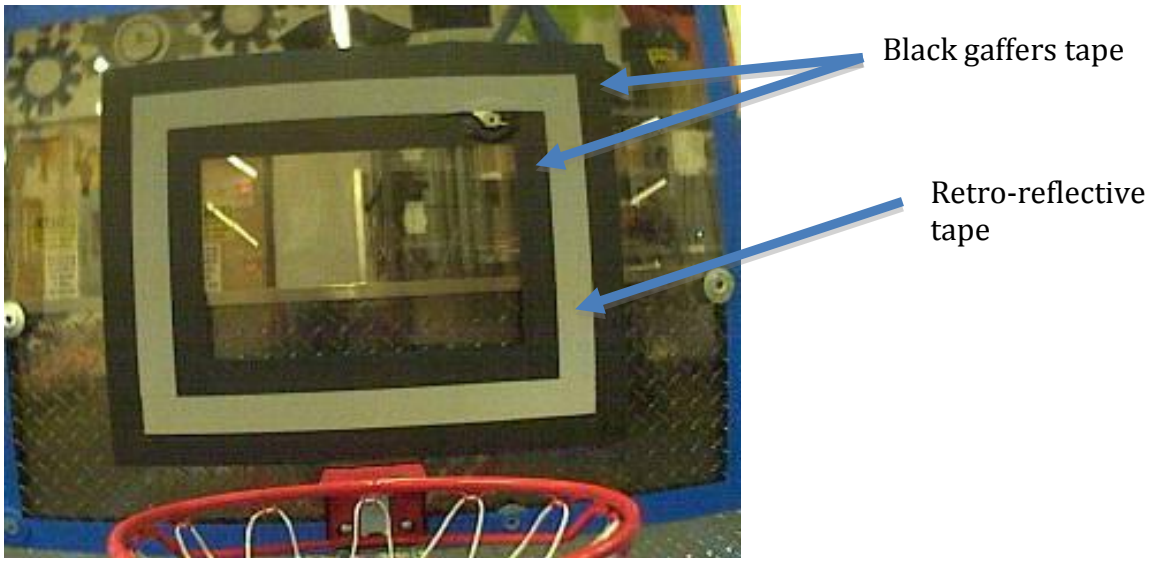
2012 Vision Information

This document describes the visual properties of the materials used to mark the field scoring elements. This document also describes several techniques for setting up the camera and processing images of the targets to extract measurements that may be useful in controlling your robot.

Materials:

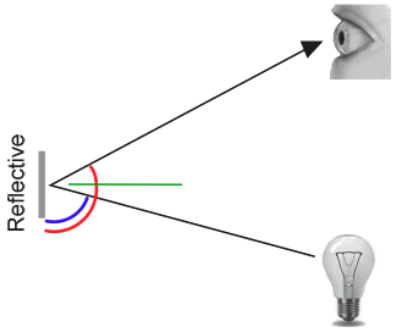
The backboard is marked using retro-reflective tape to build a rectangle, and black gaffers tape applied to both inner and outer borders. When properly lit, the retro-reflective tape produces a bright and/or color-saturated marker. The black tape helps to isolate the marker from background elements that may be similar in color or brightness.

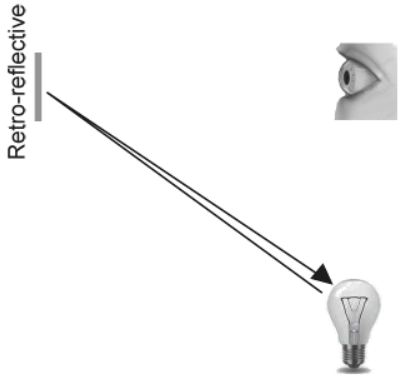


Retro-reflective Tape:

This material should be relatively familiar as it is often used to enhance nighttime visibility of road signs, bicycles, and pedestrians. The way the tape interacts with light is somewhat unexpected and easily confused with reflective materials, so a brief comparison is provided.

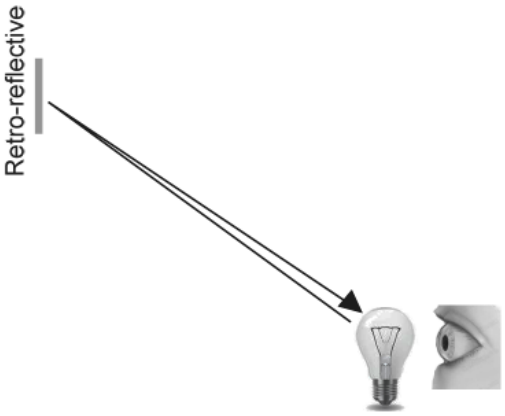
Highly reflective materials are generally mirrored so that light "bounces off" at a supplementary angle. As shown below-left, the blue and red angles sum to 180 degrees. An equivalent explanation is that the light reflects about the surface normal – the green line drawn perpendicular to the surface. Notice that a light pointed at the surface returns to the light source only if the blue angle is ~90 degrees.





