pcnt2, col, ww_in_pixels;
    double              dt, dt1, sfrc, tmp, tmp2, slope;
    OSErr               erro=0;
    unsigned long       size_x, size_y;
    uHandle             area=nil;
    dblHandle           darea=nil,temp=nil,shifts=nil,edgex=nil,Signal=nil;
    dblHandle           AveEdge=nil,AveTmp=nil,farea=nil;
    longHandle           counts=nil;

    size_x  = gStuff->inRect.right - gStuff->inRect.left;
    size_y = gStuff->inRect.bottom - gStuff->inRect.top;
```

```
nSamplesPerPixel = gStuff->imageMode;
ww_in_pixels = size_x;
```

// **Verify input selection dimensions are EVEN**

```
if (fmod((double)size_x,2.0) != 0.00)
        { ShowAlertText(SHOWALERTSTOP,SFRERR,5);
        gResult = 1;
        return 1;
        }
if (fmod((double)size_y,2.0) != 0.00)
        { ShowAlertText(SHOWALERTSTOP,SFRERR,6);
        gResult = 1;
        return 1;
}
```

// **Allocate memory**

This code not shown

// **Load the image data into the "area" array.**

```
if (TestAbort ()) {
        gResult = 1;
        return 1;
}
      err = load_area_array(globals, area); if (err != 0) {
        gResult = 1;
        return 1;
}
if (TestAbort ()) {
    gResult = 1;
        return 1;
}
      err = radiometric_conversion(globals, area, darea); if (err != 0) {
        gResult = 1;
        return 1;
    }
```

// **Extract first color channel (luminance)**

```
    for (j=0;j < size_y;j++) {
                for (i=0;i < size_x;i++) {
        (*farea)[((j*(long)size_x)+i)] =
(                  *darea)[(nSamplesPerPixel*((j*(long)size_x)+i)) + 0L];
        }
    }
    err = check_image_data( globals, farea, darea);
    if (err != 0) {
```

```
            gResult = 1;
            return 1;
        }
        if (TestAbort ()) {
            gResult = 1;
            return 1;
        }
          err = locate_centroids( globals, farea, temp, shifts); if (err != 0) {
            gResult = 1;
            return 1;
        }
//    Calculate the best fit line to the centroids
        err = fit(size_y, temp, shifts, &slope);
        if (err != 0) {
            gResult = 1;
            return 1;
        }
      if ( fabs(slope) < (1.0/(double)size_y) || slope > (double)(1.0/4.0)) { ShowAlertText(SHOWALERTSTOP,SFRERR,4);
            gResult = 1;
            return 1;
        }
/******************************************************************************/
/* Figure out how many lines to use for size_y: new window will start at top and go down that number of lines < size_y such that an integer number of x-transitions are made by the edge;  for example, if we had a slope of 10 (the edge goes down 10 lines before jumping over one pixel horizontally), and size_y = 35, the new size_y is going to be 30 (an integer multiple of 10, less than 35). */
size_y = (unsigned short)((long)(size_y*slope)*(1.0/slope));
//    reference the temp and shifts values to the new y centre
col = (long) size_y/2;
for (i=0; i < size_y; i++) {
    (*temp)[i] = (double)i-(double)col;
}
// Instead of using the values in shifts, synthesize new ones based on the best fit line.
for (i = 0; i < size_y; i++) {
    (*shifts)[i] = slope * ((*temp)[i]);
}
// compute the global MAX and MIN
dt = 99999999.9;
dt1 = -999999999.9;
pcnt = 0;
for (j = 0; j < size_y; j++) {
for (i = 0; i < size_x; i++) { dt = DMIN(dt, (*farea)[pcnt]);
```

```
dt1 = DMAX(dt1, (*farea)[pcnt]); pcnt++;
        }
}
if (TestAbort ()) {
        gResult = 1;
        return 1;
}
```

// Calculate a long paired list of x values and signal values

