PPT VISION, Inc.

IMPACT

Subpixel Resolution

PPT VISION

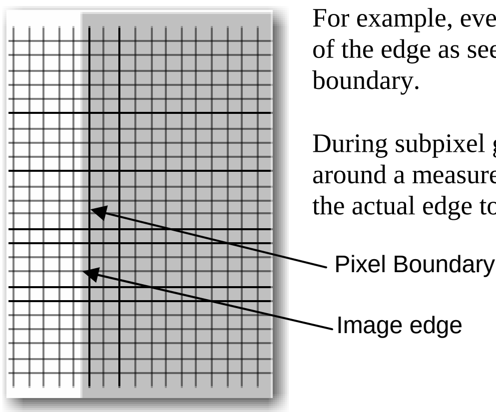
PPT VISION

Subpixel Resolution

This document describes how IMPACT Software tools use subpixel resolution interpolation to find a subpixel point. A subpixel point is used so that tools can accurately calculate an image’s edge location, the diameter of a blob, or a blob centroid's position to accuracies of less than one pixel.

Introduction

A pixel is the smallest physical unit in the video image. Although the video image contains several thousand pixels, the edges of an object, however straight, rarely fall exactly on the boundaries of the pixel in the camera.



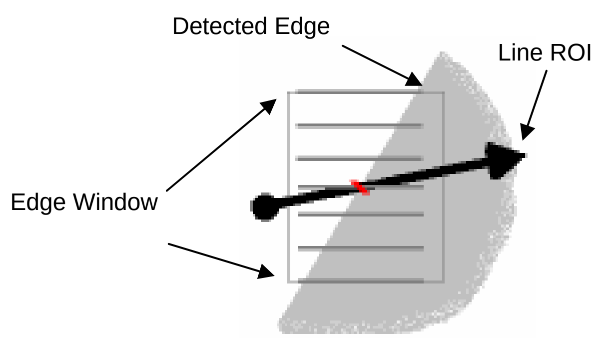
For example, even if an object appears to have a straight edge, in the extreme close-up of the edge as seen here, you can see that this edge does not fall directly on a pixel boundary.

During subpixel gauging, the system examines the grey levels of adjacent pixels around a measurement endpoint. It uses this information to interpolate the location of the actual edge to a fraction of a pixel.

Edge Window

The system collects multiple subpixel samples along the detected edge on both sides of the initial detection point. This area along the detected edge is called the edge window. The size of the edge window is determined by the Subpixel settings in the tool. These settings allow precise calculation of the subpixel edge.

The system generates each subpixel sample by examining an individual line of pixels crossing the detected edge. Although you can draw line ROIs at any angle, the vision system only samples rows or columns in the edge window. The vision system samples parallel rows of pixels if the line ROI is more horizontal than vertical, and columns if the ROI is more vertical than horizontal.

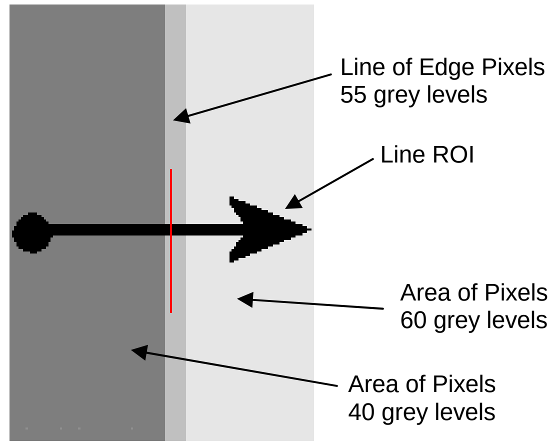


The Edge Window is the size of the area around the detected edge that is sampled. The Detected Edge is the found edge within the area.

Best-Fitting the Edge

To find the subpixel location of an edge within a single pixel, we can compare the grey level of the edge pixel to the grey levels of the pixels on either side of it. In the simplified example shown below, the edge pixel's grey level is a mix of the left hand pixel's grey level and the right hand pixel's grey level. The proportions of the mix depends on the location of the edge within the edge pixel.

The subpixel edge location is calculated by using the grey levels of the pixels around it.



This calculation is know as interpolation and is written as

subpixel location = (edge grey level – right grey level) / (left grey level – right grey level)

In the example above, the edge pixel is 55 grey levels, the left hand pixel is 40 grey levels, and the right hand pixel is 60 grey levels. Using interpolation we can calculate

subpixel location = (55 – 60) / (40 – 60)

subpixel location = (-5) / (-20)

subpixel location = 0.25

So the edge we are trying to find is located 0.25 of the distance through the pixel, looking from left to right. This is true because 0.25 of the pixel is illuminated by the left of the edge and 0.75 is illuminated by the right of the edge.

Avoid Image Saturation

Finding the subpixel edge is not possible if the image is saturated, i.e. the image light intensity is greater than 100% of the camera range. If the brighter side of the edge is saturated, then the interpolation calculation is no longer valid. If a pixel is 100% of range, then the actual light intensity falling on it may be greater than 100% and the actual light intensity is not registered. When this very bright light falls on the edge pixel, then the actual bright light intensity is mixed with the dark side intensity. Since actual bright light intensity is unknown (we know only that it is something greater than 100%), we can't accurately determine the portion of the pixel that it illuminates. As a result, the subpixel location will be biased and indicate that the bright side of the edge occupies a larger portion of the edge pixel than it actually does.

