FlexiVision

ORIGINAL INSTRUCTIONS

USER MANUAL

FlexiVision

Release	1.0
Date of release	03.2021
Notes	0.0

ARS S.r.l.
Via G. Vico, 7 – 52100 Arezzo (AR) Italia
Tel. +39 0575 398611 - Fax +39 0575 398620
info@arsautomation.com - www. arsautomation.com

FOREWORD

ARS s.r.l. is by no means liable for the consequences of incorrect operations performed by the user.

EDITOR'S NOTE

This documentation is expressly addressed to technicians. Therefore, information that can be easily retrieved by reading these texts and analysing the drawings may not be explained further.
The Editor is by no means liable for any information and data provided in this manual: all information included herein has been supplied, controlled and approved by the Manufacturer during review.
The Editor shall by no means be held responsible for the consequences resulting from the user's misuse.

GENERAL REMARKS

All operating instructions and recommendations described in this manual must be respected.
The training of the personnel in charge of using this software is of the utmost importance, both as regards the use and maintenance of the connected devices and the monitoring of all operating procedures and of all safety standards listed in the relevant instruction manuals.

***Revision**: 00*
***Edition**: 03/2021*

TABLE OF CONTENTS

1 IDENTIFICATION

1.1 Software developer identification

Developer	ARS s.r.l.
Address (registered office)	Via G. Vico, 7 52100 Arezzo (AR) - Italia Tel. +39 0575 398611 - Fax +39 0575 398620 info@arsautomation.com - www.arsautomation.com
Address (operational headquarters)	Via G. Vico, 7 52100 Arezzo (AR) - Italia Tel. +39 0575 398611 - Fax +39 0575 398620 info@arsautomation.com - www.arsautomation.com

1.2 Software identification

Model	FLEXIVISION
Release	1.0
Date of release	03/2021

PAGE INTENTIONALLY LEFT BLANK

2 PRELIMINARY INFORMATION

2.1 Addressees

This manual is destined to operators in charge of dealing with the software in all the phases of its technical life. It also contains the subjects regarding the proper use of the software, in order to maintain its functional and qualitative features unaltered over time.
All information and warnings for proper safe use are also reported.
The manual is an integral part of the software and must always accompany it in every displacement or property transfer. Once the software has been installed, the user must keep this documentation intact and make it available for consultation during the entire lifespan of the connected devices.

2.2 Supply and preservation

The manual is supplied in **electronic format.**
The manual is an integral part for the purpose of use and safety, therefore:

- **it must be kept intact** (in all its parts). Should this manual get damaged or spoilt, request a copy immediately.
- **It must always accompany the software** (even if moved, sold, leased, rented, etc.).

ARS s.r.l. shall not be held liable for software misuse and/or damages resulting from operations not indicated on the technical documents.

2.3 Updates

Should the software require functional modifications on request of the User, ARS s.r.l. shall revise or update the manual.
The user is also responsible for ensuring that, should this document be modified by ARS s.r.l., only the updated manual versions are effectively present in the points of use.

2.4 Language

The original manual has been written in **English.**
Any translations into other languages must be done from the original instructions.
ARS s.r.l. shall be responsible for the information contained in the original instructions; translations into different languages cannot be fully verified, hence should an inconsistency be detected, the text in the original language must be referred to or contact our Technical Assistance Department.

2.5 Skills and qualification required

Type	Definition
Trained Person	Person informed, educated and trained on the work and on any dangers deriving from an improper use. Also knows the importance of the safety devices, the accident-prevention standards and the safe work conditions.

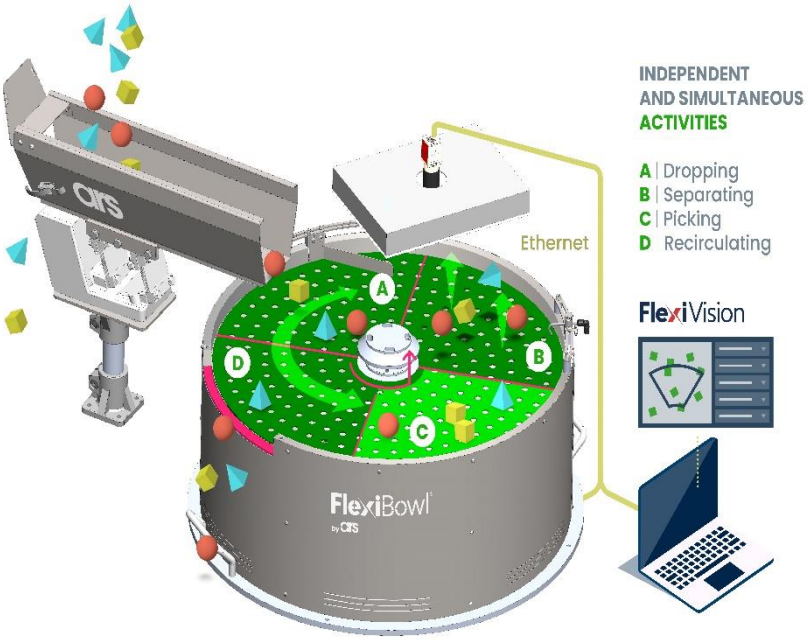
2.6 Symbols used in the manual

Symbol	Definition
	Symbol used to identify important warnings for the safety of the operator and/or machine.
	Symbol used to identify particularly important information inside the manual. The information also regards the safety of personnel involved in use of both the software and the connected devices.
	Obligation to read the external instruction manuals/booklets.

3 GENERAL DESCRIPTION

3.1 What is FlexiVision®

FlexiVision® is a PC-based vision software for robot guidance, optionally supplied to FlexiBowl® systems: thanks to this software, bulk parts can be downloaded to the feeder surface, separated, recognized and picked up.



Position	Element/section	Description
0	Connectivity	Digital I/0, TCP/IP, UDP, Ethernet-IP
1	Linear Hopper (Flow feeder)	Drops components and rear emptying
2	Diverter/ Blow Unit	Diverts components from the ring
3	Rotary Disc	Custom Discs upon request
4	Flip Unit	Separates components
5	Backlight	
6	Quick Emptying	Automatic Product Changeover
7	Quick Release	Quick Disc Change
8	Toplight	

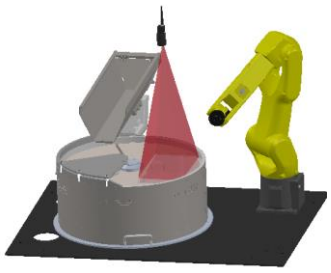
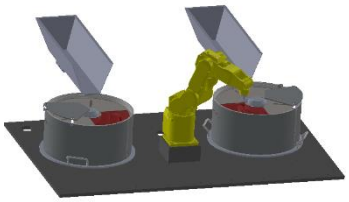
Position	Element/section	Description
9	FlexiVision System	Sends parts coordinates to the robot. Controls feeder movements and manages the parts flow from the hopper. Parts database manager
a	Dropping sector	
b	Separating sector	
c	Picking sector	
d	Recirculating sector	

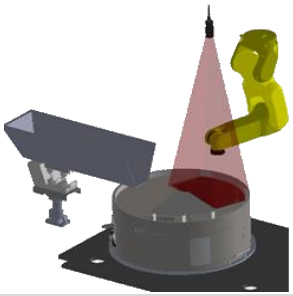
FlexiVision, developed on Cognex vision libraries, supports multiple cameras and offers a wide flexibility of use.

The camera is usually placed at the centre of the picking area.
The height is based on the FlexiBowl® dimensions and the desired resolution.

The system is compatible with a large variety of industrial robots and allows rapid development of automated cells with FlexiBowl®.

3.1.1 Typical configuration examples

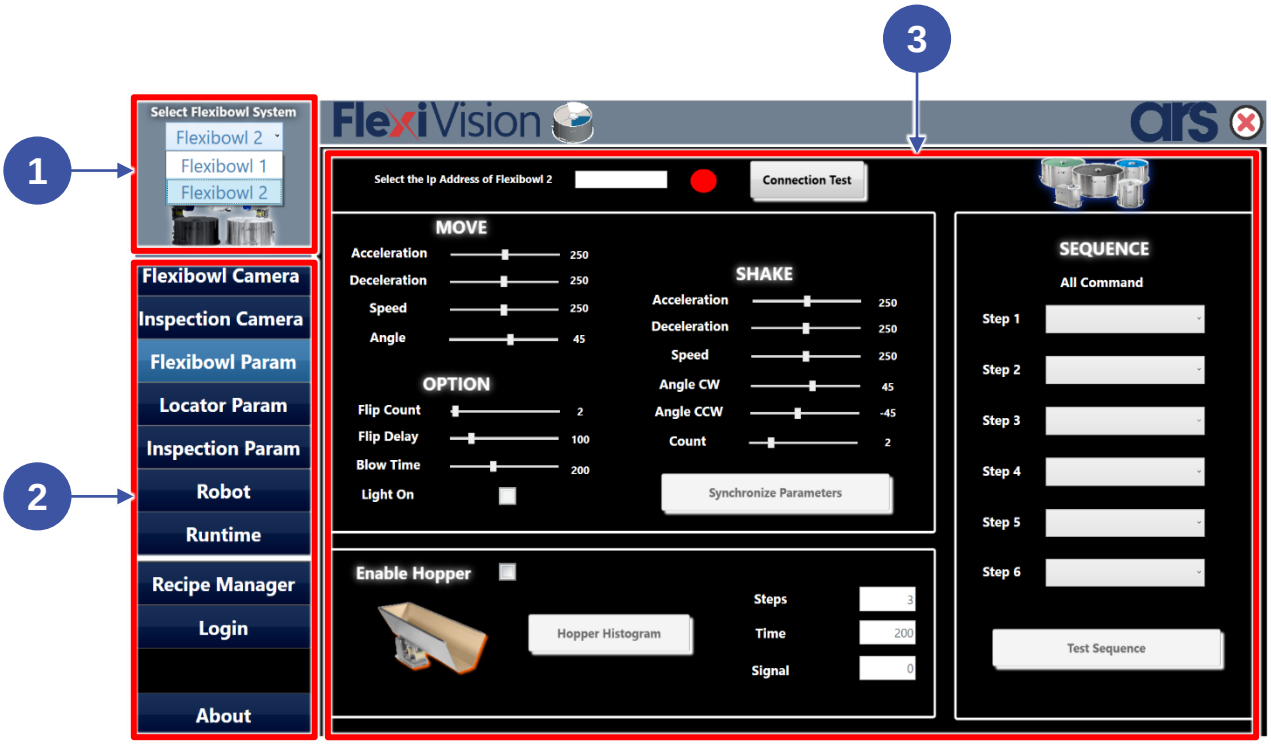
Description	Picture
Robot, 1 FlexiBowl, Camera and Bulk feeder	
Robot, 2 FlexiBowls, Camera and Bulk feeder	

Description	Picture
Top-Mount Robot, 1 FlexiBowl, Camera and Bulk feeder	

FLEXIVISION can manage a system composed of:

- **up to 4 cameras.**
- **up to 2 FlexiBowls®.**
- **up to 2 robots.**

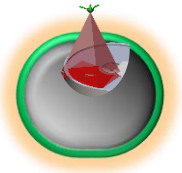
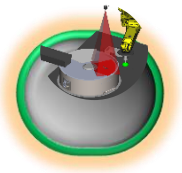
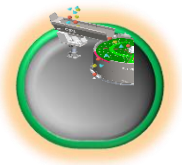
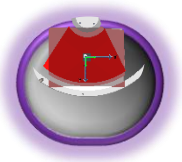
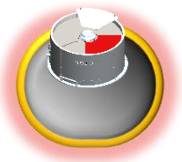
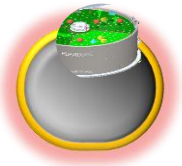
3.2 Operator interface

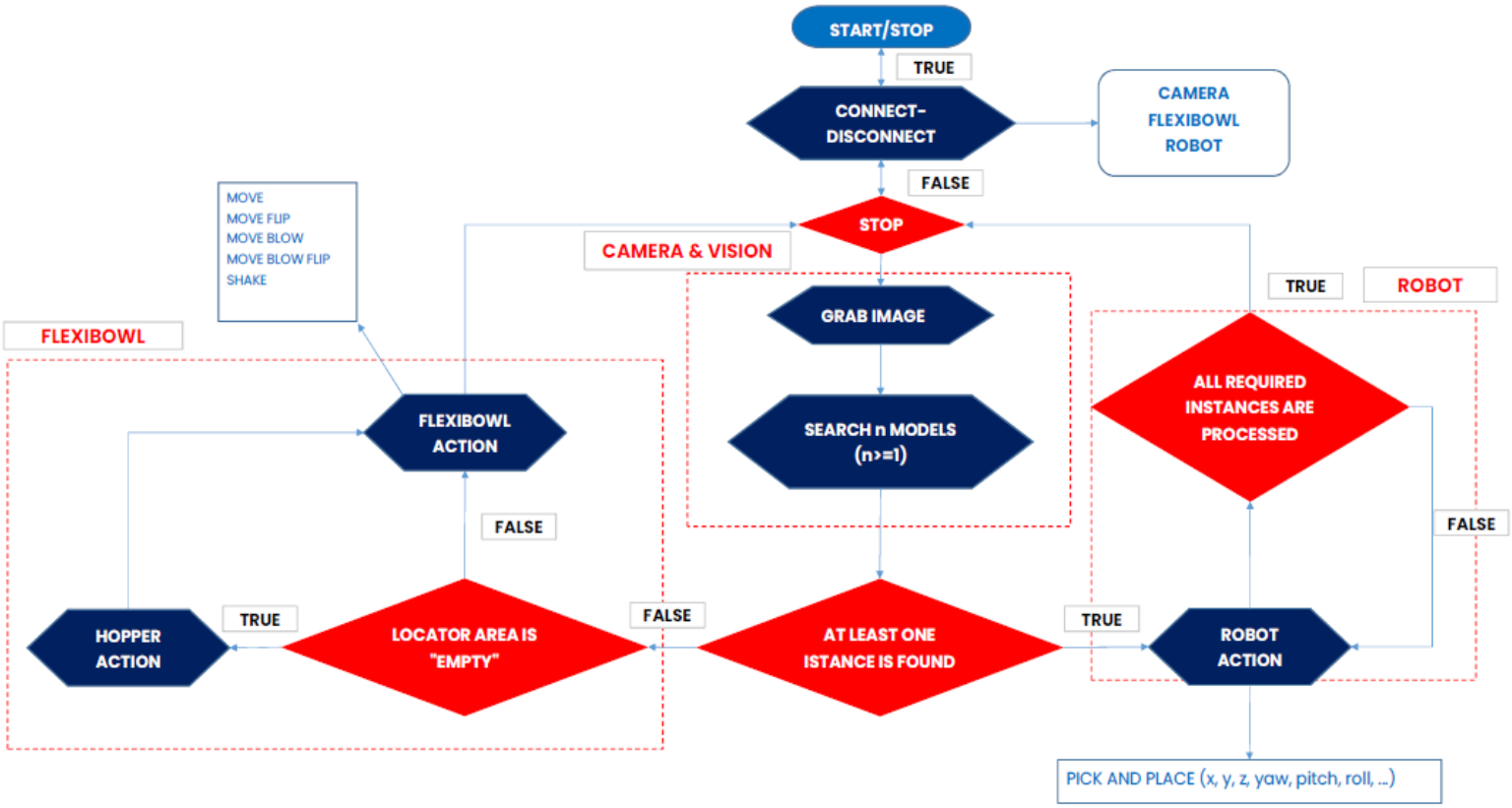


Position	Element/section	Description
1	FLEXIBOWL SELECTION A drop-down menu is available to select the Flexibowl system (1 or 2). ▪ Selected Flexibowl becomes coloured. ▪ Non-selected Flexibowl is black.	
2	OPERATION MENU: ▪ FLEXIBOWL CAMERA ▪ INSPECTION CAMERA ▪ FLEXIBOWL PARAM ▪ LOCATOR PARAM ▪ INSPECTION PARAM ▪ ROBOT ▪ RUNTIME ▪ RECIPE MANAGER ▪ ABOUT	Includes all the control and operation procedures. According to the access level, press to select and enter the required operation page.
3	CENTRAL SCREEN	The parameters or images are displayed, according to the selected operation.

3.3 General workflow

Two Parallel Tasks are recommended: one to handle robot movements, the second one to handle vision, Bulk Feeder and Flexibowl®.

Step	Action	Notes/Pictures
1	Camera locates parts	
2	Pick&Place	
3	Part dropping	
4	Check part quantities	
5	Move command	
6	Flip command	


START/STOP
TRUE
CONNECT-DISCONNECT
CAMERA
FLEXIBOWL
ROBOT
FALSE
STOP
MOVE
MOVE FLIP
MOVE BLOW
MOVE BLOW FLIP
SHAKE
CAMERA & VISION
GRAB IMAGE
TRUE
ROBOT
FLEXIBOWL
ALL REQUIRED INSTANCES ARE PROCESSED
FLEXIBOWL ACTION
SEARCH n MODELS (n>=1)
FALSE
HOPPER ACTION
TRUE
LOCATOR AREA IS "EMPTY"
FALSE
AT LEAST ONE ISTANCE IS FOUND
TRUE
ROBOT ACTION
PICK AND PLACE (x, y, z, yaw, pitch, roll, ...)

3.4 Main features

- Dynamic control of all FlexiBowl® features
- Integrated calibration
- Configuration of the hopper management algorithm
- Complete configuration of the recognition tools
- Complete configuration of inspection tools
- Saving and dynamic management of the recipe database
- Definition of multiple vision models for each single recipe
- Management of communication with the Robot
- Recognition of components position and orientation

Thanks to our software, bulk parts can be downloaded to the feeder surface, separated, recognized, and picked up. FlexiVision is developed using Cognex vision libraries and provides the possibility to add custom sequences to introduce inspection capabilities into the application. FlexiVision provides a simple programming environment to make FlexiBowl® system integration easy and with guaranteed results.

3.5 Minimum system requirements

Element	Features
Vision PC 	*SO: Windows 10 64 bit* *Processor: intel i7* *Graphics card: NVIDIA or integrated* *RAM:8/16 GB* *SSD: 250 GB* *At least, one free USB port*
Camera 	*Resolution: 2592x1944* *Frame Rate: 14 fps* *Sensor dimensions: 1/2.5"* *Sensor type: CMOS* *Protocol: GigE*

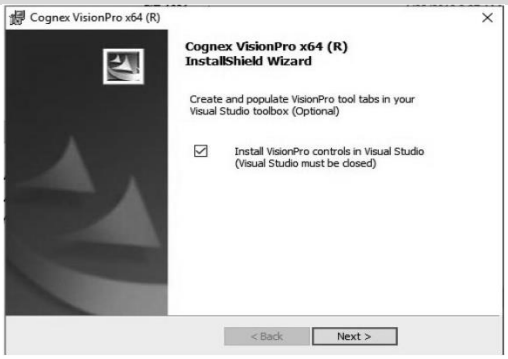
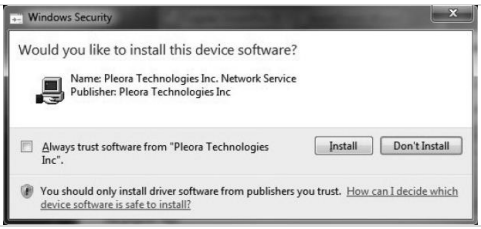
Element	Features
Robot 	*TCP/IP protocol compatibility* *String manipulation*
Switch GigE 	*4 Ethernet ports* *4 Ethernet POE ports*

4 SOFTWARE INSTALLATION

4.1 How to install Vision Pro and Designer

The following is an extract from the Cognex User Manual: for further and complete details, please refer to the whole Cognex User Manual.

Step	Action	Notes/Pictures
1	Uninstall any existing Vision Pro software.	Use the Add or Remove Programs feature of the Windows Control Panel to remove any of the following programs if they are present: • *Cognex VisionPro* • *Cognex Drivers* • *Cognex Japanese Documentation* • *Cognex Software Licensing Center.* Restart your computer if prompted.
2	Turn off the computer and install the vision hardware necessary for your vision application.	
3	Attach the necessary cameras, I/O devices and Cognex Security Key and turn on the computer.	 ATTENTION Microsoft might display the *Found New Hardware Wizard*. Select *Cancel*.
4	Install the VisionPro software. Windows 8.1/10 users: Install VisionPro from Windows Desktop. Launch the *setup.exe* application from the installation media.	 ATTENTION • You must have Administrator privileges to install VisionPro. • If your computer does not already have Microsoft Visual Studio Redistributables installed, they will automatically be installed. • If your computer requires an update to the Windows Installer, it will automatically be updated. Depending on how your system is configured, you might see a message about this update taking piace.

Step	Action	Notes/Pictures
5	(Optional) By default the Installer launches a VisionPro Edit Control extension to populate the tool palette in Visual Studio with the latest VisioPro edit controls. The extension detects which version of Visual Studio you have installed and will populate the tool palette for all installed versions by default.	 ATTENTION Be aware the extension does not support automated population of VisionPro edit controls with Visual Studio Express.
6	(Optional) Install the Asian language versions of the documentation.	ATTENTION By default the VisioPro installation will not install Asian language versions of VisionPro documentation.
7	Install the VisionPro Hardware drivers. The Cognex drivers utility will launch once the VisionPro is installed	
	The *Cognex drivers* utility will install the GigE drivers and all the frame grabber drivers: you can, as an option, select *Custom* in the *SetupType* dialog and pick the specific drivers you want to install. The drivers are Authenticode signed with a Cognex certificate.	
	The Installer might display a Windows Security warning for the GigE Vision driver from Pleora Technologies.	
	You must click *Install* to use the correct GigE Vision drivers for best performance.	

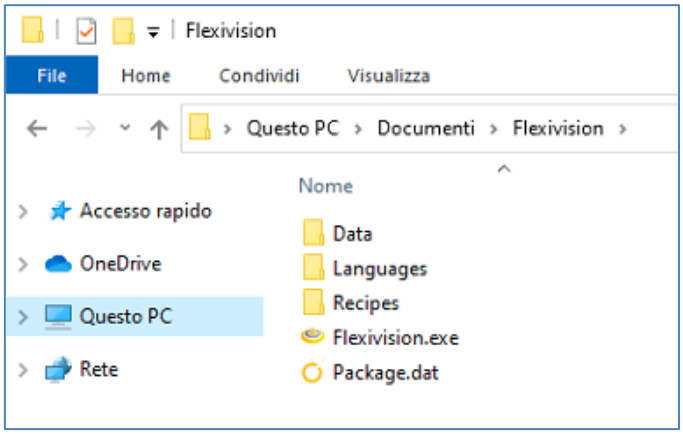
Step	Action	Notes/Pictures
8	ATTENTION If you do not install any drivers during the initial software installation and want to install one or more drivers later, re-insert the installation media and double-click on *setup.exe* in the *drivers* directory to launch the Cognex Drivers utility. Reboot the computer if prompted. If you install additional image acquisition hardware later, use the Wiindows Control Panel to select the Cognex Drivers software for repair and modifications.	

4.2 How to insert the Dongle key

Step	Action	Notes
1	To run FlexiVision, correctly insert the hardware license key in the USB port of the PC.	

4.3 Projects files

Project files are stored into the FLEXIVISION folder.



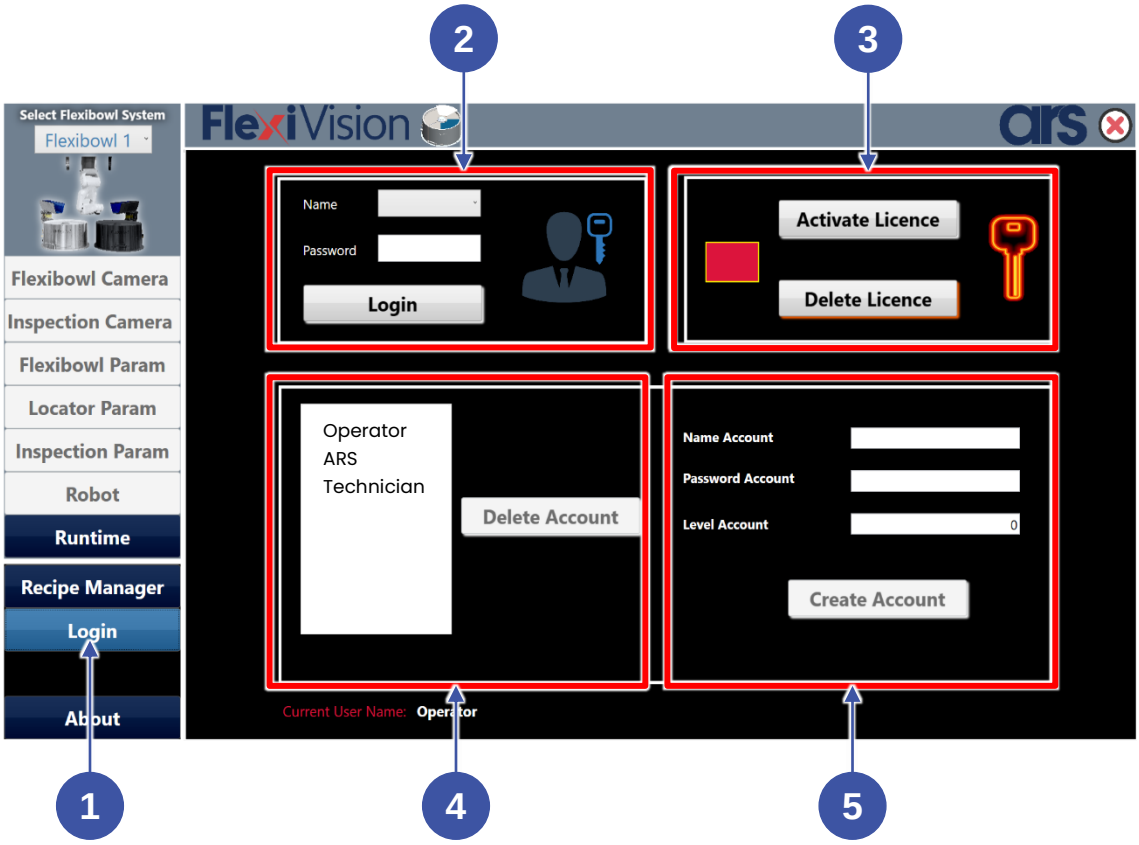
WARNING!
Do not modify and/or delete the projects files.
Risk of malfunctions.

4.4 Files back up

WARNING!
Copy the whole FlexiVision folder onto an external storage device and archive it.

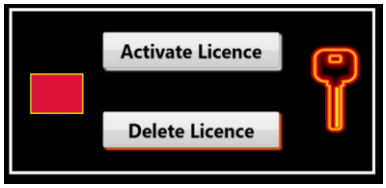
4.5 Login page

When running FLEXIVISION for the first time, the following page opens.



Position	Element/section	Description
1	OPERATION MENU	Includes all the control and operation procedures.
2	LOGIN	Allows the authentication procedure.
3	LICENSE	Is used to activate or delete a license.
4	ACCOUNT DELETE	Allows the account delete procedure.
5	ACCOUNT CREATION	Allows the account creation procedure.

NOTE
When running FLEXIVISION for the first time, the square in the LICENSE section is RED.

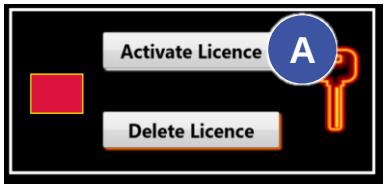


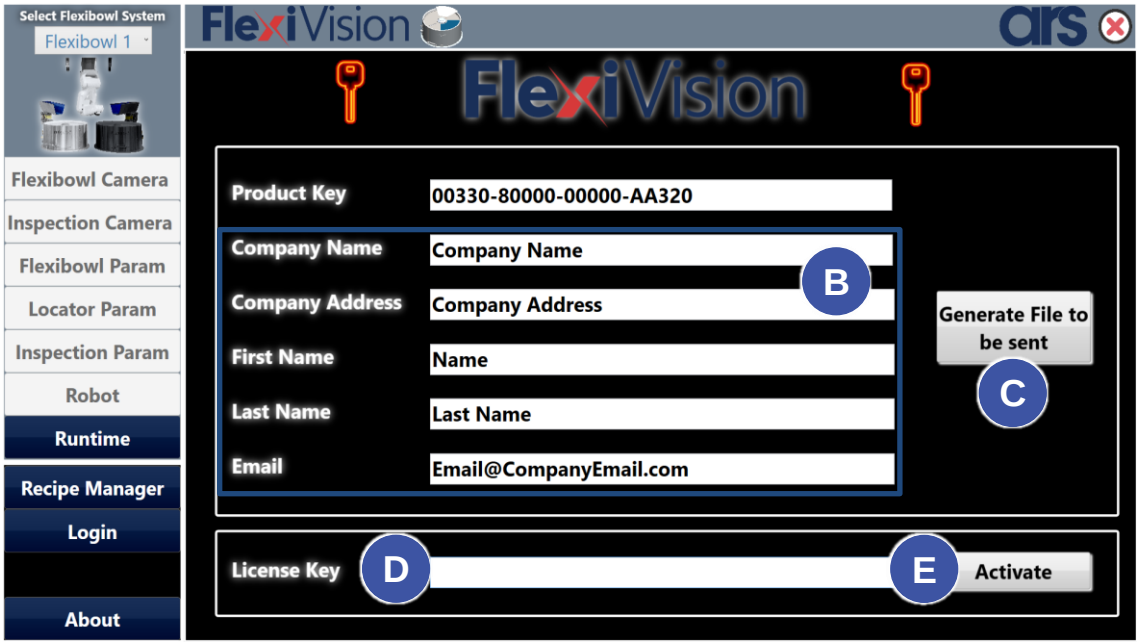
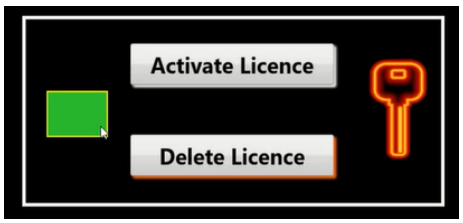
4.6 Authentication procedure

NOTE
Each user has a different access level to procedures.

Step	Action	Notes/Pictures
1	Select the user from the drop-down menu *NAME* (LOGIN section of MAIN PAGE): ▪ Operator ▪ Technician ▪ ARS	Operator ARS Technician
2	Enter the password in the related field.	Default password are: 1 for "operator"; 2 for "Technician".
3	Press *LOGIN*.	If login is successful, available functions are unlocked, according to the access level fo the user.

4.7 How to activate the licence

Step	Action	Notes/Pictures
1	Run FLEXIVISION	
2	Login as operator or technician.	See par. 4.5
3	Press *ACTIVATE LICENSE* (**A**).	

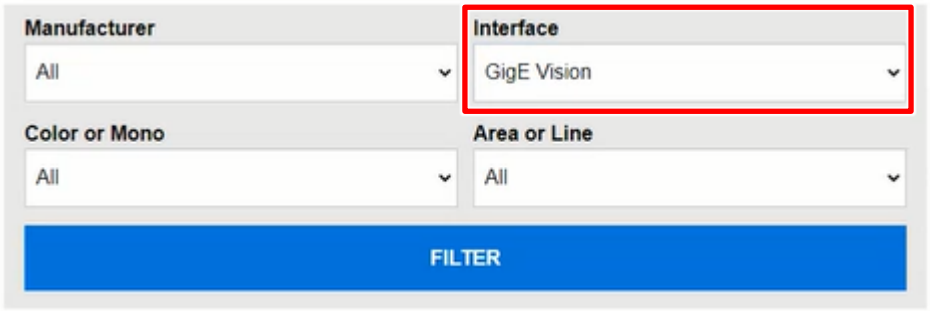
Step	Action	Notes/Pictures
4	The following page opens:  Enter the required data (company name, address, etc.) in the (**B**) fields.	
5	Press *GENERATE FILE TO BE SENT* (**C**).	A window message shall appear for file download.
6	E-mail the generated file (.xlm) to info@flexibowl.com.	ARS srl shall generate and send back a licence key.
7	Enter the licence key in the (**D**) field.	
7	Press *ACTIVATE* (**E**).	If activation is successful, the square shown in the picture below becomes GREEN. 

5 CAMERAS

5.1 Compatibility

The system is compatible with a wide range of industrial cameras.
The complete list is available on:
https://www.cognex.com/products/machine-vision/vision-software/visionpro-software/visionpro-camera-support

Find the compatible models by entering *GigE Vision* in the interface field of the filter mask, as shown in the picture below.



5.2 How to connect the camera

The following is an extract from the Cognex User Manual: for further and complete details, please refer to the whole Cognex User Manual.
For non-Cognex hardware, follow the manufacturer's installation instructions.

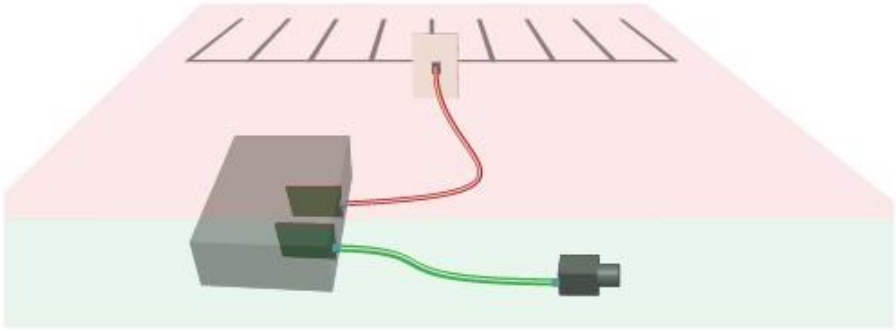
5.2.1 Security requirements

In addition to the standard software and hardware requirements listed in your product documentation, your PC must include both of the following security mechanisms to run Cognex software:

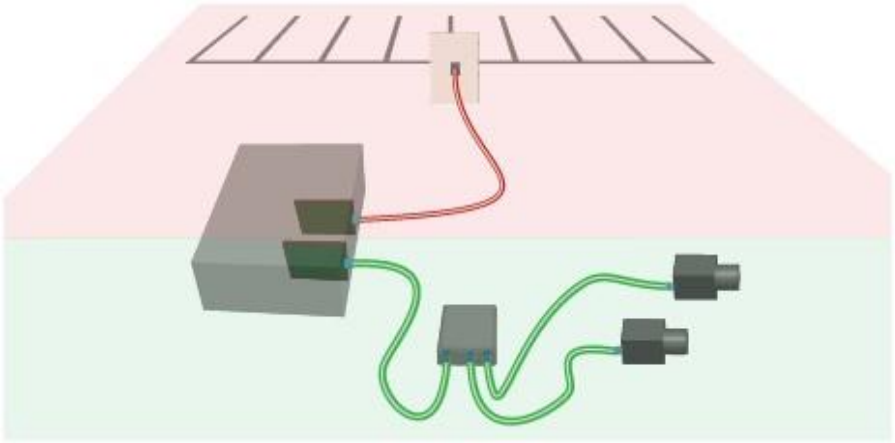
- a Cognex frame grabber
- a Cognex security key (dongle).

5.2.2 GigE Vision Camera Networks

For an application that uses only one camera, the GigE Vision camera network will consist only of a Gigabit Ethernet network adapter and the camera:



If you are using more than one camera, you can use a multi-port network adapter or a Gigabit Ethernet switch (shown):



Be aware the network bandwidth is shared among all connected the cameras when you are using a network switch.

Your PC may already have a network adapter that is used to connect your PC to a local area network or to the Internet. The network adapter(s) you use for image acquisition should be dedicated only for GigE Vision cameras and not connected to your local area network or to the Internet.

WARNING!
To avoid electromagnetic interference, any Ethernet cables you use must be shielded. Cognex strongly recommends Cat 6 or Cat 7 cables with S/STP shielding.

5.2.3 GigE Vision Network Adapters and Switches

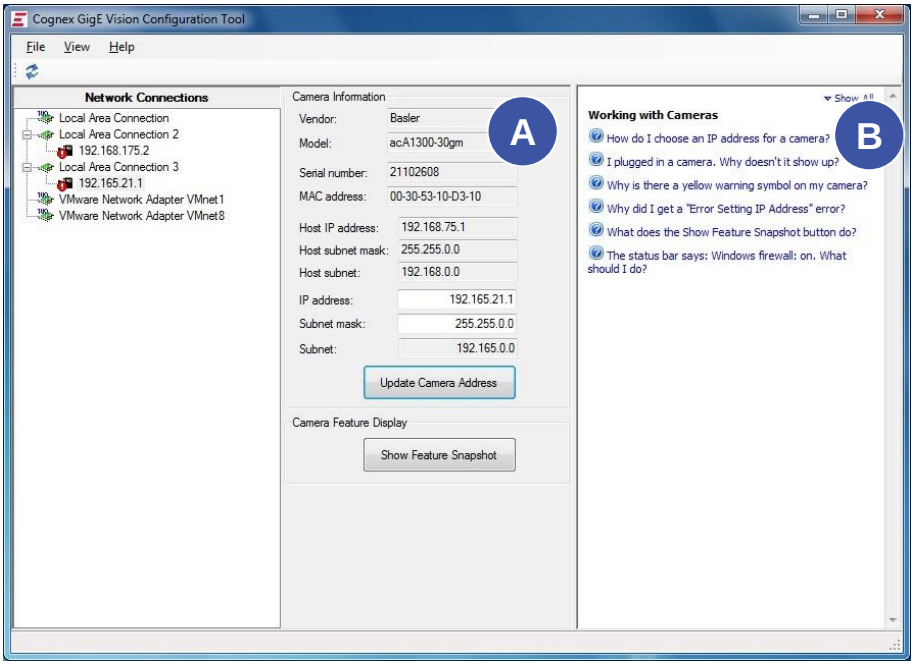
Cognex recommends Gigabit Ethernet network adapters that use the PCI Express bus, and supports a variety of multi- port adapters and Ethernet switches. Select a network adapter that support a minimum of 9000 Kbytes jumbo frame size.

