Thresholding

Thresholding uses the pixel values in an image to segment the image into two regions: a particle region, which contains the objects under inspection, and a background region.

A range of pixel values is defined, either by the user or automatically, as the threshold. Any pixel value outside the range becomes 0, and any pixel value inside the range becomes 1, or a user-defined value. Thresholding results in a binary image.

When to Use

Use thresholding to create a binary image and focus inspection on specific areas of interest.

Thresholding is often the first step in machine vision applications such as particle analysis, golden template comparison, and binary particle classification.

NI Vision supports the following thresholding methods:

- **Global Grayscale Thresholding**—Use Global Grayscale Thresholding on grayscale images with uniform lighting. Global Grayscale Thresholding can be performed with the following methods:
  - **Manual Threshold**—Enables the user to manually set the threshold range. Recommended on images with good contrast and uniform lighting.
  - **Clustering**—Thresholds the image into more than two classes. This is the most common automatic thresholding method.
  - **Entropy**—Detects small areas of interest in the image. This method is used for applications such as fault detection.
  - **Inter Variance**—Use for images in which classes are not overly disproportionate, and when the object of interest and the background contain a comparable number of pixels. For satisfactory results, the smallest class must be at least 5% of the largest one.
  - **Metric**—Calculates a value for each threshold that is determined by the surfaces representing the initial gray scale. Use this method when the object of interest and the background contain a comparable number of pixels.
  - **Moments**—Use for images that have poor contrast.
- **Global Color Thresholding**—Use Global Color Thresholding on color images with uniform lighting.

Table of Contents (/reference/en-XX/help/370281AG-01)

NI Vision 2019 for LabVIEW Help

Edition Date: March 2019

Part Number: 370281AG-01

»View Product Info (http://www.ni.com/vision)

Vision Developoment Module 2019 Help (http://www.ni.com/pdf/manuals/370281ag.zip)

Vision Development Module 2020 Help (http://www.ni.com/pdf/manuals/370281aj.zip)

- **Local Thresholding**—Use Local Thresholding to isolate regions of interest in images that exhibit non-uniform lighting changes, such as shadows or a strong illumnination gradient. Local Thresholding can be performed with the following methods:
    - **Niblack Algorithm**—Effective for applications such as display inspection and OCR images.
    - **Sauvola Algorithm**—Results in less noise and preserves the shape of the particles.
    - **Modified Sauvola Algorithm**—Less computationally intensive than the Sauvola algorithm.
    - **Background Correction Algorithm**—Reduces noise in large, empty areas.

Global Grayscale Thresholding

Global grayscale thresholding includes manual thresholding and automatic thresholding techniques.

When to Use

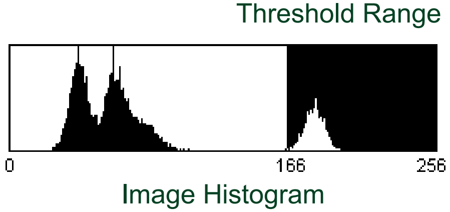
Global thresholding works best when the inspection images exhibit uniform lighting both within each image and across multiple images.

Concepts

Particles are characterized by an *intensity range*. They are composed of pixels with gray-level values belonging to a given threshold interval (overall luminosity or gray shade). All other pixels are considered to be part of the background.

Thresholding sets all pixels that belong to a range of pixel values, called the *threshold interval*, to 1 or a user-defined value, and it sets all other pixels in the image to 0. Pixels inside the threshold interval are considered part of a particle. Pixels outside the threshold interval are considered part of the background.

The following figure shows the histogram of an image. All pixels in the image whose values range from 166 to 255 are considered particle pixels.