```
pcnt = 0;
for (j = 0; j < size_y; j++) {
        for (i = 0; i < size_x; i++) {
        (*edgex)[pcnt] = (double)i -(*shifts)[j];
        (*Signal)[pcnt] = (((*farea)[((j*(long)size_x)+i)]) - dt)/(dt1-dt);
        pcnt++;
        }
}
tmpalpha = (unsigned short)ALPHA;
err = bin_to_regular_xgrid( globals, &tmpalpha, edgex, Signal, AveEdge,  counts, size_y); if (err != 0) {
        gResult = 1;
        return 1;
}
calculate_derivative( globals, (unsigned short)ALPHA, AveTmp,  AveEdge);
locate_max_PSF( globals, (unsigned short)ALPHA, AveEdge, size_x,&pcnt2);
apply_hamming_window( globals, (unsigned short)ALPHA, (unsigned short)ww_in_pixels, AveEdge, &pcnt2);
tmp = 1.0;
tmp2 =     1.0/((double)size_x*ALPHA) ;
if (TestAbort ()) {
        gResult = 1;
        return 1;
}
```

// ftwos (nx, dx, lsf(x), nf, df, sfr(f)

```
(void) ftwos(globals,(long)ALPHA*size_x, &tmp, AveEdge, (long)(size_x*ALPHA/2.0), &tmp2, AveTmp); if (TestAbort ()) {
        gResult = 1;
        return 1;
}
for (i=0; i<(long)((double)size_x*ALPHA/2.0); i++)
{
        sfrc = (*AveTmp)[i];
                                        (*FreqHdl)[i]= ((double)i/(double)size_x);
        (*dispString)[i] = (double) (sfrc/(*AveTmp)[0]); }
```

// I/O code calls here...


```
        if (TestAbort ()) {
            gResult = 1;
            return 1;
        }
    return(0);
}
#pragma                 mark            ----            pre-check           the             data            ----
/****************************************************************************/
unsigned short check_image_data(GHdl globals, dblHandle farea, dblHandle temp) {
        long        j = 0;
        double      dt, dt1, dt2, dt3;
        unsigned short              size_x, size_y;
        OSErr       erro=0;
//      --> check to make sure there is a clear black to white or white to black
//          transition on both the bottom and the top of the image
//      --> If there is not at least 20% difference between the left and right sides of the
//          image box, return with an error of 5: calling program will display an error
//          message.
//      --> If the transition left -> right is white -> black, flip the data left to right
//          (the rest of the routine assumes a black to white transition from left to right)
size_x = gStuff->inRect.right - gStuff->inRect.left;
size_y = gStuff->inRect.bottom - gStuff->inRect.top;
// Get averages of 4 pixels in each corner of the input image */
// upper right corner
dt = (*farea)[size_x-1]+(*farea)[size_x-2];
dt = dt + (*farea)[2*size_x-1]+(*farea)[2*(long)size_x-2];
dt = dt/4.0;
// lower right corner
dt1 = (*farea)[(size_x*(long)size_y-1)]+(*farea)[(size_x*(long)size_y-2)];
dt1 = dt1 + (*farea)[((size_y-1)*(long)size_x-1)] + (*farea)[((size_y-1)*(long)size_x-2)];
        dt1 = dt1/4.0;
// lower left corner
dt2 = (*farea)[((size_y-1)*(long)size_x)]+(*farea)[((size_y-1)*(long)size_x+1)];
dt2 = dt2 + (*farea)[((size_y-2)*(long)size_x)]+ (*farea)[((size_y-2)*(long)size_x+1)];
dt2 = dt2/4.0;
// upper left corner
dt3 = ((*farea)[0]+(*farea)[1]+(*farea)[size_x]+(*farea)[size_x+1])/4.0;
//      If there is not at least 20% difference between the left and right sides of the
//      image box, return with an error of 5: calling program will display an error
//      message.
if (fabs(dt-dt3) < 0.20) j=1;
if (fabs(dt1-dt2) < 0.20) j=1;
```

```
if (j) {
 if (ShowAlertText(SHOWALERTWARN,SFRERR,2) == 2) {
       gResult = 1;
              return 5;
              }
       }
```

// If the transition left -> right is white -> black, flip the data left to right // (the rest of the routine assumes a black to white transition from left to right) flip_image_data_horiz(globals, farea, temp, &dt, &dt1, &dt2, &dt3);

```
return 0;
}
/****************************************************************************/
void flip_image_data_horiz(GHdl globals, dblHandle farea, dblHandle darea,
                      double *dt, double *dt1,double *dt2,double *dt3)
{
              unsigned long            i, j;
              unsigned short  size_x, size_y;
              size_x = gStuff->inRect.right - gStuff->inRect.left;
              size_y = gStuff->inRect.bottom - gStuff->inRect.top;
```

// If the transition left -> right is white -> black,

// flip the data left to right