5.2.4 Install Your Cognex Vision Software

After installing the manufacturer's drivers for your GigE Vision network adapters, install your Cognex vision software (VisionPro) including the Cognex Drivers.
VisionPro installation include a utility for configuring your GigE Vision network adapter and camera, as well as the eBus Universal Pro driver that improves the performance of GigE Vision applications for most production environments.

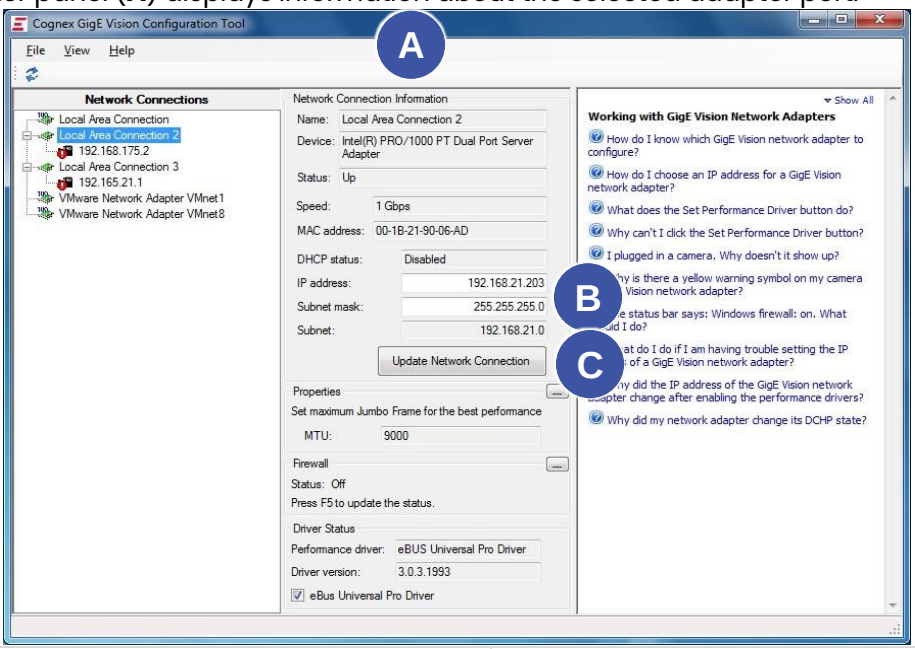
5.2.5 GigE Vision Configuration Tool

Step	Action	Notes/Pictures
1	Launch the Cognex GigE Vision Configuration Tool to assign IP addresses to each GigE Vision network adapter port and the camera connected to it.	NOTE Ensure that your GigE camera is connected to the adapter and powered on, before launching the GigE Vision Configuration Tool. Stop any applications that use CVL or VisionPro.
2	Start the GigE Vision Configuration Tool for CVL through the Start menu by choosing: ***Cognex → CVL → Cognex Utilities → GigE Vision → Cognex GigE Vision Configuration Tool***	
3	Start the GigE Vision Configuration Tool for VisionPro through the Start menu by choosing: ***Cognex → CVL → Cognex Utilities → GigE Vision → Cognex GigE Vision Configuration Tool***	
	The center panel (**A**) displays information about the available network connections and cameras. An interactive list of Questions and Answers appears on the right side of the utility (**B**) to guide you in setting IP addresses and configuring other system properties.	NOTE In most environments at least one of your Local Area Connections is associated with the network connection of the PC. Check with your network administrator if you are not sure which one it is.

Step	Action	Notes/Pictures
		

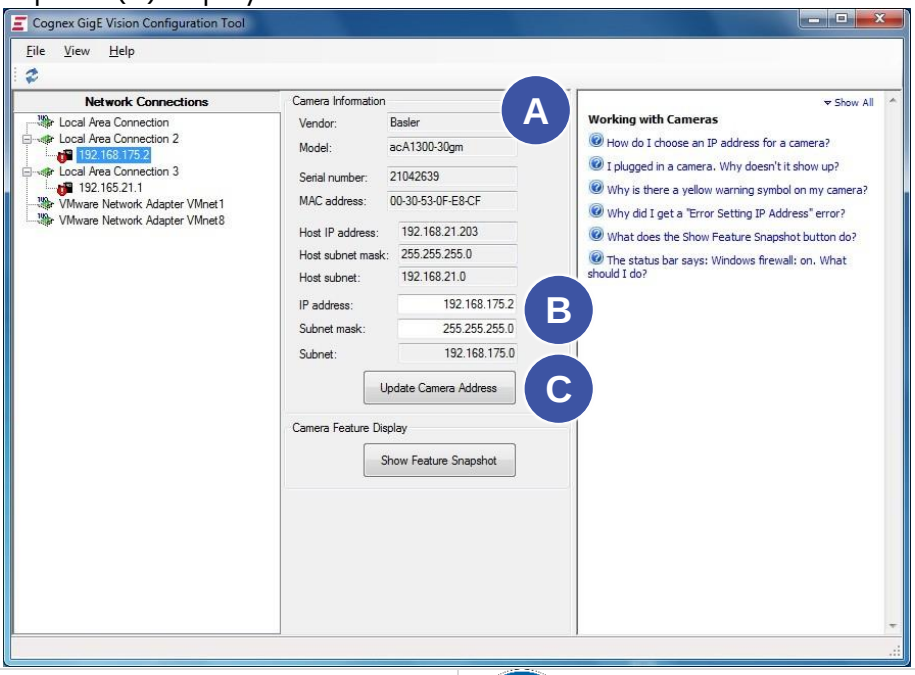
5.2.6 Network Adapter IP Address

Each GigE Vision network adapter (or each port of a multi-port adapter) must have its own IP address on its own subnet. To set the IP address for a GigE Vision network adapter:

<table>
<tr><th>Step</th><th>Action</th><th>Notes/Pictures</th></tr>
<tr><td>1</td><td>Select the Local Area Connection that corresponds to the adapter port connected to your GigE Vision camera(s).</td><td>NOTE
Refer to the embedded Questions and Answers in the utility for guidance in selecting the correct Local Area Connection.</td></tr>
<tr><td></td><td>The center panel (A) displays information about the selected adapter port:
 
</td><td></td></tr>
<tr><td>2</td><td>Enter an IP address and Subnet mask (in the B fields) for the adapter.</td><td></td></tr>
<tr><td>3</td><td>Click Update Network Connection (C). If you are not familiar with TCP/IP networking, Cognex recommends you use the following values:
<table>
<tr><th>Network adapter</th><th>IP Address</th><th>Subnet mask</th></tr>
<tr><td>1st adapter port</td><td>192.168.21.203</td><td>255.255.255.0</td></tr>
<tr><td>2nd adapter port</td><td>192.168.22.203</td><td>255.255.255.0</td></tr>
<tr><td>3rd adapter port</td><td>192.168.23.203</td><td>255.255.255.0</td></tr>
</table></td><td></td></tr>
</table>

5.2.7 Camera IP Addresses

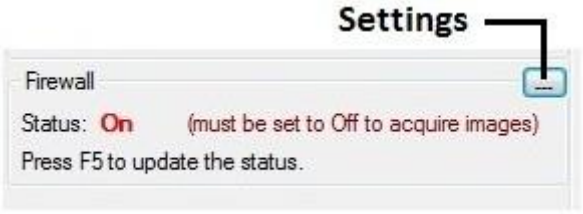
Each camera must have an IP address in the same subnet as its network adapter.
To set the IP address for a GigE Vision camera:

Step	Action	Notes/Pictures
1	Select a camera on the left.	
	The center panel (**A**) displays information about the selected camera. 	
2	Enter an IP address and Subnet mask (in the **B** fields) for the camera.	NOTE The IP address of the camera must be on the same subnet as its network adapter (or adapter port), which appears as the Host IP address. In addition, the subnet mask for the camera must be the same as the Host subnet mask. For example, with a network adapter IP address of 192.168.21.203 and a subnet mask of 255.255.255.0, the camera connected to this network adapter can be numbered from 192.168.21.1 to 192.168.21.254, excluding 192.168.21.203 (the network adapter IP address).

Step	Action	Notes/Pictures
3	Click Update Camera Address (**C**).	NOTE CVL and VisionPro order cameras according to their network addresses. Be aware that you cannot successfully change the IP address of a camera while your vision application is running. You must stop the application and restart it after changing the IP address.

5.2.8 Check Windows Firewall Status

The GigE Vision Configuration Tool indicates the Windows Firewall On/Off status for the selected adapter port.
Refer to the embedded Questions and Answers of the GigE Vision Configuration Tool for details on turning the Windows Firewall On or Off for your particular operating system.



You have several other options for turning Windows Firewall On/Off and preventing it from interfering with GigE acquisition:

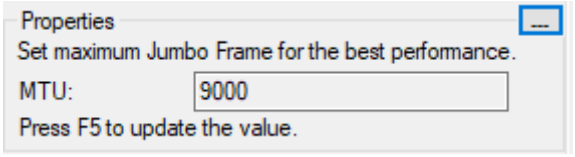
- By network domain type
- By connection
- By application.

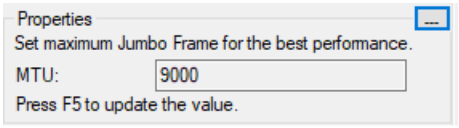
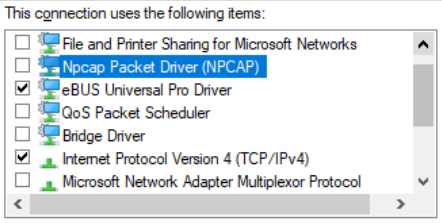
In addition, Windows allows users to customize the firewall response to various network activity. Consult your Windows documentation for details.

Be aware that the utility only detects the local setting for Windows Firewall, and not any settings determined by a group policy that overrides the local setting on this PC. In these situations, the GigE Vision Configuration Tool can report that the Windows Firewall is on for the PC while the group policy safely disables it, allowing you to ignore the warning message.

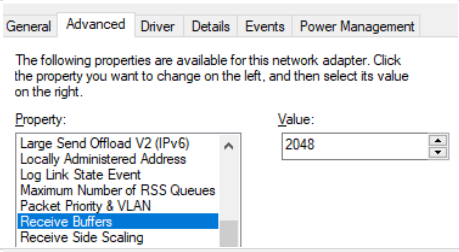
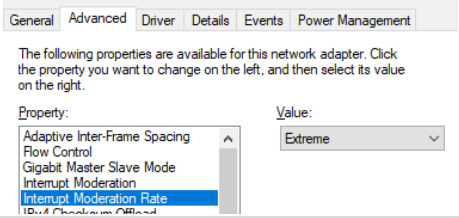
5.2.9 Change System Properties

The GigE Vision Configuration Tool displays a warning when specific system properties, such as the jumbo frame value, are not set to the most desirable levels, as shown in the following figure.



Step	Action	Notes/Pictures
1	Click the settings button to open the Windows Control Panel.	
2	Click *Configure* and then choose the *Advanced tab*.	
3	Modify the following properties as necessary: Select the Jumbo Packet property and choose a value of 9000 or greater in the dialog box.	
4	In the Networking tab, clear all the check boxes listed under *This connection uses the following items* except for *eBUS Universal Pro Driver* and *Internet Protocol Version 4 (TCP/IpV4).*	
5	Ensure any third-party Ethernet driver is either disabled on the network stack, as shown in the previous figure, or uninstalled.	

In addition, Cognex recommends you modify the following properties for this network connection, which may or may not be grouped together with the previous properties:

Step	Action	Notes/Pictures
6	Change the *Receive Buffers* property and choose the highest possible value in its *Value* list.	
7	Change the *Interrupt Moderation Rate* property to *Extreme* in its *Value* list.	

Refer to the embedded Questions and Answers of the GigE Vision Configuration Tool for more details on what system properties you should modify as necessary.

5.2.10 Disable Unused Network Clients

By default, Windows installs and enables network clients that are not required for GigE Vision. By disabling these unused clients, you can improve GigE performance.
To disable unused clients under Windows 7:

Step	Action	Notes/Pictures
1	Open the Control Panel → Network and Sharing Center. Click *Change Adapter Settings*.	
2	Right-click on the icon that represents the Gigabit Ethernet adapter you are using for the GigE Vision network and choose *Properties*.	
3	Ensure that only the following items are checked: ▪ Internet Protocol Version 4 (TCP/IPv4) ▪ Ethernet Bus Filter (eBus Universal)	
4	As stated elsewhere in this guide, ensure any third-party Ethernet drivers are disabled on the network stack or uninstalled	
5	Click *OK*.	

5.2.11 Using GigE Vision Cameras

5.2.12 Video Formats

GigE Vision cameras you obtain through Cognex use one of the following Generic GigEVision video formats:

- Mono, Mono10, Mono10 Packed, Mono12, Mono12 Packed, Mono14, or Mono16
- Bayer Color
- RGB8 Color
- YUV422 Packed.

There are no camera-specific CCF files for GigE Vision cameras.

5.2.13 Supported GigE Vision Features

Cognex vision software supports the following GigE Vision features through the Cognex vision software API.

5.2.14 GigE Vision features supported in Cognex API

- AcquisitionMode
- AcquisitionStart
- AcquisitionStop
- AcquistionFrameRateAbs
- BlackLevel
- ExposureTime
- Gain
- OffsetX
- OffsetY
- PixelFormat
- TriggerMode
- Width
- Height

See your camera documentation for a complete list of supported features.

Always set features with a Cognex API if one exists. If a Cognex API does not exist for a given feature, you can read and write directly using the VisionPro class ICogGigEAccess or the CVL class ccGigEVisionCamera.

5.2.15 Adapters and Cables

Ensure that all components in your GigE Vision network conform to Gigabit Ethernet standards and that you are using Cat 6, Cat 6a, or Cat 7 cables with S/STP shielding.

5.2.16 Troubleshooting

If you are experiencing difficulty setting the IP address of a GigE network adapter, consult the Release Information for your particular Cognex software for the latest news and information.
In most cases, image corruption or failure to create acquisition FIFOs is the result of using incompatible Gigabit Ethernet adapters. The best way to correct or avoid such problems is to use a Cognex-recommended adapter.
During the installation of the eBus Universal Pro Driver on Windows 7, Windows may display a Security Alert dialog. To permanently accept Pleora's Microsoft Authenticode certificate, when the dialog appears, click PleoraTechnologies Inc →Install Certificate → Next. Select Place all certificates in the following store.
In the Certificate store field, type Trusted Publishers. Click Next → Finish → OK. The Security Alert dialog will no longer appear.
Aborting the Cognex Driver installation can leave the network adapter in an invalid state.

Intel ProSet software may not be compatible with the eBus Universal Pro Driver that Cognex uses. Cognex recommends that you do not install Intel ProSet software or other software that behaves similarly.
The GigE Vision eBus Universal Pro Driver does not support power management. To turn off your PC, use *Shut down* rather than *Standby* or *Hibernate*.
Due to an issue with the Pacific Instruments USB driver installer, the eBus Universal Pro Driver may be removed during the installation. To fix the issue, re-install the Cognex Drivers after installing the USB drivers.
Click the refresh button or select View → Refresh to update the GigE Vision Configuration Tool with the latest IP addresses, which might not reflect the current settings after you set the IP address of a GigE network adapter or GigE Vision camera.
In some cases, a VisionPro application that uses a dual-tap GigE Vision camera can exhibit a vertical line down the center of the image while the two halves appear unbalanced. Refer to the topic Configuring a Dual-Tap GigE Vision Camera in the VisionPro online documentation for details on how to balance the taps.
Be aware that while jumbo frame support is enabled by default on most NETGEAR switches, it is disabled by default on the NETGEAR GS110TP Power over Ethernet switch. Refer to your GS110TP documentation for instructions on configuring jumbo frame support. Cognex recommends the maximum size available.

5.2.17 Precautions

To reduce the risk of injury or equipment damage, observe the following precautions when you install the Cognex product:

- Route cables and wires away from high-current wiring or high-voltage power sources to reduce the risk of damage or malfunction from the following causes: over-voltage, line noise, electrostatic discharge (ESD), power surges, or other irregularities in the power supply.
- Changes or modifications not expressly approved by the party responsible for regulatory compliance could void the user's authority to operate the equipment.
- Ensure that the cable bend radius begins at least six inches from the connector. Cable shielding can be degraded or cables can be damaged or wear out faster if a service loop or bend radius is tighter than 10X the cable diameter.
- This device is certified for office use only and if used at home, there can be frequency interference problems.
- This device should be used in accordance with the instructions in this manual.
- All specifications are for reference purposes only and can change without notice.

5.3 Camera working conditions

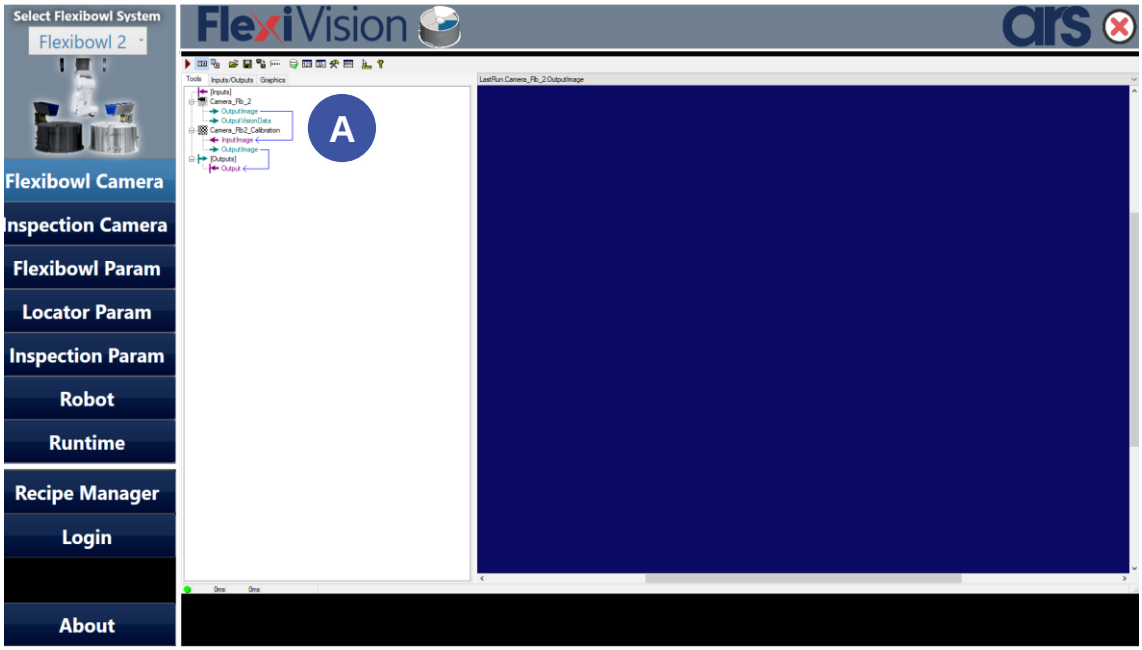
FlexiVision® is designed to work with images taken orthogonally on a work plane (the FlexiBowl plane).

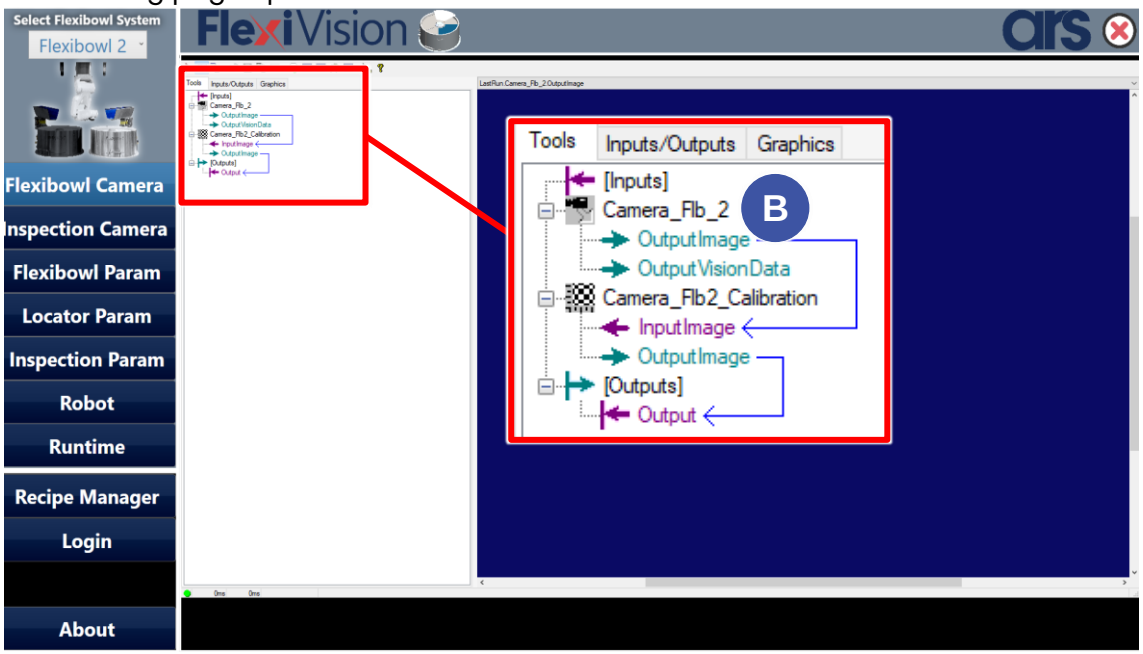
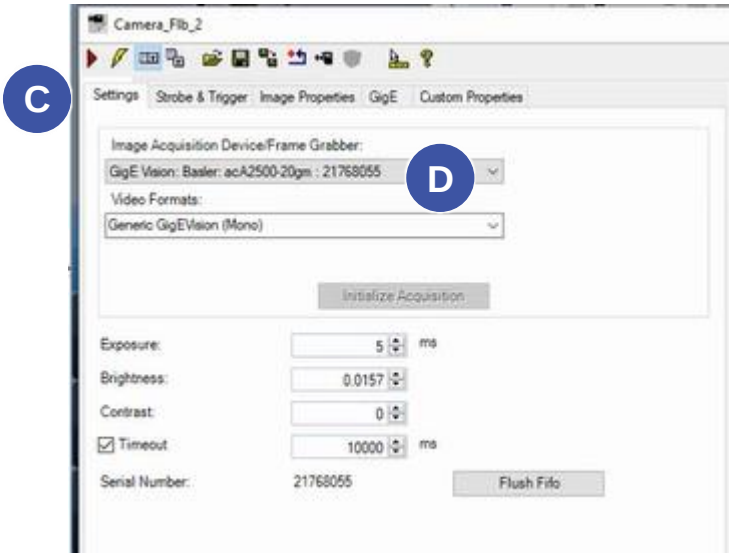
Place your camera on the top view of the plane at the right distance (e.g. 1 meter) to optimize the field of view (depending by choosed lens).
These conditions are suitable for 2D object matching performed by the locator procedure.

The camera calibration procedure (describe in the following pages) reduces the perspective effect due to the misalignment between camera sensor and work plane.

5.4 How to set the camera parameters

NOTE
This procedure can be carried out by the following users:
- **TECHNICIAN**
- **ARS**

Step	Action	Notes/Pictures
1	Run FLEXIVISION	
2	Login in.	
3	The following page opens:  Click on *Flexibowl camera* (**A**) of the operations menu.	

Step	Action	Notes/Pictures
4	The following page opens:  Doubleclick on the relevant camera field (**B**) to enter the **CogAcqFifoTool edit control.**	
5	The following page opens:  Press SETTINGS (**C**) and select the desired camera from the drop-down menu (**D**), which lists all the available image acquisition devices, connected to the system.	

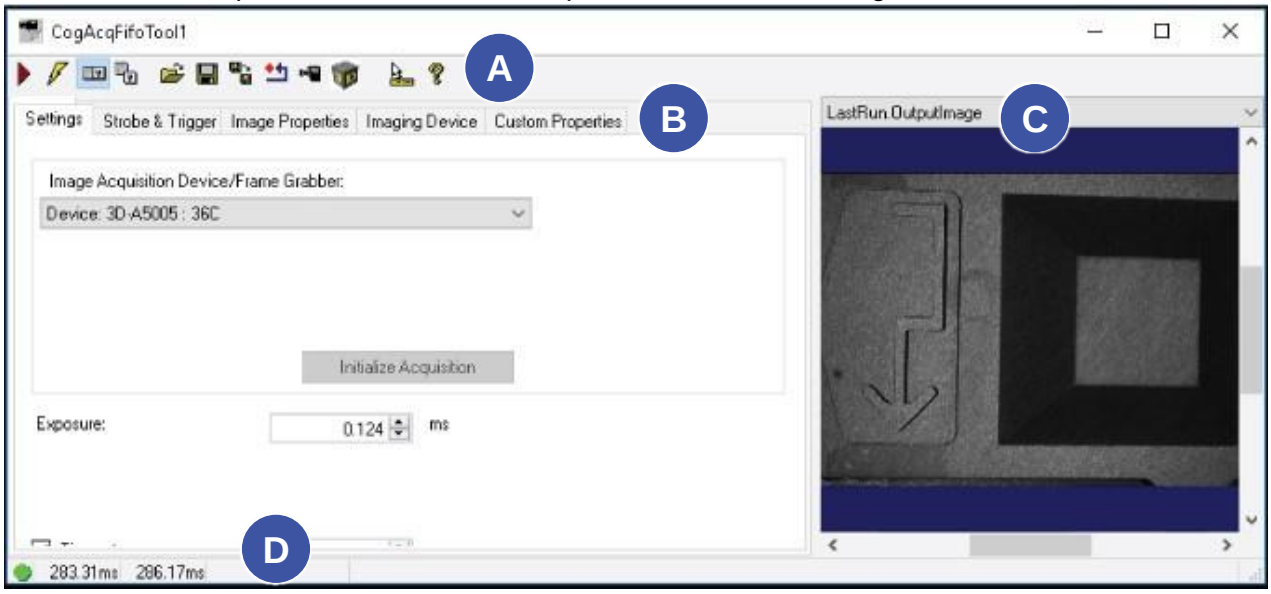
The same procedure shall be carried out for both the FLEXIBOWL CAMERA and (if present) the INSPECTION CAMERA.

5.4.1 AcqFifoTool Edit Control (camera configuration)

The following is an extract from the Cognex User Manual: for further and complete details, please refer to the whole Cognex User Manual.

The CogAcqFifoTool edit control provides a graphical user interface to the CogAcqFifoTool vision tool, which acquires images using an acquisition FIFO on a frame grabber, a GigE camera or a Cognex 3D sensor. The edit control allows you to configure various image-acquisition parameters and acquire an image.
The edit control includes the following components:

- a row of control buttons along the top (**A**) performs common operations.
- a set of function tabs (**B**) allow you to specify the trigger type, indicate the method of lighting, select a region of interest, and set other configuration parameters. The exact number, contents, and appearance of the tabs vary slightly depending on the image source you use.
- An image display window (**C**) shows the most recently acquired image currently stored in the OutputImage buffer. Right click on the image window to choose from a menu of options that include zooming in or out or enabling a pixel or subpixel grid.
- A status bar along the bottom (**D**) reports whether the tool executed successfully, the time the tool required to execute, and any error codes or messages.



See the topic VisionPro and Point Clouds for more information.

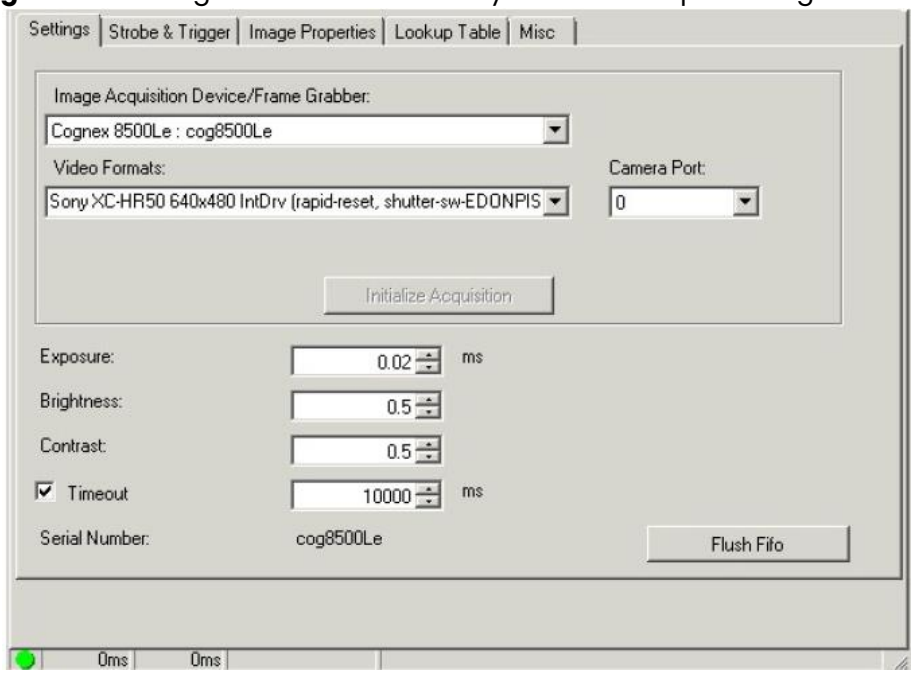
5.4.1.1 Control buttons

Button	Description	Function
	Run	Acquire a new image and make it available to other vision tools.
	Electric mode	Toggle electric mode, where the AcqFIFO tool acquires an image automatically when particular configuration parameters change. In electric mode, a lightning bolt appears next to every electric property. Electric mode only works for manual triggers, and the edit control ignores electric mode when the trigger type is set to anything other than manual.
	Local image display	Open or close the local image display window. An AcqFIFO tool supports an OutputImage buffer, which contains the last acquired image.
	Floating image display	Open one or more floating image windows, which also support the OutputImage buffer.
	Open	Open a VisionPro persistence (.vpp) file containing a set of saved properties for an Acquisition FIFO tool. VisionPro reports an error if you try to open a .vpp file for another object type or for an entire vision application.
	Save	Save the current properties of the vision tool to a VisionPro persistence (.vpp) file. The edit control allows you to choose between saving the vision tool with or without its image buffers and tool results.
	Save As	Save the current properties of the vision tool to a new VisionPro persistence (.vpp) file.
	Reset	Reset the vision tool to its default state.
	3D Display window	Open a 3D Display to view 3D range images or 3D point clouds.
	Live Video	Open a live video image window that displays a live image from the camera. Live video allows you to adjust the production environment characteristics that can affect each acquired image, such as camera position and focus, object placement, or lighting. Use Run to acquire a single image and store it in the OutputImage buffer.

Button	Description	Function
	Show ToolTips	Enable or disable the display of tooltips for individual items in the edit control.
	Help	Open this VisionPro online help file.

5.4.1.2 Settings tabs

Use the **Settings tab** to configure the video source you use to acquire images.

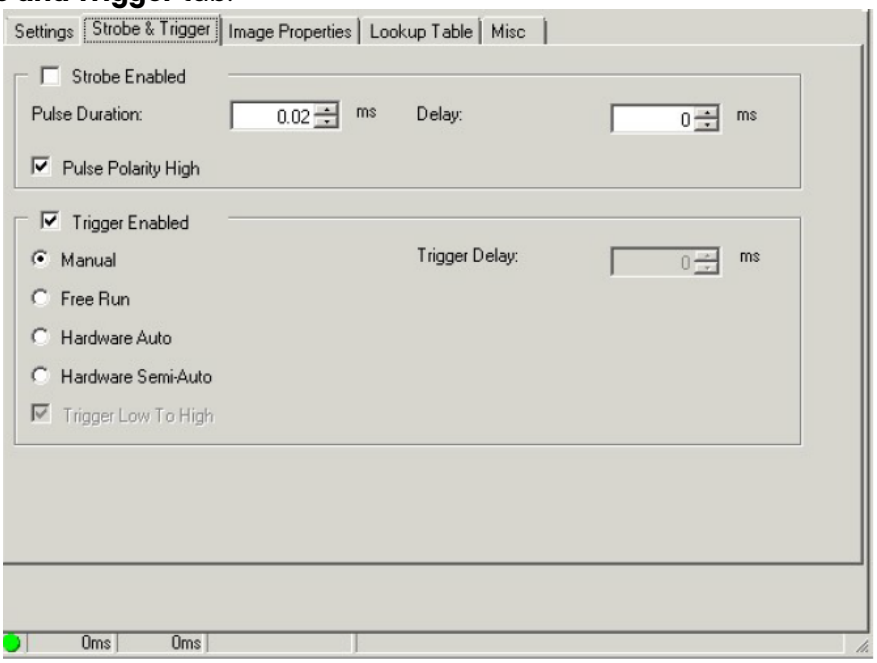


Feature	Description
Image Acquisition Device/Frame Grabber	The edit control displays the name of the frame grabber associated with this acquisition FIFO.
Video Formats	Select the video format for this acquisition FIFO. If you switch the video format later, the edit control creates a new acquisition FIFO and attaches it to the current tool.
Camera Port	Select the camera port on the frame grabber where you connected the camera.
Exposure	Choose an exposure time. You might need to experiment with objects moving past your camera at production speed in order to determine the best value. Use the value 0 to have the camera use the shortest exposure time that it supports.
Brightness	Set the brightness level for each image acquisition. You might need to experiment with different values in order to determine the best level for your vision application.

Feature	Description
Contrast	Set the contrast level for each image acquisition. You might need to experiment with different values in order to determine the best level for your vision application.
Timeout	A timeout period determines how long the acquisition FIFO waits for an image to become available before the application generates a timeout error (**CogErrConstants**). Enter a **Timeout** to specify how much time the application will wait.
Serial Number	The edit control displays the serial number of this frame grabber.
Flush Fifo	Clear all outstanding acquisition requests from the queue.

5.4.1.3 Strobe and Trigger Tab

Use the **Strobe and Trigger** tab to control an optional strobe light as well as to configure the type of trigger this acquisition channel uses to signal that an image acquisition should begin. The fields in this tab vary depending on the specific frame grabber you use. The following figure shows an example **Strobe and Trigger** tab:



Configure the following settings on the **Strobe and Trigger** tab:

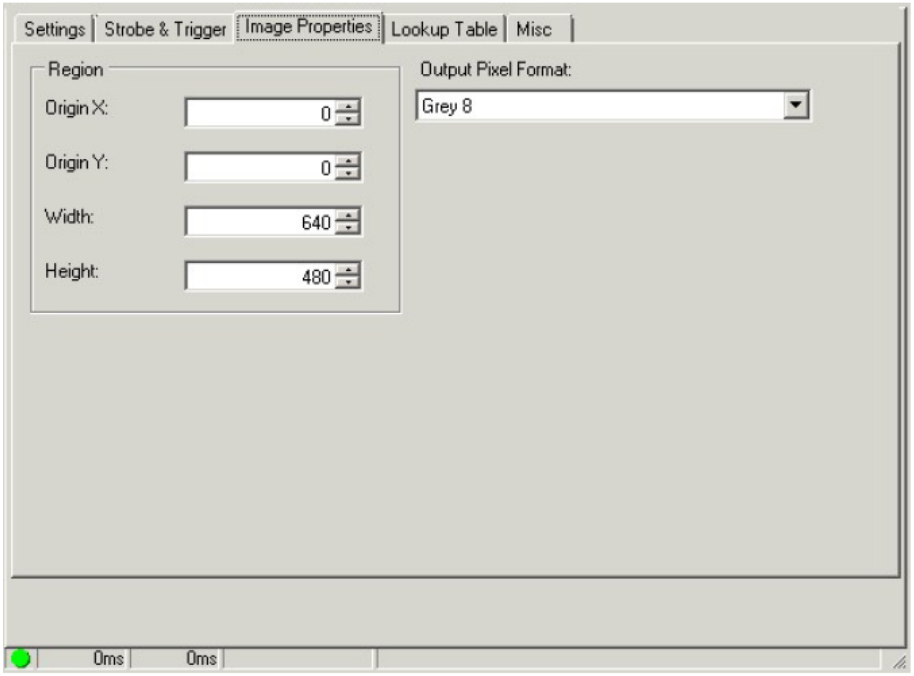
Property	Description
Strobe Enabled	Enables the strobe light for each image acquisition. With a strobe light enabled, configure the following settings: ▪ The **Pulse Duration** field sets the duration of the pulse, in milliseconds. ▪ The **Delay field** sets the delay time between the shutter pulse and the firing of the strobe. ▪ The **Pulse Polarity High** checkbox sets the strobe pulse polarity to high. Refer to your strobe hardware documentation for the correct setting.
Trigger Model	Choose one of the following incoming trigger types for this Job: ▪ Manual triggers acquire an image when you press Run. ▪ Free Run triggering allows the acquisition system to acquire images at the highest frame rate that the acquisition module can support. ▪ Hardware Auto triggers start an image acquisition when the application detects a transition on an external trigger line. Check the **Trigger Low To High** checkbox if the trigger signal is a transition from low to high. ▪ Hardware Semi-Auto triggers acquire an image when you press Run and the application detects a transition on an external trigger line. Check the **Trigger Low To High** checkbox if the trigger signal is a transition from low to high.
Min Trigger Width	Sets or gets the minimum trigger width in milliseconds. The trigger input signal must be asserted for at least this amount of time before it is recognized as a valid input trigger. QuickBuild ignores any trigger signal that does not meet this width constraint.
Min Trigger Period	Sets the minimum time between triggers in milliseconds. Only the first valid trigger within a period will initiate a camera integration cycle. Other valid triggers in that same period are missed. You can use this value to help limit the camera acquisition rate. Legal values range from 0 to 65.5. Zero specifies that there is no period requirement.
Trigger Delay	Sets the period of time, in milliseconds, between the receipt of the acquisition trigger and the start of camera integration.