Retro-reflective materials are not mirrored, but they typically have either shiny facets across the surface, or they have a pearl-like appearance. Not all faceted or pearl-like materials are retro-reflective, however. Retro-reflective materials return the majority of light back to the light source, and they do this for a wide range of angles between the surface and the light source, not just the 90 degree case. Retro-reflective materials accomplish this using small prisms, such as found on a bicycle or roadside reflector, or by using small spheres with the appropriate index of refraction that accomplish multiple internal reflections. In nature, the eyes of some animals, including house cats, also exhibit the retro-reflective effect typically referred to as night-shine. The Wikipedia articles on retro-reflection go into more detail on how retro-reflection is accomplished.

Initially, retro-reflection may not seem like a useful property for nighttime safety, but when the light and eye are near one another, as shown below, the reflected light returns to the eye, and the material shines brightly even at large distances. Due to the small angle between the driver's eyes and vehicle headlights, retro-reflective materials can greatly increase visibility of distant objects during nighttime driving.



Demonstration:

To further explore retro-reflective material properties, place a piece of the material on a wall or vertical surface, stand 10-20 feet away, and shine a small flashlight at the material. Start with the light held at your belly button, and raise it slowly until it is between your eyes. As the light nears your eyes, the intensity of the returned light will increase rapidly. If it does not, repeat using the other side of the strip. Alter the angle by moving to other locations in the room and repeating the process. The bright reflection should occur over a wide range of viewing angles, but the angle from light source to eye is key and must be quite small.

Experiment with different light sources. The material is hundreds of times more reflective than white paint; so dim light sources will work fine. For example, a red bicycle safety light demonstrates that the color of the light source determines the color of the reflected light. If possible, position several team members at different locations, each with their own light source. This shows that the effects are largely independent, and the material can simultaneously appear to be different colors to various team members. This also demonstrates that the material is largely immune to environmental lighting – the light returning to the viewer is almost entirely determined by a light source they control or one directly behind them. Using the flashlight, identify other retro-reflective articles already in your environment… on clothing, backpacks, shoes, etc.

Lighting:

We have seen that the retro-reflective tape does not shine unless a light source is directed at it, and the light source must pass very near the camera lens or the observer's eyes. While there are a number of ways to accomplish this, a very useful type of light source to investigate is the ring flash, or ring light, shown to the right. It places the light source directly on or around the camera lens and provides very even lighting. Because of their bright output and small size, LEDs are particularly useful for constructing this type of device.

As shown below, inexpensive circular arrangements of LEDs are available in a variety of colors and sizes and are easy to attach to the Axis cameras. While not designed for diffuse even lighting, they work quite well for causing retro-reflective tape to shine.

Image of various accent lights available from … http://www.superbrightleds.com/cgi-bin/store/index.cgi?action=DispPage&category=ACCENTS&Page2Disp=%2Fspecs%2FAE.htm#photos

Processing the Image for Goals:

Using the retro-reflective strip to locate the goals involves identifying hollow rectangles within a noisy image. This paper discusses two approaches – particle based processing that uses characteristics unique to the hollow rectangles and edge processing that locates edges and intersections, fits lines to the edges, and constructs polygonal descriptions of the bright borders in the image. First, we will look into the details of particle processing, followed by an edge-based approach.

Particle Processing:

The photos shown below demonstrate two lighting techniques to make target markings unique. On the left, a colored ring light causes the retro-reflective tape to shine with a unique color. On the right, a bright white light is used to make the shine very bright white compared to the background. The images on the following pages use NI Vision Assistant to demonstrate processing techniques to highlight and identify goals.