```
              if ( ((*dt) < (*dt3)) || ((*dt1) < (*dt2) ) ) {
                      for (i=0; i<size_y; i++) {
                                     for (j=0; j<size_x; j++) {
                                                    (*darea)[j] = (*farea)[(i*(long)size_x)+(size_x-1L)-j]; }
                      for (j=0; j<size_x; j++) {
                                     (*farea)[(i*(long)size_x)+j] = (*darea)[j];
                                     }
                      }
              }
}
#pragma mark ---- vector processing routines ----
/****************************************************************************/
unsigned short locate_centroids(GHdl globals, dblHandle farea,dblHandle temp, dblHandle shifts) {
              unsigned long            i, j;
              double                   dt, dt1, dt2;
              unsigned short           size_x, size_y;
 size_x = gStuff->inRect.right - gStuff->inRect.left; size_y = gStuff->inRect.bottom - gStuff->inRect.top;
```

 // Compute the first difference on each line. Interpolate to find the centroid of the first derivatives.

```
              for (j = 0; j < size_y; j++) {
              dt = 0.0;
              dt1 = 0.0;
              for (i = 0; i < size_x-1; i++) {
               dt2 = (*farea)[((j*(long)size_x)+i)+1] - (*farea)[((j*(long)size_x)+i)]; dt += dt2 * (double)i;
```



```
            dt1 += dt2;
            }
            (*shifts)[j]=dt/dt1;
            if (TestAbort ()) {
                  gResult = 1;
                  return 1;
            }
            }
//    check again to be sure we aren't too close to an edge on the corners. If the
//    black to white transition is closer than 2 pixels from either side of the
//    data box, return an error of 5; the calling program will display an error
//    message (the same one as if there were not a difference between the left and
//    right sides of the box )
if ((*shifts)[size_y-1] < 2) {
      ShowAlertText(SHOWALERTSTOP,SFRERR,3);
      return 5;
}
      if (fabs((*shifts)[0]-size_x) < 2){
          ShowAlertText(SHOWALERTSTOP,SFRERR,3);
            return 5;
            }
                              // Reference shifts to the vertical centre of the data box
j = size_y/2;
dt = (*shifts)[j];
for (i = 0; i < size_y; i++)
            {
            (*temp)[i] = (double)i - (double)j;
            (*shifts)[i] -= dt;
            }
            return 0;
}
/***********************************************************************************
/ unsigned short fit(unsigned long ndata,dblHandle x,dblHandle y, double *b)
{
            unsigned long         i;
            double                t,sxoss,sx=0.0,sy=0.0,st2=0.0,ss,sigdat,chi2,a,siga,sigb;
            *b=0.0;
            for ( i=0; i < ndata; i++ ) {
            sx += (*x)[i];
            sy += (*y)[i];
            }
            ss=(double)ndata;
```

```
        sxoss=sx/ss;
        for ( i=0; i < ndata; i++ ) {
         t = (*x)[i] - sxoss;
         st2 += t*t;
         *b += t * (*y)[i];
        }
        *b /= st2;                          //    slope
        a=(sy-sx*(*b))/ss;                  //    offset

        siga=sqrt((1.0+sx*sx/(ss*st2))/ss);
    sigb=sqrt(1.0/st2);
    chi2=0.0;
    for (i=0; i < ndata; i++) chi2 += SQR( (*y)[i] - a - (*b) * (*x)[i]); sigdat=sqrt(chi2/(ndata-2));
    siga *= sigdat;
    sigb *= sigdat;
    return 0;
}
/*****************************************************************************/
```

/* Notes: this part gets averages and puts them in a number of bins, equal to size_x times alpha. Next a long check is done in case one bin gets no values put into it: if this is the case, it will keep checking previous bins until it finds one with non-zero counts and will use that value as its current bin average. If the first bin has zero counts the program checks bins in the forward rather than reverse direction. If, in any case, the end of the array of bins is reached before finding a non-zero count, the program starts checking in the opposite direction. A bin with zero counts is not allowed, since each bin will be divided by counts at the end. */