5.4.1.4 Image Properties Tab

This section contains the following subsections:

- **Region**
- **Output Pixel Format**

Use the Image Properties tab to define a region of interest and output pixel format:



- **Region**

Use the fields in the tab to specify the origin, width, and height of the region of interest.

- **Output Pixel Format**

Use the **Output Pixel Format** list to choose one of the following pixel formats for the images the **Image Source** makes available to the vision tools you add to QuickBuild:

Property	Description
Grey 8	Grey scale images that offer 256 possible shades of grey from black to white.
Grey 16	Grey scale images that offer 16-bit encoding. See the topic **Working with 16-Bit Images** for more information. A 16-bit greyscale image supports 65,536 grey values, but you must be using a 16-bit capable camera in order to produce images that exhibit this larger range. Choosing **Grey 16** when you are using an 8-bit greyscale or 24-bit RGB camera produces images that are stored using the **Grey 16** class but support only 256 possible grey values.

Property	Description
PlanarRGB24	An image that uses 3 coincident arrays of 8-bit pixel values to represent shades of red, green and blue. Use this option with a supported color camera to generate color images for your vision application. If you select this option with a grey scale camera, the output images use the **PlanarRGB24** class with the values for red, green, and blue in each array set to identical values to generate a corresponding grey pixel value.
Automatic	Allows QuickBuild to generate images with the appropriate output pixel format based on the type of camera you are using and the video format you chose on the **Settings** tab.

5.4.1.5 Lookup Table Tab

An individual pixel in a grey-scale image can have a grey value ranging from 0 through 255. As QuickBuild captures an image, it can remap the grey value of any pixel to a different grey value through the use of a lookup table.

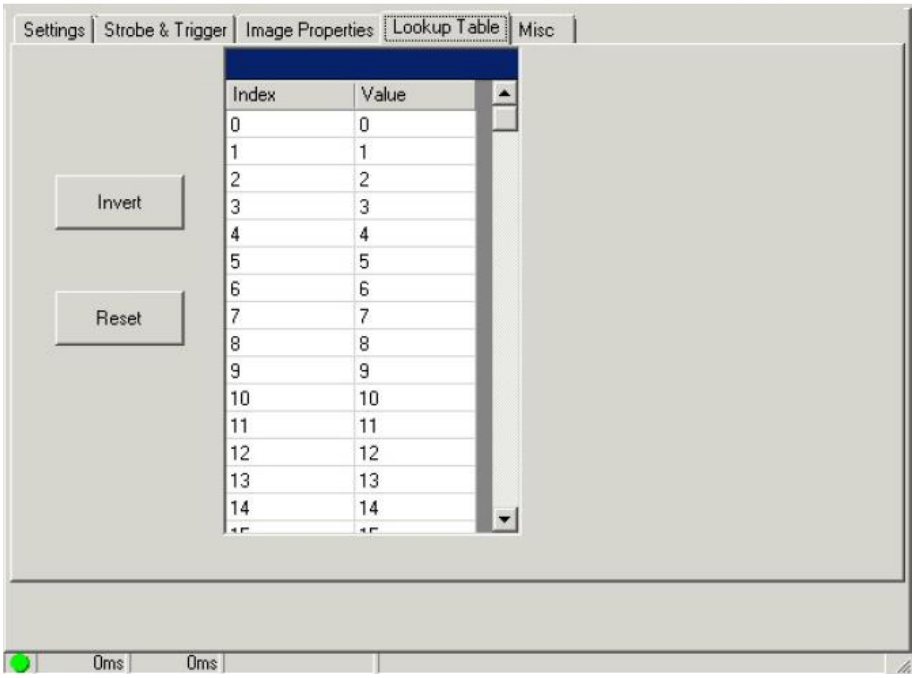
A lookup table is a 256-element array corresponding to the pixel values 0 through 255, where array element [0] corresponds to grey value 0, element [1] corresponds to grey value 1, and so on until element [255] which corresponds to grey value 255.

An acquisition channel using a lookup table evaluates each pixel in the image buffer and changes the grey value based upon the value for the corresponding element in the array. For example, if table element [50] has the value 75, any pixel with a grey value of 50 is given the new grey value of 75 before the image is made available for analysis by any other vision tool.

QuickBuild actually uses a lookup table regardless of whether you have set explicit values for the elements in the array. By default, however, it uses an identity lookup table, which does not change the grey values in the image. In an identity lookup table, element [0] is set to 0, element [1] is set to 1, and so on.

If your acquisition device supports a Lookup Table tab, you can redefine the values in the lookup table. For example, you might generate a new lookup table and choose a specific grey value as the midpoint between light and dark features in an image, and then map all darker pixels to some low value and all lighter values to some high value. This essentially binarizes each acquired image so that all features appear as either black or white.

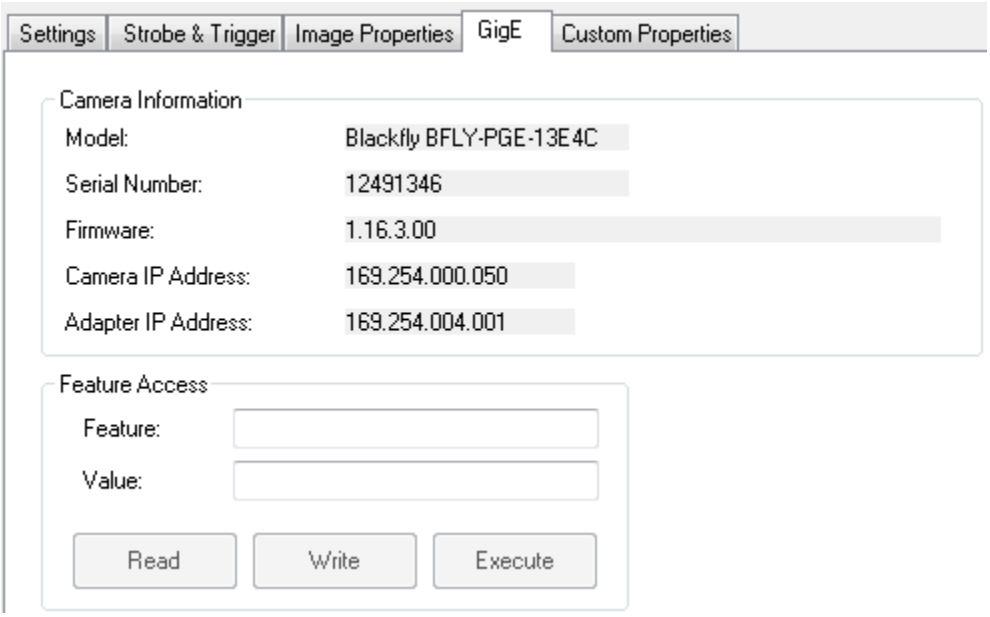
The following figure shows the default Lookup Table tab:



- Click inside the **Value** cell to change the value of any incoming grey value.
- Click **Invert** to swap dark values for light value and light values for dark values.
- Click **Reset** to set all values to their identity defaults.

5.4.1.6 GigE Tab

Use the GigE tab to view basic information about the connected GigE Vision camera and modify various GigE Vision properties. Refer to the GigE Vision Cameras User's Guide for more information. The following figure shows an example GigE tab:



The **Camera Information** section of the tab presents basic information regarding the camera and adapter you are using. Use the **Feature** Access area to examine and modify the XML-defined properties for the GigE Vision camera you are using.

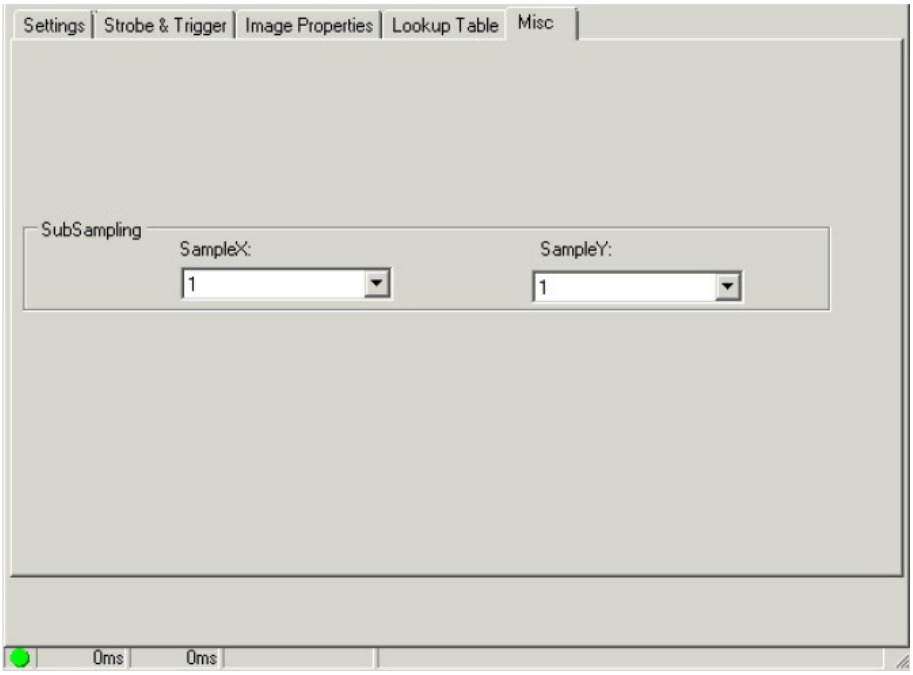
- Enter the XML node name in the **Feature** field.
- Clicking **Read** will read the value of the node and update the **Value** field.
- Clicking **Write** will attempt to write whatever value is in the **Value** field to the node.
- Clicking **Execute** will attempt to execute the command feature specified in the Feature field.

QuickBuild will display a dialog box for any errors that occur, such an invalid entry for the **Feature** or **Value** fields.

Refer to the documentation for the GigE Vision camera you are using for a list of supported XML nodes.

5.4.1.7 Misc Tab

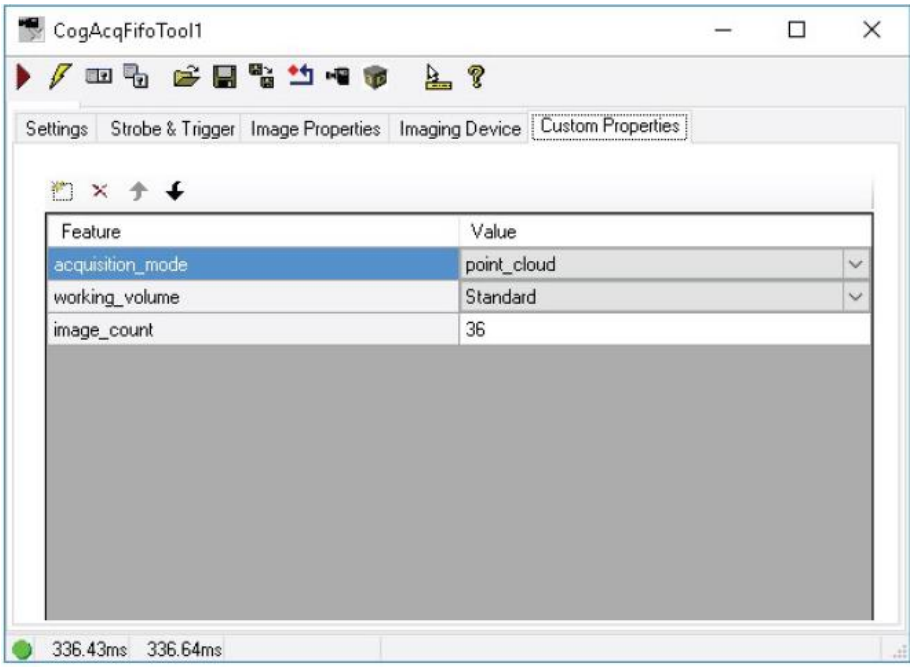
Use the **Misc** tab to select a subsampling rate and reduce the image size, which can result in faster acquisitions.



Specify a subsampling ratio for **Sample X and Sample Y**. For example, setting **SampleX** to 8 specifies an 8:1 reduction in the number of pixels along the x-axis.

5.4.1.8 Custom PropertiesTab

Use the Custom Properties tab to set any custom properties your image acquisition device supports:



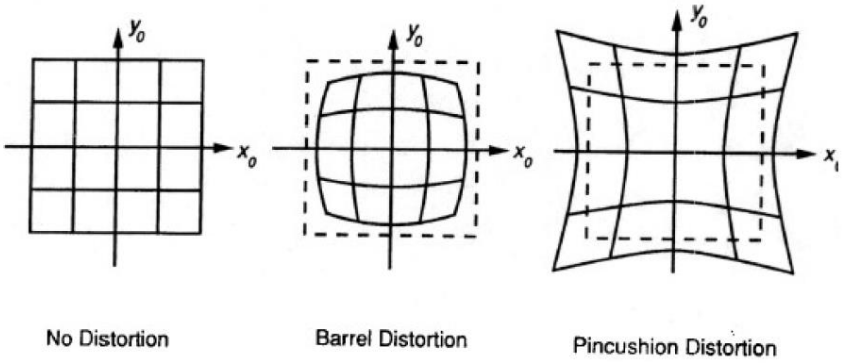
If you are capturing point clouds, see the section "VisionPro and Point Clouds" for details on the custom properties a Cognex 3D sensor supports.

5.5 How to calibrate the camera

FlexiVision requires a camera calibration before starting to work with images.

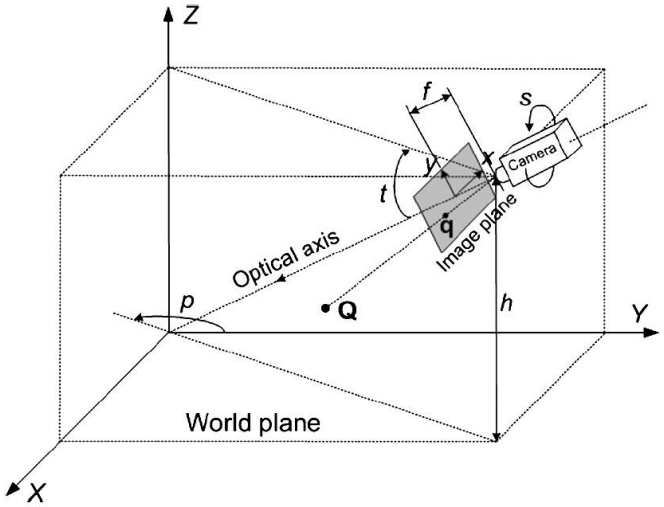
5.5.1 Why calibration is necessary

Every pair of camera sensor and lens applies its own particular distortions on taken image.
So, every pair of camera and lens needs a specific calibration.



Distortions are produced also by additional filters and the focus/aperture setup.

You have to find the optimal camera position and lens setup before starting the calibration procedure.
The calibration must be repeated every time a camera physical feature is modified.

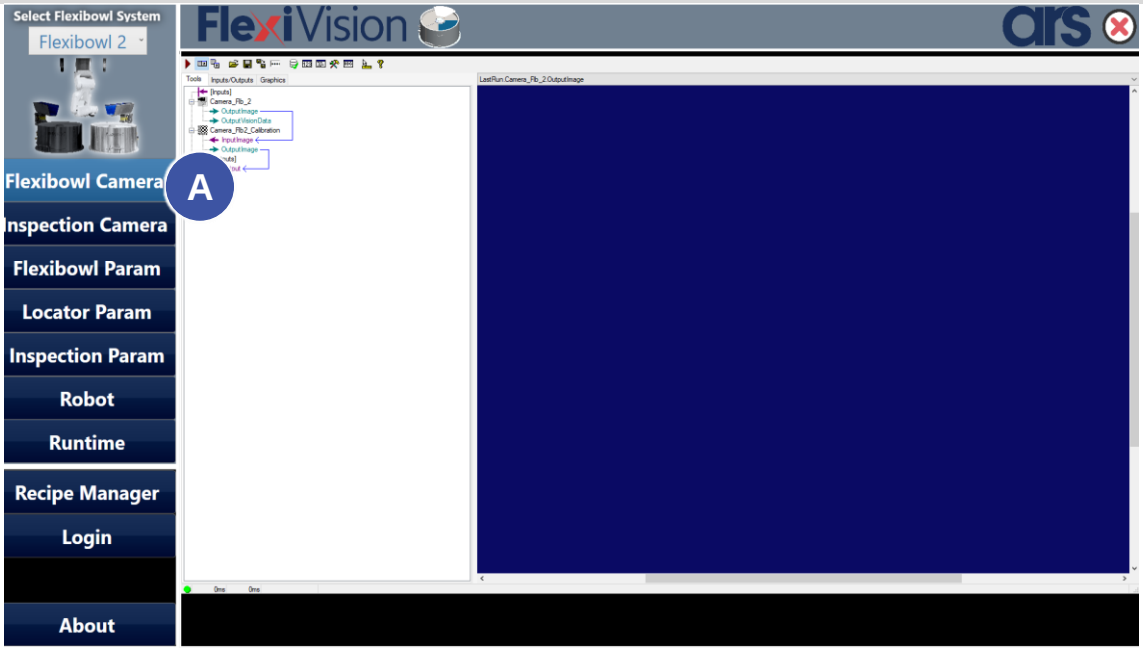


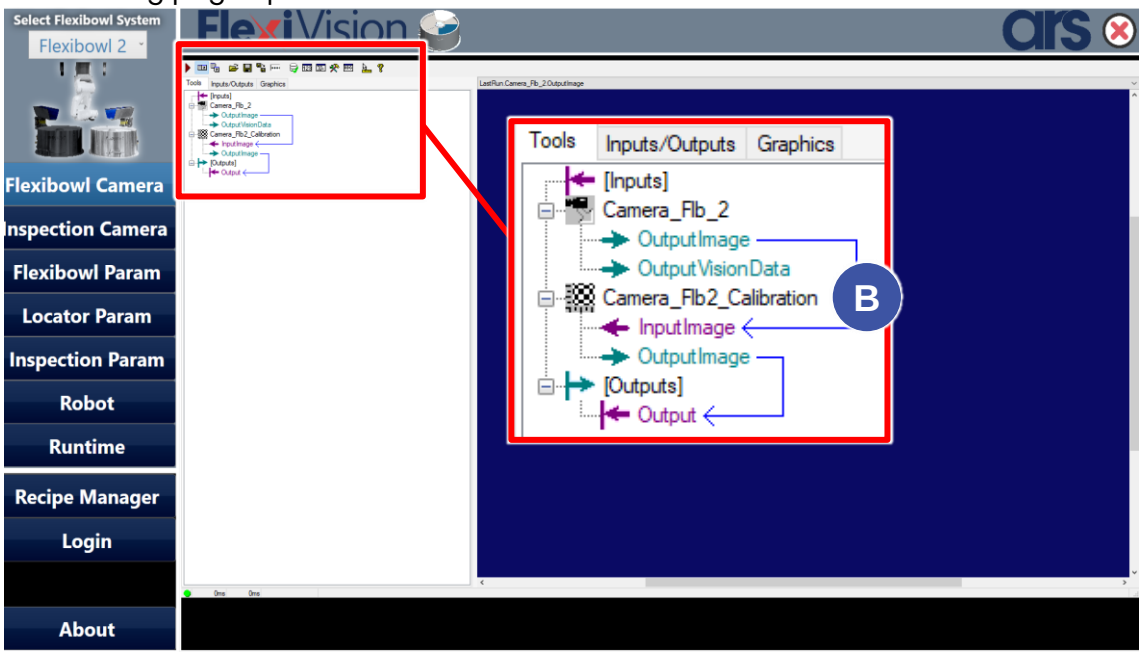
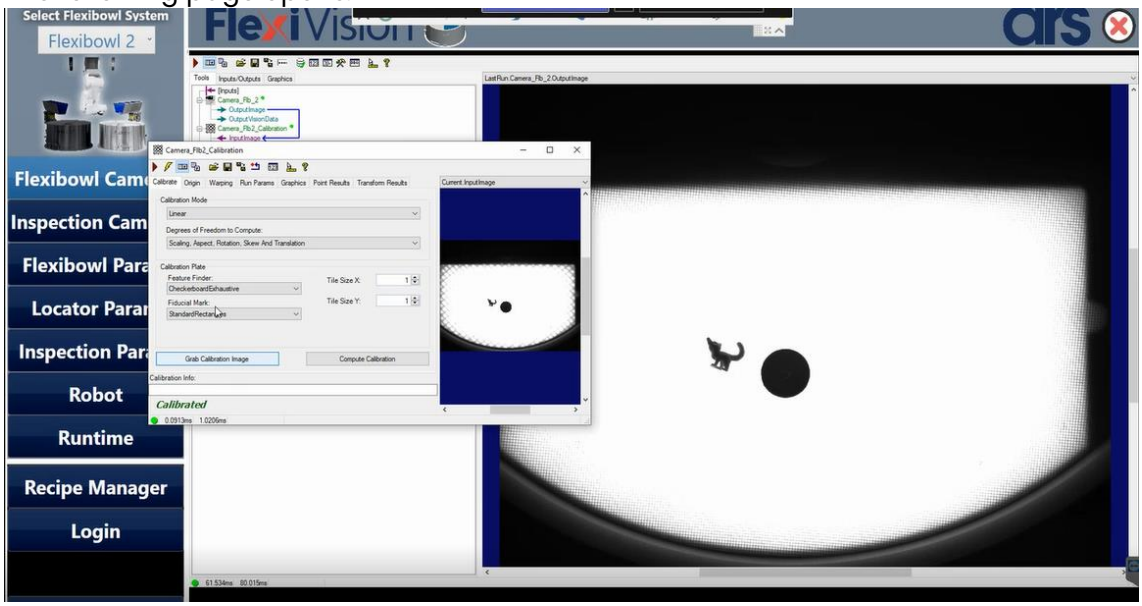
With camera calibration it is possible to get the internal intrinsic (sensor resolution, focal length, lens distortions, ...) and external (position and orientation) camera parameters.
Through the calibration, every frame got by the camera can be "undistorted", that means lens distortions and perspective effect can be corrected in relation to the specific work context conditions. This operation is needed in order to get accurate results during any further image processing and pattern matching tasks: if the image is well undistorted, the metric dimensions of each pixel become constant and well known, in other words measurable.

5.5.2 Flexibowl camera page

Once the camera parameters have been set, proceed as described in the following.

NOTE
This procedure can be carried out by the following users:
- **TECHNICIAN**
- **ARS**

Step	Action	Notes/Pictures
1	 Click on *Flexibowl camera* (**A**) of the operations menu.	

Step	Action	Notes/Pictures
2	The following page opens:  Doubleclick on the relevant camera field (**B**)	
3	The following page opens:  It is now possible to start the calibration procedure by the CogCalibCheckerboardTool**.**	

The same procedure shall be carried out for both the FLEXIBOWL CAMERA and (if present) the INSPECTION CAMERA.

For the checkerboard assembly on the FLEXIBOWL plate, please refer to the instruction

5.5.3 CogCalibCheckerboardTool

FLEXIVISION checkerboard is supplied by ARS s.r.l. as an option.

The following is an extract from the Cognex User Manual: for further and complete details, please refer to the whole Cognex User Manual.

The Checkerboard Calibration tool edit control provides a graphical user interface to the **CogCalibCheckerboardTool** tool, which you use to calibrate an application so that it returns results in meaningful, real-world values.

By adding a Checkerboard Calibration tool to your application, the tools you use to analyze an image can return results in a specific unit of measurement such as inches, centimeters, millimeters, and so on.

The Checkerboard Calibration tool locates the vertices in an image of a checkerboard-style calibration plate (or the dots in a grid-of-dots-style plate) and determines the best-fit 2D transformation between the found image locations and the physical dimensions of the plate. The tool can generate either a linear transformation, or a nonlinear one that also accounts for optical and perspective distortion. The Checkerboard Calibration tool also supports nonlinear transformation using a linescan camera.

The Checkerboard Calibration tool supports both checkerboard and grid-of-dots calibration plates.
Cognex recommends the use of checkerboard calibration plates with the CogCalibCheckerboardTool.
Support for grid-of-dots plates is provided for compatibility purposes.

As part of configuring a Checkerboard Calibration tool, you define the grid spacing (either the tile height and width or the dot spacing) in the unit of measurement you want to use.

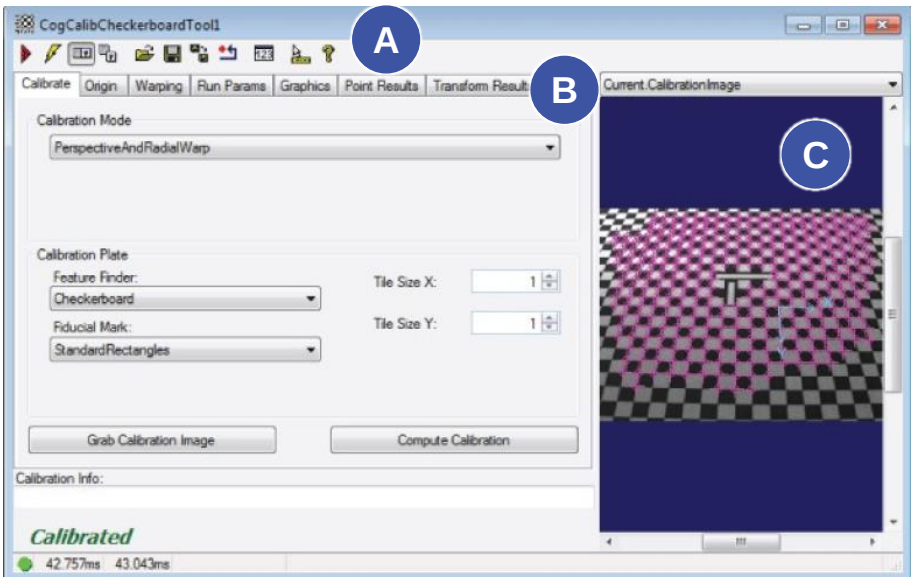
Once calculated, the 2D transformation can be used on subsequent image acquisitions to map the uncalibrated coordinate space of the input image to a raw calibrated coordinate space. By further specifying a precise location and orientation for the origin of this space, you allow the tool to generate a final calibrated space which can then be passed onto other vision tools. Vision tools that use this calibrated output image return point locations in terms of the unit of measurement you define. If the 2D transformation is nonlinear, the tool also warps the pixels of the input image to correct for the distortion.

As you develop your vision application, you typically capture an image of a calibration plate, acquired from an Acquisition FIFO tool, and pass it to a Checkerboard Calibration tool. Once the calibration transformation has been calculated and you are satisfied with the final calibrated coordinate space, you use the same Acquisition FIFO tool to capture images of objects you want to inspect. As the application operates, the Checkerboard Calibration tool takes the stored 2D transformation and attaches it to the coordinate space tree of the input image.

Images of the calibration plate you use for calibration must be greyscale images. However, run-time images that use the calibration tool can be either color or greyscale.

You will need to recalibrate your application if you switch to a different type of camera or if you alter the distance between the camera and the object under inspection. In either case, the number of grid points (checkerboard tiles or dots) required to span a specific distance in uncalibrated space changes, and you must let the Checkerboard Calibration tool determine the new mapping.

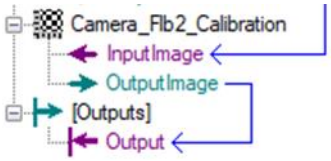
The following figure shows the Checkerboard Calibration tool edit control:



The edit control offers the following features:

- A row of control buttons (**A**) at the top left provide access to the most common operations.
- A set of function tabs (**B**) allow you to determine the type of mapping (linear or non-linear), define the ratio between the grid spacing and the unit of measurement you want to use, and name the new calibrated coordinate space.
- An image display window (**C**) displays acquired images and the output image the Checkerboard.
- Calibration tool generates.

You can experiment with the edit control by using QuickBuild to create a CogCalibCheckerboard tool. A new CogCalibCheckerboard tool appears with the input terminals for the image of the checkerboard pattern and the output image using the calibrated coordinate space, as shown in the following figure:



In a QuickBuild application, the Checkerboard Calibration tool accepts the output image from an image-acquisition tool and then passes its own output image to other vision tools that will use the calibrated coordinate space.

This topic contains the following sections:

- Control Buttons
- Calibration Info
- Calibrate Tab
- Origin Tab
- Warping Tab
- Run Params Tab
- Graphics Tab
- Point Results Tab
- Transform Results Tab

5.5.3.1 Control buttons

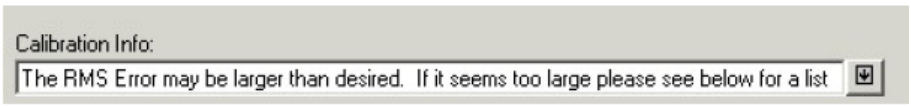
Button	Description	Function
	Run	Generate an output image by mapping the latest input image to the calibrated coordinate space. You do not need to run the Checkerboard Calibration tool to initially generate the calibrated coordinate space, but the tool must run as the application executes in order to generate the output image which can be used by other vision tools. The tool cannot run until it has been calibrated.
	Electric mode	Toggle electric mode, where the Checkerboard Calibration tool executes automatically when particular configuration parameters change. In electric mode, a lightning bolt appears next to every electric property.

Button	Description	Function
	Local image display	Open or close the local image display window. A Checkerboard Calibration tool supports the following image buffers: ▪ The Current.InputImage buffer contains the last image received from an image-acquisition tool. To configure a Checkerboard Calibration tool and generate a calibrated coordinate space, the input image must be of a checkerboard calibration plate before you to copy it to the Current.CalibrationImage buffer. As the application executes, the tool accepts each image of the objects you want to inspect and adds the calibrated coordinate space to the space tree of the input image. ▪ The Current.CalibrationImage buffer contains the image of the checkerboard pattern copied from the Current.InputImage buffer as you configure the tool, with graphics to indicate known vertices. ▪ The LastRun.OutputImage buffer contains the output image generated each time the tool runs. If the 2D transformation is linear, then the output image resembles the input image. If the 2D transformation is nonlinear, then the tool warps the output image to remove the effects of any perspective and radial distortion present in the calibration image. ▪ The LastRun.InputImage buffer contains the last image processed by the tool along with any graphics you enable from the Graphics tab. In addition, if you use the tool to generate a nonlinear 2D transformation, the tool generates a Current.UndistortedCalibrationImage buffer, which contains a warped image of the calibration plate to remove any known distortion. You can also use this undistorted image to specify a **destination rectangle**.
	Floating image display	Open one or more floating image windows, which support the same image buffers as the local image display window.
	Open	Open a VisionPro persistence (.vpp) file that contains a set of saved properties for this vision tool object type. VisionPro reports an error if you try to open a .vpp file for another object type.

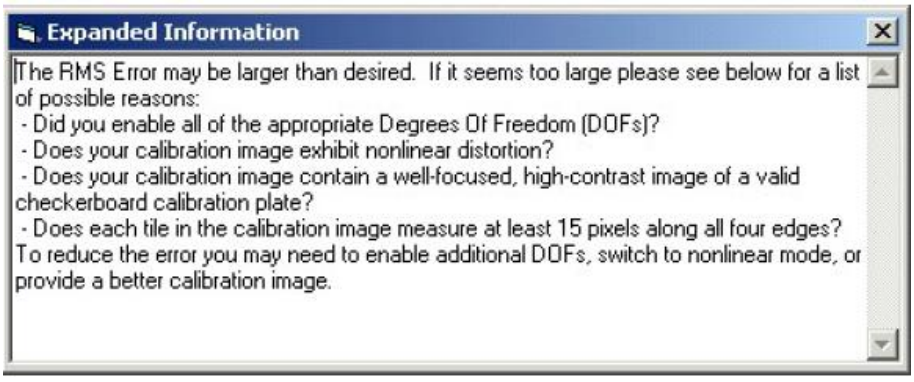
Button	Description	Function
	Save	Save the current properties of the vision tool to a VisionPro persistence (.vpp) file. The edit control allows you to choose between saving the vision tool with or without its image buffers and tool results.
	Save As	Save the current properties of the vision tool to a new VisionPro persistence (.vpp) file.
	Reset	Reset the vision tool to its default state.
	Show Floating Results	Enable or disable the display of tooltips for individual items in the edit control.
	Show ToolTips	Enable or disable the display of tooltips for individual items in the edit control.
	Help	Open this VisionPro online help file.

5.5.3.2 Calibration Info

The edit control includes a **GetInfoStrings** text box along the bottom, which displays any diagnostic text strings from the last attempt to calculate a 2D transformation.
The following figure shows an example Calibration Info box:



Additional information appears in a separate box, as shown in the following figure:



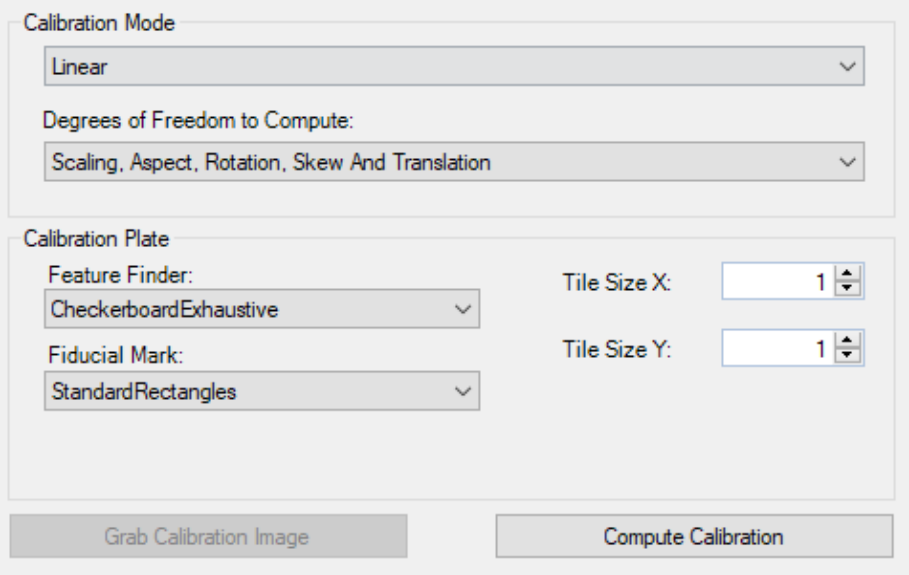
The text box is empty if the tool is not calibrated.
If the tool did not produce any message strings during the last calibration attempt, the box appears empty. All the tabs on the edit control display this message box.

5.5.3.3 Calibrate Tab

This section contains the following subsections.

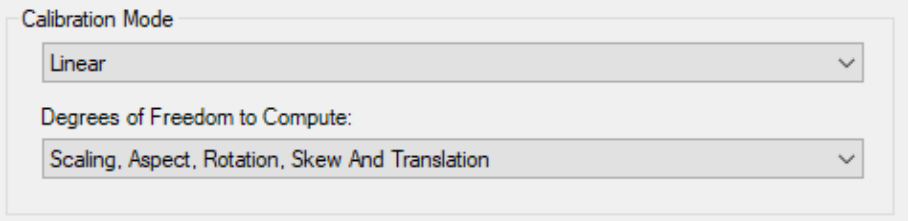
- Calibration Mode
- Calibration Plate
- Grab Calibration Image
- Compute Calibration

Use the Calibrate tab to select the type of 2D transformation to generate and define the checkerboard tiles in terms of the physical units you want to use in your application. The following figure shows the default Calibrate tab:



NOTE
Tile size Depend on which calibration mode and calibration plate type is selected; additional controls may be visible on the Calibrate tab.

- **Calibration Mode**



Select one of the following calibration **Calibration Mode** values.

Mode	Description
NoDistortionWarp	This mode will model perspective distortion only; any nonlinear optical distortion is ignored. By comparing the residual error values produced using this computation mode with the residual error values from **ThreeParamRadialWarp** or SineTanLawProjectionWarp you can improve your understanding of the individual sources of residual error.
ThreeParamRadialWarp	This model calibrates for nonlinear optical distortion and perspective distortion. When compared with **PerspectiveAndRadialWarp**, this mode adds additional coefficients that properly model the location of the optical center.  **NOTE** **This mode is recommended for lenses with minimal to moderate distortion, typically those with focal lengths greater than 6mm**.
SineTanLawProjectionWarp	This model calibrates for nonlinear optical distortion and perspective distortion. When compared with **ThreeParamRadialWarp**, this model uses a computation model that is appropriate for lenses with moderate to severe distortion, typically those with focal lengths less than 6mm.
Linescan2DWarp	The calibration or fixturing computation will generate a nonlinear transformation between coordinate spaces suitable for use with a linescan camera. The transformation will calibrate for perspective and radial distortion.  **NOTE** **This mode accomodates linescan configurations where the motion stage is tilted with respect to the direction of motion.** **You may optionally specify the "LinescanDistanceToTarget" (in physical units) from the camera's image sensor plane to the surface of the calibration plate. If you supply this value, it should be accurate to within +/- 10%. Supplying this distance improves the robustness and accuracy of the computed calibration.**

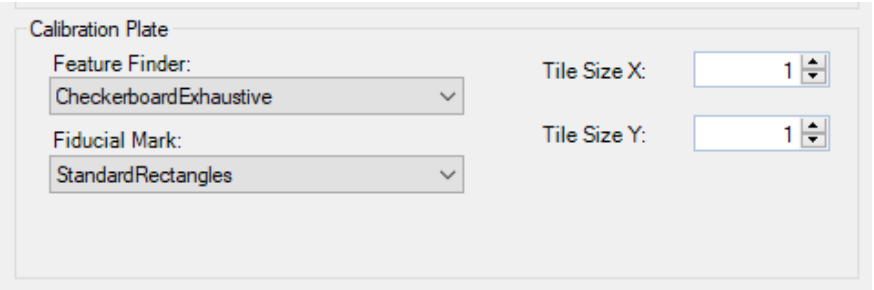
Mode	Description
LinescanWarp	The calibration or fixturing computation will generate a nonlinear transformation between coordinate spaces suitable for use with a linescan camera. The transformation will calibrate for perspective and radial distortion.  **NOTE** **You may optionally specify the "LinescanDistanceToTarget" (in physical units) from the camera's image sensor plane to the surface of the calibration plate. If you supply this value, it should be accurate to within +/- 10%. Supplying this distance improves the robustness and accuracy of the computed calibration.**
PerspectiveAndRadialWarp	This model calibrates for nonlinear optical distortion and perspective distortion. This method assumes that the optical center precisely corresponds to the image sensor center.  **NOTE** **Cognex recommends using "ThreeParamRadialWarp" or "SineTanLawProjectionWarp".**
Linear	This model calibrates for linear distortion (aspect, skew, and shear) only.  **NOTE** **If you select this mode, you can specify which "DOFsToCompute" to allow as the tool calculates the best-fit transformation between uncalibrated points and the raw calibrated points.**

NOTE
If you are calibrating a linescan camera, you can choose between the nonlinear linescan calibration with one-dimensional warping or two-dimensional warping. See the topic on "Linescan Camera Calibration" for the differences between the warping functions.