Original images:

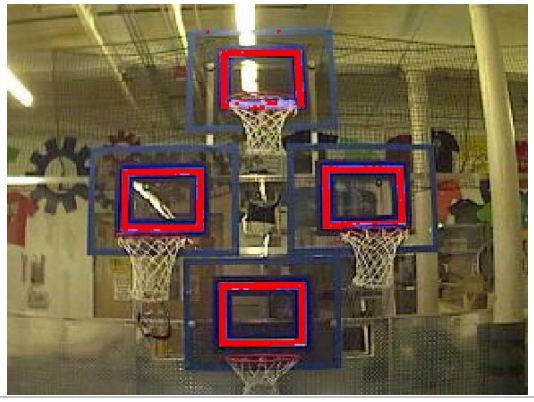
Masking the image:

In this initial step, pixel values are compared to constant color or brightness values to create a binary mask shown below as a red overlay. This single step eliminates most of the pixels that are not part of a target's retro-reflective tape. Color based masking works well, provided the color is relatively saturated, bright, and consistent. Color inequalities are generally more accurate when specified using the HSL (Hue, Saturation, and Luminance) or HSV (Hue, Saturation, and Value) color space rather than the RGB (Red, Green, and Blue) space. This is especially true when the color range is quite large in one or more dimension.

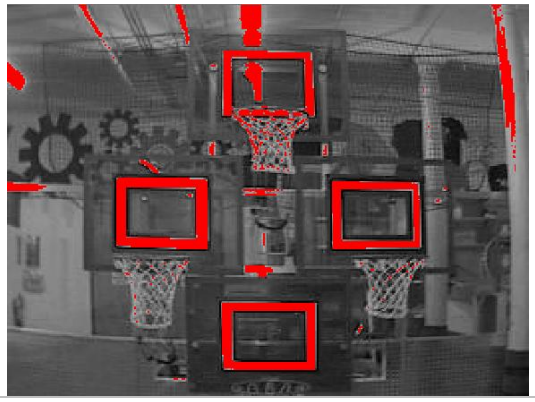
What is HSL/HSV?

The hue or tone of the color is commonly seen on the artist's color wheel and contains the colors of the rainbow: red, orange, yellow, green, blue, indigo, and violet. The hue is specified using a radial angle on the wheel, but in imaging, the circle typically contains only 256 units, starting with red at zero, cycling through the rainbow, and wrapping back to red at the upper end. Saturation of a color specifies amount of color, or the ratio of the hue color to a shade of gray. Higher ratio means more colorful, less gray. Zero saturation has no hue and is completely gray. Lightness or Value indicates the shade of gray that the hue is blended with. Black is 0 and white is 255.

HSL color threshold:
Hue range is between136 and 182
Saturation is above 45
Luminance is above 116



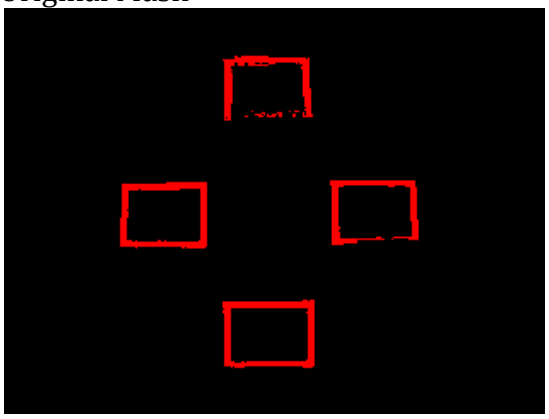
Color image converted to grayscale
Luminance above 168



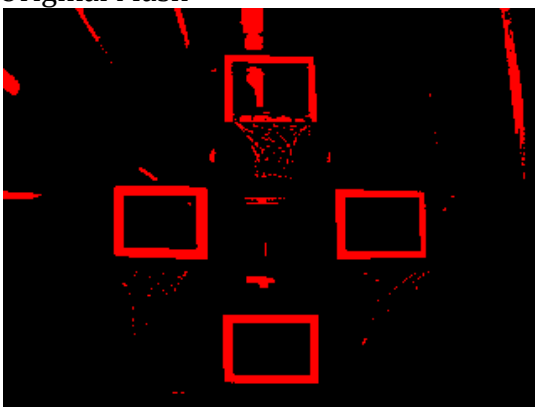
Identifying Rectangles:
The mask image contains groups of adjacent pixels referred to as particles or blobs. The next step is to identify characteristics of a target particles that distinguish it from noisy streaks as shown in the right mask, and characteristics that hold up to breaks in the rectangle caused by the rim, net, or other objects. It is also preferred if the characteristics do not degrade too badly when the target is distorted as the camera moves to the side of the field and views the targets from different angles and distances.