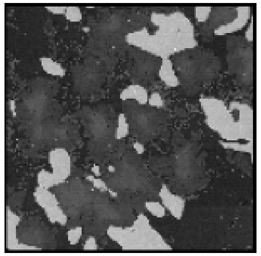


Manual Threshold

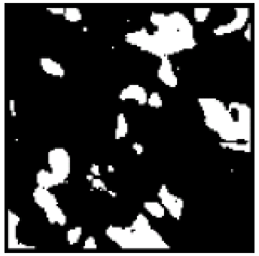
The threshold interval in a manual threshold has two user-defined parameters: lower threshold and upper threshold. All pixels that have gray-level values equal to or greater than the lower threshold and equal to or smaller than the upper threshold are selected as pixels belonging to particles in the image.

Manual Thresholding Example

This example uses the following source image.



Highlighting the pixels that belong to the threshold interval [166, 255] (the brightest areas) produces the following image.



Automatic Threshold

NI Vision has five automatic thresholding techniques.

- Clustering
- Entropy
- Inter Variance
- Metric
- Moments

In contrast to manual thresholding, these techniques do not require that you set the lower and upper threshold values. These techniques are well suited for conditions in which the light intensity varies from image to image.

Clustering is the only multi-class thresholding method available. Clustering operates on multiple classes so you can create tertiary or higher-level images.

The other four methods—entropy, metric, moments, and interclass variance—are reserved for strictly binary thresholding techniques. The choice of which algorithm to apply depends on the type of image to threshold.

Depending on your source image, it is sometimes useful to invert the original grayscale image before applying an automatic threshold function, such as entropy and moments. This is especially true for cases in which the background is brighter than the foreground.

Clustering

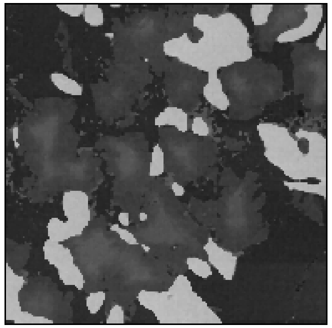
Clustering is the most frequently used automatic thresholding method. Use the clustering method when you need to threshold the image into more than two classes.

Clustering sorts the histogram of the image within a discrete number of classes corresponding to the number of phases perceived in an image. The gray values are determined, and a *barycenter* is determined for each class. This process repeats until it obtains a value that represents the center of mass for each phase or class.

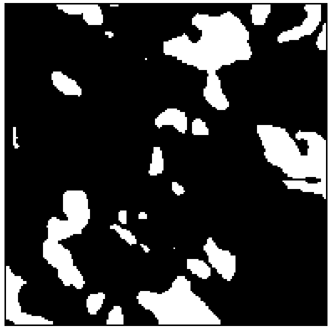
Example of Clustering

This example uses a clustering technique in two and three phases on an image. Notice that the results from this function are generally independent of the lighting conditions as well as the histogram values from the image.

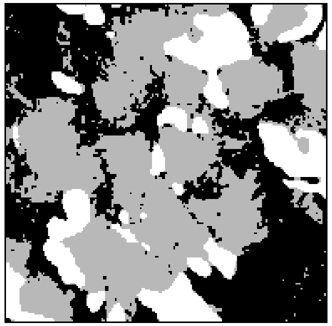
This example uses the following original image.



Clustering in two phases produces the following image.



Clustering in three phases produces the following image.



Entropy

Based on a classical image analysis technique, entropy is best for detecting particles that are present in minuscule proportions on the image. For example, this function would be suitable for fault detection.

Inter Variance

Inter variance is based on discriminant analysis. An optimal threshold is determined by maximizing the interclass variation with respect to the threshold.

Metric

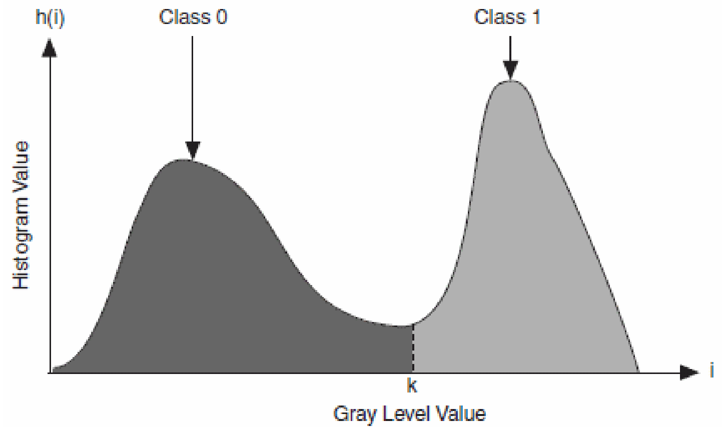
For each threshold, a value determined by the surfaces representing the initial gray scale is calculated. The optimal threshold corresponds to the smallest value.