Select one of the following calibration **Degrees of Freedom to Compute** values.

Option	Description
None	The transformation returned by the N point to N point fitting will be the identity transform. It will have unity scale and aspect. It will contain no rotation, skew, or translation. No points are required for this DOF computation.
TranslationX	The transformation computed by the N point to N point fitting will contain only translation in the X direction. At least one pair of points is required for this DOF computation.
TranslationY	The transformation computed by the N point to N point fitting will contain only translation in the Y direction. At least one pair of points is required for this DOF computation.
Translation	The transformation computed by the N point to N point fitting will contain only translation. At least one pair of points is required for this DOF computation.
RotationAndTranslation	The transformation computed by the N point to N point fitting will contain only rigid rotation and translation. At least two pairs of points are required for this DOF computation.
ScalingRotationAndTranslation	The transformation computed by the N point to N point fitting will contain uniform scaling, rigid rotation, and translation. At least two pairs of points are required for this DOF computation.
ScalingAspectRotationAndTranslation	The transformation computed by the N point to N point fitting will contain nonuniform scaling, rigid rotation, and translation. At least three pairs of points are required for this DOF computation.
ScalingAspectRotationSkewAndTranslation	The transformation computed by the N point to N point fitting will contain nonuniform scaling, skewed rotation, and translation. At least three pairs of points are required for this DOF computation.

- **Calibration Plate**



Use the following options to describe the attributes of the calibration plate (grid type spacing and fiducial type) and to control the method the tool uses to locate grid points in the image.

NOTE
If you are specified any of the DataMatrix fiducial types, you can enable a timeout for the calibration operation. If the "DataMatrix Timeout" checkbox is checked, the tool will terminate the calibration operation after the specified timeout value (DataMatrixTimeoutSeconds) has passed.

Parameter	Description
Feature Finder	This value specifies how the tool will search the calibration image for calibration vertex points. This method must match the type of calibration plate that you are using. ▪ **Checkerboard** VisionPro supports this mode for existing applications but does not recommend this for new applications. For checkerboard plates with Cognex 'L' fiducial marks or without a fiducial mark, use **CheckerboardExhaustive** or **CheckerboardEfficient**. For checkerboard plates with DataMatrix fiducials, use **CheckerboardExhaustiveMultiRegion** or **CheckerboardEfficientMultiRegion.** ▪ **DotGrid** VisionPro supports this mode for existing applications that use a grid-of-dots calibration plate. Cognex recommends checkerboard plates for all calibration applications. ▪ **CheckerboardExhaustive** This mode provides the most robust feature extraction possible for checkerboard plates (with or without the Cognex "L" fiducial marks) but requires the longest time to execute. Choose the **CheckerboardEfficient** mode for generally faster processing times. ▪ **CheckerboardExhaustiveMultiRegion** This mode provides the most robust feature extraction possible for checkerboard calibration plates with DataMatrix fiducial marks but requires the longest time to execute. Choose the

Parameter	Description
	CheckerboardEfficientMultiRegion mode for generally faster processing times. This method can tolerate partial occlusion of plate features from reflections or other issues, as long as the fiducial marks are visible. ▪ **CheckerboardEfficient** This mode provides generally good calibration results for checkerboard plates (with or without the Cognex "L" fiducial marks) in less time than **CheckerboardExhaustive** mode, provided certain conditions apply. See the section **Feature Extraction Modes** for details. ▪ **CheckerboardEfficientMultiRegion** This mode provides generally good calibration results for checkerboard plates with DataMatrix fiducial marks in less time than **CheckerboardExhaustiveMultiRegion** mode, provided certain conditions apply. See the section **Feature Extraction Modes** for details.
Fiducial Mark	This value specifies the type and characteristics of the fiducial marks on the calibration plate. **NOTE** **Not all combinations of feature finding and fiducial marks are valid. If you enter an invalid combination, the tool will generate an error at calibration time.** ▪ **DataMatrixWithGridPitch** This specifies that the plate includes Cognex-compatible DataMatrix fiducial marks, each of which specifies the coordinates of the mark in the plate's 2D coordinate system and the grid pitch in mm. If you specify this mode, the tool ignores any grid pitch values that you supply. ▪ **DataMatrix** This specifies that the plate includes Cognex-compatible DataMatrix fiducial marks, each of which specifies the coordinates of the mark in the plate's 2D coordinate system. If you specify this mode, you must specify the grid pitch explicitly (**PhysicalTileSizeX** and **PhysicalTileSizeY**). If you use this mode and the fiducial marks include grid pitch information, that information is ignored; the tool uses the values that you specify. ▪ **DotGridAxes** The use of this method is supported for legacy applications that use a grid-of-dots calibration plate. The origin and grid pitch are specified by a pattern of extra dots on the plate.

Parameter	Description
	▪ **StandardRectangles** This specifies that the plate uses a single standard Cognex 'L' fiducial mark. ▪ **None** Use this value for calibration plates with no fiducial marks. The Calibration tool will use the tile vertex or dot center that is closest to the center of the image as the origin and will assign the positive x-axis and y-axis to the grid lines which are closest in angle to the x-axis and y-axis in the pixel coordinate space of the calibration image
Tile Size X **Tile Size Y**	Enter the grid spacing for the calibration plate. For a checkerboard-style plate, this is the tile size. For a grid-of-dots calibration plate, this is the spacing between dot centers in the X- and Y-direction. Enter the grid spacing using real-world units of measurement you want to use for your vision application. For example, if you want to use inches for your application and your calibration plate uses tiles that are one-half inch in size, enter a value of 0.5. **NOTE** **If you are using a calibration plate that specifieds grid spacing, the tool replaces any grid spacing values that you have entered with those defined by the plate markings. The units (inches or mm) specified by the plate markings are discarded.**

- **Grab Calibration Image**

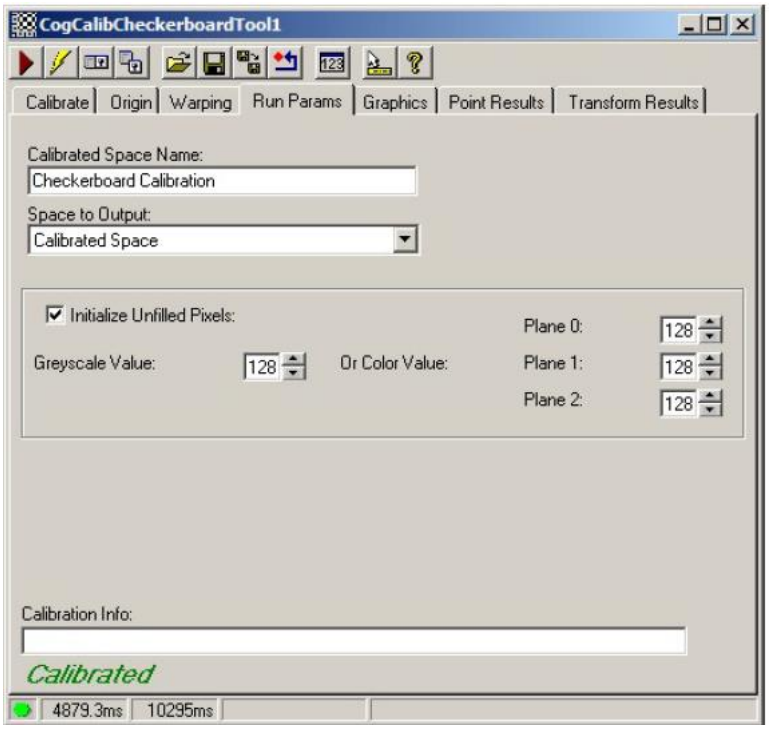
Click **Grab Calibration** to copy the current image stored in the Current.InputImage buffer into the Current.Calibration buffer. To configure the tool, you must copy an image of a calibration plate into the Current.Calibration buffer.

- **Compute Calibration**

Click **Calibrate** to have the tool calculate the best-fit 2D transformation, linear or nonlinear, based on the Current.Calibration image and the current set of parameters.

5.5.3.4 Run Params Tab

Use the Run Params tab to specify the name of the final calibrated coordinate space and the grey value of any undefined pixels in the output image after the tool warps the image. The following figure shows the default Run Params tab:

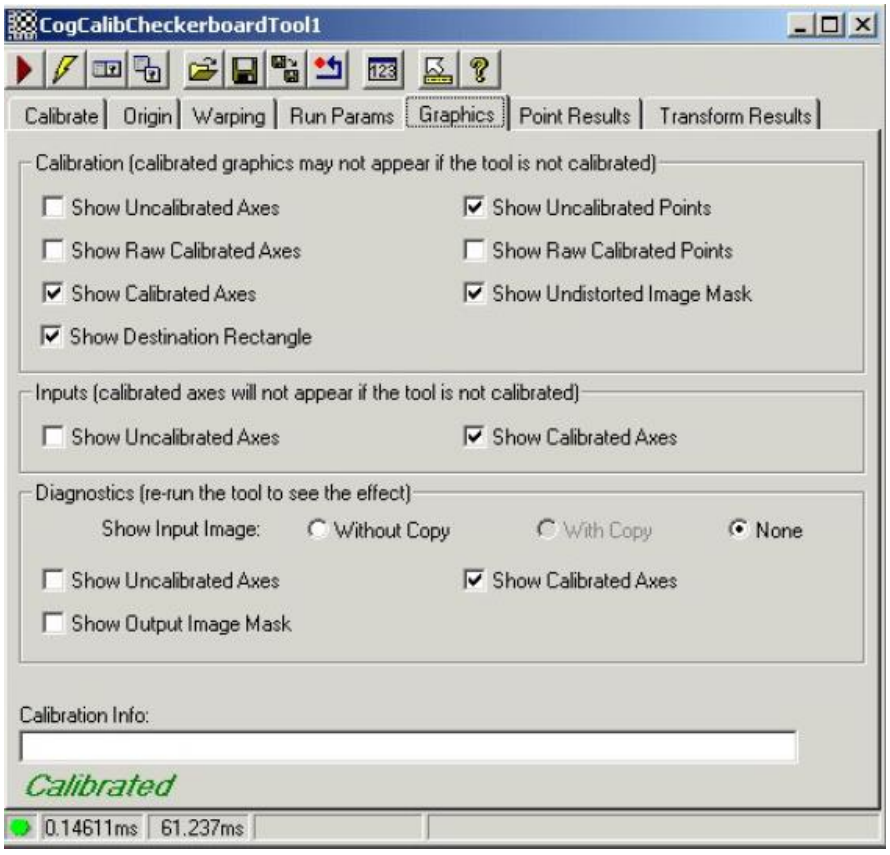


The Run Params tab contains the following parameters:

Parameter	Description
CalibratedSpaceName	Choose a valid non-qualified name for the calibrated coordinate space that will be attached to the coordinate space tree of the output image.
SpaceToOutput	Choose whether the selected space of the output image will be a fully-qualified copy of the calibrated space name, or a fully qualified copy of the uncalibrated space name.
UnfilledPelValue **UnfilledPelValueEnabled** **UnfilledPelPlane0Value** **UnfilledPelPlane1Value** **UnfilledPelPlane2Value**	Select the value for unfilled pixels in the output image after the tool performs an image warping. If you leave the **Initialize Unfilled Pixels**: checkbox unchecked, unfilled pixels will not get initialized. For monochrome images, choose the **Greyscale Value**. For color images, choose a value for each color plane; RGB or HSI depending on your image type. **Plane 0** for R or H, **Plane 1** for G or S, and Plane 2 for B or I.

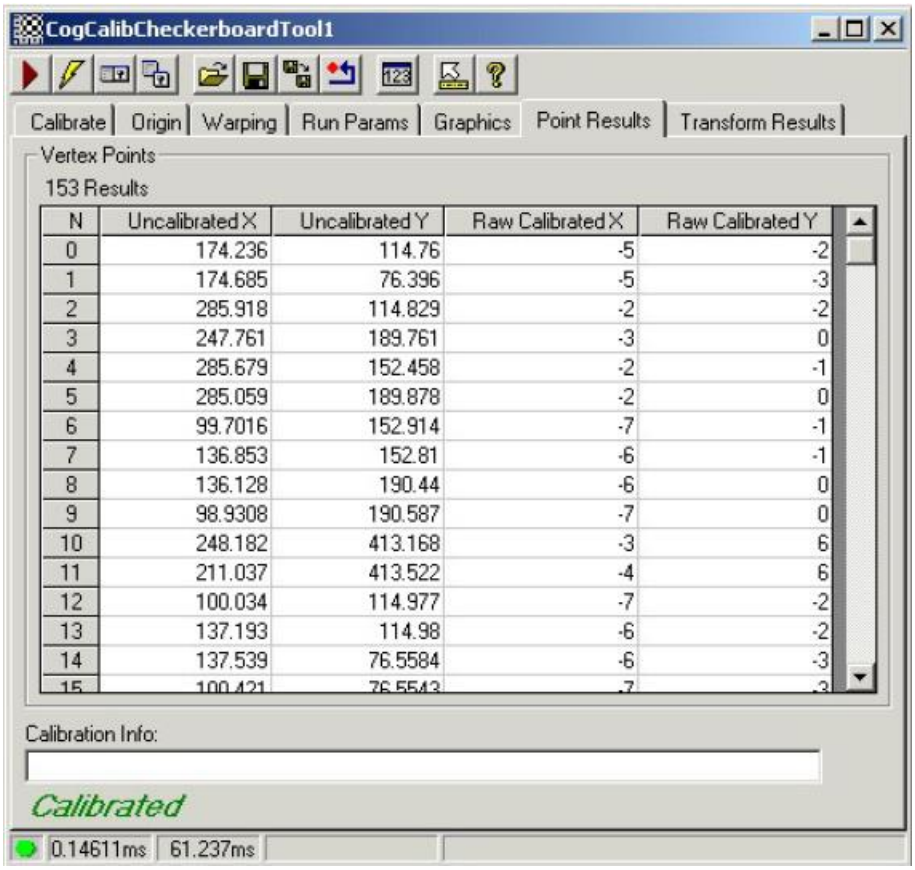
5.5.3.5 Graphics Tab

Use the Graphics tab to determine what graphics the tool will use. The following figure shows the default Graphics tab for a tool configured with a nonlinear calibration:



5.5.3.6 Point Results Tab

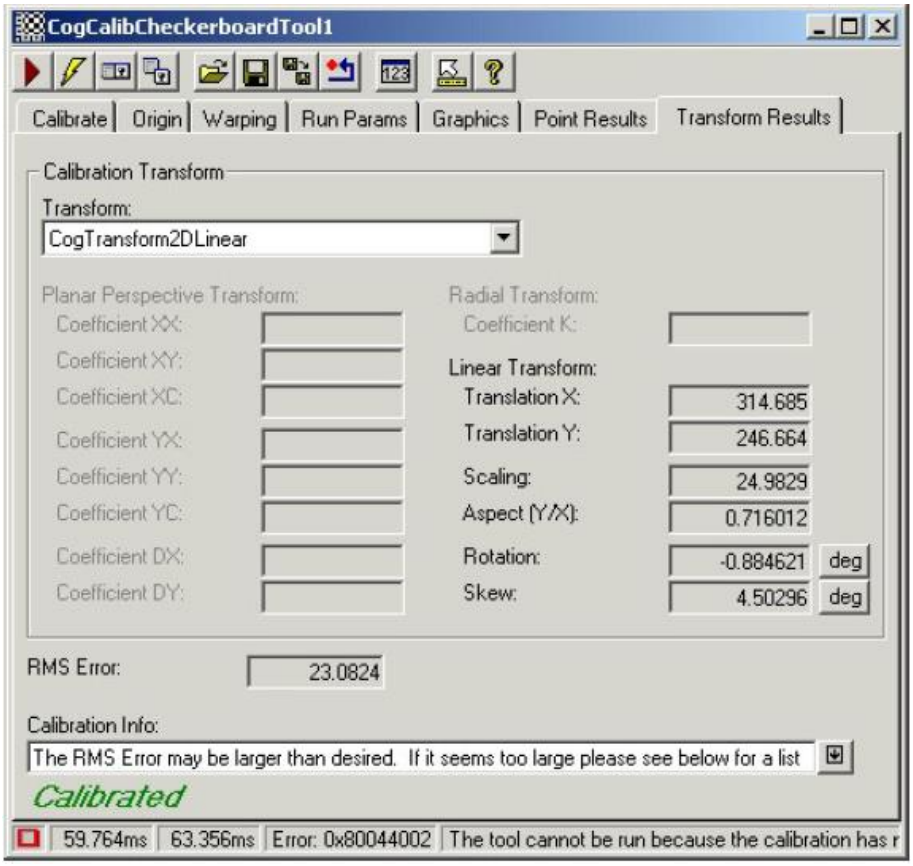
Use the Point Results tab to view all the vertex points found in the calibration plate. The following figure shows an example Point Results tab:



All fields in the Point Results tab are read only. The uncalibrated point results are reported in the selected space coordinates of the calibration image, while the raw calibrated point results are reported in the raw calibrated coordinate space.
Select any point result to highlight the point on the Current.CalibrationImage buffer (and on the Current.UndistortedCalibrationImage buffer is warping is in effect).

5.5.3.7 Transform Results Tab

Use the Transform tab to view details of the 2D transformation as calculated by the tool. The following figure shows an example Transform Results tab:



The Transform Results tab contains the following fields:

Field	Description
Transform list	The list displays one or more of the following transform types that the tool has calculated: ▪ **CogTransform2DCameraCalibration** if the computation method is **ThreeParamRadialWarp**, **SineTanLawProjectionWarp**, or **NoDistortionWarp**. ▪ **CogTransform2DLinear** for a linear transform with or without origin adjustment ▪ **CogTransform2DPerspectiveAndRadial** for a nonlinear transform without origin adjustment ▪ **CogTransform2DPerspectiveAndRadial** and **CogTransform2DLinear** for a nonlinear transform with origin adjustment.

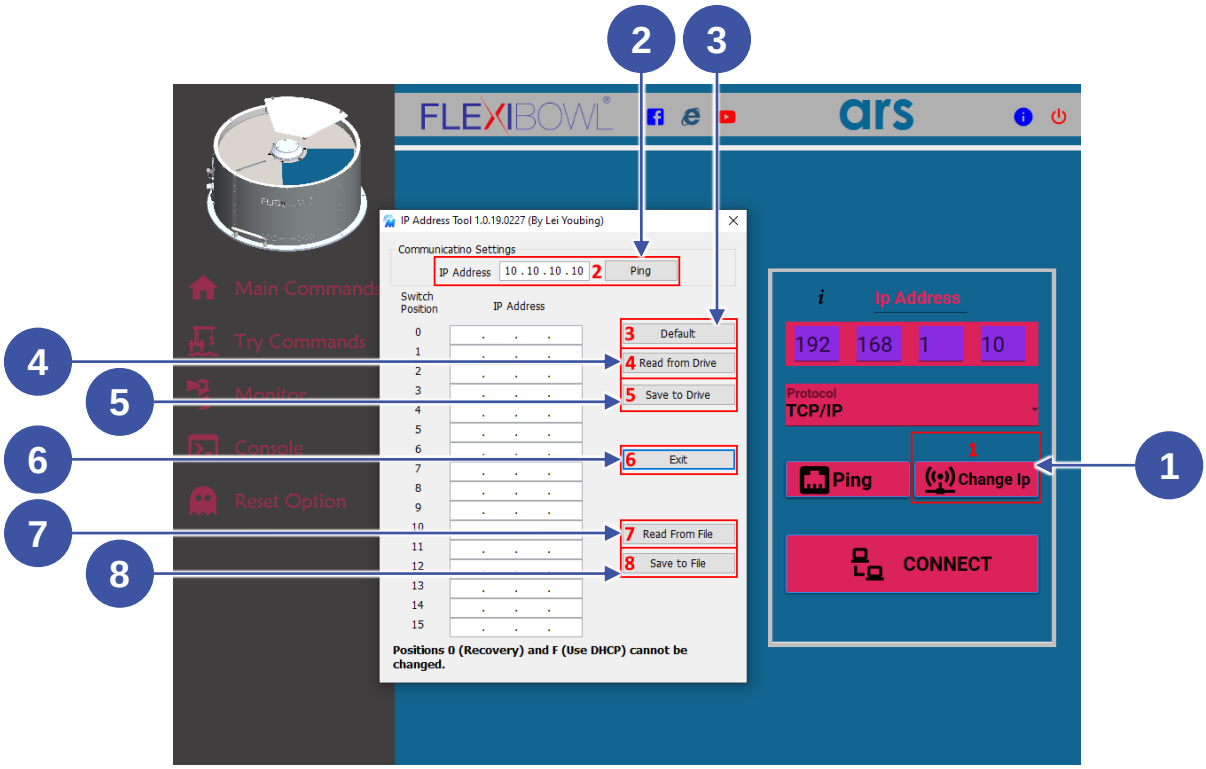
Field	Description
Planar Perspective Transform	These values describe the planar perspective properties of the uncalibrated to raw calibrated transform. These fields are disabled if the computation method is **Linear, ThreeParamRadialWarp, SineTanLawProjectionWarp** or **NoDistortionWarp**
Radial Transform	These values describe the radial distortion property of the uncalibrated to raw calibrated transform. These fields are disabled if the computation method is **Linear, ThreeParamRadialWarp, SineTanLawProjectionWarp,** or **NoDistortionWarp**.
Linear Transform	These values vary depending on the type of 2D transformation the tool calculated, in the following ways: ▪ For a linear transformation these values indicate the entire transform from calibrated to uncalibrated space. ▪ For a nonlinear transformation without an adjustment, it is the final linear portion of the perspective and radial transform that maps from raw calibrated to uncalibrated space. ▪ For a nonlinear transformation with an adjustment, these values represent both the transform described previously and the linear adjustment transform that maps from raw calibrated space to calibrated space, depending upon the selection made in the Transform list. ▪ For **ThreeParamRadialWarp, SineTanLawProjectionWarp,** and **NoDistortionWarp**, these values are not available.
ComputedRMSError	This is the error between the uncalibrated points and the mapped raw calibrated points, expressed in uncalibrated space. In most cases, a large RMS error indicates that you have specified a linear transformation when the calibration image exhibits significant perspective or radial distortion.

PAGE INTENTIONALLY LEFT BLANK

6 FLEXIBOWL®

6.1 How to set the IP address

With FLEXIBOWL running, press the **Change IP** button in the following page.
The table shown in the picture is used to set a different IP address (even if in the same class).



Position	Element	Description
1	Change IP	Opens a window that allows to change the IP address.
2	Ping	Allows to ping the Flexibowl® with the address specified in the IP address.
3	Default	Enters the FLEXIBOWL® default addresses in the table.
4	Read from Drive	Reads the current set IP addresses.
5	Save to Drive	Saves the current set IP addresses.

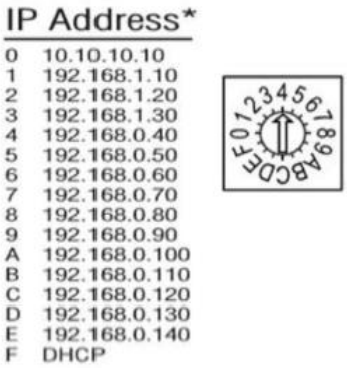
Position	Element	Description
6	Exit	Exit from this page.
7	Read from file	Reads the file with the IP addresses, if previously geneerated by pressing the SAVE TO FILE key.
8	Save to File	Generates a file containing the current table of IP addresses.

NOTE
Restart Flexibowl® to make the changes effective.

NOTE
IP addresses related to position O and F cannot be modified.

6.1.1 IP address recovery

If the IP address is lost, the rotary dip switch can be used to interface again with the Flexibowl (see following picture):



CAUTION!
Disconnect the power supply before taking the cover off.

For **FLEXIBOWL®200 and 350**, proceed as follows to place the dip switch in a position different from the current one:

Step	Action	Notes/Pictures
1	Unplug the power cable from the control panel.	
2	Take the cover off the FlexiBowl®.	
3	Find the driver.	
4	Use a flat screwdriver to select the correct dip switch position	IP Address* 0 10.10.10.10 1 192.168.1.10 2 192.168.1.20 3 192.168.1.30 4 192.168.0.40 5 192.168.0.50 6 192.168.0.60 7 192.168.0.70 8 192.168.0.80 9 192.168.0.90 A 192.168.0.100 B 192.168.0.110 C 192.168.0.120 D 192.168.0.130 E 192.168.0.140 F DHCP 
5	Riassemble all the components.	

For **FLEXIBOWL®500, 650 and 800**, proceed as follows to place the dip switch in a position different from the current one:

Step	Action	Notes/Pictures
1	Unplug the power cable from the control panel.	
2	Take the cover off the FlexiBowl®.	
3	Remove the connector between the backlight and the connection board.	

Step	Action	Notes/Pictures
4	Unscrew the socket head cap screws fixing the backlight to the FlexiBowl®. **Note** **Keep the removed screws for reassembly.**	
5	Use a flat screwdriver to select the correct dip switch position	IP Address* 0 10.10.10.10 1 192.168.1.10 2 192.168.1.20 3 192.168.1.30 4 192.168.0.40 5 192.168.0.50 6 192.168.0.60 7 192.168.0.70 8 192.168.0.80 9 192.168.0.90 A 192.168.0.100 B 192.168.0.110 C 192.168.0.120 D 192.168.0.130 E 192.168.0.140 F DHCP 
6	Riassemble all the components.	

6.2 How to connect the Flexibowl® to Flexivision

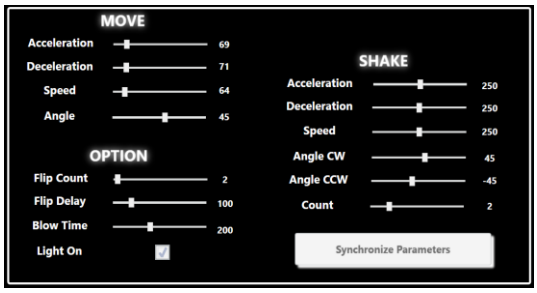
6.2.1 Flexibowl Param page

By pressing the *FLEXIBOWL PARAM* key on the OPERATION MENU, the following page opens.

Position	Element/section	Description
1	OPERATION MENU	
2	FLEXIBOWL IP ADDRESS	
3	CONNECTION STATUS led	Led GREEN: Flexibowl connected Led RED: Flexibowl not connected
4	TEST CONNECTION pushbutton	
5	FLEXIBOWL PARAMETERS	
6	SYNCHRONIZE PARAMETERS pushbutton	
7	FLEXIBOWL MOVEMENT SEQUENCE	

Position	Element/section	Description
8	TEST SEQUENCE pushbutton	
9	HOPPER ACTIVATION check box	
10	HOPPER HYSTOGRAM pushbutton	
11	HOPPER HYSTOGRAM PARAMETERS	

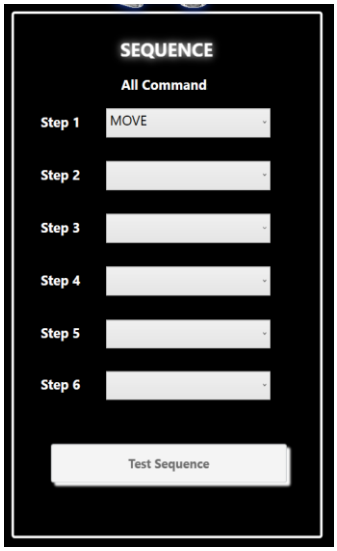
6.2.1.1 FLEXIBOWL Parameters

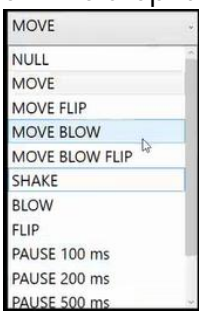


ID	Element	Description
1	MOVE - Acceleration	Acceleration value used at each MOVE command
2	MOVE - Deceleration	Deceleration value used at each MOVE command
3	MOVE – Speed	Speed value (rpm) used at each MOVE command
4	MOVE - Angle	Angle at which FlexiBowl® moves at each MOVE command
5	SHAKE - Acceleration	Acceleration value used at each SHAKE command
6	SHAKE - Deceleration	Deceleration value used at each SHAKE command
7	MOVE – Speed	Speed value (rpm) used at each SHAKE command
8	MOVE – Angle CW	Clockwise Angle with which FlexiBowl® moves at each SHAKE command
9	MOVE – Angle CCW	Counterclockwise Angle with which FlexiBowl® moves at each SHAKE command
10	OPTION – Flip Count	Number of Flip activations that will be carried out

ID	Element	Description
11	OPTION – Flip Delay	Time (in milliseconds) between a flip activation and deactivation
12	OPTION – Blow time	Time (in milliseconds) of blow time activation
13	OPTION – Light on	Press to enable/disable the backlight

6.2.1.2 Robot sequence



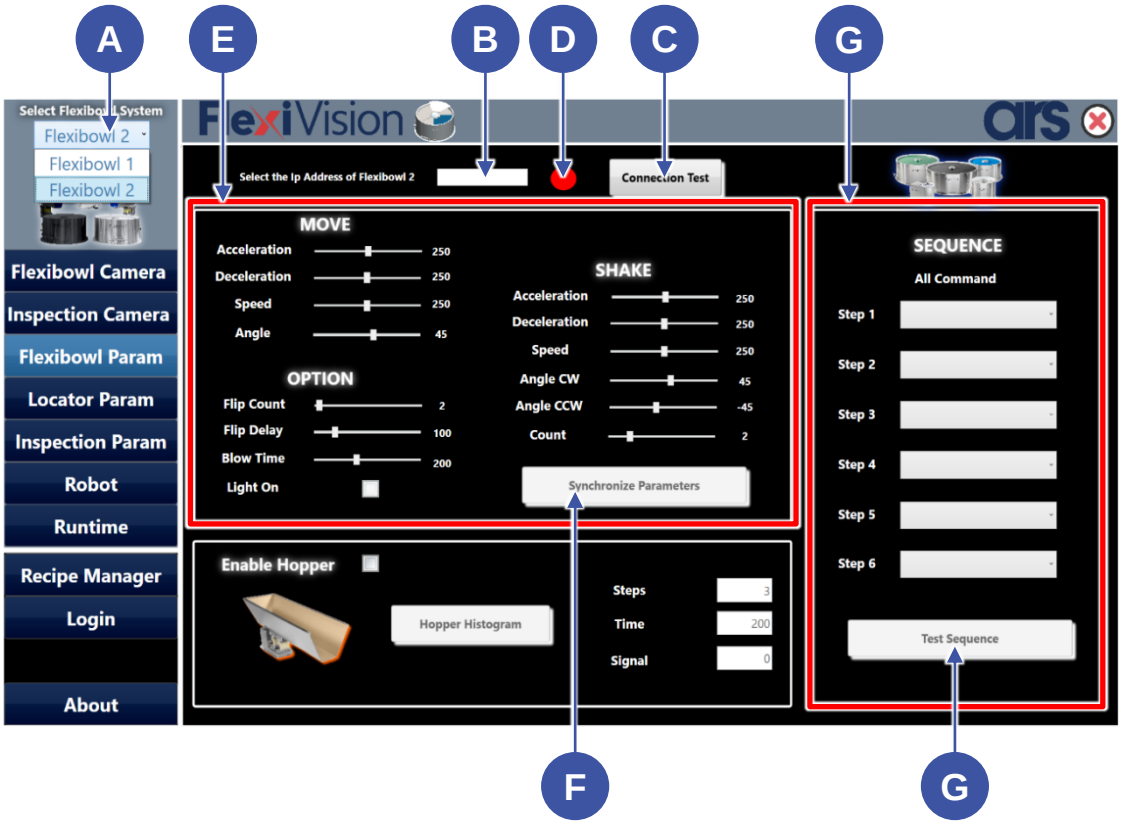
ID	Element	Description
1	SEQUENCE	For each step of the sequence, select the movement from the drop-down menu.  If during image acquisition the camera does not find any object, FlexiVision will automatically recall this sequence to move the FlexiBowl® plate to present other components.
2	TEST SEQUENCE	Press to test the sequence

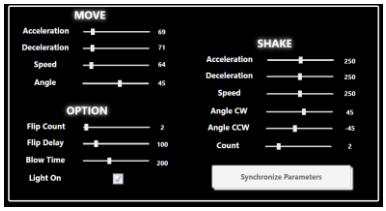
6.2.1.3 HOPPER parameters setting

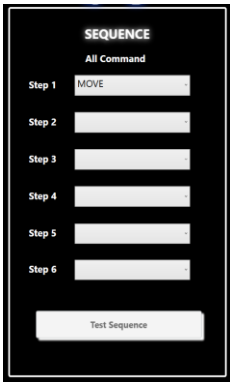
ID	Element	Description
1	ENABLE HOPPER	Enables/disables the vibrating hopper control.
2	HOPPER HISTOGRAM pushbutton	Enters the hopper **CogHistogramTool**.
3	STEPS	Sets the no. of steps (forward movements) which pass between the backlit imaging acquiring area and the hopper.
4	TIME	Sets the hopper vibration time
5	SIGNAL	Sets the signal number to be sent to the robot.

NOTE
Flexivision does not directly control the vibrating hopper by I/O but sends a string to the robot for it to do.

6.2.2 Connection to FLEXIBOWL



Step	Action	Notes/Pictures
1	Select the FLEXIBOWL from the drop down menu (**A**).	
2	Enter the FLEXIBOWL IP ADDRESS in the (**B**) field.	
3	Press the CONNECTION TEST button (**C**). If connection is successful, led (**D**) turns to green. If connection is not successful, led (**D**) remains red.	
4	Set the FLEXIBOWL parameters value (**E**), by the slide bars.	

Step	Action	Notes/Pictures
5	Enable or disable the backlight.	
6	Press the SYNCHRONIZE PARAMETERS pushbutton.	
7	Select, for each step, the ROBOT MOVEMENT SEQUENCE: if the camera does not find any object during image acquisition, FlexiVision will automatically recall this sequence to move the Flexibowl plate.	

7 ROBOT

7.1 General requirements

The robot must have the ability to open and manage one or more communication servers using the TCP / IP protocol, in order to receive and send strings by a task parallel to the main one, so as to receive information from the vision system while the main job is running.

7.1.1 Standard data structure

Data	Terminator char
String (command)	CHR (13)

Command	Action
"set_Recipe=recipe name"	The recipe corresponding to the sent "recipe name" is loaded.
"get_Recipe"	The name of the recipe currently loaded on FlexiVision is shown. **Return**:" recipe name".
"start_Locator"	Starts the parts localization process by recalling the FlexiBowl® handling routine in case there are no parts that can be picked up. **Return**: "Pattern1;x;y;r".
"stop_Locator"	Stops the process of locating the object with the aid of the FlexiBowl®.
"turn_Locator"	If no parts are picked up, by this command the operator can make the Flexibowl rotate and the "start_Locator" routine start. **Return**: "Pattern1;x;y;r".
"test_Locator"	Starts the process of locating the object without the aid of the FlexiBowl®. **Return**: "Pattern1;x;y;r".
"start_Control"	Starts the inspection cycle. Return: "Control1;x;y;r".
"state_Locator"	Locator status diagnostics is shown: **Return**: ▪ "Locator is Running" ▪ "Locator is in Error" ▪ "Locator is not Running".
"start_Empty"	Start the FlexiBowl® Quick-Emptying sequence. **Return**: "start_Empty ended"

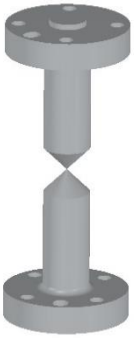
If hopper should be activated you will receive the string "Hopper;signalnumber;time"

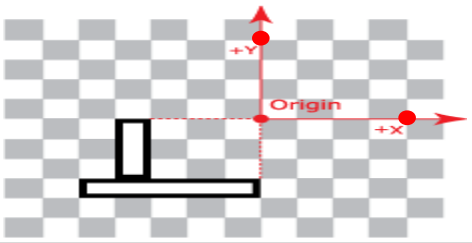
7.2 Robot-Tool creation and calibration

NOTE
For more details about the procedure for creating working tools and working frames, refer to the robot instruction manual.

