Sightech Vision Systems, Inc.

PC-Eyebot

Learning Modes Explained – Feature Types and Sizes

By controlling the way our self-learning vision actually “sees”, we can specialize the inspection process. Given the needs of an application, we will discuss how to choose the best Feature Types and Sizes.

Published 2005

Sightech Vision Systems, Inc.
6580 Via del Oro
San Jose, CA 95126
Tel: 408.282.3770 Fax: 408.413-2600
Email: sales@Sightech.com
Web: www.Sightech.com

Table of Contents:

Selecting the Best Feature Type.

Feature Type is the most important parameter:

The Feature Type is the most important selection when determining how to optimize how the vision system operates. Selecting the Feature Type establishes the principal framework used by the system to View the real world. These types determine whether we want to learn color, shape, gray-levels, widths, behavior, etc. Below are the Feature Type choices, descriptions, and uses.

Kinds of Input Image Data that can be used by a Feature:

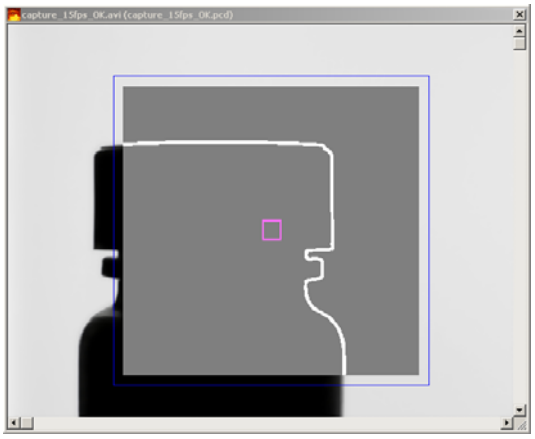
a) Gray-level – this is an 8-bit value per pixel that represents the intensity.
b) Binary image – this is a black-or-white (binary) representation of the image. Edge convolutions and comparing the image, pixel-by-pixel, to an intensity threshold are common way to create a binary image.
c) Color image data. PC-Eyebot uses 24bit RGB format.
d) Behavior – image movement from frame-to-frame (coming soon).
e) 3-Dimensional – multiple camera images from different perspectives (future development)
f) The X, Y, XY, or R (radius) values.

Binary transform of the image:

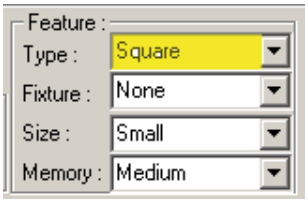
It is important that neural processes reduce the incoming visual data as much as possible and emphasize the part of the data that pertains to the goals of the inspection tasks. An important facility for image data reduction is the binary transform of the image where one stream of the image is reduced from 24 bits to one bit (either 1 or 0) for each pixel - a 24:1 reduction. The primary methods used for this reduction are gray-level conversion, "thresholding", and/or "edge-conversions". There are many choices for this image pre-processing – we call these "Transforms". Many of the Features define their Views to include this binary representation of the image data.

The binary image data has the following benefits:

a) High data content – more information per bit
b) Good for shape representation
c) Binary transform may be picked to optimize for application
d) Correct transform adjustment can mask undesired image data from vision process – it can therefore emphasize the data that is important for the desired decision making.
e) The edge-detection transforms are relatively insensitive to small light level changes. This reduces the negative effect of stray lighting changes.



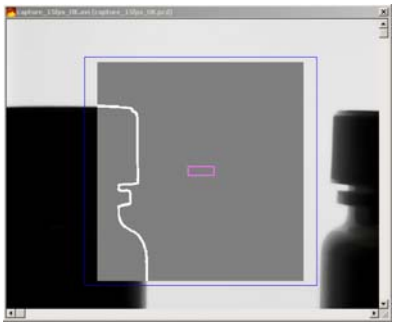
How to choose the Feature Type



The type may be selected from the drop-down list: Feature→Type.