The example code that ships with LabVIEW applies a convex hull operation to the mask. This operation fills interior voids and tends to repair breaks in the rectangular tape. The definition of the operation is the same as the string or rubber band rule used for robot bumpers. Each particle is surrounded by a virtual rubber band, and all pixels within the band are added to the particle. A particle shaped like the letter V becomes a downward pointing triangle, a C becomes a filled-backwards D, etc. The next images show the original mask and the convex hull results.

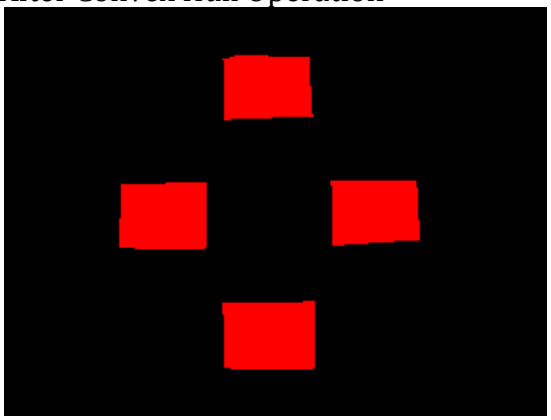
Original Mask



Original Mask



After Convex Hull Operation



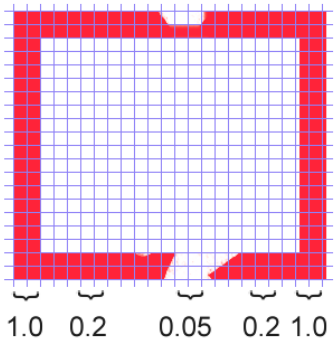
After Convex Hull Operation

As shown above, the primary role of the convex hull is to fill and repair the rectangles. After the convex hull operation, a particle report operation is used to examine the area and bounding rectangle edges for the particles. These are used to compute several scored terms to help pick the shapes that are most rectangular.

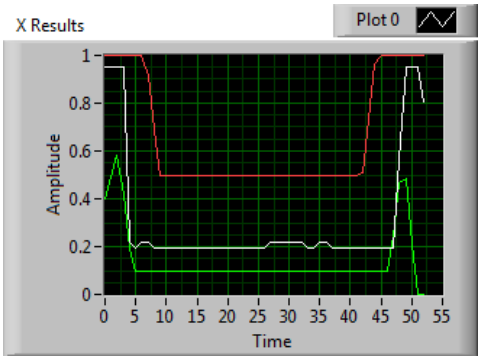
The rectangle score is calculated as (Particle Area / Bounding Box Area) * 100. A perfectly rectangular and level particle scores 100, a circular particle scores ($\pi/4$) * 100, or about 78, and the score drops even further for diagonal lines and other streaks, rotated, and distorted images.

The aspect ratio score is based on (Bounding Box Width / Bounding Box Height). The backboard target ratio is 24/18, or 1.33. The score is normalized to return 100 when the ratio is 4/3 and drops linearly as the ratio varies below or above.

The edge score describes whether the particle is appropriately hollow. As shown below, it is calculated using the row and column averages across the bounding box extracted from the original image and comparing that to a profile mask. The score ranges from 0 to 100 based on the number of values within the row or column averages that are between the upper and lower limit values.

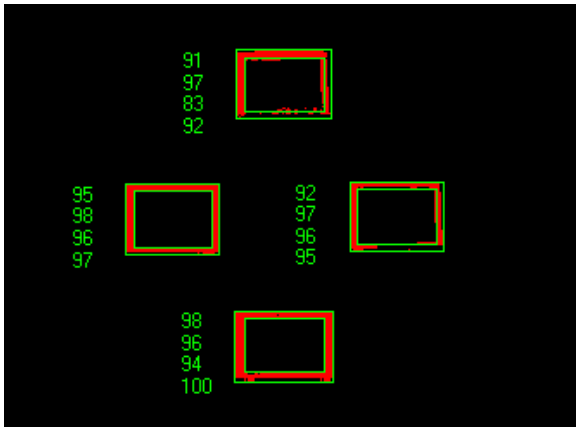


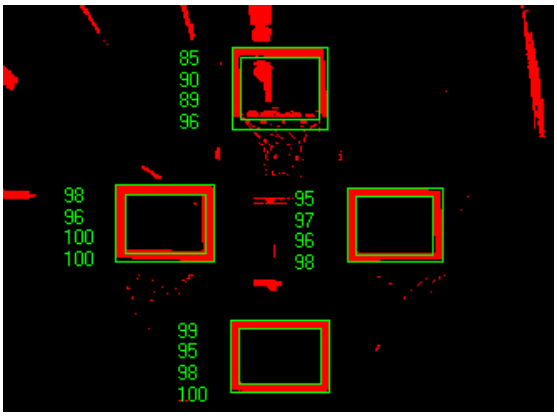
Column averages for a particle rectangle.