This operation requires two calibration tips:

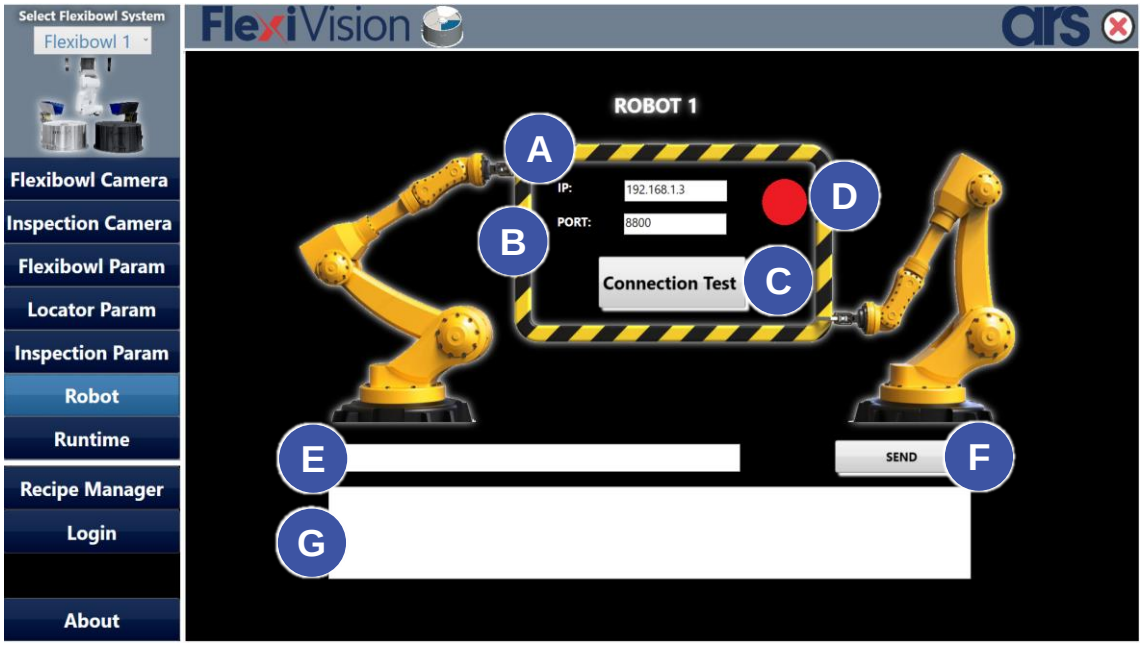
- the first shall be applied on the robot flange;
- the second one shall be positioned on the FlexiBowl® or on a planar surface, in order to carry out the tool procedure.

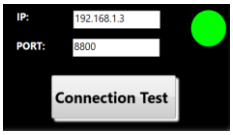


Step	Action	Notes/Pictures
1	Remove any tool on the robot flange and replace it with the first calibration tip.	
2	Place the second calibration tip on the FlexiBowl® or on a planar surface.	
3	Create the tool following the procedure written in the robot's manual.	
4	Create a 3-point frame (origin, x, y) on the robot.	
5	Open the dialogue window to create the calibration working frame, keeping care not to activate the tool created previously.	
6	For registration of calibration points, position the robot on the intersection of the squares of the calibration grid, as shown in the picture.	

7.3 How to connect the robot

By pressing the *ROBOT* key on the OPERATION MENU, the following page opens.



Step	Action	Notes/Pictures
1	Ensure that the robot server is up.	
2	Enter the ROBOT IP ADDRESS in the (**A**) field.	
3	Enter the ROBOT port in the (**B**) field.	
4	Press the CONNECTION TEST button (**C**). If connection is successful, led (**D**) turns to green. If connection is not successful, led (**D**) remains red.	 
5	Enter a string in the (**E**) field and press SEND (**F**) to test the communication with the robot.	
6	The message sent by the robot is displayed in field (**G**).	

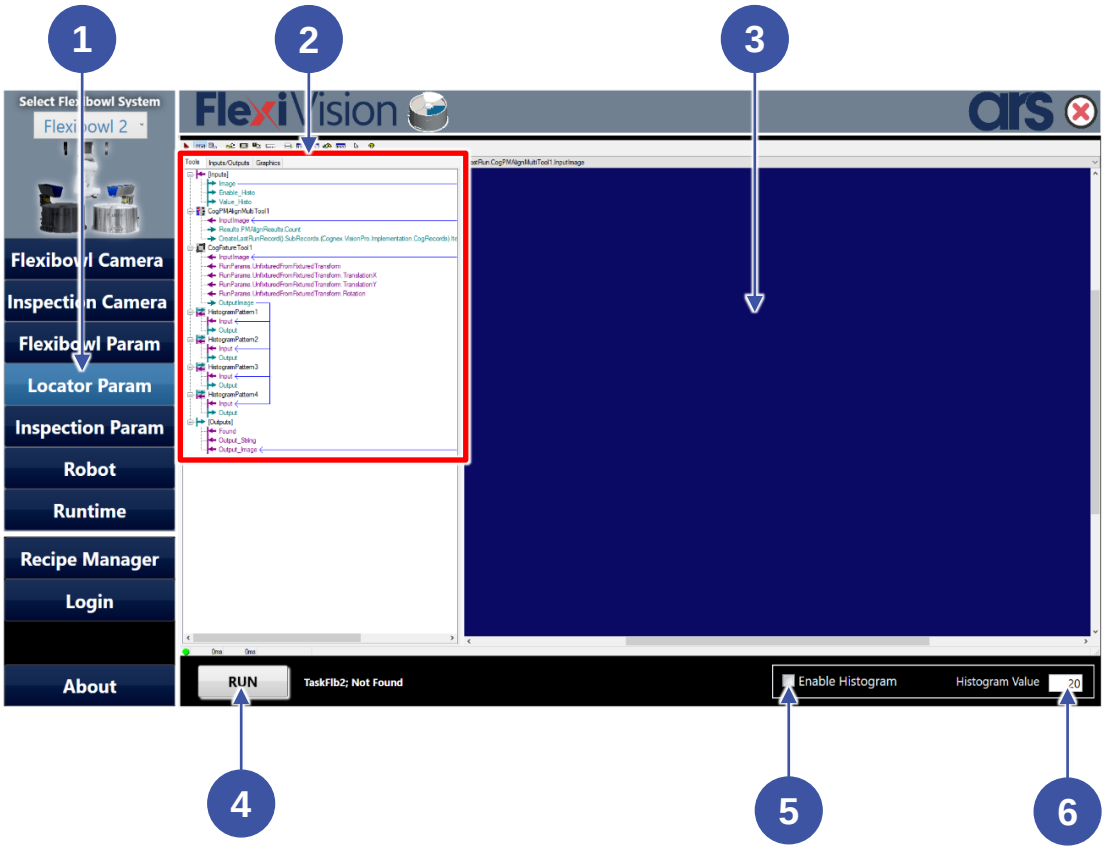
7.4 Example

Contact ARS at info@flexibowl.com to receive communication examples.

8 LOCATOR

8.1 Locator Param page

By pressing LOCATOR PARAM in the OPERATIONS menu, the following page opens:



Position	Element/section	Description
1	OPERATION MENU	Includes all the control and operation procedures.
2	**PMAlign Edit Control** TOOL	
3	LAST INPUT IMAGE	
4	RUN key	
5	ENABLE HISTOGRAM check box	
6	HISTOGRAM VALUE	

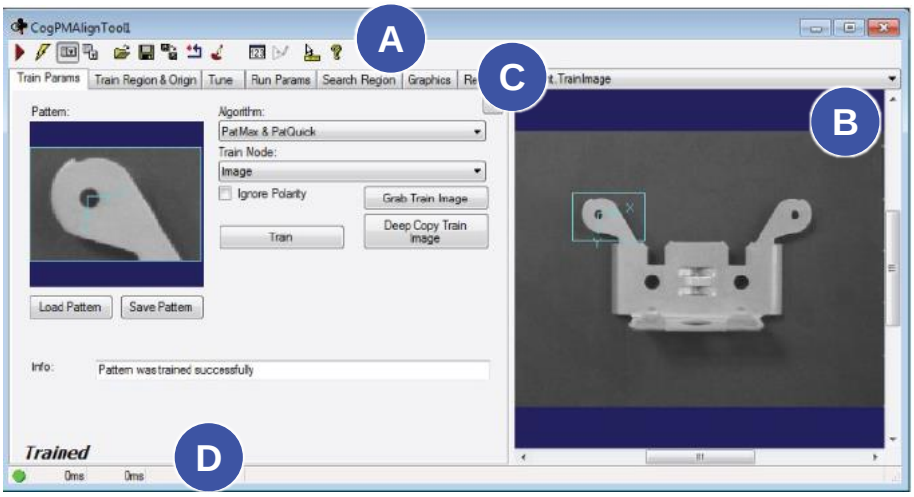
8.2 Elements of standard Locator

Standard locator is composed of the elements describe in the following.

8.2.1 PMAlign Edit Control

The following is an extract from the Cognex User Manual: for further and complete details, please refer to the whole Cognex User Manual.

The PMAlign edit control provides a graphical user interface to the **CogPMAlignTool** and its component, which allows you to train a pattern and then have the tool search for it in successive input images. You can specify the type of algorithm to use when performing the pattern training or the pattern search, and choose between creating a trained pattern from an image or from a collection of shape models. The pattern search can be constrained by an optional search region within the input image. The following figure shows an example PMAlign edit control:



The PMAlign edit control includes the following components:

- A row of control buttons at the top left (A).
- A tool display window (B) that can display the PMAlign tool image buffers: Current.TrainImage, Current.InputImage, and LastRun.InputImage. These buffers contain the trained pattern, the search image in which the PMAlign tool searches for the pattern, and the same image with the results of the search. Right click the tool display to bring up menu options that include zooming in or out of the image or showing a pixel or subpixel grid.
- A set of tabs organized by function (C). These functions include parameter settings to run the tool, parameter settings for the search region of interest, display settings for the tool displays, and training results. Pressing the **Control + Tab keys** scrolls through the set of tabs.
- A status bar at the bottom left of the control (D). A green circle indicates that the tool ran successfully; red means the tool ran unsuccessfully. The status bar also displays the time to run the tool and any error codes or messages. The first time that the status bar displays is the raw tool execution time. The second includes the time needed to update the edit control.

This topic contains the following sections.

- Control Buttons
- PMAlign Edit Control Buffers
- Train Params Tab
- Train Region and Origin Tab
- Tune Tab
- Run Params Tab
- Search Region Tab
- Graphics Tab
- Results Tab

8.2.1.1 Control buttons

Button	Description	Function
	Run	**Run** the PMAlign tool. You must have a trained pattern, an input image, and specified run parameters. PMAlign searches for the trained pattern in the input image. You may constrain the pattern search to a search region within the input image.
	Electric mode	Toggles electric mode. When selected, the PMAlign tool runs automatically if certain parameters have changed. These parameters are indicated by electric bolt icons that appear when the tool is in electric mode.
	Local image display	Opens the local tool display window, which can display the **Current.InputImage,** **Current.PatternTrainImage,** or **LastRun.InputImage** buffer.
	Floating image display	Opens one or more floating tool display windows. You can display the **Current.InputImage, Current.PatternTrainImage,** or **LastRun.InputImage buffer**. Unlike the local tool display, you can resize or move the position of the floating tool display window.
	Open	Loads a VisionPro persistence (.vpp) file, which contains a set of saved properties for this vision tool object type. Loading a persistence file for another object type throws an error and the load is unsuccessful. For more information about VisionPro persistence features, see the topic Persistence in VisionPro.
	Save	Saves the current properties of the underlying tool to a VisionPro persistence file. You have the option to save either the entire tool or the tool without its images or results.

Button	Description	Function
	Save As	Saves the current properties of the underlying tool to a new VisionPro persistence file.
	Reset	Resets the underlying tool to a default state.
	Image mask editor	Opens the **Image Mask Editor** for creating a mask to add to the training image.
	Results	Opens a new, separate results window, allowing you to view run results without turning to the **Results** tab.
	Model maker	Launches the **Model Maker** for editing shape models when you are using Synthetic PatMax.
	Show ToolTips	Enable or disable the display of tooltips for individual items in the edit control.
	Help	Access the VisionPro Software Documentation.

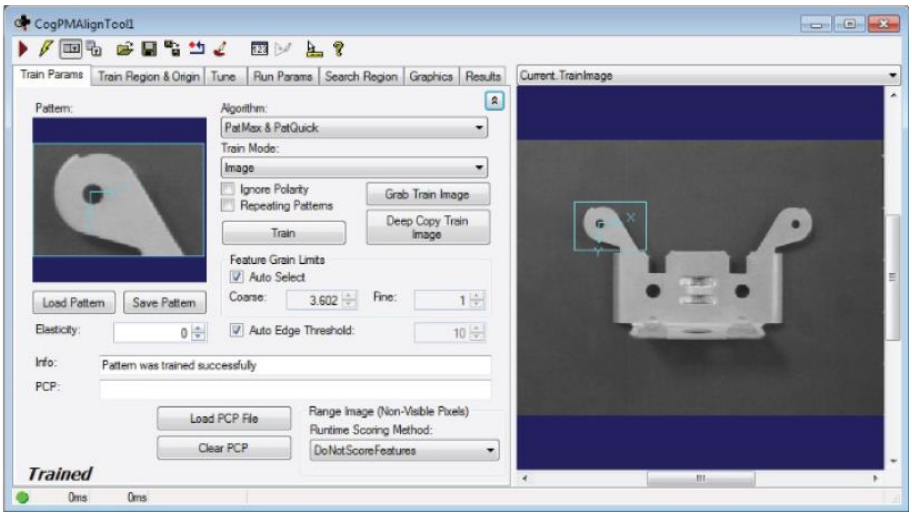
8.2.1.2 PMAlign Edit Control Buffers

The PMAlign edit control has three image buffers. Two of the buffers use the underlying PMAlign tool's InputImage and TrainImage; the third buffer displays the last input image that the PMAlign tool ran on and the results of that search. All three buffers can be shown in both the local and floating tool display windows.

- The Current.**InputImage** provides the input images to the PMAlign tool.
- The Current.***TrainImage*** contains the training image.
- The LastRun.InputImage buffer displays the last image on which the tool most recently ran. Use the Graphics tab to highlights the search area and the results of the search.

When you run the PMAlign tool, the tool searches the Current.InputImage for the pattern in the Current.TrainImage buffer, and stores the results of this search on the **Results** tab.

8.2.1.3 Train Params Tab



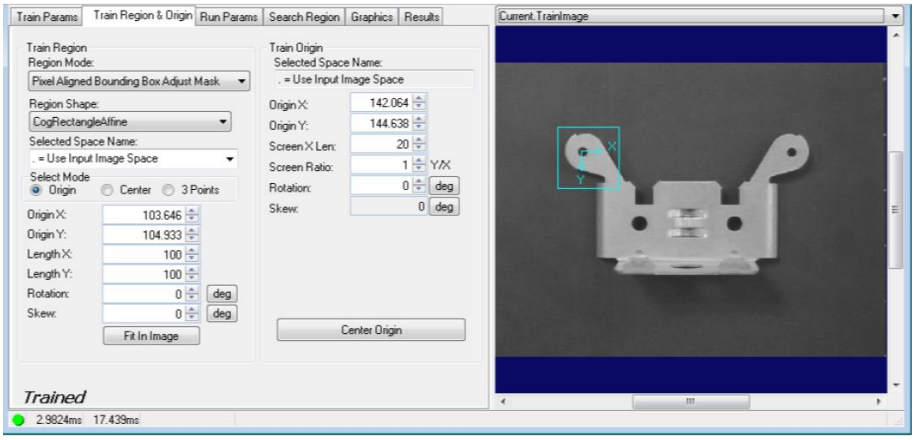
Use the **Train Params** tab to set training parameters and to train the search pattern. Clicking the **Advanced Feature** button on the far right of the tab displays less commonly used features.

Feature	Description
Pattern	Displays the trained pattern, created either from an image or from a collection of shape models. This is specified by the **Train Region**, which is highlighted by a blue border, within the **TrainImage.** You can set the train region using the **Train Region and Origin** tab or by resizing its display in Current.TrainImage. The message text at the bottom of the tab indicates whether the PMAlign tool is trained.
Load Pattern	Opens a VisionPro persistence file, which has a VPP extension, that contains a trained pattern.
Save Pattern	Saves the current trained pattern into a VisionPro persistence file, which has a VPP extension.
Algorithm	Select the search algorithm to use for training: ▪ **PatMax:** Requires more time to execute but is more accurate and returns additional score information ▪ **PatQuick:** Faster but less accurate than PatMax ▪ **PatMax and PatQuick** ▪ **PatFlex:** Locate patterns that have undergone a variety of nonlinear deformations ▪ **PatMax - High Sensitivity:** Appropriate for images with very low contrast or significant video noise or image degradation ▪ **Perspective PatMax**: Locate 2D features that have undergone perspective distortion
Train Mode	Choose whether the PMAlign pattern should be trained based on the pixel content of the training image or trained based on the shape models you create and modify with the Model Maker.

Feature	Description
Ignore Polarity	If enabled, pattern polarity is ignored. If disabled, only patterns with polarity matching the trained pattern will be found. You must allow the tool to ignore polarity if you are using a trained pattern made of shape models and any of the models have an undefined polarity.
Repeating Patterns	Use this parameter when the pattern you want to train contains elements that repeat, such as a grid or a set of bars or a pattern of parallel lines. This parameter is valid only when you choose **PatMax** from the **Algorithm** pulldown list.
Train	Trains the desired pattern as specified by the **TrainRegion** in the **TrainImage** or by the current collection of shape models. If the tool already has a trained pattern then the tool will untrain and then retrain. When the pattern is successfully trained, the text at the bottom of the control says, "Trained."
Grab Train Image	Sets the **TrainImage** property to a reference to the **InputImage**. This button is not enabled unless there is an image in Current.InputImage. It is also not enabled if you are creating a trained pattern from shape models and a transform.
Deep Copy Train Image	Clones the **InputImage** and assigns a reference to that deep copy to the TrainImage property. Perform a deep copy to prevent the PMAlign tool from becoming untrained if any Fixture tool updates the coordinate space tree of the **TrainImage** as your vision application runs.
Feature Grain Limits	If **Auto Select** is not selected, then the **Coarse** and **Fine** grain limits that you supply are used.
Elasticity	Specifies the Elasticity property, which is the amount of variance, in pixels, allowed by the PMAlign tool. In general, you should specify a nonzero elasticity value if you expect inconsistent variation in patterns in run-time images.
Auto Edge Threshold	Disable the automatic value if you want to set a different threshold for the absolute minimum value of edge magnitude, below which the edge direction will be randomized.
Info	Train-time diagnostic message containing information about this pattern.
PCP	The PCP (PatMax Customizable Pack) string associated with this pattern. Appears only if a PCP file, which configures PMAlign parameters for specific applications, has been loaded.
Load PCP File	Loads a PCP file.
Clear PCP	Clears the underlying tool of the configurations derived from the currently loaded PCP file.

Feature	Description
Runtime Scoring Method	Instructs the pattern how to treat non-visible pixels at run-time when the input image is of type **CogImage16Range**.

8.2.1.4 Train Region and Origin Tab



Use the **Train Region and Origin** tab to define the **TrainRegion**, which defines the area of the **TrainImage** buffer that becomes the trained pattern. You can also define the train region graphically in the Current.TrainImage buffer. It may be easier to first specify the training region graphically, then use this tab to fine tune the train region parameters. The PMAlign edit control updates the train region values so that the values on this tab always match the shape of the train region in the Current.TrainImage buffer.
Some of the parameters on this tab, such as **Rotation and Skew**, can be specified in degrees (default) or radians. The underlying tool keeps the values in radians but the edit control converts them to degrees when appropriate.

Train Region Features

Feature	Description
TrainRegionMode	Defines the bounding box for the region. ▪ **Pixel Aligned Bounding Box** encloses the defined region within a rectangle. This means that the portions of the image that are outside the defined region but within the bounding rectangle are included. ▪ **Pixel Aligned Bounding Box Adjust Mask** encloses the defined region within a rectangle, but masks out the image portions that are outside the region but inside the enclosing rectangle. The result is that the defined region is closer to what you specify. If you are training a pattern from a collection of shape models, you must use a Pixel Aligned Bounding Box, as a **Pixel Aligned Bounding Box Adjust Mask** region is not supported for shape training.

Feature	Description
TrainRegion	Select the shape of the input region. Selecting **None=Use entire image** means that the tool uses the entire input image. A PMAlign tool supports the following input region shapes: • CogCircle • CogEllipse • CogPolygon • CogRectangle • CogRectangleAffine • CogCircularAnnulusSection • CogEllipticalAnnulusSection The set of region-defining parameters that appear depend on the region shape you use. For more information on using a polygon as an input region, see the topic **Using Polygon Input Regions**.
SelectedSpaceName	The coordinate space in which the training region is interpreted. For information, see **Coordinate Space Names**. The selected space name of the training region is ignored when training from shape models and a transform.
Select Mode	Available when **Region Shape** is **cogRectangle** or **cogRectangleAffine**. Selects the set of parameters that define the rectangle. If **cogRectangleAffine** is chosen, note that the angles of rotation and skew can be specified in degrees or radians, although the underlying tool keeps the measurements in radians.
FitToImage	Centers the train region within Current.TrainImage.

Train Origin Features

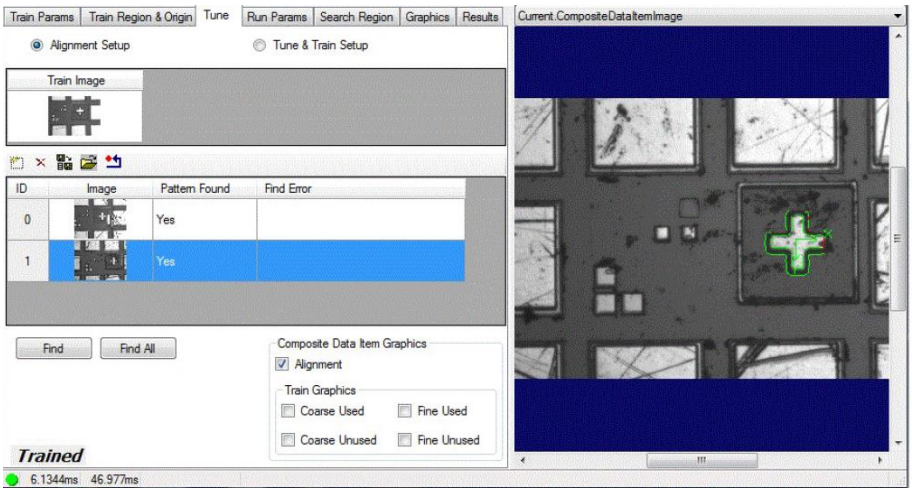
Feature	Description
Origin X **Origin Y** **Screen X** **Len** **Screen Aspect** **Rotation** **Skew**	Values that define the location and orientation of the train region's origin. These values will change if you modify the origin graphically. Note that the angles of rotation and skew can be specified in degrees or radians. Equivalent to the **Origin** property.
Center Origin	Sets the train region's origin at the center of the train region.

8.2.1.5 Tune Tab

The **Tune** tab lets you create and configure the **CogPMAlignComposite** object, which manages composite training for this pattern. The controls in this tab are used for both the **Alignment and Tuning** phases of composite training. You switch between the two phases by clicking the **Alignment Setup and Tune &Train Setup** option buttons at the top of the tab. The controls are different depending on which phase is selected.

NOTE
For information on AutoTune and composite model training, please see the topic **PatMax AutoTune**.

8.2.1.6 Tune Tab: Alignment Phase

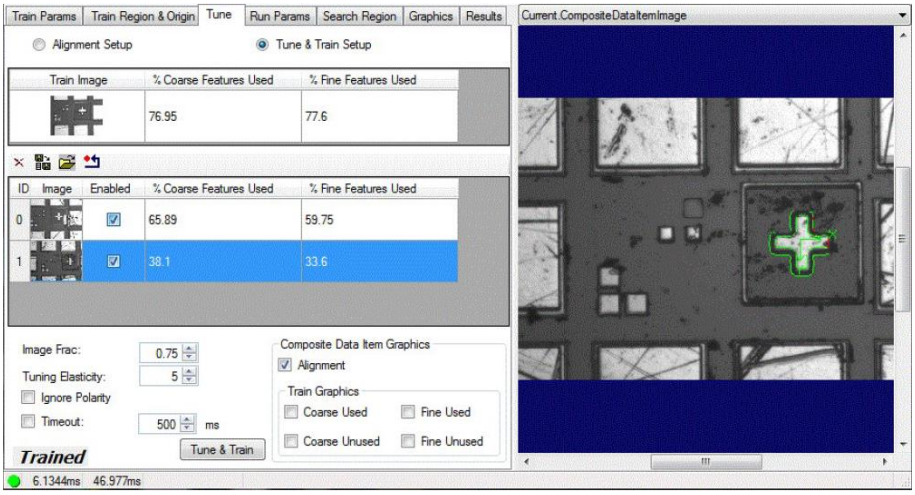


During the alignment setup phase of tuning, you use the controls on the **Tune** tab to manage the collection of **Images** and alignment **Poses** stored within this pattern's **CogPMAlignComposite** object's **CompositeData** collection.

Feature	Description
Image List Buttons 	Use the image list buttons to manage the collection of images and alignment poses used for tuning. In addition to adding and removing images (the first two buttons), the third and fourth buttons allow you to save and load tuning data collections to and from files.
Find and **Find All** buttons.	Clicking **Find** attempts to locate the trained pattern in the selected image; **Find All** attempts to locate the trained pattern in all the images.

Feature	Description
Composite Data Item Graphics selectors.	During the initial alignment phase, check the **Alignment** check box to view the alignment graphics for the selected image. If you have tuned the pattern, then the tuning graphics are available as well. NOTE You must select **Current.CompositeDataItemImage** in the in the drop-down box of the image display associated with an image in the image collection list.

8.2.1.7 Tune Tab: Tuning Phase

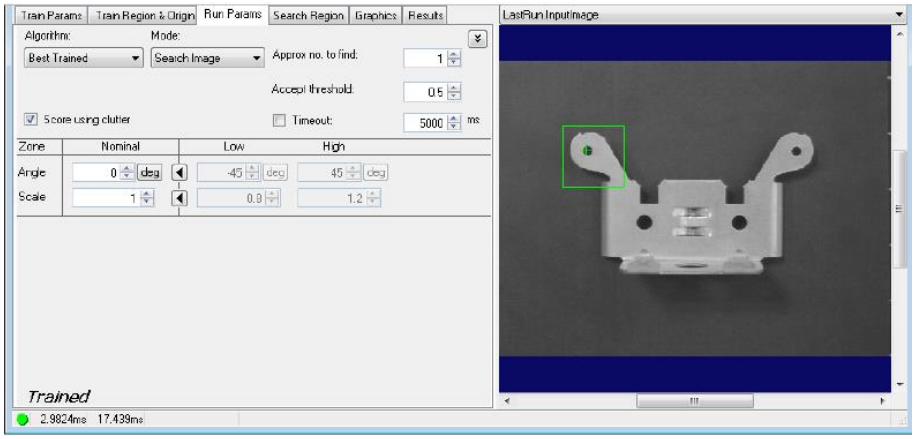


During the tuning phase, you use the controls on the **Tune** tab to configure the tuning parameters, to select which of the images to use for tuning, and to review the tuning results.

Feature	Description
Image List Buttons	The add button is not available during the tuning phase, but you can still save and load the collection of tuning data.
Enabled check box.	Check the Enabled check box for an image to use features from that image when tuning the pattern. Changing the Enabled state of an image untrains the pattern. **NOTE** If the **Enabled** check box is checked for any image, then clicking the **Tune & Train** button or the **Train** in the **Train Params** tab will retrain the pattern using features from all the enabled images (not just the **TrainImage** for the pattern).

Feature	Description
PercentCoarseFeaturesUsed **PercentFineFeaturesUsed**	These columns display the percentage of coarse and fine features used in the final pattern from each enabled image. You can display both the used and discarded coarse and fine features by checking the appropriate check boxes in the **Composite Data Item Graphics** section.
ImageFraction	Use the **Image Frac** control to determine what fraction of the tuning images need to contain a given feature for it to be included in the tuned pattern.
TuningElasticity	Use the **Tuning Elasticity** to set the alignment tolerance (in pixels) for image features. Features must be within the specified distance to be counted as the same feature.
IgnorePolarity check box.	Check this option to ignore feature polarity in all tuning images. The same setting for ignore polarity is used for all tuning images, including the **TrainImage** for the pattern.  **NOTE** If you check **Ignore Polarity,** and perform composite training, you must also check **Ignore Polarity** on the **Train Params** tab. Failure to do so will result in a run-time error.
Timeout	Sets the timeout for the overall tuning and training operation.
Tune & Train	Initiates the tuning and training operation.
Composite Data Item Graphics selectors.	During the tuning phase, use the check boxes in the **Train Graphics** frame to view the features used and discarded during tuning. You can also view the alignment graphics.  **NOTE** You must select **Current.CompositeDataItemImage** in the in the drop-down box of the image display associated with an image in the image collection list.

8.2.1.8 Run Params Tab



Use the **Run Params** tab to specify how to perform the pattern search. These parameters include the run algorithm to use, thresholds and limits, and the amount of rotating and scaling allowable during the pattern search.

Clicking the **Advanced Feature** button  on the far right of the tab displays less commonly used features.

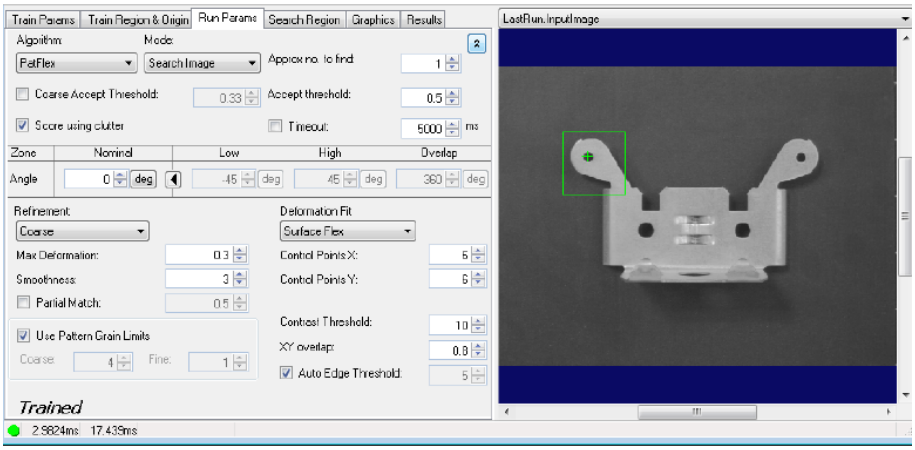
Feature	Description
RunAlgorithm	Selects the search algorithm. The Patmax algorithm is more accurate than Patquick and can return additional score information, but requires more processing time. You can also specify **BestTrained**, in which case the tool will run using the highest-accuracy algorithm for which it has been trained. If you specified **PatMax - High Sensitivity** for the trained algorithm, specify either **PatMax** or **Best Trained**. In either case, PatMax high sensitivity mode is used. If you specified **Perspective PatMax** for the trained algorithm, specify either **Perspective PatMax** or **Best Trained**.

Feature	Description
RunMode	By default, the PMAlign tool uses **Search Image** mode to search the entire image for the coarse features that indicate the presence of the pattern it is trained to locate. Switch to **Refine Start Pose** mode to give this PMAlign tool a specific StartPose, which is a two-dimensional linear transformation that defines the starting location of the search. The StartPose typically comes from another vision tool that has already executed and generated results about known features in the image.  **NOTE** To use **Refine Start Pose** mode you must add an input terminal to expose the StartPose property for this PMAlign tool.
CoarseAcceptThreshold	The PMAlign tool uses a default Coarse Accept threshold to refine early search results based on the coarse features of the pattern. You can enable the **Coarse Accept threshold** and specify a different value, forcing the tool to consider more (lower value) or fewer (higher value) potential matches in each run-time image. The threshold value cannot exceed the current value for **Accept threshold.** The edit control will correct the value for **Accept threshold** automatically if you set a value for **Coarse Accept threshold** higher than **Accept threshold**.
ScoreUsingClutter	If checked, then the PatMax algorithm considers extraneous or clutter features when computing the score of a pattern instance. Considering clutter features usually results in lower scores. Available for the PatMax algorithm only. You may need to disable this feature if the trained pattern consists of shape models.
ApproximateNumberToFind	Specifies the number of results to look for.
AcceptThreshold	Specifies the acceptance threshold for the result score. Only results with scores greater than or equal to this value are accepted.
TimeoutEnabled	If checked, then the timeout value limits the execution time of the PMAlign inspection.

Feature	Description
ZoneAngle	Specifies the angle of rotation that is allowable when PMAlign performs a pattern search. You can specify a nominal value that the PMAlign results must match exactly or you can click the  arrow to specify a range of values for the results.
ZoneScale	Specifies the scale value to be used when PMAlign performs a pattern search. You can specify a nominal value that a searched pattern must match exactly or you can click the  arrow to specify a range of values for the results.
ZoneScaleX	Specifies the scale value in the X direction to be used when PMAlign performs a pattern search. You can specify a nominal value that a searched pattern must match exactly or you can click the  arrow to specify a range of values for the results.
ZoneScaleY	Specifies the scale value in the Y direction to be used when PMAlign performs a pattern search. You can specify a nominal value that a searched pattern must match exactly or you can click the  arrow to specify a range of values for the results.
PartialMatchEnabled	Only available for the PatFlex algorithm, PatFlex will find results that match only a fraction of the full pattern with a score better than the AcceptThreshold.
PartialMatchCoverageThreshold	Only available for the PatFlex algorithm, this specifies the minimum fraction of the pattern that must be matched in a valid PatFlex result.
GrainLimitsUsePattern	If checked, then the granularity limits determined by the training pattern are used. If not checked, then the **GrainLimitCoarse** and **GrainLimitFine** values specified are in effect.
ContrastThreshold	Defines the minimum acceptable contrast for a pattern instance. Only pattern instances where the average difference in pixel values across all feature boundaries exceeds the contrast threshold are considered by PMAlign.
XYOverlap	Result candidates overlap in area if the percentage of area overlap is greater than XYOverlap. PMAlign discards the weaker pattern instance when two pattern instances overlap for all degrees of freedom as well as area by the specified overlap percentage.

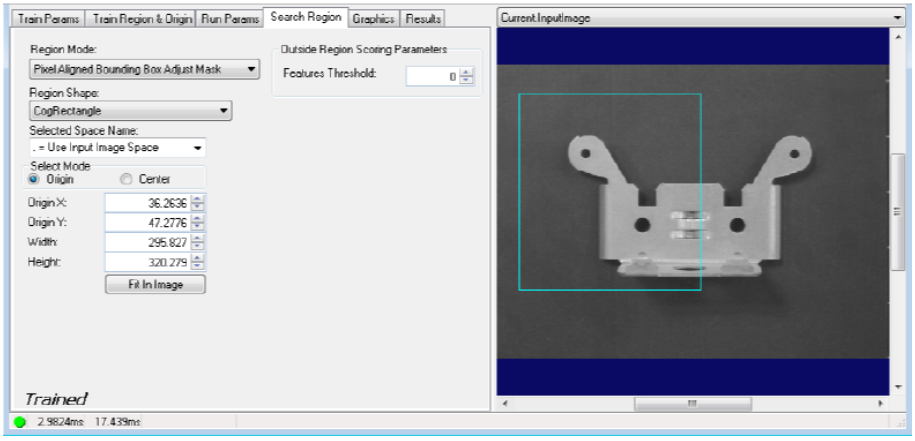
Feature	Description
AutoEdgeThresholdEnabled	Disable the automatic value if you want to set a different threshold for the absolute minimum value of edge magnitude, below which the edge direction will be randomized.

If you use the PatFlex search algorithm, the Run Params tab presents several different parameters, as shown:



Feature	Description
Refinement	The amount of refinement done on the deformation transform. A value of 'None' may contain some error. Higher levels of refinement will be more accurate at the expense of time.
MaxDeformationRate	The maximum amount of deformation expected in run-time image.
Smoothness	The smoothness value used in fitting the deformation transform to the runtime input image.
DeformationFit	Controls the type of fit PatFlex will use to model deformation in the runtime input image.
ControlPointsX	The number of control points in the X direction. Increasing the number of control points allows the transform to better match patterns with areas of heavily local deformation (i.e., sharper features).
ControlPointsY	The number of control points in the Y direction. Increasing the number of control points allows the transform to better match patterns with areas of heavy local deformation (i.e., sharper features).

8.2.1.9 Search Region Tab

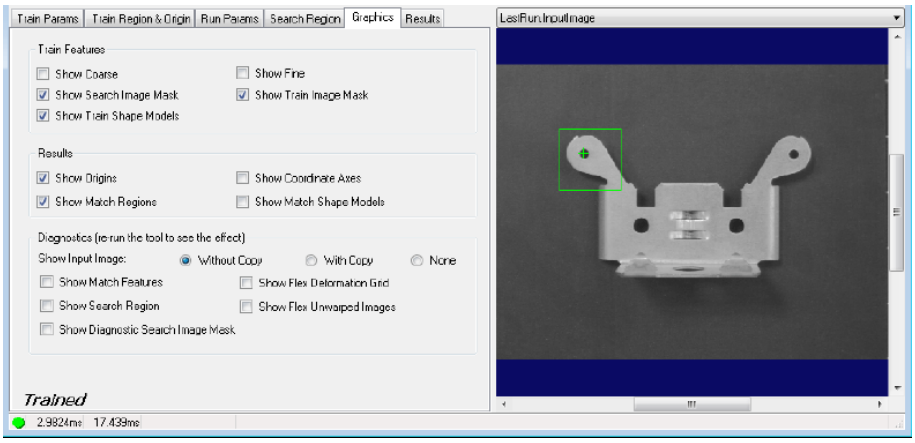


Use the Search Region tab to define the **SearchRegionMode**, the area of the **InputImage** buffer to which the pattern search is constrained. The search region appears with a blue border in the Current.InputImage, and you can graphically define the search region in this buffer. It may be easier to specify the search region graphically, then use this tab to fine tune the search region parameters. When you resize the search region in the InputImage, the values on this tab change; likewise, changing the parameter values causes the search region to alter its size and shape.