In this example, the left hand pixel is 50 grey levels, the right hand pixel is 120 grey levels, and the edge falls midway through the edge pixel (the subpixel location = 0.5). The right hand pixel will appear to be 100 (saturated) but the edge pixel grey level will be calculated as

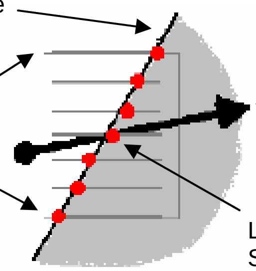
0.5 * (50 + 120) = 85

because it cannot properly integrate the bright side of the edge without saturating. The interpolation formula gives us

(85 - 100) / (50 - 100) = 0.3

rather than the correct 0.5.

Once the system has performed subpixel analysis to determine the subpixel location of the detected edge along each line, an edge is then best-fit to the subpixel location in each line of pixels. IMPACT tools can be programmed to drop outlying samples, i.e. those which lie farther away from the fitted line than the specified distance. Once an outlying sample is dropped, the tool refits the edge to the remaining sample points. Finally, the subpixel location of the edge is determined using the best-fit edge generated above.



Determining the accuracy of blob calculations

Blob calculations can be used to report information with subpixel accuracy. If your inspection relies on subpixel results, you may be concerned with the tolerance limits for these calculations. This section lists the accuracies for IMPACT calculations, and explains how these results are calculated.

Centroid Calculations

The centroid of a blob can be used to report an object's position with subpixel accuracy. But what is that accuracy? For a circle, the short answer is ± 0.6/sqrt (diameter) pixels. The centroid calculation is more accurate with bigger circles.

Blob Diameter	Location Accuracy
10 pixels	± 0.183 pixels
100 pixels	± 0.06 pixels

A longer answer is that the centroid is an unbiased estimator with a Gaussian distribution and a standard deviation of 0.2/sqrt (diameter). If we need 3 sigma error bound, then our accuracy is ± 0.6/sqrt (diameter) pixels.

Derivation

The *binary X centroid* is the average value of the X coordinate of the perimeter pixels. The interior pixels contribute to the centroid calculation, but have no effect on its accuracy. Each perimeter pixel position is known to pixel accuracy. The subpixel accuracy comes from the averaging operation.

Each *blob perimeter pixel position* is an estimate of the true circle edge position, with uniform error distribution between -0.5 and +0.5 pixels. In other words, if a perimeter pixel is at X, then the true circle edge can be anywhere from X-0.5 to X+0.5 with equal probability, but not outside that range. This assumes a fairly sharp edge and a threshold near the center of the edge gray levels. If the threshold is off center, the end result will be the same because there are an equal number of rising and falling edge pixels on the blob.

The centroid is found by averaging together the N perimeter pixel positions. The standard deviation of the centroid is given by:

standard deviation centroid = standard deviation pixel / sqrt (N)

The standard deviation of a uniform distribution of width 1.0 is 0.29. The number of significant samples averaged for the X or Y centroid is 2*diameter since there are two edges on each row or column going through the circle. So:

standard deviation centroid = 0.29 / sqrt (2*diameter) = 0.2 / sqrt (diameter)

The central limit theorem tells us that averaging a large number of measurements will result in a Gaussian distribution. We now use the Gaussian error function to tell us what accuracy we have. We know that our value is within three standard deviations 99.7% of the time. We know that our value is within two standard deviations 95.4% of the time, and so on.

Circular Diameter Calculations

The diameter of a blob can be used to report a circular object's size with subpixel accuracy. But what is that accuracy?

For a 3 sigma error bound, the accuracy of circular blob diameter calculations is ± 0.78/sqrt (diameter) pixels. The diameter calculation is more accurate with bigger circles.

Blob Diameter	Location Accuracy
10 pixels	± 0.247 pixels
100 pixels	± 0.078 pixels

Derivation

If σ_p^2 represents the variance of a pixel location, then the variance of the length calculation of a horizontal line segment is,

$$\sigma_l^2 = \sigma_p^2 + \sigma_p^2 = 2\sigma_p^2 \qquad (1)$$

The area of a circular blob is the sum of the length of all the d line segments that make up the circle, where d is the diameter of the circle. The variance of the area is,

$$\sigma_A^2 = d\sigma_l^2 = 2d\sigma_p^2 \qquad (2)$$

The relationship between the area and the diameter of a circle is,

$$d = \sqrt{\frac{4A}{\pi}} = \sqrt{\frac{4}{\pi}}A^{0.5} \qquad (3)$$

Through error propagation, the variance of the circle diameter therefore is,

$$\sigma_d^2 = \frac{0.5^2 d^2}{A^2}\sigma_A^2 \qquad (4)$$
$$= \frac{0.8106}{d}\sigma_p^2$$

and the standard deviation is,

$$\sigma_d = \frac{0.9}{\sqrt{d}}\sigma_p \qquad (5)$$

Assuming a fairly sharp edge and a threshold near the center of the edge gray levels, the pixel standard deviation can be computed using a uniform distribution of width 1.0, which is 0.29. The standard deviation of the circle diameter thus becomes,

$$\sigma_d = \frac{0.9 \times 0.29}{\sqrt{d}} = \frac{0.26}{\sqrt{d}} \qquad (6)$$

Reference

Haralick, R.M. and L.G. Shapiro. 1992. Computer and Robot Vision. Vol 2.
Chapter 20 - Accuracy. AW.