Moments

This technique is suited for images that have poor contrast. The moments method is based on the hypothesis that the observed image is a blurred version of the theoretically binary original. The blurring that is produced from the acquisition process, caused by electronic noise or slight defocalization, is treated as if the statistical moments of average and variance were the same for both the blurred image and the original image. This function recalculates a theoretical binary image.

In-Depth Discussion

All automatic thresholding methods use the histogram of an image to determine the threshold. The following figure explains the notations used to describe the parameters of the histogram. These notations are used throughout this section to show how each automatic thresholding method calculates the threshold value for an image.



- *i* represents the gray level value
- *k* represents the gray level value chosen as the threshold
- *h*(*i*) represents the number of pixels in the image at each gray level value
- *N* represents the total number of gray levels in the image (256 for an 8-bit image)
- *n* represents the total number of pixels in the image

Use the automatic thresholding techniques to determine the threshold pixel value *k* such that all gray-level values less than or equal to *k* belong to one class 0 and the other gray level values belong to another class 1.

Clustering

The threshold value is the pixel value *k* for which the following condition is true:

$$\frac{\mu_1 + \mu_2}{2} = k$$

where μ_1 is the mean of all pixel values that lie between 0 and *k*, and μ_2 is the mean of all the pixel values that lie between *k* + 1 and 255.

Entropy

In this method, the threshold value is obtained by applying information theory to the histogram data. In information theory, the entropy of the histogram signifies the amount of information associated with the histogram. Let

$$p(i) = \frac{h(i)}{\sum_{i=0}^{N-1} h(i)}$$

represent the probability of occurrence of the gray level *i*. The entropy of a histogram of an image with gray levels in the range [0, N – 1] is given by

$$H = \sum_{i=0}^{N-1} p(i)\log_2 p_b(i)$$

If *k* is the value of the threshold, then the two entropies

$$H_b = -\sum_{i=0}^{k} P_b(i)\log_2 p_b(i)$$

$$H_w = -\sum_{i=k+1}^{N-1} P_w(i)\log_2 p_w(i)$$

represent the measures of the entropy (information) associated with the black and white pixels in the image after thresholding. $P_b(i)$ is the probability of the background, and $P_b(w)$ is the probability of the object.

The optimal threshold value is gray-level value that maximizes the entropy in the thresholded image given by

$$H_b + H_w$$

Simplified, the threshold value is the pixel value *k* at which the following expression is maximized:

$$-\frac{1}{\sum_{i=0}^{k} h(i)} \sum_{i=0}^{k} \log_2 (h(i)+1)h(i) - \frac{1}{\sum_{i=k+1}^{N-1} h(i)} \sum_{i=k+1}^{N-1} \log_2 (h(i)+1)h(i) + \log_2 \left(\sum_{i=0}^{k} h(i) \sum_{i=k+1}^{N-1} h(i) \right)$$

Inter Variance

The threshold value is the pixel value *k* at which the following expression is maximized:

$$\sigma_B^2(k) = \frac{[\mu_T\omega(k) - \mu(k)]^2}{\omega(k)[1-\omega(k)]}$$

where

$$\mu(k) = \sum_{i=0}^{k} ip(i)$$

$$\mu_T = \sum_{i=0}^{N-1} ip(i)$$

$$\mathrm{w}(k) = \sum_{i=0}^{k} \mathrm{p(i)}$$

where $\sigma_B^2(k)$ is the Interclass Variance.

$\mu(k)$ is the mean of the class containing bins 0 to k.