Feature	Description
SearchRegionMode	Defines the bounding box for the region. ▪ Pixel **Aligned Bounding Box** encloses the defined region within a rectangle. This means that the portions of the image that are outside the defined region but within the bounding rectangle are included. ▪ • **Pixel Aligned Bounding Box Adjust** Mask encloses the defined region within a rectangle, but masks out the image portions that are outside the region but inside the enclosing rectangle. The result is that the defined region is closer to what you specify.
Outside Region Scoring Parameters	Choose a **Features Threshold** value that specifies the percentage of features in the trained pattern that can be outside the search region without penalizing the score. The default value of 0 means that the PMAlign tool expects all the features of the trained pattern to be located within the search region, while a value of 0.3 means that up to 30% of the trained pattern can be outside the search region without affecting the final score. Use this feature when you want to allow some portion of the features in the trained pattern to exist outside the search region without affecting the overall score given to the features the search region still contains.

Feature	Description
SearchRegion	The shape of the search region. Selecting "None=Use entire image" means that the entire Current.InputImage becomes the search region. The set of region-defining parameters depends on the selected **Region Shape**.
SelectedSpaceName	The coordinate space in which the search region is interpreted. For more information, see **Coordinate Space Names**.
Select Mode	Available when **Region Shape** is cogRectangle or cogRectangleAffine. Selects the set of parameters that define the rectangle. If cogRectangleAffine is chosen, note that the angles of rotation and skew can be specified in degrees or radians, although the underlying tool keeps the measurements in radians.
FitToImage	Centers the search region within Current.InputImage.

8.2.1.10 Graphics Tab

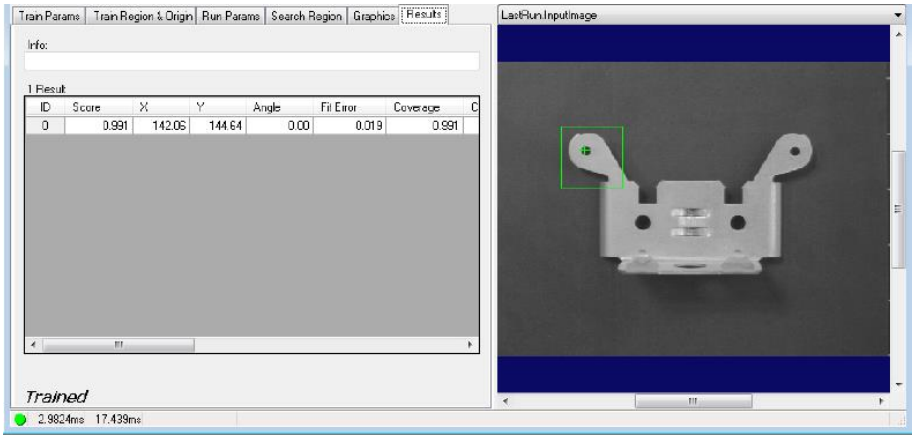


Use the Graphics tab to choose the results graphics that are displayed in the LastRun.InputImage display.

Feature	Description
Train feature display	You can show these features in the Current.TrainImage buffer, which contains the trained pattern: ▪ Show features that were trained at the coarse granularity limit. Features are displayed in yellow. Equivalent to **CreateGraphicsCoarse** method. ▪ Show features that were trained at the fine granularity limit. Features are displayed in green. Equivalent to **CreateGraphicsFine** method. The preceding features appear only if the pattern is trained successfully. ▪ Show the shape models. ▪ Show Search Image Mask shows a graphic representing the run-time mask, if you supplied one. The graphic is shown on the Current.InputImage display. ▪ • Show Train Image Mask shows a graphic representing the training-time mask, if you supplied one. The graphic is shown on the Current.TrainImage display.
Result graphics display	You can show these features in the LastRun.InputImage buffer, which contains the image that the PMAlign tool last searched, and the results of that search. Uses the **CreateResultGraphics** method to generate these results. ▪ Train pattern's origin ▪ Train pattern's coordinate axes. ▪ The match region. ▪ The shape models of found features
Diagnostics display	Displays the following features in the LastRun.InputImage buffer. Uses the **CreateResultGraphics** method to generate these results. ▪ Show match features displays the found features that were used to match the trained pattern. Not all trained features may be present. For each matched feature, the color of the graphic indicates the quality of the match (red indicates poor matches, yellow fair matches, and green good matches). ▪ Show search region displays the search region as defined in Current.InputImage buffer (or in the Search Region tab). ▪ Show Diagnostic Search Image Mask displays the run-time mask, if one was specified. The Show Input Image option buttons let you specify whether a reference to the input image or a deep copy of the input image is displayed for the LastRun.InputImage. You can also specify that no image be displayed. The Show Flex Deformation Grid check box will display a grid that represents the computed deformation transformation. You must

Feature	Description
	specify PatFlex for the run-time algorithm, and you must re-run the tool after checking this box to see the result. The Show Flex Unwarped Images check box will display an unwarped version of the input image. The unwarping is most accurate within the region that corresponds to the trained pattern. You must specify PatFlex for the run-time algorithm, and you must re-run the tool after checking this box to see the result.

8.2.1.11 Results Tab



The **Results** tab displays the results of the most recent pattern searches. This corresponds to the **CogPMAlignResult** interface. Use the slider control below the results grid to display the complete set of results.

Feature	Description
GetInfoStrings	Displays the run-time diagnostic message text string(s) for the last PMAlign result. If there are no messages then the text box will be empty.
Results Grid	Displays the following information for each result. ▪ **Score** for this result. The range is 0.0 to 1.0 with higher values indicating a closer match. ▪ **FitError**, a measure of how closely the found pattern matches the features of the trained pattern, without regard to missing features. The range is zero (perfect fit) to infinity (poor fit). Used only for the PatMax algorithm. ▪ **Coverage**, the percentage of the features in the trained pattern that are found in the search result. Range is 0.0 to 1.0. Used only for the PatMax algorithm.

Feature	Description
	▪ **Clutter**, the number of extraneous features present in the result divided by the number of features in the trained pattern. Range from 0.0 to infinity. Used only for the PatMax algorithm. ▪ **MaxCoarseAcceptThreshold**, the maximum coarse accept threshold that would allow this result to be found. The coordinates, angle and scaling factors are retrieved with the **GetPose** method. ▪ X, Y are the coordinates of the pattern's origin as measured in the selected space of the InputImage. ▪ Angle is the rotation of pattern space as measured in the selected space of the InputImage. ▪ Scale, ScaleX and ScaleY are the scaling factors between pattern space and the selected space of the InputImage.

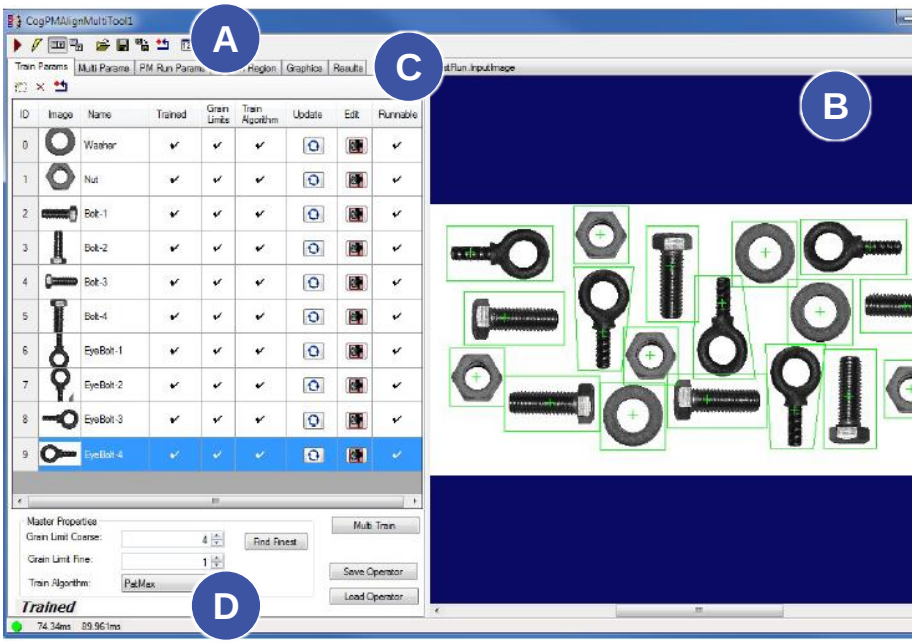
8.2.2 PMAlignMulti Tool Edit Control

The following is an extract from the Cognex User Manual: for further and complete details, please refer to the whole Cognex User Manual.

This topic contains the following sections.

- Control Buttons
- PMAlignMulti Edit Control Buffers
- Train Params Tab
- Multi Params Tab
- PM Run Params Tab
- Search Region Tab
- Graphics Tab
- Results Tab

The PMAlignMulti edit control provides a graphical user interface to the **CogPMAlignMultiTool** and its components, which allows you to locate instances of multiple patterns in a single runtime image. You can configure the **CogPMAlignMultiTool** by adding patterns, specifying a region of interest, and selecting runtime parameters. For more information, see the **Multi-Model PatMax Theory** and **Using a Multi-Model PatMax Tool** topics. The following figure shows an example PMAlignMulti edit control:



The PMAlignMulti edit control includes the following components:

- A row of control buttons at the top left (**A**).
- A tool display window (**B**) that can display the PMAlignMulti tool image buffers: Current.PatternTrainImage, Current.InputImage, and LastRun.InputImage. These buffers contain the trained patterns, the search image in which the PMAlignMulti tool searches for the pattern, and the same image with the results of the search. Right click the tool display to bring up menu options that include zooming in or out of the image or showing a pixel or subpixel grid.
- A set of tabs organized by function (**C**). These include the patterns you added to the tool and (PMAlign) training parameters for the patterns, Multi-Model specific parameters, parameter settings for the search region of interest, display settings for the tool displays, and search results. Pressing the Control + Tab keys scrolls through the set of tabs.
- A status bar at the bottom left of the control (**D**). A green circle indicates that the tool ran successfully; red means the tool ran unsuccessfully. The status bar also displays the time to run the tool and any error codes or messages. The first time that the status bar displays is the raw tool execution time. The second includes the time needed to update the edit control.

8.2.2.1 Control buttons

Button	Description	Function
	Run	Runs the PMAlignMulti tool. You must have at least one runnable pattern in the queue, an input image, and specified run parameters. PMAlignMulti searches for the patterns in the queue in the input image. You may constrain the pattern search to a search region within the input image.
	Electric mode	Toggles electric mode. When selected, the PMAlignMulti tool runs automatically if certain parameters have changed. These parameters are indicated by electric bolt icons that appear when the tool is in electric mode.
	Local image display	Opens the local tool display window, which can display the Current.InputImage, Current.PatternTrainImage, or LastRun.InputImage buffer.
	Floating image display	Opens one or more floating tool display windows. You can display the Current.InputImage, Current.PatternTrainImage, or LastRun.InputImage buffer. Unlike the local tool display, you can resize or move the position of the floating tool display window.
	Open	Loads a VisionPro persistence (.vpp) file, which contains a set of saved properties for this vision tool object type. Loading a persistence file for another object type throws an error and the load is unsuccessful. For more information about VisionPro persistence features, see the topic Persistence in VisionPro.
	Save	Saves the current properties of the underlying tool to a VisionPro persistence file. You have the option to save either the entire tool or the tool without its images or results.
	Save As	Saves the current properties of the underlying tool to a new VisionPro persistence file.
	Reset	Resets the underlying tool to a default state.
	Results	Opens a new, separate results window, allowing you to view run results without turning to the Results tab.
	Show ToolTips	Enable or disable the display of tooltips for individual items in the edit control.
	Help	Access the VisionPro Software Documentation.

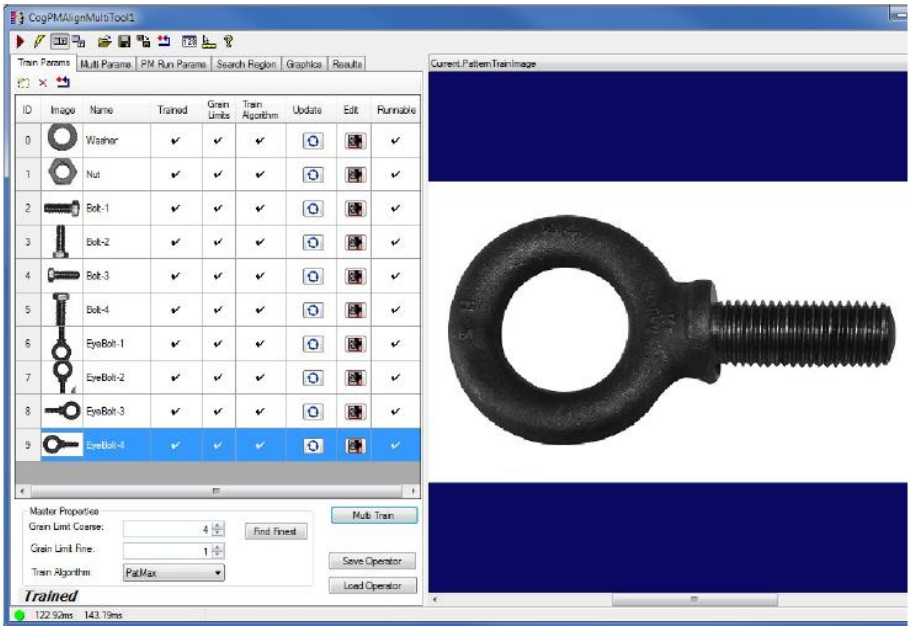
8.2.2.2 PMAlignMulti Edit Control Buffers

The PMAlignMulti edit control has three image buffers. One of the buffers uses the underlying PMAlignMulti tool's InputImage; the second buffer displays the training image for the selected pattern; the third buffer displays the last input image that the PMAlignMulti tool ran on and the results of that search. All three buffers can be shown in both the local and floating tool display windows.

- The Current.InputImage provides the input images to the PMAlignMulti tool.
- The Current.PatternTrainImage contains the training image for the pattern you select on the Train Params tab.
- The LastRun.InputImage buffer displays the last image on which the tool most recently ran. Use the Graphics tab to highlight the search area and the results of the search.

When you run the PMAlignMulti tool, the tool searches the Current.InputImage for the runnable patterns, and shows the results of this search on the Results tab.

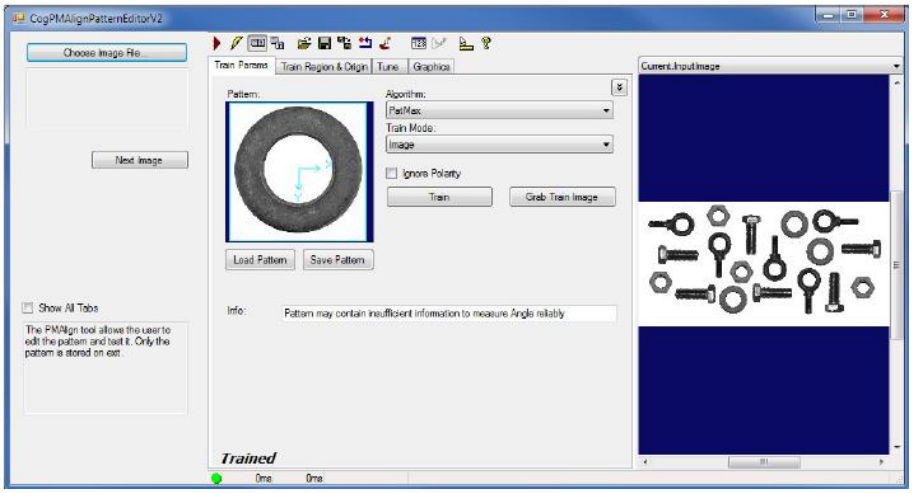
8.2.2.3 Train Params Tab



Use the **Train Params** tab to add and configure patterns for the Multi-Model, set training parameters for the Multi-Model, and to train the Multi-Model.

Feature	Description
Patterns Grid	Allows you to add, delete, or modify patterns (**Pattern**) in the PMAlignMulti tool. An ID number specifies the index of the pattern in the collection, the training image of the pattern is displayed as a thumbnail, and you can specify a custom name for the pattern (which does not have to be unique but it is recommended to be unique). Check marks mark whether an added pattern is trained, its granularity limits match those of the Multi-Model, its training algorithm matches that of the Multi-Model, and it is runnable. If the pattern is runnable, it can be used in the queue, that is, the pattern has been trained in the Multi-Model tool and can be located at run time. Use the Update button to quickly apply to a pattern the granularity and algorithm settings specified in the Master Properties for the Multi-Model. After copying the Multi-Model operator properties to the pattern, it retrains the pattern. Note that the update operation will also set the **GrainLimitAutoSelect** to false and will cause the PMAlignMulti tool to become untrained. Click the Edit button to configure and train the associated pattern in the **pattern editor**.
GrainLimitCoarse	For the Multi-Model, it specifies the coarse grain limit that must be used by all the patterns in the operator collection.
GrainLimitFine	For the Multi-Model, it specifies the fine grain limit that must be used by all the patterns in the operator collection.
TrainAlgorithm	For the Multi-Model, it specifies the train algorithm that must be used by all the patterns in the operator collection.
Find Finest	If the grain limits of some patterns do not match, it is recommended that the Multi-Model tool use the finest grain limits. The Find Finest button is provided to look through all of the patterns and update the Multi-Model operator properties with the finest values.
Multi Train button	Allows you to train the Multi-Model tool. The trained state is shown in bold on all tabs of the control. The patterns that have been trained will have a check mark in the Runnable column. If you modify the grain limits or the train algorithm, the tool will become untrained and the patterns will have an x in the Runnable column.
Save/Load Operator buttons	Allow you to save the Multi-Model operator and restore it. Note that in addition to the properties on the Train Params tab, the Search Order Queue and the **ResultStatisticWindowLength** are also saved and restored because they are properties of the **CogPMAlignMulti** class. If the Multi-Model was saved trained, it is restored trained.

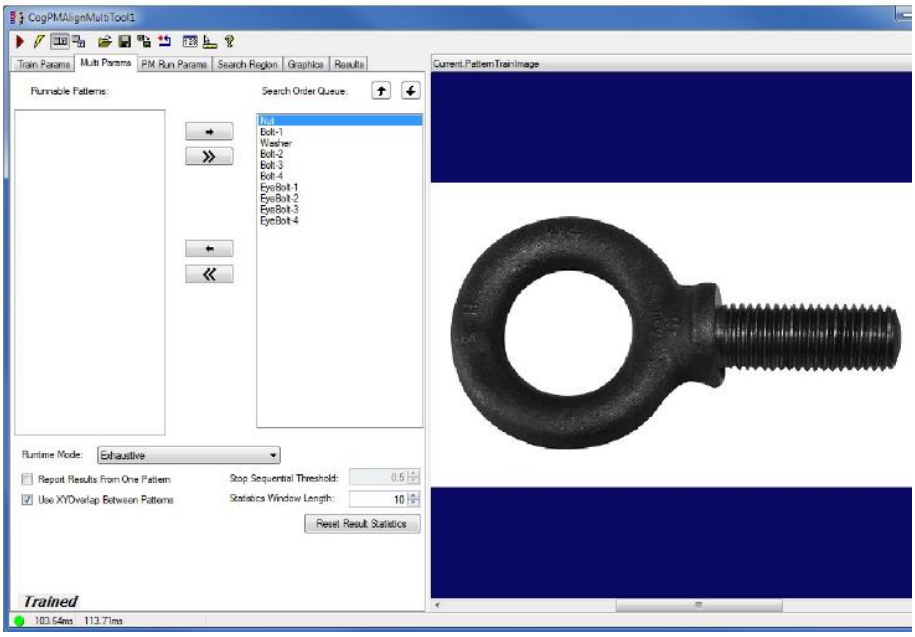
8.2.2.4 Pattern Editor



The pattern editor GUI is similar to the PMAlign tool's GUI described in the **PMAlign Edit Control** topic. Click the **Choose Image** File button in the pattern editor to choose the image file to be used for training the pattern. Click the **Next Image** button to cycle through the images in a cdb or an idb file.

Note that by default three of the CogPMAlignEditV2 tabs are not visible in the pattern editor, including the Run Params tab. You specify the run-time parameters for all patterns on the PM Run Params tab of the main CogPMAlignMultiTool edit control. You can show the three hidden tabs by selecting the **Show All Tabs** check box. You can use the runtime parameters you specify on the hidden tabs to test the pattern. Note that no changes to the run-time parameters you specify for the pattern on the hidden tabs are saved, they are not propagated back to the Multi-Model tool.

8.2.2.5 Multi Params Tab

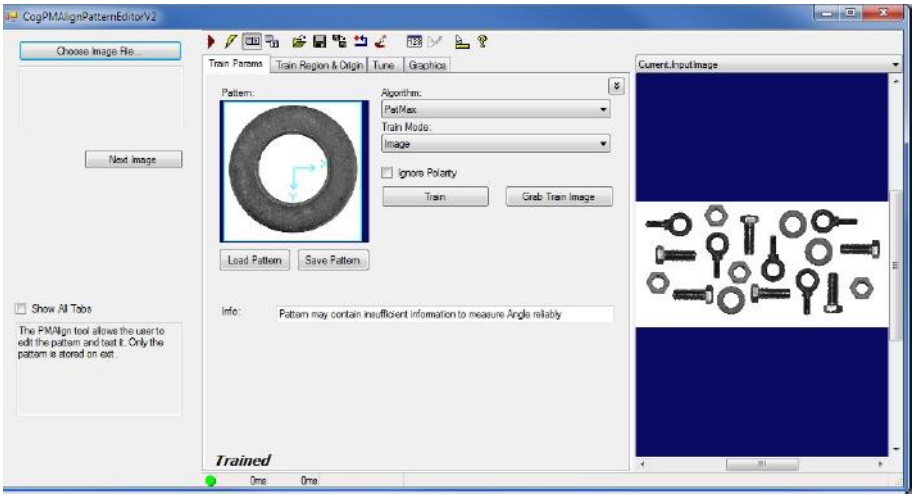


Use the **Multi Params** tab to add and remove the patterns from the Search Order Queue, to re-order the patterns in the Search Order Queue, and to adjust the Multi-Model parameters. For more information on the Multi-Model run-time parameters, see the **Run-Time Parameters Detailed** section.

Feature	Description
Patterns Grid	Allows you to add, delete, or modify patterns (**Pattern**) in the PMAlignMulti tool. An ID number specifies the index of the pattern in the collection, the training image of the pattern is displayed as a thumbnail, and you can specify a custom name for the pattern (which does not have to be unique but it is recommended to be unique). Check marks mark whether an added pattern is trained, its granularity limits match those of the Multi-Model, its training algorithm matches that of the Multi-Model, and it is runnable. If the pattern is runnable, it can be used in the queue, that is, the pattern has been trained in the Multi-Model tool and can be located at run time. Use the Update button to quickly apply to a pattern the granularity and algorithm settings specified in the Master Properties for the Multi-Model. After copying the Multi-Model operator properties to the pattern, it retrains the pattern. Note that the update operation will also set the **GrainLimitAutoSelect** to false and will cause the PMAlignMulti tool to become untrained. Click the Edit button to configure and train the associated pattern in the **pattern editor**.
GrainLimitCoarse	For the Multi-Model, it specifies the coarse grain limit that must be used by all the patterns in the operator collection.
GrainLimitFine	For the Multi-Model, it specifies the fine grain limit that must be used by all the patterns in the operator collection.
TrainAlgorithm	For the Multi-Model, it specifies the train algorithm that must be used by all the patterns in the operator collection.
Find Finest	If the grain limits of some patterns do not match, it is recommended that the Multi-Model tool use the finest grain limits. The Find Finest button is provided to look through all of the patterns and update the Multi-Model operator properties with the finest values.
Multi Train button	Allows you to train the Multi-Model tool. The trained state is shown in bold on all tabs of the control. The patterns that have been trained will have a check mark in the Runnable column. If you modify the grain limits or the train algorithm, the tool will become untrained and the patterns will have an x in the Runnable column.

Feature	Description
Save/Load Operator buttons	Allow you to save the Multi-Model operator and restore it. Note that in addition to the properties on the Train Params tab, the Search Order Queue and the **ResultStatisticWindowLength** are also saved and restored because they are properties of the **CogPMAlignMulti** class. If the Multi-Model was saved trained, it is restored trained.

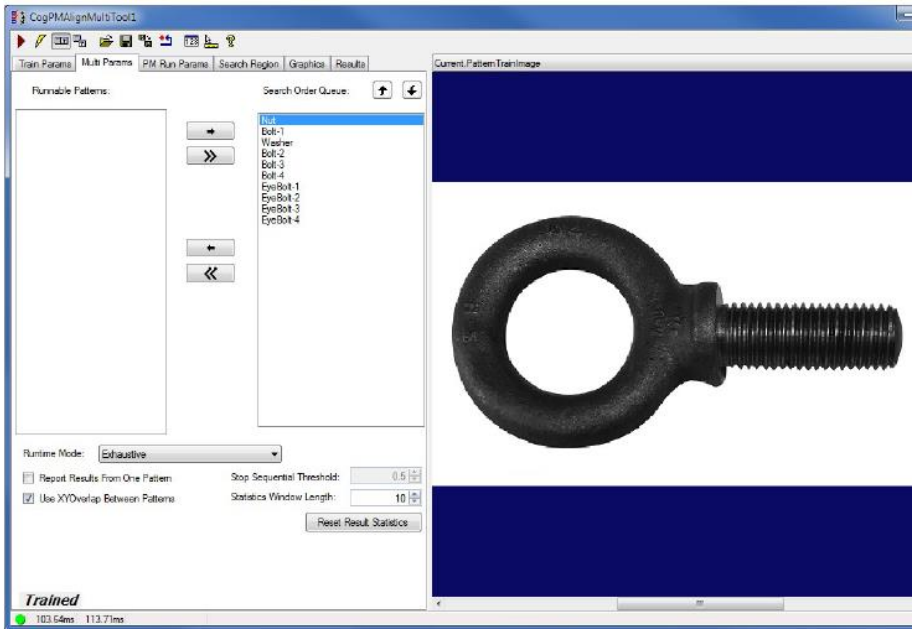
8.2.2.6 Pattern Editor



The pattern editor GUI is similar to the PMAlign tool's GUI described in the **PMAlign Edit Control** topic. Click the **Choose Image File** button in the pattern editor to choose the image file to be used for training the pattern. Click the Next Image button to cycle through the images in a cdb or an idb file.

Note that by default three of the CogPMAlignEditV2 tabs are not visible in the pattern editor, including the Run Params tab. You specify the run-time parameters for all patterns on the PM Run Params tab of the main CogPMAlignMultiTool edit control. You can show the three hidden tabs by selecting the **Show All Tabs** check box. You can use the runtime parameters you specify on the hidden tabs to test the pattern. Note that no changes to the run-time parameters you specify for the pattern on the hidden tabs are saved, they are not propagated back to the Multi-Model tool.

8.2.2.7 Multi Params Tab

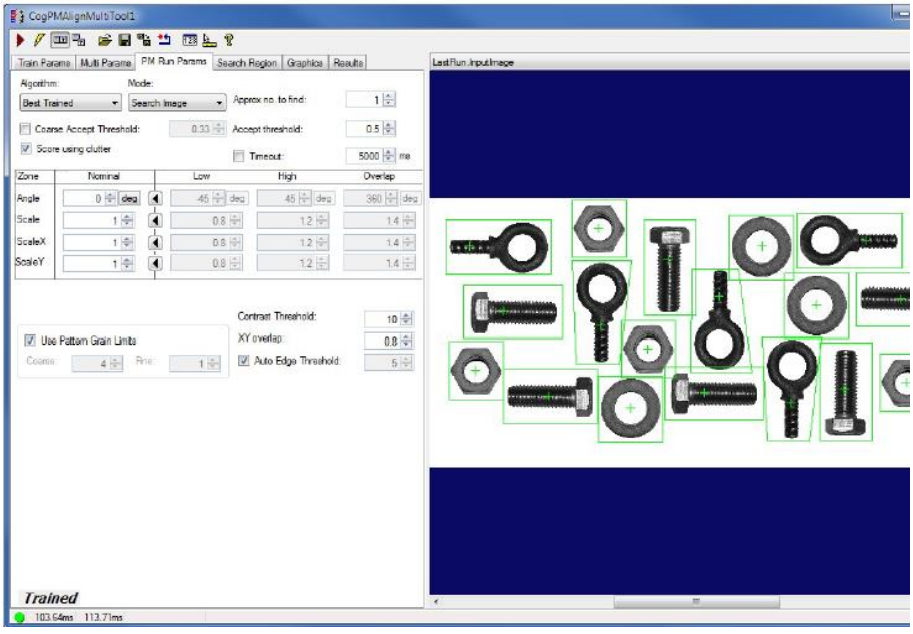


Use the **Multi Params** tab to add and remove the patterns from the Search Order Queue, to re-order the patterns in the Search Order Queue, and to adjust the Multi-Model parameters. For more information on the Multi-Model run-time parameters, see the **Run-Time Parameters Detailed** section.

Feature	Description
Runnable Patterns	The list of patterns that are runnable and Multi-Model trained.
Search Order Queue	The list of patterns that are in the Search Order Queue. The patterns can be ordered using the up and down arrow buttons. The patterns can be added to the queue or removed from the queue using the following buttons: ▪ The right arrow button adds the selected pattern to the queue. ▪ The left arrow button removes the selected pattern from the queue. ▪ The right double arrows button adds all the runnable patterns to the queue. ▪ The left double arrows button removes all the patterns from the queue.
RuntimeMode	The run-time mode. For more information, see the **Run-Time Parameters Detailed** section.
ReportResultsFromOnePatternOnly	A Boolean to choose whether the results returned are from a single pattern. For more information, see the **Run-Time Parameters Detailed** section.

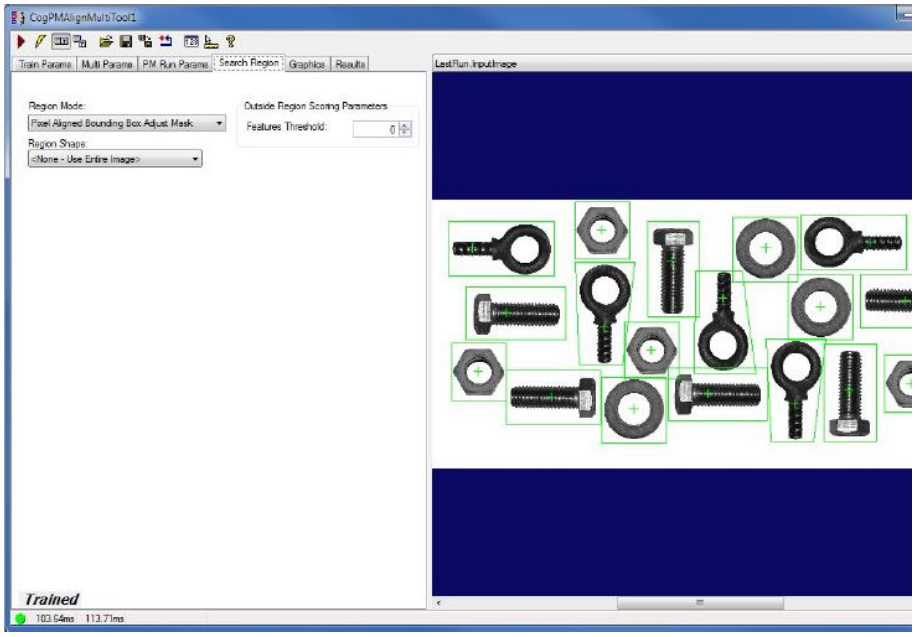
Feature	Description
UseXYOverlapBetweenPatterns	A Boolean whether to use the **XYOverlap** of **CogPMAlignRunParams** between patterns. If true, the overlap value from the **CogPMAlignRunParams** is used to discard overlapping instances of different patterns. Otherwise, the overlap value is only used to discard overlapping instances of a single pattern.
StopSequentialThreshold	This property is only used in Sequential search as one of the stopping condition criteria. Sequential searches will stop when the total number of found instances (with a score at or above this threshold) from a single pattern meets or exceeds the **ApproximateNumberToFind**. It does not alter the PatMax search algorithm. The value must be greater than or equal to the **AcceptThreshold**, otherwise a run-time error will be reported. For more information, see the **Run-Time Parameters Detailed** section.
Statistics Window Length	**ResultStatisticWindowLength** is the window over which statistic data is collected. The data is used in SequentialMostSuccessful runtime mode to re-order the patterns in the queue.
ResetResultStatistics	Resets the result statistics window. It only affects the SequentialMostSuccessful mode.

8.2.2.8 PM Run Params Tab



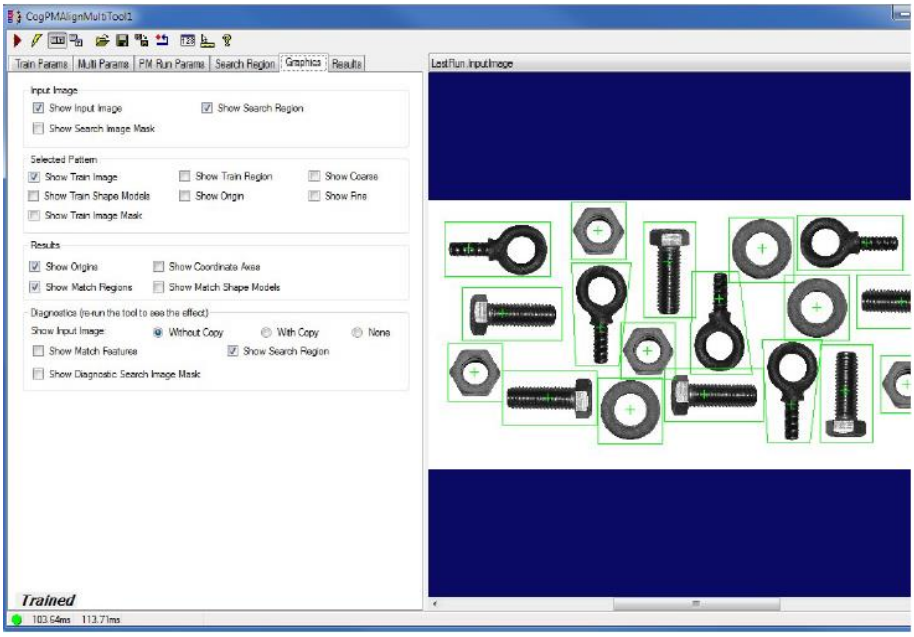
The **PM Run Params** tab is based on the **Run Params tab** found in the **CogPMAlignEditV2** control. Some of the items are not present because the Multi-Model tool supports only the PatMax and PatQuick algorithms.

8.2.2.9 Search Region Tab



The **Search Region** tab is a copy of the **Search Region tab** found in the **CogPMAlignEditV2** control.

8.2.2.10 Graphics Tab

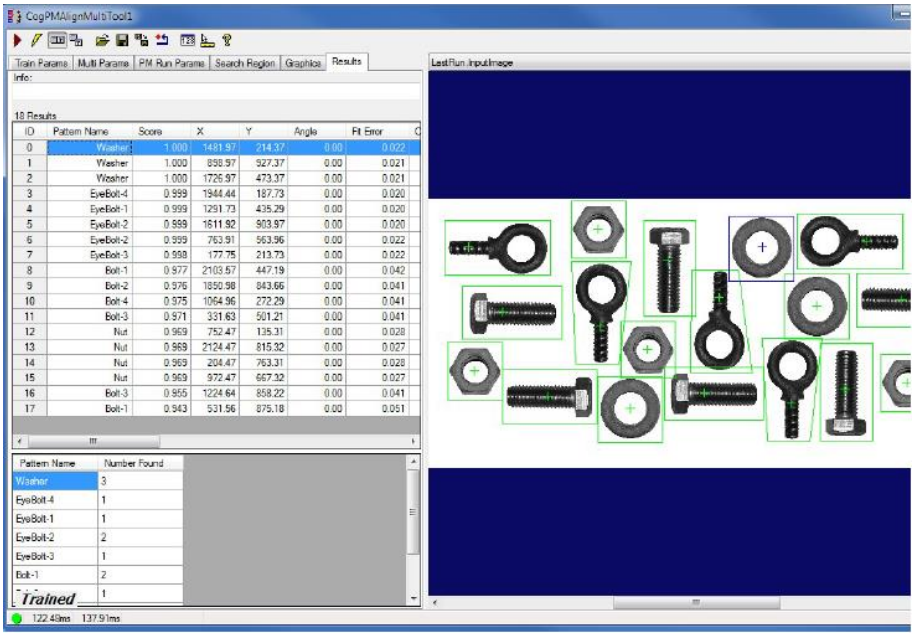


The **Graphics** tab is similar to but slightly different from the **Graphics tab** found in the CogPMAlignEditV2 control. It contains similar options but the Train Features graphics are applied to the selected pattern. The new Current.PatternTrainImage record is added to show the train graphics of the selected pattern.