White line is the average, red is upper limit, and green is lower limit..

The images below show the image mask with target and score annotations. For the largest particles, numbers are displayed to the left, showing the rectangle percentage score, the aspect ratio score, the X and Y Edge scores. If all scores are above the limits, the particle is surrounded with a green inner and outer rectangle. If a score falls below the limit, its number is drawn in orange rather than green. This aids in determining why a particle fails to be considered a target, and you can decide to adjust the score limit or try and improve the image mask.



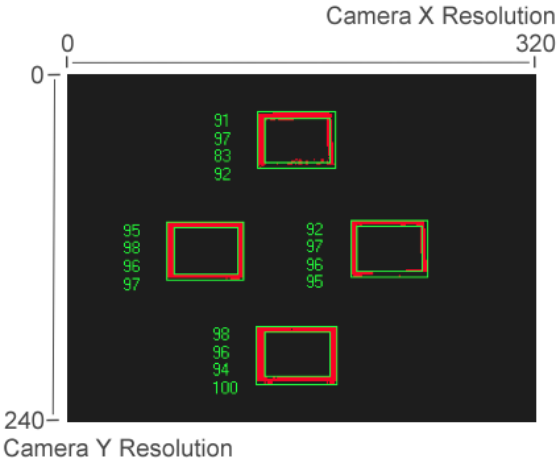


Measurements:

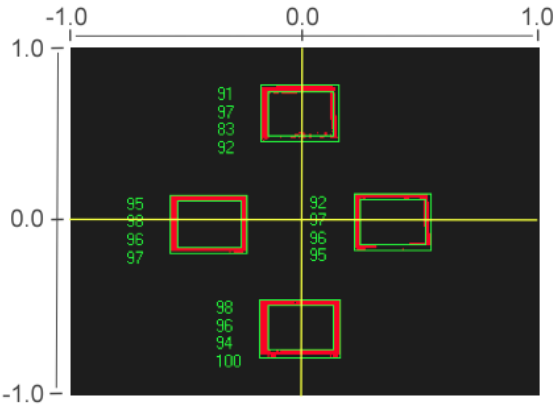
If a particle scores well enough to be considered a target, it makes sense to calculate some real-world measurements such as position and distance. The LabVIEW example code includes these basic measurements.

Position:

The target position is well described by both the particle and the bounding box, but all coordinates are in pixels with 0,0 being at the top left of the screen and the right and bottom edges determined by the camera resolution. This is a useful system for pixel math, but not nearly as useful for driving a robot. You can change it to something that may be more useful.



Pixel Coordinate System – $P_{x,y}$



Aiming Coordinate System – $A_{x,y}$

To convert a point from the pixel system to the aiming system, you can use the formula shown below.

$$A_{x,y} = (P_{x,y} - \frac{resolution_{x,y}}{2}) / \frac{resolution_{x,y}}{2}$$

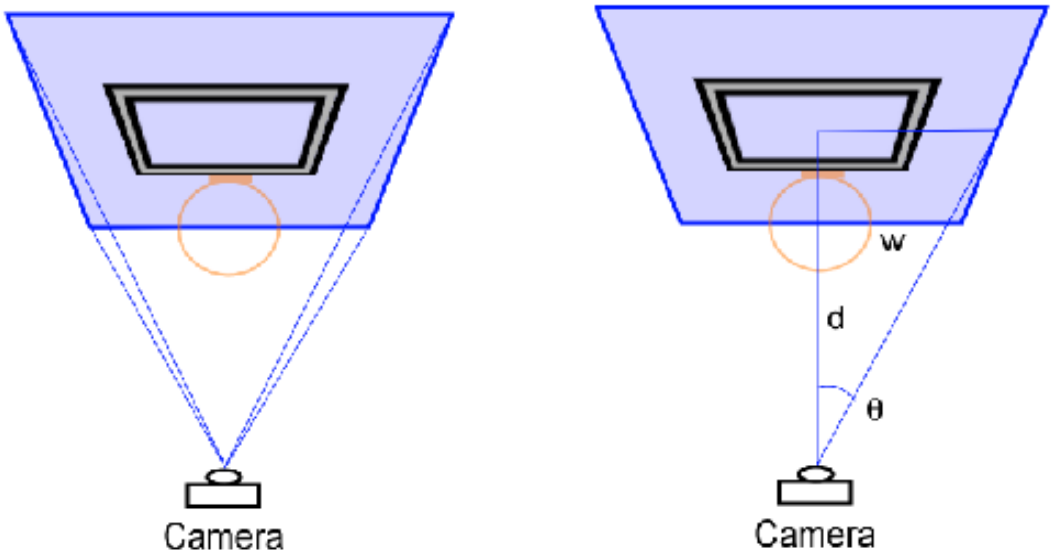
The resulting coordinates are close to what you want, but the Y axis is inverted, so you can correct this by multiplying the point by [1,-1]. This coordinate system is useful because it has a centered origin and the scale is similar to joystick outputs and RobotDrive inputs.