μ_T is the overall mean

$\omega(k)$ is the class probability

Metric

The threshold value is the pixel value *k* at which the following expression is minimized:

$$\sum_{i=0}^{k} h(i)\left|i-\mu_1\right| + \sum_{i=k+1}^{N-1} h(i)\left|i-\mu 2\right|$$

where μ_1 is the mean of all pixel values in the image that lie between 0, and *k*, and
μ_2 is the mean of all the pixel values in the image that lie between *k* + 1 and 255.

Moments

In this method the threshold value is computed in such a way that the moments of the image to be thresholded are preserved in the binary output image.

The *k*th moment *m* of an image is calculated as

$$m_k = \frac{1}{n} \sum_{i=0}^{i=N-1} i^k h(i)$$

where *n* is the total number of pixels in the image.

Global Color Thresholding

Color thresholding converts a color image into a binary image.

When to Use

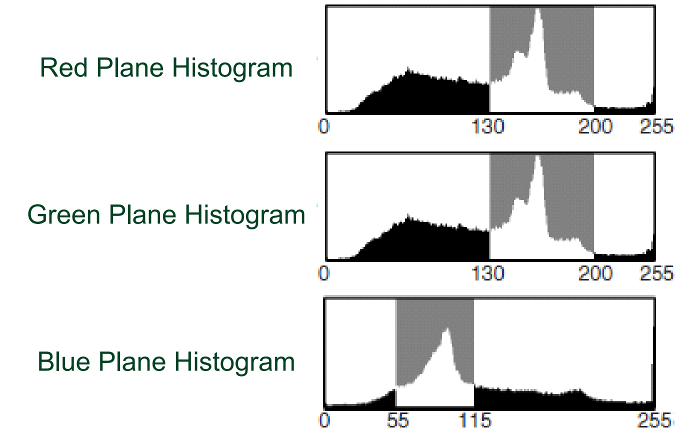
Threshold a color image when you need to isolate features for analysis and processing or to remove unnecessary features.

> **Note** Before performing a color threshold, you may need to enhance your image with lookup tables or the equalize function.

Concepts

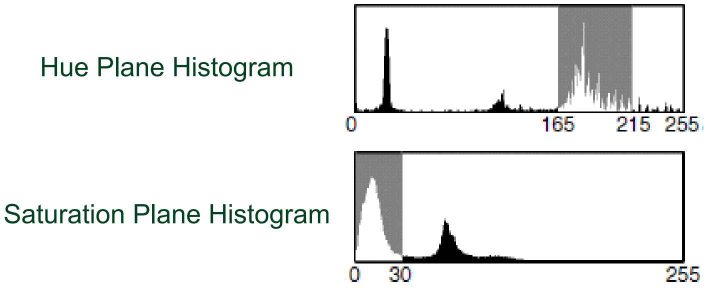
To threshold a color image, specify a threshold interval for each of the three color components. A pixel in the output image is set to 1 if and only if its color components fall within the specified ranges. Otherwise, the pixel value is set to 0.

The following figure shows the histograms of each plane of a color image stored in RGB format. The gray shaded region indicates the threshold range for each of the color planes. For a pixel in the color image to be set to 1 in the binary image, its red value should lie between 130 and 200, its green value should lie between 100 and 150, and its blue value should lie between 55 and 115.

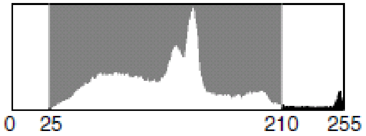


To threshold an RGB image, first determine the red, green, and blue values of the pixels that constitute the objects you want to analyze after thresholding. Then, specify a threshold range for each color plane that encompasses the color values of interest. You must choose correct ranges for all three color planes to isolate a color of interest.

The following figure shows the histograms of each plane of a color image stored in HSL format. The gray shaded region indicates the threshold range for each of the color planes. For a pixel in the color image to be set to 1 in the binary image, its hue value should lie between 165 and 215, its saturation value should lie between 0 and 30, and its luminance value should lie between 25 and 210.