Feature	Description
Input image display	You can show these features in the Current.InputImage buffer: ▪ The input image in the Current.InputImage buffer. ▪ The search region. ▪ Search Image Mask showing a graphic representing the run-time mask, if you supplied one. The graphic is shown on the Current.InputImage display.
Selected pattern display	You can show these features in the Current.PatternTrainImage buffer: ▪ The training image for the pattern. ▪ The shape models. ▪ Train Image Mask showing a graphic representing the training-time mask, if you supplied one. The graphic is shown on the Current.PatternTrainImage display. ▪ The training region. ▪ Train pattern's origin. ▪ Features that were trained at the coarse granularity limit. Features are displayed in yellow. Equivalent to **CreateGraphicsCoarse** method. ▪ Features that were trained at the fine granularity limit. Features are displayed in green. Equivalent to **CreateGraphicsFine** method.
Results graphics display	You can show these features in the LastRun.InputImage buffer, which contains the image that the PMAlignMulti tool last searched, and the results of that search. Uses the **CreateResultGraphics** method to generate these results. ▪ Train patterns' origins. ▪ Train patterns' coordinate axes. ▪ The match regions. ▪ The shape models of found features.

Feature	Description
Diagnostics display	Displays the following features in the LastRun.InputImage buffer. Uses the **CreateResultGraphics** method to generate these results. ▪ Show match features displays the found features that were used to match the trained pattern. Not all trained features may be present. For each matched feature, the color of the graphic indicates the quality of the match (red indicates poor matches, yellow fair matches, and green good matches). ▪ Show search region displays the search region as defined in Current.InputImage buffer (or in the Search Region tab). ▪ Show Diagnostic Search Image Mask displays the run-time mask, if one was specified. The Show Input Image option buttons let you specify whether a reference to the input image or a deep copy of the input image is displayed for the LastRun.InputImage. You can also specify that no image be displayed.

8.2.2.11 Results Tab



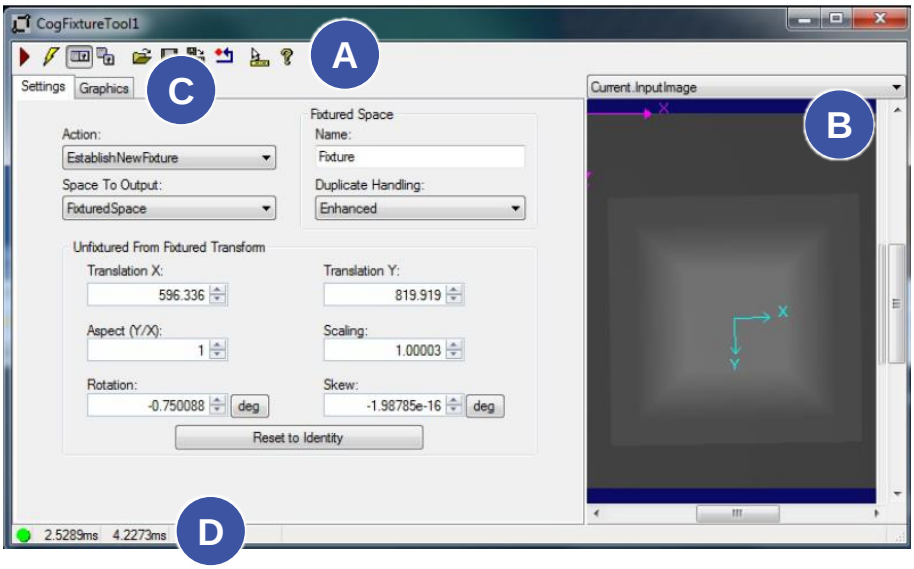
The Results tab is the same as the **Results tab** of the **CogPMAlignEditV2** control but with an extra column to show the name of the pattern and another grid to provide a summary of the results found. This second grid shows the number of instances of the patterns found in the runtime image.

8.2.3 Fixture Edit Control

This topic contains the following sections.

- Control Buttons
- Image Buffers
- Settings Tab
- Graphics Tab

The Fixture edit control provides a graphical user interface (GUI) for the simple **CogFixtureTool** vision tool and its components.



The Fixture tool attaches a fixtured coordinate space to an input image and provides an updated image as output for use by other tools. You must supply a **nonqualified coordinate space name** for the fixtured space and a **CogTransform2D** that defines the fixtured coordinate space relative to the unfixtured space.

The Fixture tool acquires the necessary information to perform the fixturing operation from the input image and run-time parameters you supply. The unfixtured space name is the selected space name in the input image. You obtain the 2D transformation from another vision tool, such as **CogPMAlignTool**. The edit control allows you to edit the various components of this transform before attaching it to the coordinate space tree of the specified image.

A Fixture edit control exposes the following default tool input and outputs for creating data links:

- InputImage
- UnfixturedFromFixturedTransform
- TranslationX
- TranslationY
- Rotation
- OutputImage

Note that although the Fixture edit control exposes the UnfixturedFromFixtured **2D transformation** as a default input, if you want to adjust the translation and rotation components of the fixturing transformation, you should use the specific inputs for translation and rotation. Using the **UnfixturedFromFixturedTransform** input from another vision tool may introduce unexpected

discrepancies in your fixturing. For example, the **CogPMAlignTool** tool may add a small scaling factor to its result transformation, which can cause problems if you have obtained a scaling measurement by previously calibrating your input image. You almost never need to adjust any other components of the transformation other than translation and rotation. Whichever way you choose to receive transformation data, either from the UnfixturedFromFixturedTransform input or from the individual inputs for the translation and rotation components, you must not connect both types of transformation input at the same time.

The Fixture edit control includes the following components:

- A row of control buttons (**A**) at the top left.
- A tool display window (**B**) that displays the Fixture tool image buffers, which include Current.InputImage and LastRun.OutputImage. These buffers contain the input image to which the tool attaches the fixtured coordinate space and the output image from the tool's last run. Right click the tool display to bring up menu options that include zooming in or out of the image or showing a pixel or subpixel grid.
- A set of tabs organized by function (**C**). These functions include parameter settings to run the tool and settings to display tool graphics, such as fixtured and unfixtured axes. Pressing the **Control + Tab** keys scrolls through the set of tabs.
- A status bar at the bottom left of the control (**D**). A green circle indicates that the tool ran successfully; red means the tool ran unsuccessfully. The status bar also displays the time to run the tool and any error codes or messages. The first time that the status bar displays is the raw tool execution time. The second includes the time needed to update the edit control. Controls only update when they are visible.

8.2.3.1 Control buttons

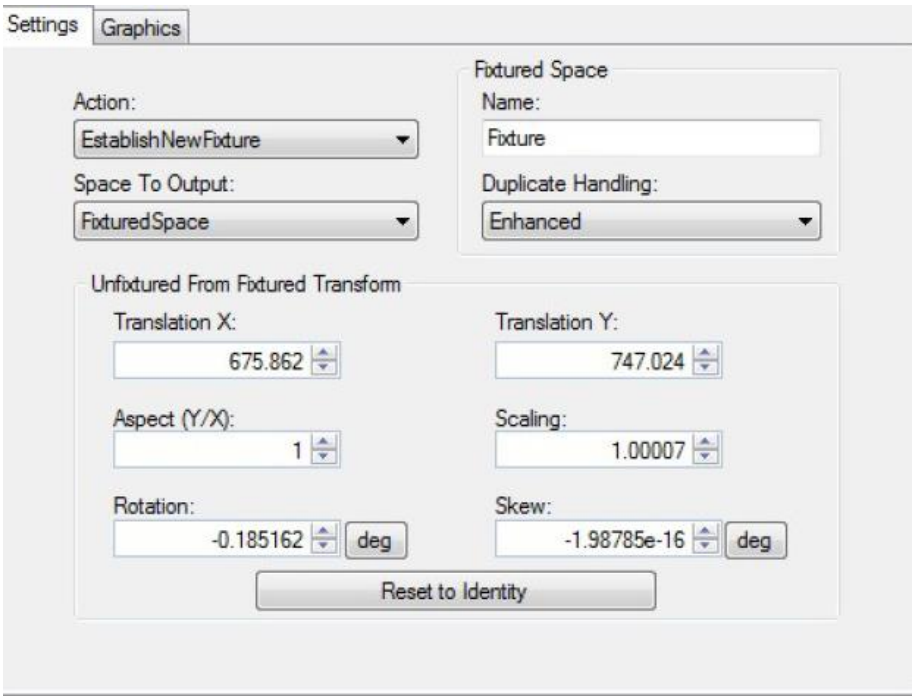
Button	Description	Function
	Run	Runs the Fixture tool. You must have an image available in the Current.InputImage buffer (equivalent to the **InputImage**). This button invokes the **Run** method.
	Electric mode	Toggles electric mode. When selected, the Fixture tool runs automatically if certain parameters have changed. When the edit control is in electric mode, these parameters are indicated by electric bolt icons.
	Local image display	Opens or closes the local tool display window. This window has a selection box that you use to specify the image buffer you want to view.
	Floating image display	Opens one or more floating tool display windows, providing an additional tool display window. As with the local tool display window, you can specify the image buffer to view.

Button	Description	Function
	Open	Loads a VisionPro persistence (.vpp) file, which contains a set of saved properties for this vision tool object type. Loading a persistence file for another object type throws an error and the load is unsuccessful. For more information about VisionPro persistence features, see the topic Persistence in VisionPro.
	Save	Saves the current properties of the underlying tool to a VisionPro persistence file. You have the option to save either the entire tool or the tool without its images or results.
	Save As	Saves the current properties of the underlying tool to a new VisionPro persistence file.
	Reset	Resets the underlying tool to a default state.
	Show ToolTips	Enable or disable the display of tooltips for individual items in the edit control.
	Help	Opens this help topic.

8.2.3.2 Image Buffers

The Fixture edit control has two image buffers. The Current.InputImage buffer uses the image supplied by the underlying Fixture tool's **InputImage** property. The LastRun.OutputImage buffer holds the output image that results from the fixturing operation.
The output image is a new COM object that references the same pixels and space tree, as the input image. Because of this, the newly attached fixture space is accessible from either image. The space was attached to the shared tree, not just to the tree associated with the output image. The selected space names, however, are not shared. Each image has its own selected space name.

8.2.3.3 Settings Tab

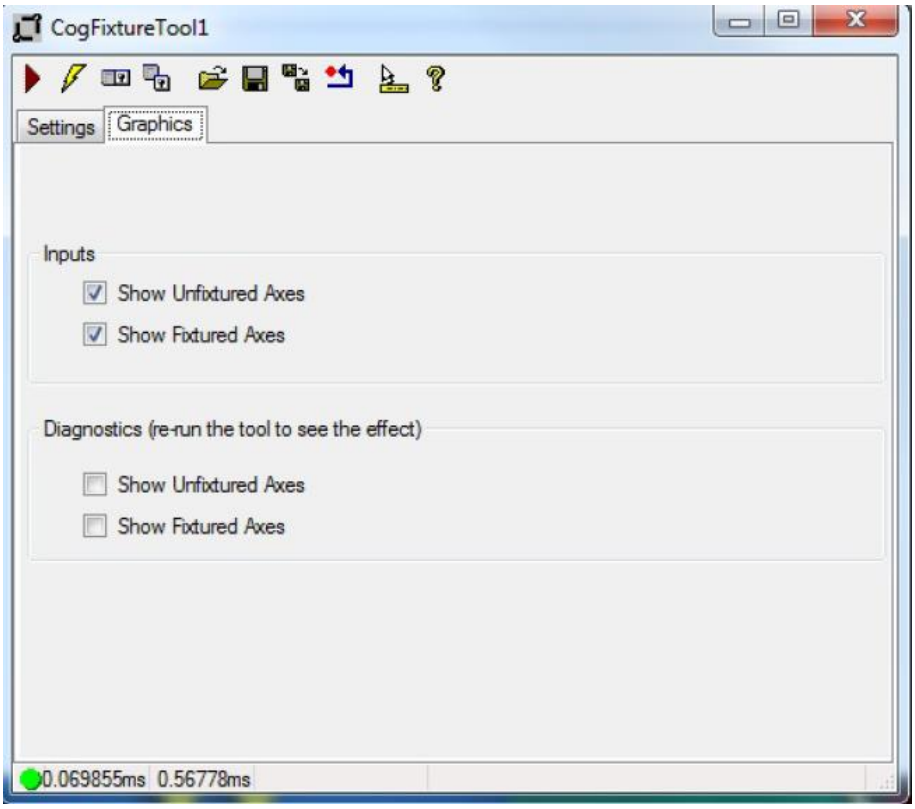


These parameters include the name of the fixtured space, the selected space name of the output image, and the transformation that maps points from the fixtured space to the unfixtured space. When the edit control is in electric mode, electric bolt icons indicate parameters whose changes cause the tool to run automatically.

Feature	Description
Action	Determines how the Fixture tool should process the input image.
FixturedSpaceName	Name of the fixtured coordinate space to attach to the **CogCoordinateSpaceTree** of the input image. This must be a valid **nonqualified space name**. Equivalent to the **FixturedSpaceName** property.
Duplicate Handling	Specifies how **fixtured space name collisions** are handled. Select **FixturedSpaceNameDuplicateHandling** to generate a warning when one tool overwrites the space created by another tool, **FixturedSpaceNameDuplicateHandling** to silently ignore collisions.
Space to Output	Specifies whether the selected space name of the output image will be a copy of the fixtured space name or a copy of the unfixtured space name. The output selected space name is always a **fully-qualified space name**. Equivalent to the **SpaceToOutput** property.

Feature	Description
Unfixtured from Fixtured Transform	Changes the values of the components of the **UnfixturedFromFixturedTransform** that will be attached to the input image's coordinate space tree. In a typical vision application, you calibrate an image before using it as input to the Fixture tool and have therefore obtained a desired scaling measurement from the calibration. You should not need to adjust the Scaling component of the UnfixturedFromFixtured transform. Doing so may affect the accuracy and validity of the resulting fixtured space. Note that you can specify the rotation and skew components in either degrees or radians, although the underlying tool maintains these values in radians.
Reset to Identity	Creates a new linear identity transform and sets it as the tool's UnfixturedFromFixtured transform.

8.2.3.4 Graphics Tab



When the control is in electric mode, electric bolt icons display parameters that cause the tool to run automatically if parameter values change.

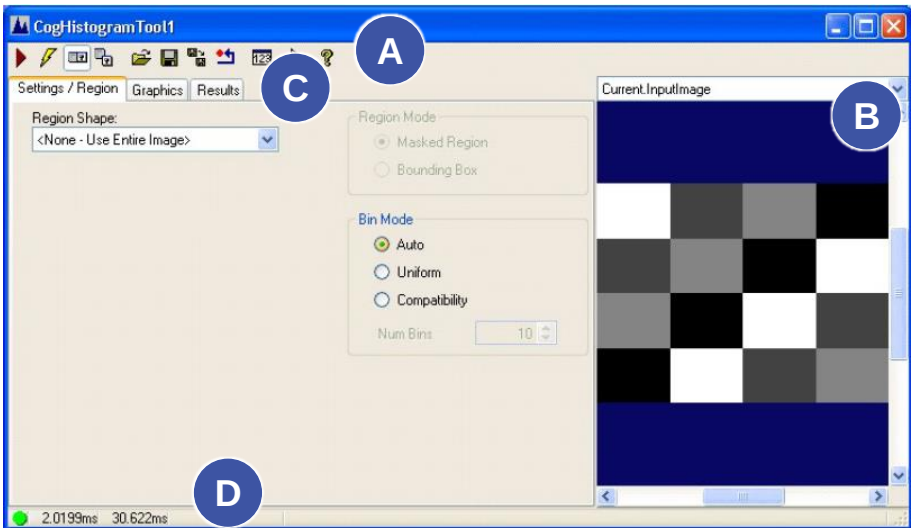
Feature	Description
Inputs	You can display the following results graphics: ▪ Show Unfixtured Axes displays noninteractive coordinate axes that represent unfixtured space in the Current.InputImage. Equivalent to enabling the **CogFixtureCurrentRecordConstants** bit of the tool's CurrentRecord. ▪ • Show Fixtured Axes displays interactive coordinate axes that represent fixtured space in the Current.InputImage. Equivalent to enabling the **CogFixtureCurrentRecordConstants** bit of the tool's CurrentRecord. Manipulating the fixtured axes changes the translation and rotation components of the **UnfixturedFromFixturedTransform**.
Diagnostics	You can display the following results graphics: ▪ Show Unfixtured Axes displays noninteractive coordinate axes that represent unfixtured space in the LastRun.OutputImage. Equivalent to enabling the **CogFixtureLastRunRecordDiagConstants** bit of the tool's LastRunRecord. ▪ • Show Fixtured Axes displays noninteractive coordinate axes that represent fixtured space in the LastRun.OutputImage. Equivalent to enabling the **CogFixtureLastRunRecordDiagConstants** bit of the tool's CurrentRecord.

8.2.4 Histogram edit control tool

The following are extracts from the Cognex User Manuals: for further and complete details, please refer to the whole Cognex User Manual.

This topic contains the following sections.

- Control Buttons
- Histogram Edit Control Buffers
- Settings / Region Tab
- Graphics Tab
- Results Tab



The Histogram edit control provides a graphical user interface (GUI) for the **CogHistogramTool** tool. The Histogram tool computes basic statistical measures of the pixels in an image such as the mean and median values, standard deviation, and variance. The tool can also display a graphical representation of the image histogram.

The Histogram edit control includes the following components:

- A row of control buttons at the top left (**A**).
- A tool display window (**B**) that can display the Histogram tool image buffers: Current.InputImage, LastRun.InputImage, and LastRun.Histogram. These buffers contain the image on which the tool will operate (including any specified input region), the last image upon which the tool operated, and a graphical display of the histogram.
- A set of tabs organized by function (**C**). These functions include parameter settings to run the tool and for the input region, display settings for the tool displays, and the tool results. Pressing the **Control + Tab** keys scrolls through the set of tabs.
- A status bar at the bottom left of the control (**D**). A green circle indicates that the tool ran successfully; red means the tool ran unsuccessfully. The status bar also displays the time to run the tool and any error codes or messages. The first time that the status bar displays is the raw tool execution time. The second includes the time needed to update the edit control. Controls only update when they are visible.

8.2.4.1 Control buttons

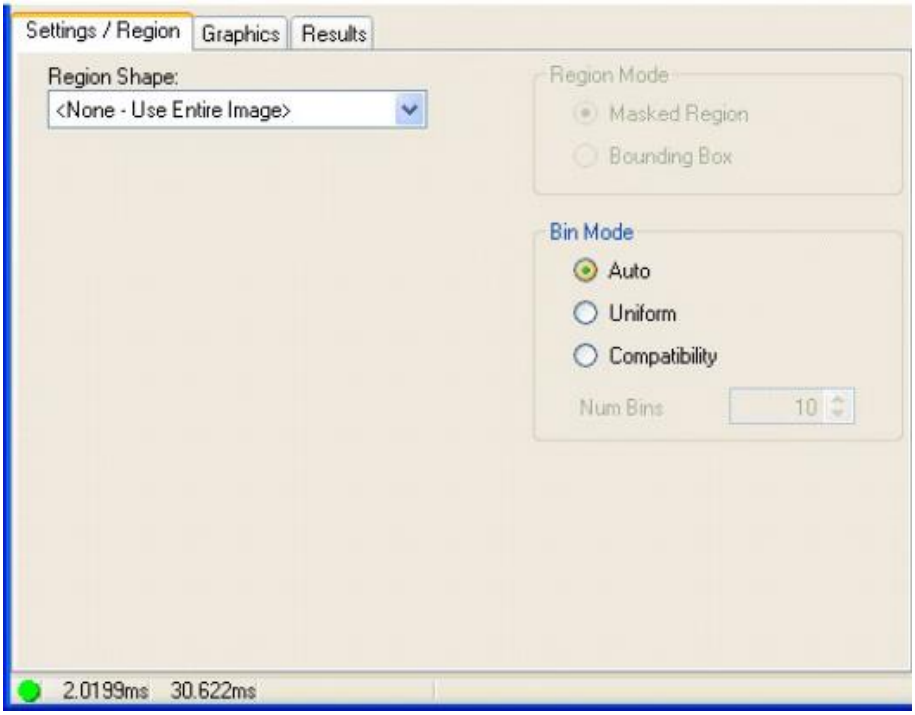
Button	Description	Function
	Run	Runs the Histogram tool. You must have an image available in the Current.InputImage buffer (equivalent to the **InputImage**). This button invokes the **Run** method.
	Electric mode	Toggles electric mode. When selected, the Histogram tool runs automatically if certain parameters have changed. When the edit control is in electric mode, these parameters are indicated by electric bolt icons.
	Local image display	Opens or closes the local tool display window. This window has a selection box that you use to specify the image buffer you want to view.
	Floating image display	Opens one or more floating tool display windows, providing an additional tool display window. As with the local tool display window, you can specify the image buffer to view.
	Open	Loads a VisionPro persistence (.vpp) file, which contains a set of saved properties for this vision tool object type. Loading a persistence file for another object type throws an error and the load is unsuccessful. For more information about VisionPro persistence features, see the topic Persistence in VisionPro.
	Save	Saves the current properties of the underlying tool to a VisionPro persistence file. You have the option to save either the entire tool or the tool without its images or results.
	Save As	Saves the current properties of the underlying tool to a new VisionPro persistence file.
	Reset	Resets the underlying tool to a default state.
	Results	Opens a new, separate results window, allowing you to view run results without turning to the Results tab.
	Show ToolTips	Enable or disable the display of tooltips for individual items in the edit control.
	Help	Opens this help topic.

8.2.4.2 Histogram Edit Control Buffers

The Histogram edit control has three image buffers. The first buffer shows the Histogram tool's **InputImage**, the second buffer displays the last input image that the Histogram tool ran on, and the third buffer shows the histogram graphic. All three buffers can be shown in both the local and floating tool display windows.

- The Current.InputImage provides the input images to the Histogram tool. This is the Histogram tool's **InputImage** buffer. If you specify a **Region**, the region selection graphic is shown in this buffer.
- The LastRun.InputImage buffer displays the image on which the tool most recently ran. Use the Graphics tab to highlight and select the input region.
- The LastRun.Histogram contains the histogram graphic. It is the same graphic produced by calling the **CreateResultGraphics** function.

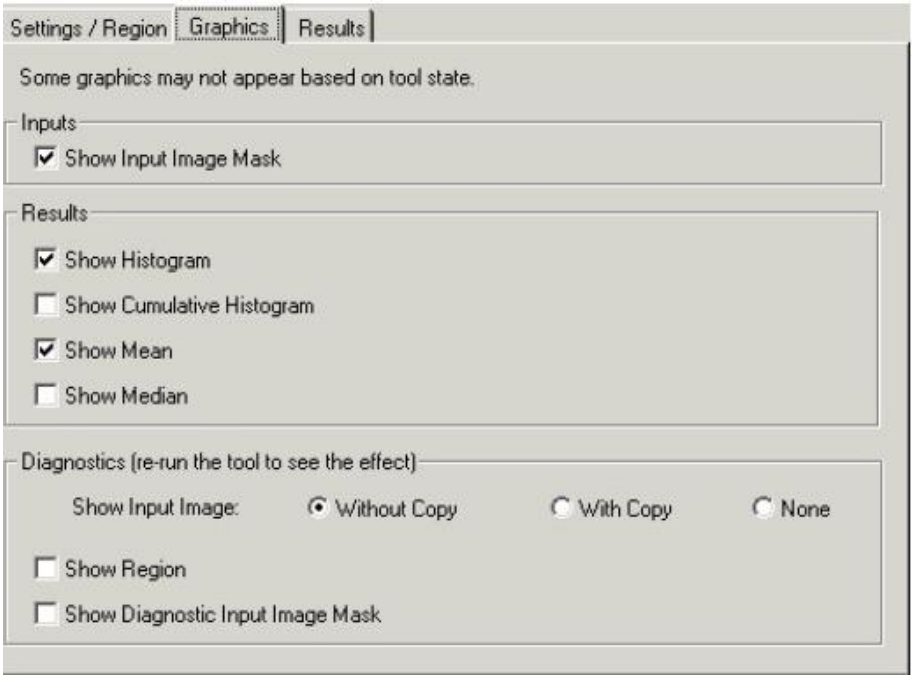
8.2.4.3 Settings / Region Tab



You use the Settings / Region tab to specify the input region for the tool and to set the tool's operating parameters.

Feature	Description
Region	Select the shape of the input region. Selecting "None=Use entire image" means that the tool uses the entire input image. A Histogram tool supports the following input region shapes: • CogCircle • CogEllipse • CogPolygon • CogRectangle • CogRectangleAffine The set of region-defining parameters that appear depend on the region shape you use. For more information on using a polygon as an input region, see the topic **Using Polygon Input Regions**.
RegionMode	Defines how the tool interprets the region you specify. ▪ Masked Region specifies that only the pixels that lie within the region are analyzed. ▪ Bounding Box specifies that pixels that lie within a pixel-aligned bounding box that encloses the region are processed.
SelectedSpaceName	The coordinate space in which the region is interpreted. For information, see **Coordinate Space Names**.
Select Mode	Available when Region Shape is CogRectangle or CogRectangleAffine. Selects the set of parameters that define the rectangle. If cogRectangleAffine is chosen, note that the angles of rotation and skew can be specified in degrees or radians, although the underlying tool keeps the measurements in radians.
BinMode	Specifies how the tool processes the bins it has been configured to use. The default setting of **Auto** allows the tool to ignore any setting for **Num Bins** and use the full dynamic range of the current input image (256 bins for an 8-bit image, 1024 bins for a 10-bit image, and so on up to 65,536 bins for a 16-bit image).
NumBins	Specifies the size of the histogram. By default, the histogram contains one bin for each possible pixel value in the input image (256 for 8-bit grey-scale images). You can specify a smaller value, in which case pixels with values greater than or equal to the number of bins are included in the last bin.

8.2.4.4 Graphics Tab

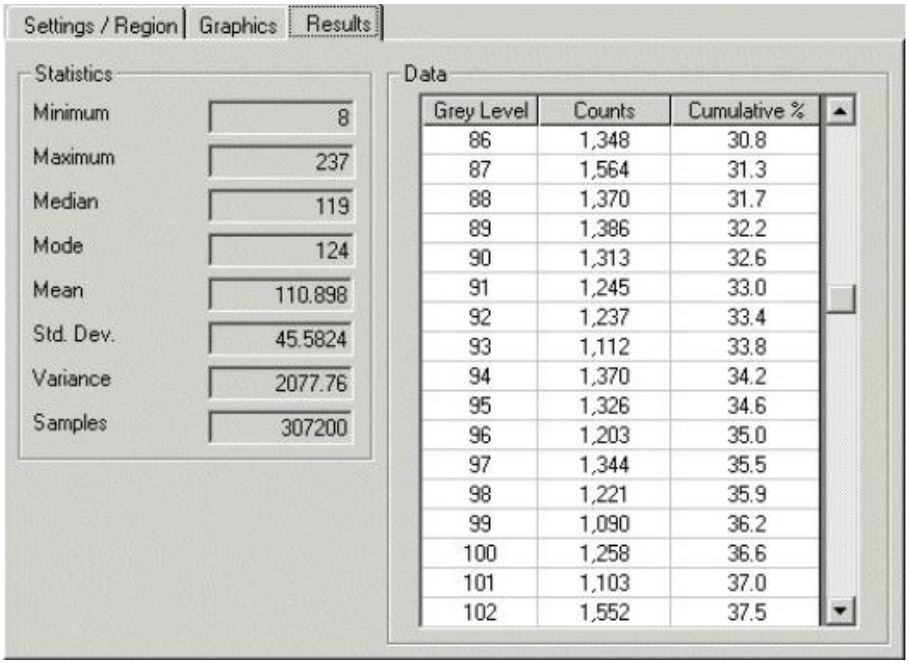


You use the Graphics tab to specify the graphics that the tool generates and displays.

Feature	Description
Inputs	Show **InputImageMask**. If you supply a mask image, the area of masked pixels is displayed on the Current.InputImage.
Results	Use these controls to determine which graphics are displayed in the LastRun.Histogram window. These settings are equivalent to the **CogHistogramResultGraphicConstants** enumerations supplied to **CreateResultGraphics** function. ▪ Show Histogram displays the graphical histogram (in yellow). ▪ Show Cumulative Histogram displays the cumulative histogram (in blue). ▪ Show Mean displays a vertical line at the mean pixel value (in white). ▪ Show Median displays a vertical line at the median pixel value (in yellow).

Feature	Description
Diagnostics display	Displays the following features in the LastRun.InputImage buffer. Uses the **CreateResultGraphics** method to generate these results. ▪ Show Input Image determines whether or not the input image is recorded as part of the diagnostic record, and whether the image is copied to the record or saved in the record as a reference. ▪ Show Region displays the search region (if one is specified). ▪ Show Diagnostic Input Image Mask displays the **InputImageMask** if one is specified.

8.2.4.5 Results Tab



The Results tab displays the results of the most recent run of the tool. This corresponds to the **CogHistogramResult** interface.

Feature	Description
Statistics	Displays the following histogram statistics. ▪ The **Minimum** histogram bin index that contains a non-zero pixel count ▪ The **Maximum** histogram bin index that contains a non-zero pixel count. ▪ The histogram bin index of the bin containing the **Median** (middle) pixel value in the histogram. This is the histogram bin index where half of the distribution of pixel values is above and half below it. ▪ The histogram bin index of the bin containing the **Mode** (most common) pixel value in the histogram. This is the histogram bin index with the greatest pixel count. ▪ The arithmetic **Mean** (average) pixel value in the image. ▪ The **StandardDeviation** of the pixel values in the image. ▪ The **Variance** of the pixel values in the image. ▪ The **NumSamples** in the histogram.
Data	Displays the following information about each bin in the histogram. ▪ The number of pixels in that bin (the Count). ▪ The cumulative percentage of image pixels at that bin. By default, the number of bins in the image corresponds to the maximum possible pixel value in the image; 256 for 8-bit greyscale images up to 65,536 for 16-bit greyscale images.

8.3 Standard scripts

```
List<ICogGraphic> grafica = new List<ICogGraphic>();
//ICogRecord Record_Output = null;
public override bool GroupRun(ref string message, ref CogToolResultConstants result)
{

  grafica.Clear();
  double Value_Histo;
  bool Enable_Histo;
  double dispX, dispY;

  CogToolBlockTerminal m_CogInput_Enable_Histo = mToolBlock.Inputs["Enable_Histo"] as CogToolBlockTerminal;
  CogToolBlockTerminal m_CogInput_Value_Histo = mToolBlock.Inputs["Value_Histo"] as CogToolBlockTerminal;
  CogToolBlockTerminal m_CogOutput_Found = mToolBlock.Outputs["Found"] as CogToolBlockTerminal;
  CogToolBlockTerminal m_CogOutput_Output_String = mToolBlock.Outputs["Output_String"] as CogToolBlockTerminal;
  CogPMAlignMultiTool m_CogPMAlignMultiTool = mToolBlock.Tools["CogPMAlignMultiTool1"] as CogPMAlignMultiTool;
  CogFixtureTool m_cogFixtureTool = mToolBlock.Tools["CogFixtureTool1"] as CogFixtureTool;

  m_CogOutput_Found.Value = false;

  Enable_Histo = (bool) m_CogInput_Enable_Histo.Value;
  Value_Histo = (double) m_CogInput_Value_Histo.Value;

  CogTransform2DLinear xform;

  ////////////////////////////////////////////////////

  mToolBlock.RunTool(m_CogPMAlignMultiTool, ref message, ref result);
  if(m_CogPMAlignMultiTool.Results.PMAlignResults.Count < 1)
  {
    m_CogOutput_Found.Value = false;
    m_CogOutput_Output_String.Value = string.Format("{0};{1};{2};{3};\n\r",
      "Null",
      "Null",
      "Null",
      "Null");

    return false;

}

  //int count = 0;
  foreach (CogPMAlignResult m_CogPMAlignMultiResult  in m_CogPMAlignMultiTool.Results.PMAlignResults)
  {
    //se non è abilitato l'histogram di controllo
    if(!Enable_Histo)
    {
        m_CogOutput_Found.Value = true;
        m_CogOutput_Output_String.Value = string.Format("{0};{1};{2};{3};\n\r",
        m_CogPMAlignMultiResult.ModelName,
        m_CogPMAlignMultiResult.GetPose().TranslationX.ToString("0.##"),
        m_CogPMAlignMultiResult.GetPose().TranslationY.ToString("0.##"),
        ((m_CogPMAlignMultiResult.GetPose().Rotation * 180) / Math.PI).ToString("0.

      return false;
    }
    else
    {
      CogToolBlock m_cogToolBlock = new CogToolBlock();
      string str = m_CogPMAlignMultiResult.ModelName;
      switch (str) {
        case "Pattern1":
          m_cogToolBlock = mToolBlock.Tools["HistogramPattern1"] as CogToolBlock;
          break;
        case "Pattern2":
          m_cogToolBlock = mToolBlock.Tools["HistogramPattern2"] as CogToolBlock;
          break;
        case "Pattern3":
          m_cogToolBlock = mToolBlock.Tools["HistogramPattern3"] as CogToolBlock;
          break;
        case "Pattern4":
          m_cogToolBlock = mToolBlock.Tools["HistogramPattern4"] as CogToolBlock;
          break;
        default:
          Console.WriteLine("Nothing");
          break;
      }
```

Support variables

Toolboxes inputs and outputs
Declare which tools shall be used

Variables inizialization

Run Locator Tool.
If no objects are found, exit from Tool Block

process every single result

Check if Histogram control is enabled; if "Histogram" checkbox is disabled, set as output string the first found value.

If the "Histogram" checkbox is enabled, apply histogram control to each result of the locator

Select the tool related to the model name (pattern1, patter2…)

```
        m_cogFixtureTool.RunParams.UnfixturedFromFixturedTransform = m_CogPMAlignMultiResult.GetPose();
        mToolBlock.RunTool(m_cogFixtureTool, ref message, ref result);

        int[] StandardDeviationResult = new int[m_cogToolBlock.Tools.Count];
        //eseguo gli Histogram
        for(int j = 0; j < m_cogToolBlock.Tools.Count;j++)
        {
          CogHistogramTool m_cogHistogramTool = (CogHistogramTool) m_cogToolBlock.Tools[j];
          mToolBlock.RunTool(m_cogHistogramTool, ref message, ref result);
          CogRectangleAffine r = new CogRectangleAffine(m_cogHistogramTool.Region as CogRectangleAffine);
          CogHistogramResult m_cogHistogramResult = m_cogHistogramTool.Result;
          if(m_cogHistogramResult.StandardDeviation < Value_Histo)
          {
            StandardDeviationResult[j] = 1;
            r.Interactive = false;
            r.Color = CogColorConstants.Green;
            r.LineWidthInScreenPixels = 2;
            // r.SelectedSpaceName = "Fixture";
            xform = m_CogPMAlignMultiResult.GetPose();
            xform.MapPoint(r.CenterX, r.CenterY, out dispX, out dispY);
            r.CenterX = dispX;
            r.CenterY = dispY;
            r.Rotation = m_CogPMAlignMultiResult.GetPose().Rotation + r.Rotation;
            grafica.Add(r);
          }
          else
          {
            //se non ok
            StandardDeviationResult[j] = 0;
            r.Interactive = false;
            r.Color = CogColorConstants.Red;
            r.LineWidthInScreenPixels = 2;
            //r.SelectedSpaceName = "Fixture";
            xform = m_CogPMAlignMultiResult.GetPose();
            xform.MapPoint(r.CenterX, r.CenterY, out dispX, out dispY);
            r.CenterX = dispX;
            r.CenterY = dispY;
            r.Rotation = m_CogPMAlignMultiResult.GetPose().Rotation + r.Rotation;
            grafica.Add(r);
          }
        }

        //Controllo i risultati degli Histogram
        int sum = 0;
        for(int t = 0; t < m_cogToolBlock.Tools.Count;t++)
        {
          sum = sum + StandardDeviationResult[t];
        }

        if(sum == m_cogToolBlock.Tools.Count)
        {
          m_CogOutput_Found.Value = true;
          m_CogOutput_Output_String.Value = string.Format("{0};{1};{2};{3};\n\r",
            m_CogPMAlignMultiResult.ModelName,
            m_CogPMAlignMultiResult.GetPose().TranslationX.ToString("0.##"),
            m_CogPMAlignMultiResult.GetPose().TranslationY.ToString("0.##"),
            ((m_CogPMAlignMultiResult.GetPose().Rotation * 180) / Math.PI).ToString("0.##"));
        }
      }
    }

    return false;

}
```

Run the FixtureTool tool, to set the origin of the pattern found as image origin.
Run the tool block containing the histograms and analyze the results.

According to the number of histogram tools of the toolblock, analyze the results and check if the standard deviation is below the set value.

If found value is lower than the set threshold value, rectangles are green coloured.

If found value is higher than the set threshold value, rectangles are red coloured.

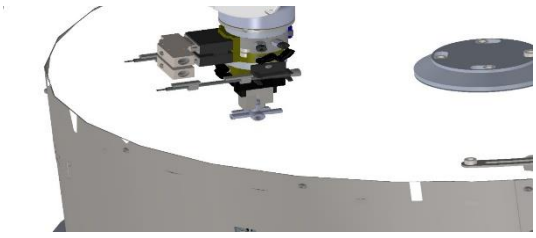
Analyze all the tool block histogram results: if they are all green, the item can be picked up.

Set the output string to be sent to the robot.

8.4 How to create a Locator Params procedure – Example

Two objects lay on the Flexibowl plate: the following procedure describes how to check them through Histograms, to verify the possibility of picking them up with the robot gripper without hitting other parts.

