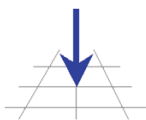
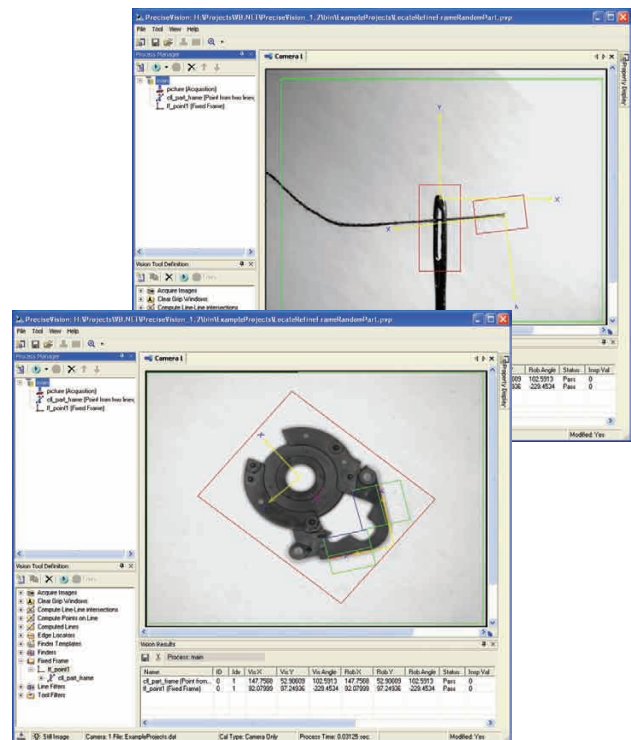


PRECISEVISION

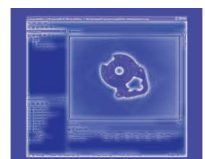
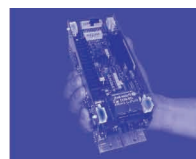
Can a vision system be so easy to use that a vision guided motion program fits on a single piece of paper?

Precise Automation software makes it a reality. PreciseVision software works seamlessly with any Guidance Controller including those in Precise robots. PreciseVision's built-in interface to Guidance Controllers makes incorporating vision information into a motion application extremely simple and enables the robot to respond almost instantly to new vision information.

PreciseVision provides a simple solution for applications requiring locating parts, correcting for manufacturing tolerances, inspecting parts or visually closing a position loop (visual servoing). It features an easy-to-use graphical programming interface, patented object locator, visual inspection, visual measurement tools and more.



**PRECISE
AUTOMATION**



<i>General Specifications</i>	<i>Range & Features</i>
Programming interface	Windows PC application that enables vision tools to be graphically positioned on camera images. Tools can be chained together to produce more complex vision processing.
Visual tools	
Object identification and locating	Patented Object "Finder" Tool – locates and identifies translated, rotated and scaled objects in grayscale images in a matter of milliseconds. Extremely easy to use for objects with both simple and complex 2D grayscale features. Blob Finder – Locates isolated regions in binary images and returns the region's statistics. Provides an extremely fast alternative to the Finder for simple binary images
Feature measurement	Line Fitter Tool – Provides an easy method for locating linear edges in the camera image. This tool can be used together with others to refine the position or orientation of an object. Arc Fitter Tool – Locates the center and/or the radius of a circle or an arc. Edge Locator Tool – Searches for edges along a linear path through an image. Rotating Edge Locator Tool – Arc version of the Edge Locator Tool.
Inspection	Pixel Counting Tool – Returns statistical data on the grayscale or binary pixels within a rectangular region of an image. Histogrammer Tool – Returns statistics on the distribution of grayscale values within a region of an image. Pixel Color Window Tool - Tests if the color of a region matches a trained color.
Computed results	Clear Grip Tool – Defines a bounding area that can be used to represent robotic gripper fingers. Tests to ensure that the regions for the fingers are free of objects. Computed "Feature" Tools – A group of tools that can compute geometric relationships between the results of other tools, e.g. the point of intersection of two lines, the line represented by two points, etc. Tool Filter – Applies a filter to a tool that returns multiple results, e.g. the Finder, and returns only the results that satisfy specific criteria.
Image Acquisition Processing	Acquire Tool – Performs the basic image acquisition function from a camera. This tool allows the user to specify which physical camera to utilize and to easily store images to a disk file for later replay. Motion Sensing Tool – Indicates when all objects in a rectangular region are stationary. Convolving Tool – Applies a number of low-level filtering operators to a region of an image, e.g. low pass, Laplacian, Gaussian, sharpening, etc. Morphology Tool – Modifies all pixels in an image region based on the value of neighboring pixels. Typical operations include image "dilation" and "erosion".
Tool linking	One of the most powerful features of PreciseVision™ is the ability to link one vision tool to another. This provides the required features to build up custom vision algorithms without the need to write code.
Features	Built-in robot-to-vision calibration procedures for many configurations including stationary and arm-mounted cameras; gain/offset configuration from within a vision process.
Cameras	Directly interfaces to DALSA GigE Ethernet Cameras or IDS uEye USB Cameras. Includes support for standard 640x480 and higher resolution, grayscale and color cameras.

SAMPLE VISION-GUIDED MOTION CONTROL PROGRAM

On command from the robot program, a picture is taken and the image is analyzed. If a specified vision tool finds a part, the robot moves above the part, reaches down to touch the part, moves back up, and finally moves to a safe location.

```

1: Public Sub MAIN
2:   Dim vis As New Vision
3:   Dim vResult As New VisResult
4:
5:   Robot.Attached = 1
6:   Move.Loc (safe, vsProfile)
7:
8:   vis.Process("Main") ' Run vision process "Main"
9:   If vis.ResultCount("part1") = 0 Then
10:     Console.WriteLine("Vision object not found")
11:     Goto done
12:   End If
13:   vResult = vis.Result ("part1", 1) ' Get results
14:
15:   ' Create a reference frame object and set it
16:   ' equal to the returned vision location
17:   Dim vsRefFrame As New RefFrame
18:   vsRefFrame.Loc.PosWrtRef = vResult.Loc
19:
20:   ' Pickup point is relative to new frame
21:   vsRelPoint.RefFrame = vsRefFrame
22:
23:   Move.Approach (vsRelPoint, vsProfile)
24:   Move.Loc (vsRelPoint, vsProfile)
25:   Move.Approach (vsRelPoint, vsProfile)
26:
27:   ' Move back to safe location
28:   Move.Loc (safe, vsProfile)
29:
30: done:
31: End Sub

```



automate with ease