Luminance Plane Histogram



The hue plane contains the main color information in an image. To threshold an HSL image, first determine the hue values of the pixels that you want to analyze after thresholding. In some applications, you may need to select colors with the same hue value but various saturation values. Because the luminance plane contains only information about the intensity levels in the image, you can set the luminance threshold range to include all the luminance values, thus making the thresholding process independent from intensity information.

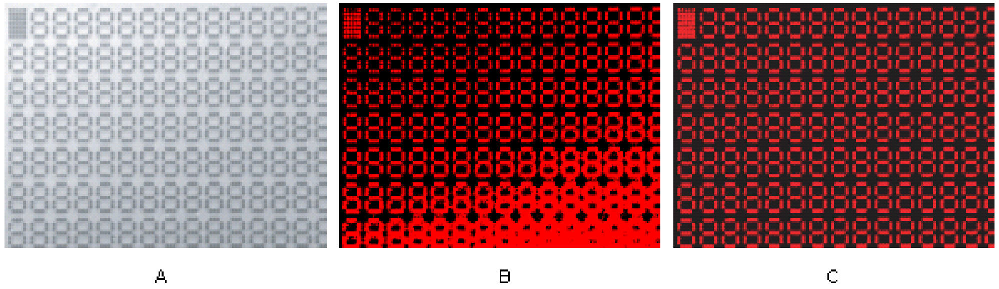
Local Thresholding

Local thresholding, also known as locally adaptive thresholding, is like global grayscale thresholding in that both create a binary image by segmenting a grayscale image into a particle region and a background region. Unlike global grayscale thresholding, which categorizes a pixel as part of a particle or the background based on a single threshold value derived from the intensity statistics of the entire image, local thresholding categorizes a pixel based on the intensity statistics of its neighboring pixels.

When to Use

Use local thresholding to isolate objects of interest from the background in images that exhibit nonuniform lighting changes. Nonuniform lighting changes, such as those resulting from a strong illumination gradient or shadows, often make global thresholding ineffective.

The following figure shows the effect of global thresholding and local thresholding on an image with nonuniform lighting changes. Figure A shows the original inspection image of LCD digits. Figure B shows how a global threshold segments the inspection image. Notice that many of the nondigit pixels in the bottom, right corner are erroneously selected as particles. Figure C shows how a local threshold segments the inspection image. Only pixels belonging to LCD digits are selected as particles.



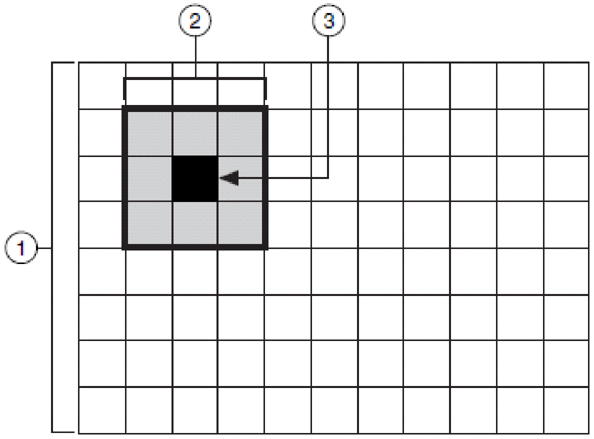
Concepts

The local thresholding algorithm calculates local pixel intensity statistics—such as range, variance, surface fitting parameters, or their logical combinations—for each pixel in an inspection image. The result of this calculation is the *local threshold value* for the pixel under consideration. The algorithm compares the original intensity value of the pixel under consideration to its local threshold value and determines whether the pixel belongs to a particle or the background.

A user-defined window specifies which neighboring pixels are considered in the statistical calculation. The default window size is 32 × 32.

> **Note** Even-numbered window dimensions and odd-numbered window dimensions produce the same center pixel. For example, in the following figure, the pixel under consideration is the same for a 4 × 4 local thresholding window as it is for a 3 × 3 local thresholding window.