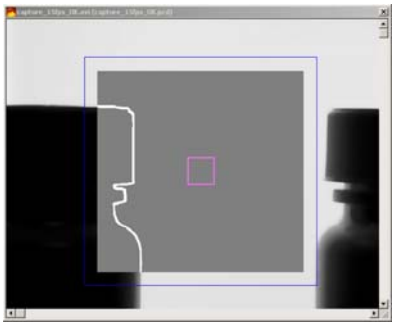
Feature Type - Rectangle

This type uses the binary image for its image data input. Its View consists of a N-by-M View, where N is larger than M so that the View consists of a horizontal rectangle. This Feature emphasizes shape information.



Feature Type - Square

This type uses the binary image for its image data input. Its View consists of a N-by-N View that emphasizes shape information – angle as well. This Feature does not allow color information.



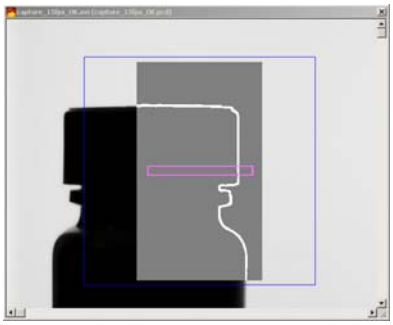
Feature Type - Circular

This type uses the binary image for its image data input. Its View consists of a View that is circular in its format. This Feature emphasizes shape information, but attempts not to emphasize the angle of the shape. This feature imparts rotational freedom to the system characteristic.

Feature Type - Burst

This type uses the binary image, or the multi-resolution binary image, for its image data input. Its View consists of a N-by-N View that is circular in its format that does emphasize angle. This Feature defines finer detail in the center and more general (blurred) detail further from the Feature center. This is why it is called Burst.

Feature Type - Wide

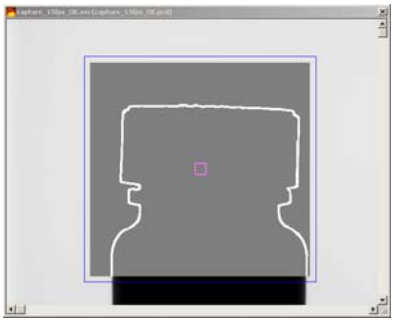


This type uses the binary image for its image data input. Its View consists of a N-by-M View, where N is much larger than M so that the View consists of a wide thin horizontal rectangle. This Feature emphasizes width information.

Feature Type – Color-10

This type uses both the binary image and color for its image data input. Its View consists of a square N-by-N View combined by the color of the Feature. Though it combines both color and shape information, it emphasizes shape over color.

Feature Type – Color-50

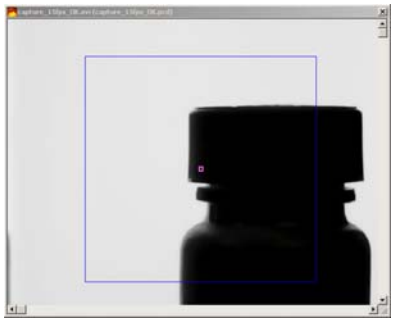


This type uses both the binary image and color for its image data input. Its View consists of a square N-by-N View combined by the color of the Feature. It combines both color and shape information in equal amounts.

Feature Type – Color-90

This type uses both the binary image and color for its image data input. Its View consists of a square N-by-N View combined by the color of the Feature. Though it combines both color and shape information, it emphasizes color over shape.

Feature Type – Spectrum



This type uses only color for its image data input. Its View has no shape content at all, so it has total rotational freedom. It is, however, more sophisticated than a the color of a single pixel. Instead, it creates a color from a small cluster of pixels. To minimize memory usage and algorithmic work, It filters out color-edge transitions. This Feature emphasizes at least moderately uniform color areas. This cluster is N-by-N sized.