```
unsigned short bin_to_regular_xgrid(GHdl globals, unsigned short *alpha,dblHandle edgex,dblHandle Signal, dblHandle AveEdge, longHandle counts,unsigned long size_y)
{
    long        i, j, k,bin_number;
    unsigned short          size_x;
    size_x = gStuff->inRect.right - gStuff->inRect.left;
    for (i=0; i<(size_x*(*alpha)); i++)  {
        (*AveEdge)[i] = 0;
        (*counts)[i] = 0;
    }
    for (i=0; i<(size_x*(long)size_y); i++) {
      bin_number = (long) ((*alpha)*((*edgex)[i]));
      if (bin_number >= 0)  {
                          if (bin_number <= (size_x*(*alpha) - 1) )  {
           if (bin_number == 0) {
           }
           (*AveEdge)[bin_number] = (*AveEdge)[bin_number] +  (*Signal)[i];
        (*counts)[bin_number] = ((*counts)[bin_number])+1;
        }
       }
```

```
                    if (TestAbort ()) { gResult = 1; return 1;
        }
    }
    for (i=0; i<(size_x*(*alpha)); i++) {
     if (TestAbort ()) {
        gResult = 1;
        return 1;
        }
        j = 0;
        k = 1;
        if ((*counts)[i] == 0) {
         if (i == 0) {
            while (!j) {
                if ((*counts)[i+k] != 0) {
                    (*AveEdge)[i] = (*AveEdge)[i+k]/((double) (*counts)[i+k]);
                      j = 1;
                    }
                    else k++;
                }
                  } else {
    while (!j && ((i-k) >= 0) )
    {       if ( (*counts)[i-k] != 0){
                  (*AveEdge)[i] = (*AveEdge)[i-k]/((double) (*counts)[i-k]);
                  j = 1;
                  } else k++;
                  }
                  if ( (i-k) < 0 ) {
                        k = 1;
                        while (!j){
                              if ((*counts)[i+k] != 0){
                              (*AveEdge)[i] = (*AveEdge)[i+k]/((double) (*counts)[i+k]);
                              j = 1;
                        } else k++;
                  }
            }
        }
    } else (*AveEdge)[i] = ((*AveEdge)[i])/ ((double) (*counts)[i]);
    }
    return 0;
}
/*********************************************************************************/
void calculate_derivative(GHdl globals, unsigned short alpha,dblHandle AveTmp, dblHandle AveEdge)
```

```
{
        unsigned long       i;
        unsigned short  size_x;
        size_x = gStuff->inRect.right - gStuff->inRect.left;
        for (i=0; i< size_x*alpha; i++) (*AveTmp)[i] = (*AveEdge)[i];
            for (i=1; i< (size_x*alpha-1); i++)(*AveEdge)[i] = ((*AveTmp)[i+1]-(*AveTmp)[i-1])/2.00; (*AveEdge)[0] =
            (*AveEdge)[1];
        (*AveEdge)[size_x*alpha-1] = (*AveEdge)[size_x*alpha-2];
}
/*****************************************************************************/
void locate_max_PSF(GHdl globals,unsigned short alpha, dblHandle AveEdge, long size_x,long *pcnt2) {
        unsigned long       i;
    double                          dt=0.0,dt_new=0.0;
        long                        scnt2 = 0L, left = -1L,right = -1L;
// find maximim value in Point Spread Function array */
    for (i=0; i<size_x*alpha; i++) {
        dt_new = fabs((*AveEdge)[i]);
        if ( dt_new > dt) {
            (*pcnt2) = (long) i;
            dt = dt_new;
            }
        }
// find leftmost and rightmost occurrence of maximum */
 for (i=0; i<size_x*alpha; i++) {
    dt_new = fabs((*AveEdge)[i]);
    if ( dt_new == dt ) {
        if ( left < 0 ) left = i;
        right = i;
            }
    }
//  find centre of maxima */
    (*pcnt2) = (right+left)/2;
}
/*****************************************************************************/
void apply_hamming_window(GHdl globals, unsigned short alpha,unsigned short newxwidth,
        dblHandle AveEdge, long *pcnt2)
{
    long            i,j,begin, end, edge_offset;
    double                  sfrc;
    unsigned short  size_x;
    size_x = gStuff->inRect.right - gStuff->inRect.left;
// Shift the AvgEdge[i] vector to centre the lsf in the transform window
```