The window size should be approximately the size of the smallest object you want to separate from the background. The following figure shows a simplified local thresholding window.



1 Image
2 Local Thresholding Window
3 Pixel under Consideration

> **Note** The pixel intensities of all of the pixels in the window, including the pixel under consideration, are used to calculate the local threshold value.

A typical local thresholding function requires a large amount of computation time. Also, the time a typical local thresholding function takes to complete often varies depending on the window size. This lack of *determinism* prevents local thresholding from being used in real-time applications. The NI Vision local thresholding function uses a fully optimized, efficient algorithm implementation whose computation speed is independent of the window size. This significantly reduces the computation cost and makes using the function in a real-time segmentation applications possible.

The following sections describe the algorithms available in the NI Vision local thresholding function.

> **Note** You must specify whether you are looking for dark objects on a light background or light objects on a dark background regardless of which algorithm you use.

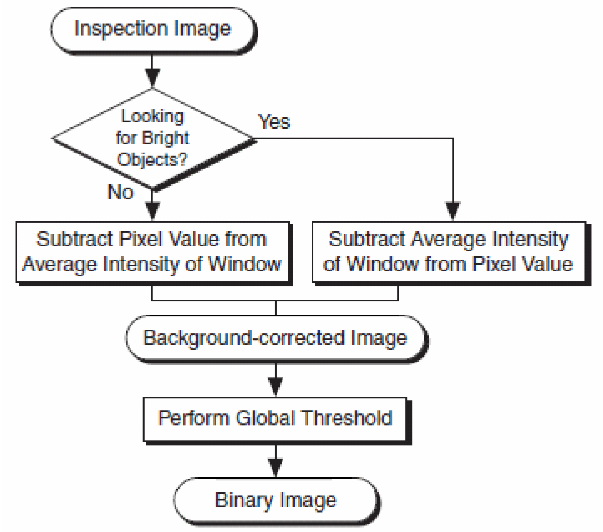
Niblack Algorithm

This algorithm has been experimentally shown to be the best among eleven locally adaptive thresholding algorithms, based on a goal-directed evaluation from OCR and map image segmentation applications. The algorithm is effective for many image thresholding applications, such as display inspection and OCR.

The Niblack algorithm is sensitive to the window size and produces noisy segmentation results in areas of the image with a large, uniform background. To solve this problem, the NI Vision local thresholding function computes a deviation factor that the algorithm uses to correctly categorize pixels.

Background Correction Algorithm

This algorithm combines the local and global thresholding concepts for image segmentation. The following figure illustrates the background correction algorithm.


Inspection Image
Looking for Bright Objects?
Yes
No
Subtract Pixel Value from Average Intensity of Window
Subtract Average Intensity of Window from Pixel Value
Background-corrected Image
Perform Global Threshold
Binary Image

The background-corrected image is thresholded using the interclass variance automatic thresholding method described in the automatic threshold section of this chapter.

In-Depth Discussion

The following sections provide an in-depth discussion of the algorithms used by each thresholding method.

Niblack

In the Niblack algorithm, the local threshold value *T*(*i*, *j*) at pixel (*i*, *j*) is calculated as

$$T(i, j) = m(i, j) + k.\omega\ (i, j)$$

where m(*i*, *j*) is the local sample mean, *k* is the deviation Niblack/Sauvola factor, and ?(*i*, *j*) is the standard deviation.

Each image pixel I(*i*, *j*) is categorized as a particle or background pixel based on the following:

$$\text{if } I(i, j) > T(i, j),\ I(i, j) = \text{particle}$$

$$\text{else } I(i, j) = \text{background}$$

Tip Setting *k* to 0 to increases the computation speed of the Niblack algorithm.