Feature Type – Coloration

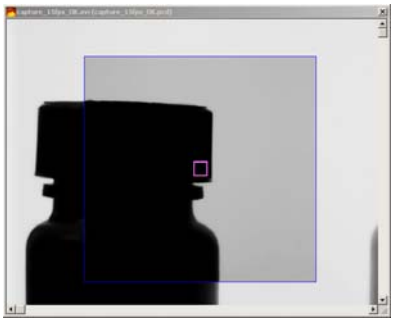
This advanced type uses only color for its image data input – but a more complex View of color. Its View has no shape content at all, so it has total rotational freedom. It defines a cluster of pixels, but extracts more than one color. It is specially designed to extract the "Shape of Color" from the image. It particularly emphasizes color transitions, including color edges, etc. It is

quite different from Spectrum in the regard. This Feature is good for otherwise very difficult applications that need advanced discrimination. This is a very powerful mode.

Feature Type – Graylevel

Instead of color information, this type uses both the binary image data and graylevel information for its image data input. Therefore this Feature is sensitive to shapes as well as graylevel shading. It defines a small cluster of pixels for establishing the graylevel data view.

Feature Type – Texture



Instead of color information, this type uses a cluster of only graylevel information for its image data input. Note that this Feature defines colorless data, but does not involve the binary image transform. This Feature is very good at inspecting fabric patterns, fiber bundles, filters, etc. This is an advanced feature that exhibits an extremely high level of sensitivity to small defects in fabric-oriented inspection. For the correct applications, this feature is very powerful.

Feature Type – Behavior

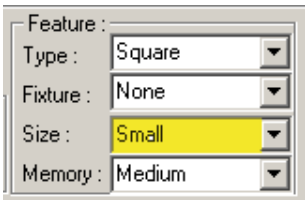
This advanced type combines shape-based information along with temporal information. This allows the vision system to learn "how things move" as well as "how they look". This Feature Type is useful for monitoring a repetitive process. If process degradation takes place, it will be easily detected.

Feature Type – Special_1 … Special_5

These Feature types are reserved for special development. We can combine shape, graylevel, color, and/or behavior information and can customize new data reduction methods to meet the demanding inspection needs of our customers.

Selecting the Feature Size.

Selecting the size:

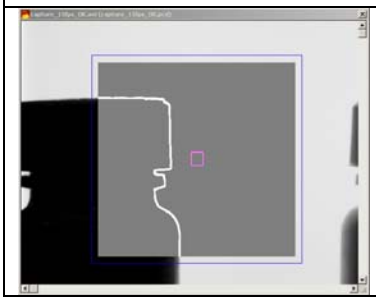
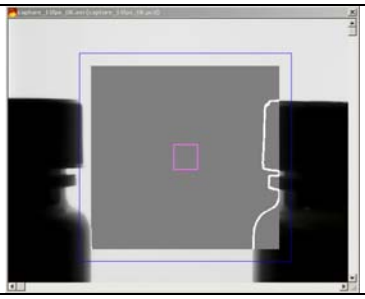
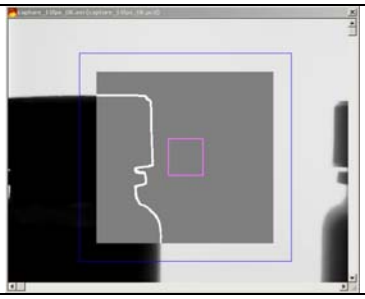


One of 3 choices of Feature size may be picked – Small, Medium, and Large. The above diagram shows Feature→Size→Small selected.

How to check how the size appears:

The actual feature size may be observed in the camera window while in VIEW mode. It will appear as a pink-colored outline in the center – see the images below. This shows the actual size of the feature – not where it is located.

What size should we use?

Small	**Medium**	**Large**
		

Choose a feature size so the feature fits over a typical defect part of the image. In the above example, we want to detect a slightly loose bottle cap. You can see that the Medium feature size in the center image is correct for the desired inspection – the feature will correctly guide learning so the shape of the small gap between the cap and the bottle can be very accurately judged.