Distance:

The target distance is computed with knowledge about the target size and the camera optics. The approach uses information about the camera lens view angle and the width of the camera field of view. Shown below-left, a given camera takes in light within the blue pyramid extending from the focal point of the lens. Unless the lens is modified, the view angle is constant. As shown to the right, the values are related through the following trigonometric relationship.

$$\tan\Theta = w/d$$

Refer to the following URLs for the datasheets for the Axis cameras. http://www.axis.com/files/datasheet/ds_206_33168_en_0904_lo.pdf and http://www.axis.com/files/datasheet/ds_m10_40705_en_1009_lo.pdf. These give rough view angles for the lenses of 47˚ for the M1011, and 54˚ for the 206 model. Remember that this is for the entire field of view, and therefore the angle is 2Θ and the width is 2w.



The next step is to use the information you have about the target to find the width of the field of view – the blue rectangle shown above. This is possible because you know the target rectangle size in both pixels and feet, and you know the FOV rectangle width in pixels. You can use the following relationship.

$$T_{ft}/T_{pixel} = FOV_{ft}/FOV_{pix}$$

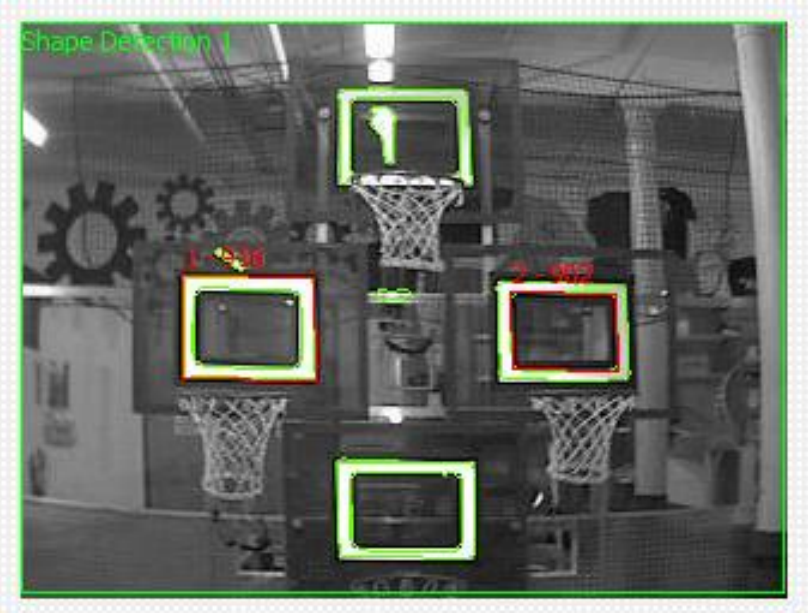
The target width measures 2 ft wide, and in the example images used earlier, the target rectangle measures 56 pixels when the camera resolution is 320x240. This means that the blue rectangle width is 2*320/56 or 11.4 ft. Half of the width is 6.7 ft, and the camera used is the M1011, so the view angle is ~47˚, making Θ equal to 23.5˚.

Putting this information to use, the distance to the target is equal to 6.7/tan23.5˚ or 15.4 ft. Notice that the datasheets give approximate view angle information. When testing, the distance to the target tended to be a bit short. Using a tape measure, the view angles of 48˚ for the 206 and 43.5˚ for the M1011 gave better results for calculating distance. To perform your own calibration, eliminate perspective distortion by placing the camera and target in the same horizontal and vertical plane, and use a tape measure between the camera lens and the target center point.