Sauvola

In the Sauvola algorithm, the windowed standard deviation is normalized by dividing the windowed standard deviation by the dynamic range of the standard deviation (*R*). This results in less noise and preserves the shape of the particles. In the Sauvola algorithm, the local threshold value *T*(*i, j*) at pixel (*i, j*) is calculated as

$$T(i,j) = m(i,j) * \left[1 + k * \left(1 - \frac{\omega\,(i,j)}{R}\right)\right]$$

where *m*(*i, j*) and *w*(*i, j*) are the mean and standard deviation calculated in a window, *k* is the Niblack/Sauvola deviation factor, and *R* is the Sauvola deviation range.

Modified Sauvola

In the Modified Sauvola algorithm, the windowed mean deviation is normalized by dividing it by the dynamic range of the standard deviation (*R*). This method uses mean deviation instead of standard deviation, making it less computationally intensive than the Sauvola algorithm. In the Modified Sauvola algorithm, the local threshold value *T*(*i, j*) at pixel (*i, j*) is calculated as

Modified Sauvola algorithm

Modified Sauvola algorithm

where *m*(*i, j*) is the local mean and *d*(*i, j*) are the local mean deviation, *k* is the Niblack/Sauvola deviation factor, and *R* is the Sauvola deviation range.

Background Correction

In the background correction algorithm, the background-corrected image *B*(*i*, *j*) is calculated as

$$B(i, j) = I(i, j) - m(i, j)$$

where *m*(*i*, *j*) is the local mean at pixel (*i*, *j*).

Thresholding Considerations

A critical and frequent problem in segmenting an image into particle and background regions occurs when the boundaries are not sharply demarcated. In such a case, the determination of a correct threshold interval becomes subjective. Therefore, you may want to enhance your images before thresholding to outline where the correct borders lie. You can use lookup tables, filters, FFTs, or equalize functions to enhance your images. Observing the intensity profile of a line crossing a boundary area is also helpful in selecting a correct threshold value. Finally, keep in mind that morphological transformations can help you retouch the shape of binary particles and, therefore, correct unsatisfactory selections that occurred during thresholding.

us/innovations/automotive.html)

Aerospace, Defense, & Government (//www.ni.com/en-us/innovations/aerospace-defense.html)

Academic & Research (//www.ni.com/en-us/innovations/academic-research.html)

Wireless (//www.ni.com/en-us/innovations/wireless.html)

Electronics (//www.ni.com/en-us/innovations/electronics.html)

Energy (//www.ni.com/en-us/innovations/energy.html)

Industrial Machinery (//www.ni.com/en-us/innovations/industrial-machinery.html)

Heavy Equipment (//www.ni.com/en-us/innovations/transportation-and-heavy-equipment.html)

Partners (//www.ni.com/en-us/partners.html)

Activate a Product (//sine.ni.com/myproducts/app/main.xhtml)

Retrieve a Quote (//sine.ni.com/apps/utf8/niwq.retrieve_quote)

Terms of Service (//www.ni.com/en-us/about-ni/legal/service-terms.html)

Careers (//www.ni.com/en-us/about-ni/careers.html)

Investor Relations (//investor.ni.com/)

Newsroom (//www.ni.com/en-us/about-ni/newsroom.html)

Corporate Impact (//www.ni.com/en-us/about-ni/corporate-impact.html)

Supply Chain & Quality (//www.ni.com/en-us/about-ni/corporate-quality.html)

Events (//www.ni.com/en-us/events.html)

Discussion Forums (//forums.ni.com/t5/Discussion-Forums/ct-p/discussion-forums)

Submit a Service Request (//sine.ni.com/srm/app/getassistance)

□ (//twitter.com/NI_India_)

□ (//www.youtube.com/user/NatInstIndia)

□ (//www.facebook.com/NationalInstruments/)

□ (//www.instagram.com/niglobal/?hl=en)

(https://www.ni.com/global-gateway?rtrn=https%3A%2F%2Fzone.ni.com%2Freference%2Fen-XX%2Fhelp%2F370281AG-01%2Fnivisionconcepts%2Fthresholding%2F)

LEGAL (//www.ni.com/en-us/about-ni/legal.html) | PRIVACY (//www.ni.com/en-us/about-ni/legal/privacy-statement.html)