Edge Detection:

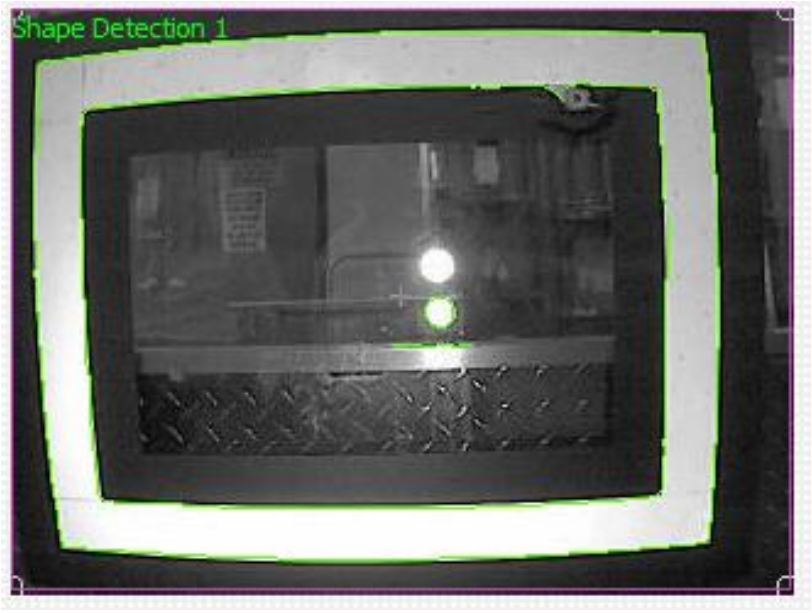
An alternative image processing approach searches out edges in the image, where an edge is defined as the border between an adjacent bright and dark area. This technique locates the strong edges on the borders of the target formed by the tape shine and the black gaffers tape. The edges can then be mapped to geometric shapes for further processing.

In the image shown below, NI Vision Assistant has been used to locate edges within the monochrome image. The edges are also called contours and are traced in green. The contours that form rectangles are also scored and marked in red. Rectangles with scores below 900 are not shown.



The pixel conversions and distance calculation shown above are equally valid for mapping results of edge detection to real-world coordinates. We have seen edge detection can work quite well in a simple image above, but there are some interesting conditions to be aware of. Notice that each rectangle above has both an inner and outer edge. On the left, the outer scored higher. On the right, the inner scored higher. If you lower the score limit, you start to receive a number of rectangles for the same target.

The following image shows a target with contours marked in green. It is not marked as a rectangle, however, because the lines that connect the corners are not actually straight. The camera lens distorted the image – turning real-world lines into arcs. This particular distortion is often called a barrel distortion. You can measure the lens distortion and correct the images taken with that lens before or during processing, but the calibration process is rather involved and the correction requires considerable processing overhead. The camera is clearly very close to the target, and probably isn't typical.



Another challenge is shown in the image below. The camera is located to the right, a bit below the target, and the plane of the sensor is not parallel to the plane of the target. Since objects farther from the camera are smaller than closer objects, the length of opposite rectangle edges are not equal, and they do not meet at 90 degrees. This form of distortion is called perspective distortion. Knowing that the quadrilateral is actually a rectangle, it is possible to determine the distance from camera to each edge and locate the camera in 3D space, or use it to correct the image distortion. The image to the left shows the contours and the lack of red indicates that the shape algorithm does not consider the edges to form a rectangle. Shown to the right, the contours are fit with lines, and with some work, it is possible to identify the shared points and reconstruct the quadrilateral and therefore the perspective rectangle.

Camera Settings:

It is very difficult to achieve good image processing results without good images. With a light mounted near the camera lens, you should be able to use any basic example, the dashboard, or framework code to view the results. It is also possible to open NI Vision Assistant or a web browser to view camera images and experiment with camera settings.

Focus:

The Axis M1011 has a fixed-focus lens and no adjustment is needed. The Axis 206 camera has a black bezel that rotates to move the lens in and out. Ensure that the images you are processing are relatively sharp and focused for the distances needed on your robot.

Compression:

The Axis camera returns images in BMP, JPEG, or MJPG format. BMP images are quite large and take more time to transmit to the CompactRIO and laptop. Therefore the WPILib implementations typically use JPG or MJPG – motion JPEG. The compression setting ranges from 0 to 100, with 0 being very high quality images with very little compression, and 100 being very low quality images with very high compression. The camera default is 30, and it is a good compromise, with few artifacts that degrade image processing. Due to implementation details within the VxWorks memory manager, you may notice a performance benefit if you keep the image sizes consistently below 16 kBytes. Due to the way that the JPEG compression algorithm works, image size is dependent on the amount of detail or pattern contained in the image. The images below are an example of how compression impacts size, but do not predict the size of arbitrary images.

320x240 Color Image: Compression set to 0. Image file size is 20,715 bytes. High quality, but large in size. May be slower to compress and decompress -- which impacts frame rate.

320x240 Color Image: Compression set to 30. Image file size is 6,450 bytes. Good quality, relatively small in size. Some image artifacts are present on edges.

320x240 Color Image: Compression set to 100. Image file size is 2,222 bytes. Poor quality for processing. Notice blocky artifacts and rough edges.

Size:
Image sizes shared by the supported cameras are 160x120, 320x240, and 640x480. The M1011 has additional sizes, but they are not built into WPILib. The largest image size has four times as many pixels that are one-fourth the size of the middle size image. The large image has sixteen times as many pixels as the small image.

From the example above, we can see that at ~15 feet, a 24 inch target is 56 pixels wide in a medium sized image. The tape used on the target is 2 inches wide, and for good processing, you want that 2 inch feature to be at least two pixels wide. Using the distance equations above, you can see that a medium size image is fine up to the point where the field of view is around 320 inches or almost 27 feet – the width of the FRC field. This occurs at around 30 feet, or mid-field. The small image size should be usable for processing to a distance of about 15 feet.

Image size also impacts the time to decode and to process. Smaller images are roughly four times faster than the next size up. If the robot or target is moving, it is quite important to minimize image processing time since this adds to the delay between the target location and perceived location. If both robot and target are stationary, processing time is typically less important.

Color Enable:
The Axis cameras typically return color images, but are capable of disabling color and returning a monochrome or grayscale image. The resulting image is a bit smaller in file size and considerably quicker to decode. If processing is carried out only on the brightness or luminance of the image, and the color of the ring light is not used, this may be a useful technique for increasing the frame rate or lowering the CPU usage.

Laptop or CompactRIO:
The LabVIEW example shows that it is possible to process the camera images on either the laptop or CompactRIO using the same code. What are the advantages and tradeoffs? The table, shown below, sums up the more important considerations.

	CompactRIO	Laptop
CPU speed	400MHz	At least 1.6 GHz
Vector instructions	None	MMX or SSE
Image data	Located in RAM	Transmitted via TCP
NI IMAQ Vision	Supported	Supported
OpenCV	N/A	Supported

The CPU speed and vector instructions often mean that image processing can take 1/10th the time on a laptop than on the CompactRIO. To take advantage of this and distribute the processing, you need to send the image from the robot to the dashboard laptop, process it, and then send the results back to the robot. Transmitting data and images also takes time, so the best location to process images

depends on all of these factors. You may want to take measurements or experiment to determine the best approach for your team.

WB and Exposure:
If the color of the light shine is used to identify the marker, it is key to control the camera settings that affect the image coloring. The most important setting is white balance. White balance controls how the camera blends the component colors of the sensor in order to produce an image that matches the color processing of the human brain. The camera has five or six named presets, an auto setting that constantly adapts to the environment, and a hold setting for custom calibration. The easiest approach is to use a named preset; one that maintains the saturation of the target and does not introduce problems by tinting neutral objects with the color of the light source. To custom-calibrate the white balance, place a known neutral object in front of the camera. A sheet of white paper is a reasonable object to use. Set the white balance setting to auto, wait for the camera to update its filters (ten seconds or so), and switch the white balance to hold.

The brightness or exposure of the image also has an impact on the colors that are reported. The issue is that as overall brightness increases, color saturation starts to drop. The following example shows how this occurs. A saturated red object placed in front of the camera returns an RGB measurement high in red and low in the other two components – e.g. (220, 20, 30). As overall white lighting increases, the RGB value increases to (240, 40, 50), then (255, 80, 90), then (255, 120, 130), and then (255, 160, 170). Once the red component is maximized, additional light can only increase the blue and green, and dilutes the red percentage in the measured color, lowering the saturation value of the color. If the point is to identify the red object, it is useful to adjust the exposure to avoid diluting your principle color. The desired image often looks somewhat dark except for the colored shine.

There are two approaches to control camera exposure times. One is to allow the camera to compute the exposure settings automatically based on its sensors, and then adjust the camera's brightness setting to a small number to lower the exposure time. The brightness setting acts similar to the exposure compensation setting on SLR cameras. The other approach is to calibrate the camera to use a custom exposure setting. Change the exposure setting to auto, expose the camera to bright lights so that it computes a short exposure, and then change the exposure setting to hold. Both approaches result in an overall dark image with brightly saturated target colors that stand out from the background and are easier to mask.