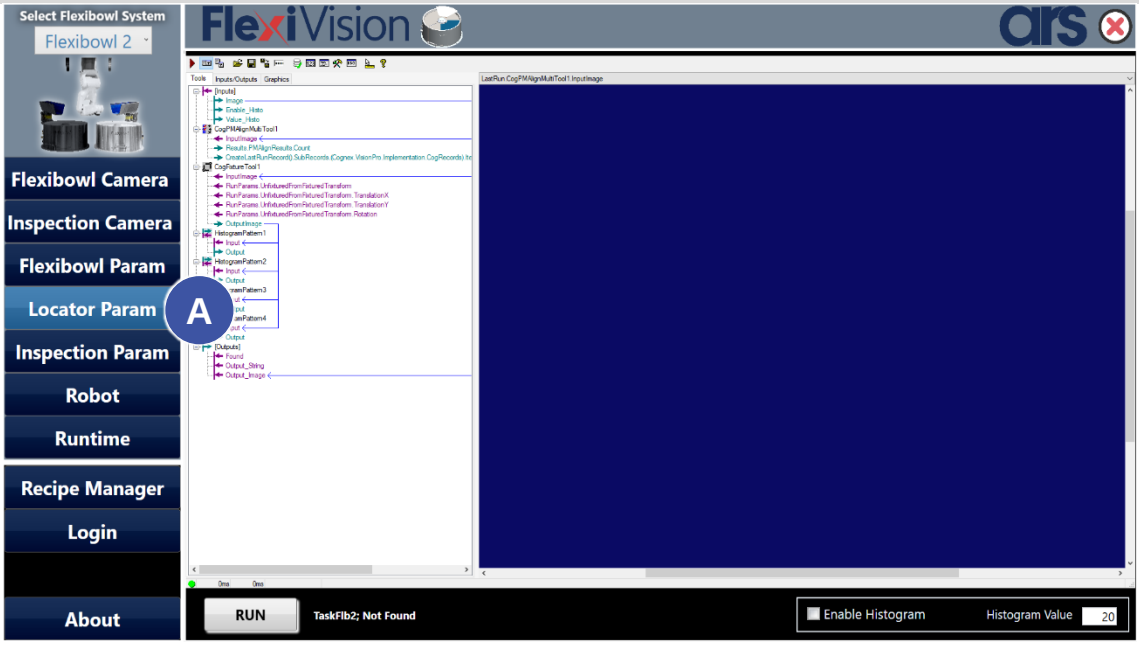
8.4.1 Preliminary operations

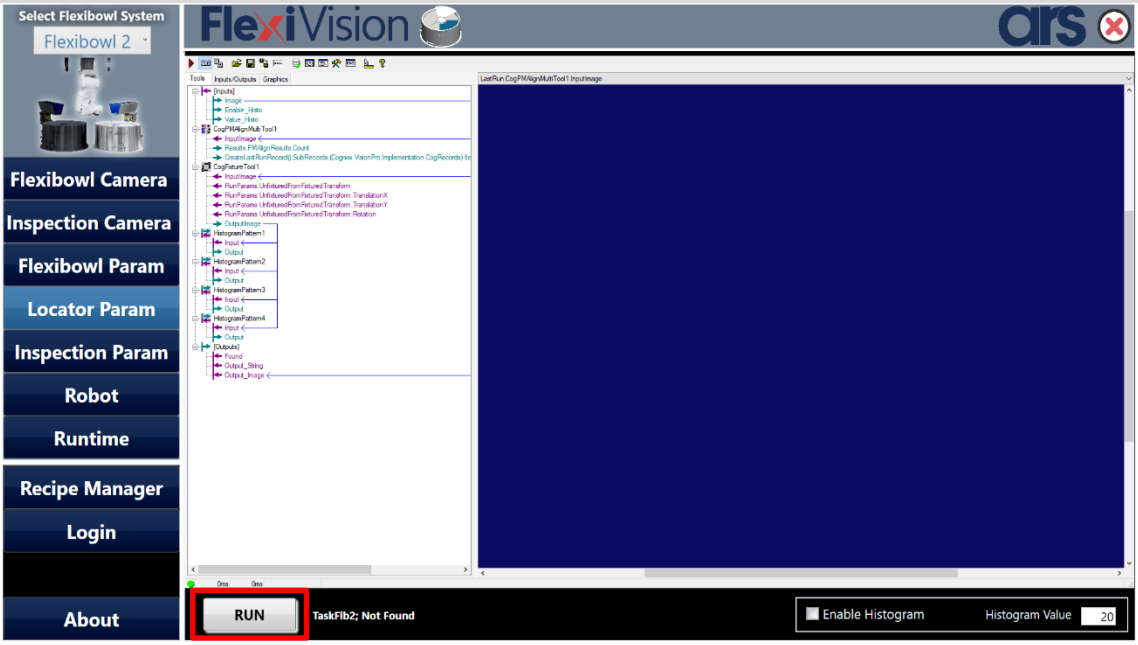
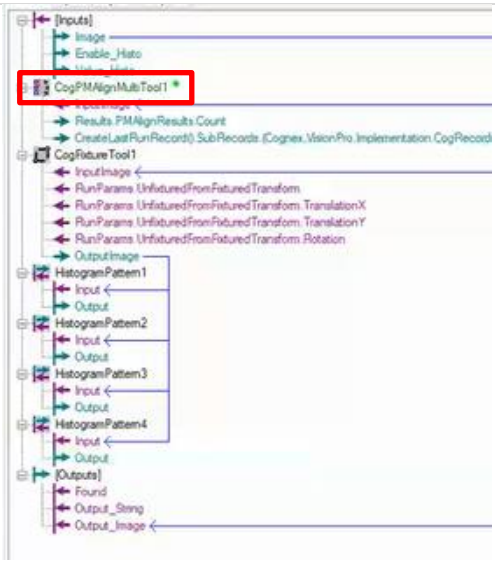
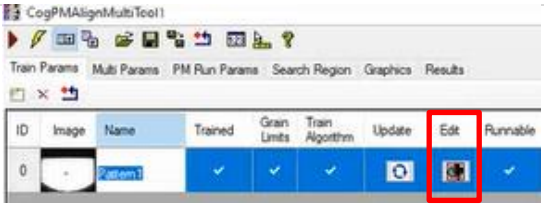
Step	Action	Notes/Pictures
1	Leave one item only on the FlexiBowl® plate	
2	Assemble the robot picking tool (gripper or vacuum) on the robot head	**NOTE** Refer to the robot Instruction manual for details.
3	Activate the calibration frame on the robot	**NOTE** Refer to the robot Instruction manual for details.
4	Set the robot rotation to "0°"	**NOTE** Refer to the robot Instruction manual for details.
5	Move the robot manually to the pick up position and align it to the item to be picked up.	**NOTE** If the picking tool is a gripper, it is possible to open/close its teeth to center the item mechanically. 

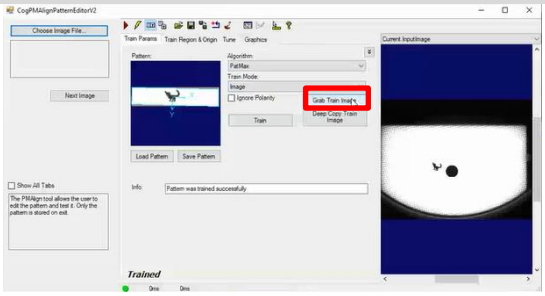
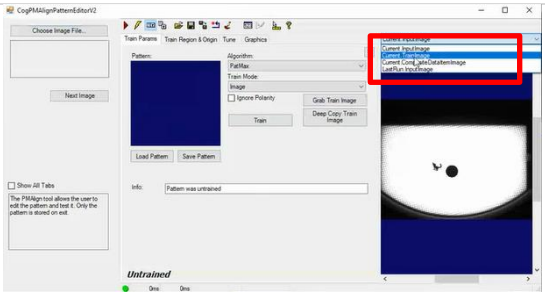
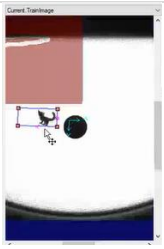
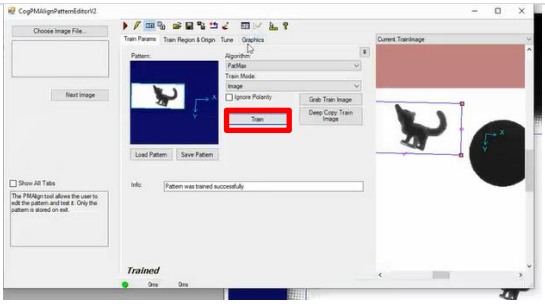
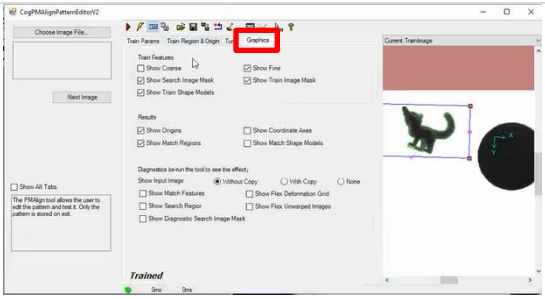
Step	Action	Notes/Pictures
6	Once robot gripper and item are properly aligned, note the pick up height ("Z" coordinate) and move the robot gripper away from the image acquisition zone. The "Z" coordinate shall be entered in the robot working recipe.	This operation allows to align camera and robot rotation.

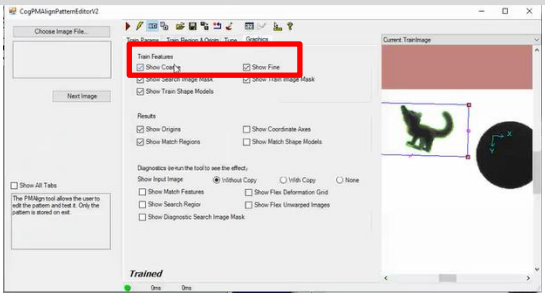
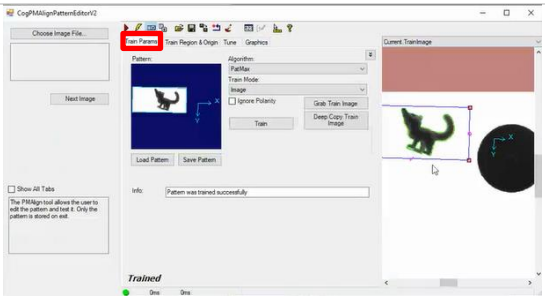
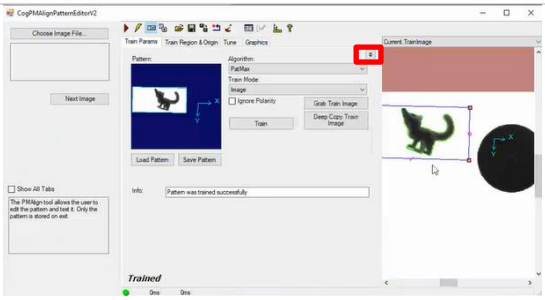
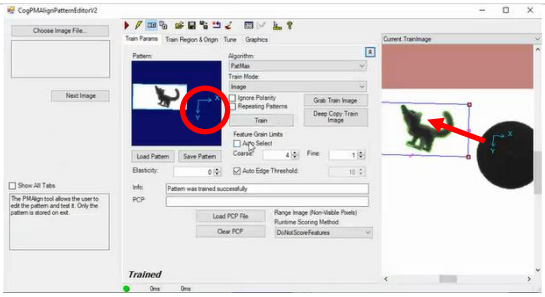
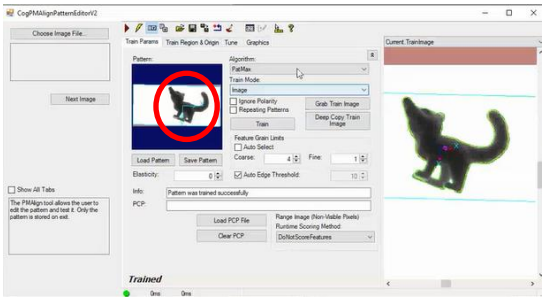
8.4.2 How to carry out the Locator Params procedure for Pattern 1

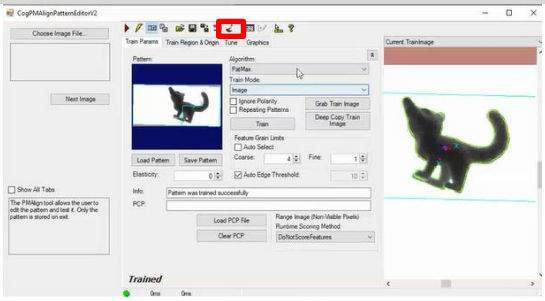
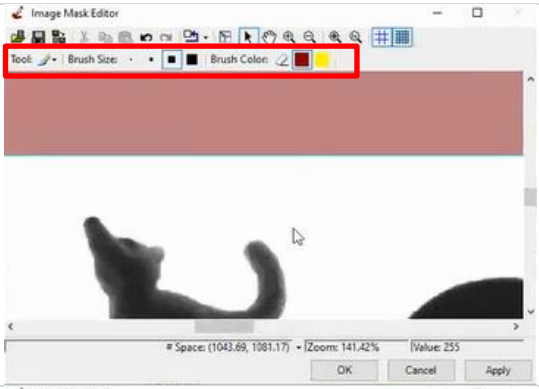
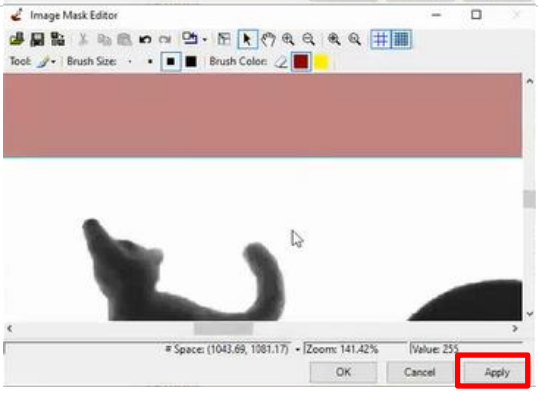
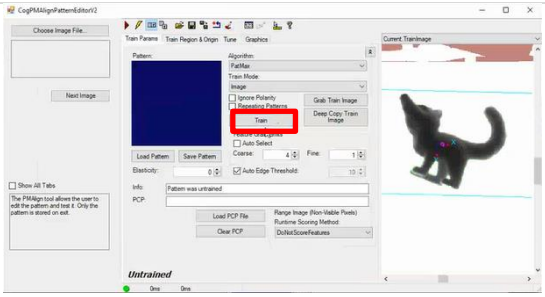
NOTE
This procedure can be carried out by the following users:
- **TECHNICIAN**
- **ARS**

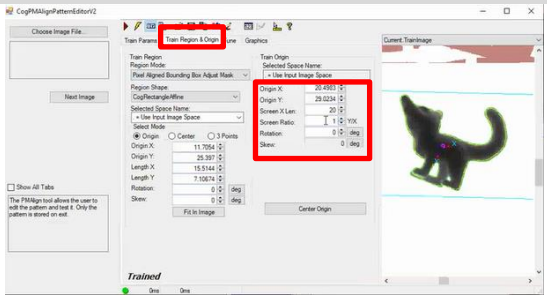
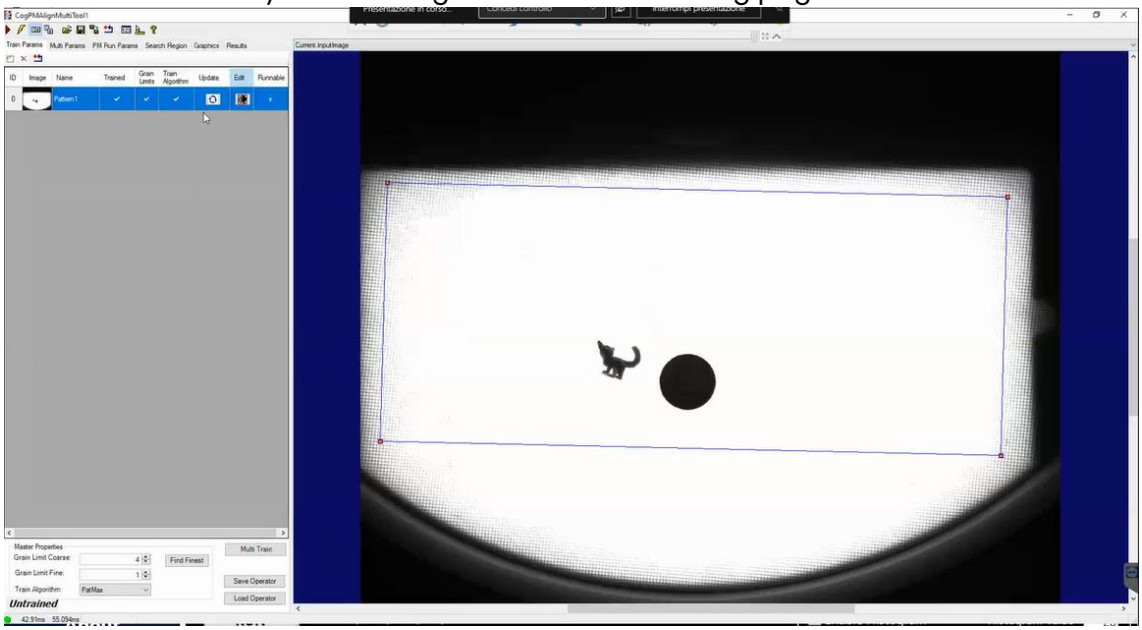
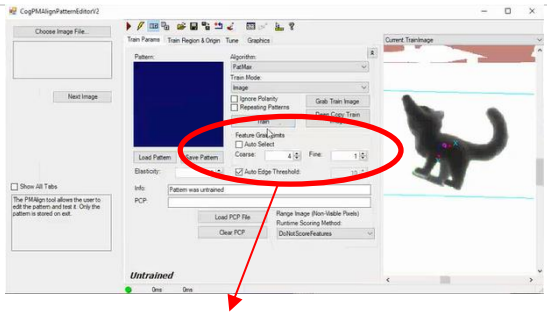
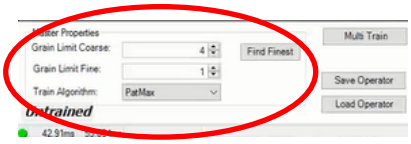
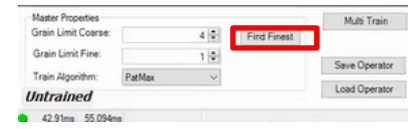
Step	Action	Notes/Pictures
1	 Click on *Locator Param* (**A**) of the operations menu.	

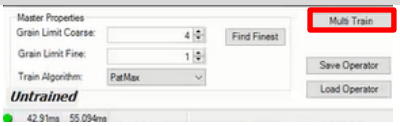
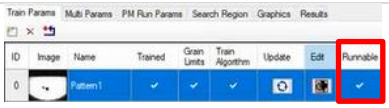
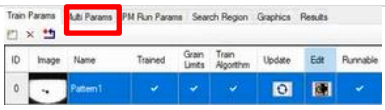
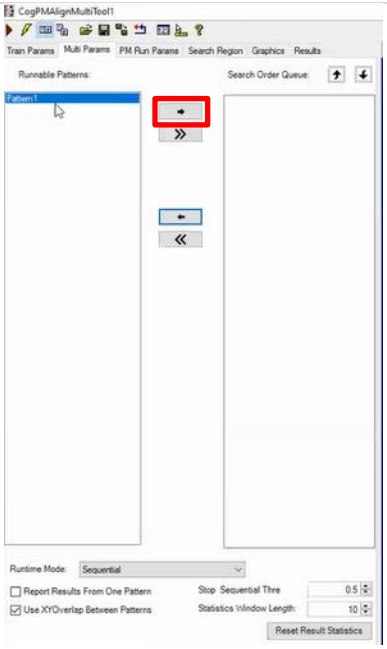
Step	Action	Notes/Pictures
2	 Update the image by pressing the **RUN** button.	
3	Doubleclick on **CogPMAlignMultiTool1**.	
4	The following mask appears on the left side. Doubleclick on **Edit** to edit PATTERN 1.	

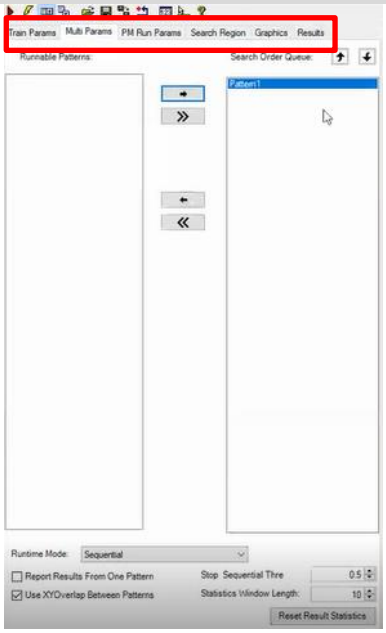
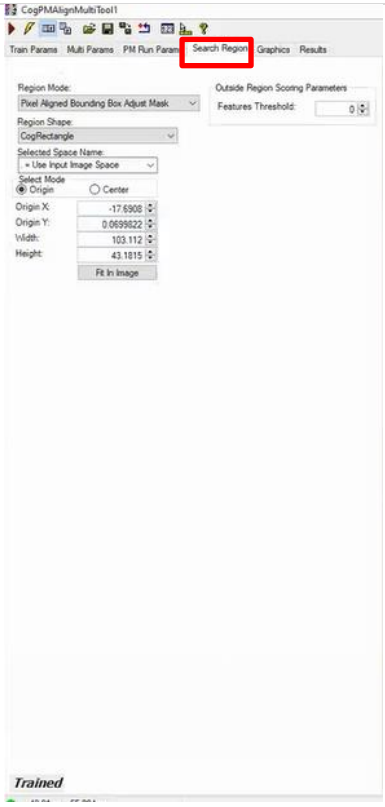
Step	Action	Notes/Pictures
5	The following window opens. Press **Grab Train Image**.	
6	From the drop down menu, select **Current Train Image.**	
7	Draw or resize the region of interest for the pattern.	
8	Press **TRAIN** to create the region.	
9	Enter the **Graphics** menu.	

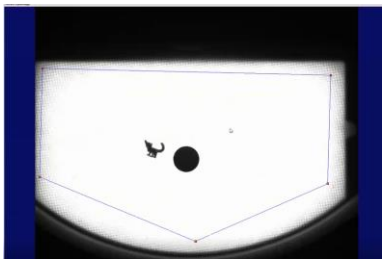
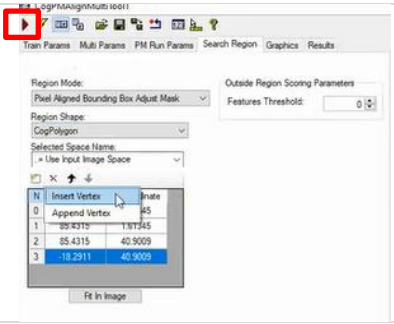
Step	Action	Notes/Pictures
10	On the check boxes, select **Show fine** and **Show coarse**.	
11	Go back to the **Train Params** menu: lines which have been created are now visible.	
12	Press the ⏬ arrows to enter the additional features (see previous paragraphs for details).	
13	Drag (0, x,y) to the desired position. These are the coordinates for robot pick up position.	 

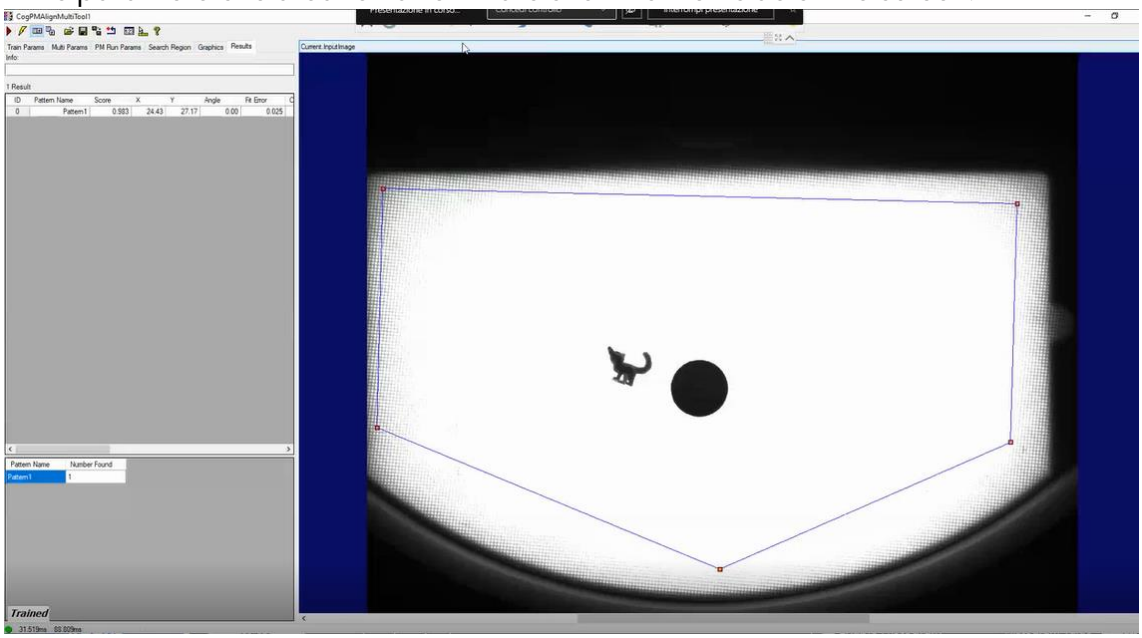
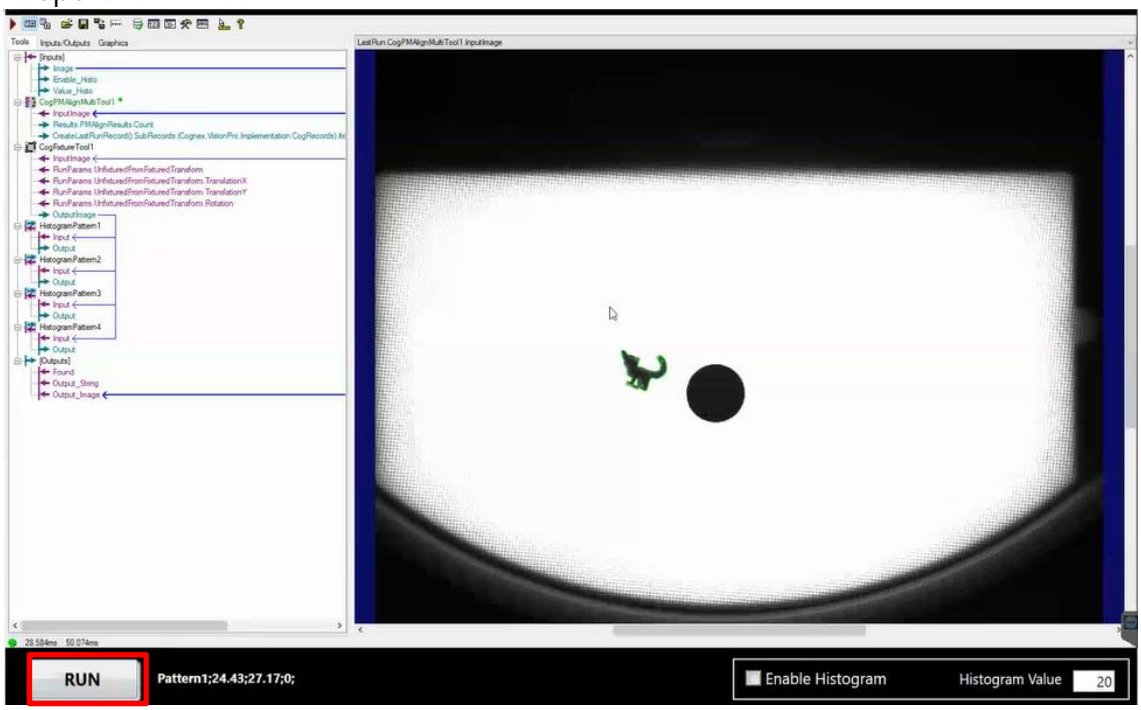
Step	Action	Notes/Pictures
14	Press the the **Image Mask Editor**  from the control buttons	
15	A window mask appears, with tools (in the top part) to delete or mask details which are not useful for the pattern recognition (e.g. use rubber for delete; use red brush to mask details).	
16	Press **Apply** to make the operations effective.	
17	Press **Train**.	

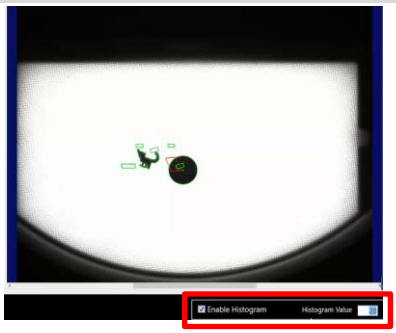
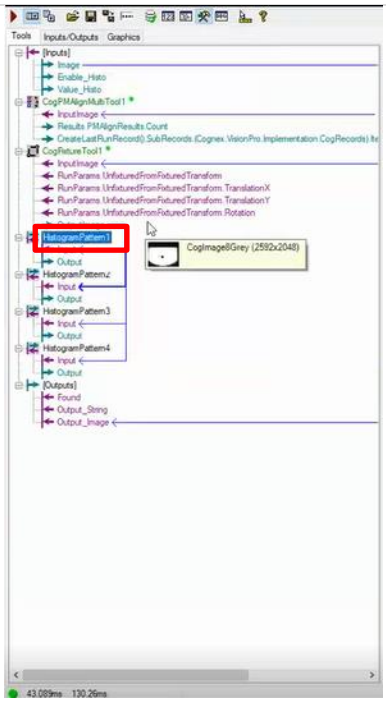
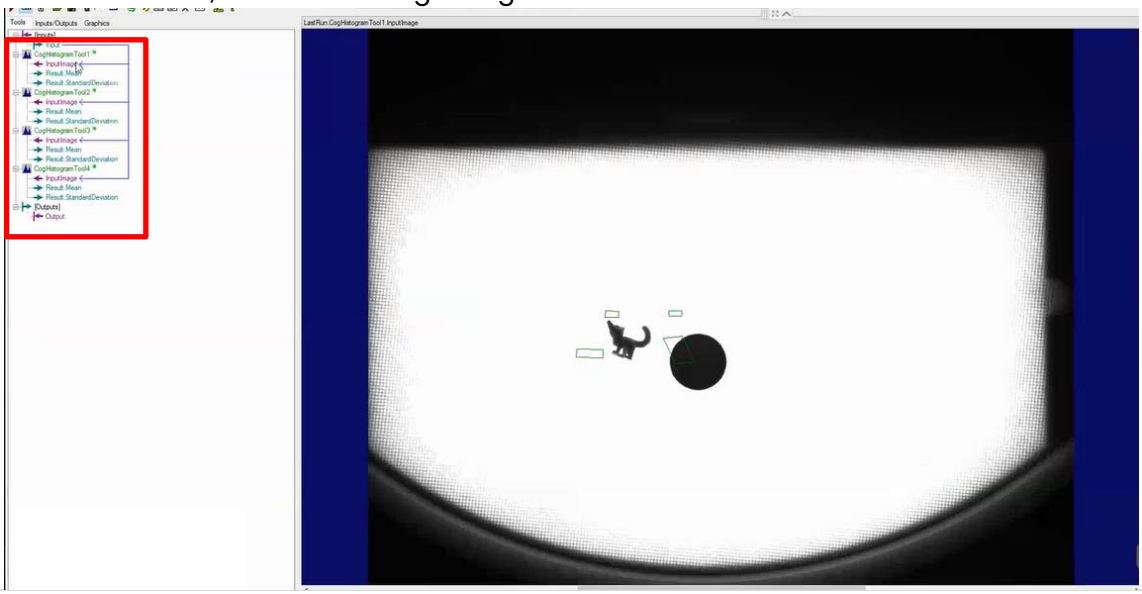
Step	Action	Notes/Pictures
18	If necessary, enter the **Train Region and Origin** menu to change the position of origin coordinates or rotation, by the side fields. Positions are in pixels.	
19	Close the window by the ✕ and go back to the following page.	
20	The Master Properties shown in the previous page left bottom (Grain Limit Coarse, Grain Limit Fine, train Algorithm) must equal the one listed in the Train Params page. Enter them in the relevant fields or, alternatively, enter high values in the Grain Limit Coarse and Grain Limit Fine of the Master Properties, and press **Find Finest**. The system automatically shall set them to the values of the Train Params mask.	  

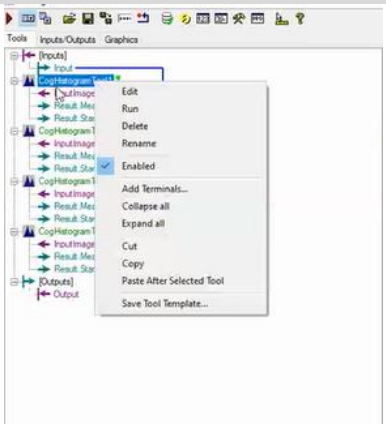
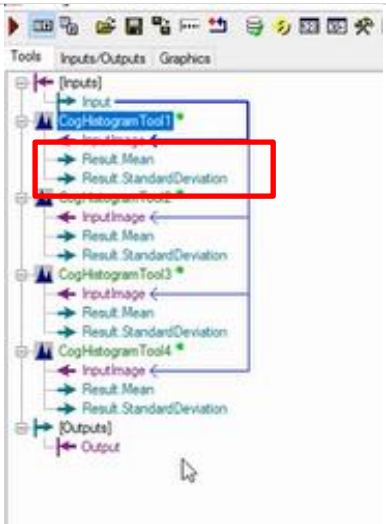
Step	Action	Notes/Pictures
21	Press **Multi Train**.	
22	A flag appears in the Runnable field (left top of the previous page): Pattern 1 can be used.	
23	Enter the Multi Params menu	
24	Select the pattern from the Runnable Patterns column and press →: the pattern is moved to the Search Order Queue column. **NOTE** Each time a parameter is changed, the pattern becomes not runnable. The procedure listed above needs to be repeate (Multi Train).	

Step	Action	Notes/Pictures
25	If necessary, modify the pattern features by the relevant menu.	
26	By the **Search Region** menu, it is possible to redefine the region for image acquisition (e.g. to avoid collisions during robot pick up): the system shall give the coordinates of the elements which are inside this region.	

Step	Action	Notes/Pictures
27	The region can be modified by dragging its vertexes. **NOTE** It is recommended to select the polygon (as region shape) and to add vertexes, in order to delimit an area as close as possible to the Flexibowl semicircle.	 
28	Press Run .	

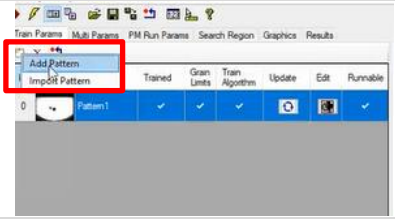
Step	Action	Notes/Pictures
29	All the parameters related to Pattern 1 are shown on left side of the screen. 	
30	Close the window by ✕, press **RUN** on the bottom of the screen and the following page will open: 	

Step	Action	Notes/Pictures
31	By the check box, enable the Hystogram (to make all of them visible) or disable it. Histograms are used to measure the brightness threshold for an image or section of an image in order to determine the presence/ absence of a part or to determine relative fill levels. In the **Hystogram value** field, set the reject threshold, related to grey scale.	
32	Doubleclick on **HystogramPattern1**: the histograms related to pattern 1 only are shown.	
33	On the left menu, the related CogHistogramTools are shown. 	

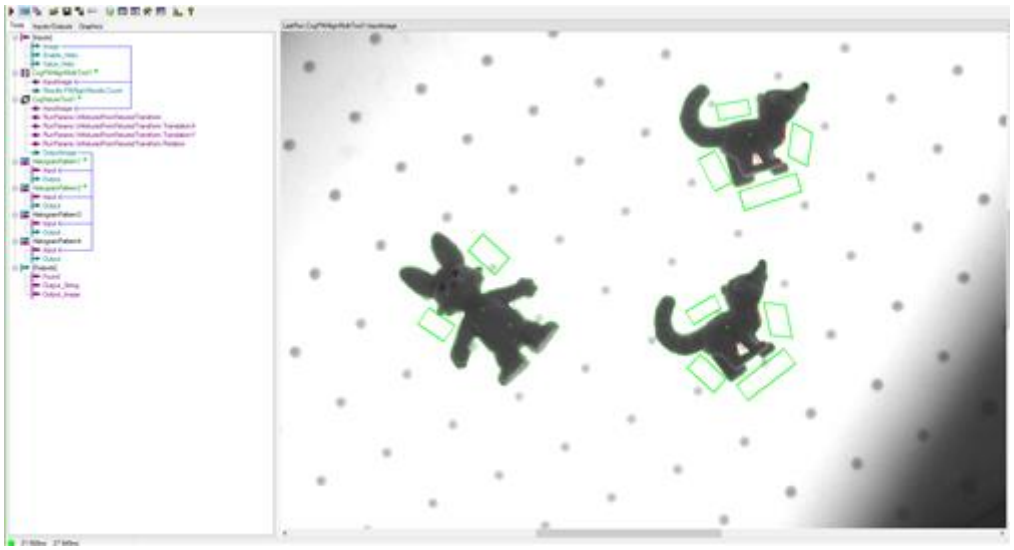
Step	Action	Notes/Pictures
34	Move on each CogHistogramTool, right-click to open the menu shown aside.	
35	Doubleclick on each CogHistogramTool to move them and/or to change their shape and dimensions. 	
36	Doubleclick on **Results.Mean** or on **Result.StandardDeviation** to show the rresults related to each Histogram.	

8.4.3 How to add patterns

NOTE
The FLEXIVISION standard configuration includes four patterns for each recipe.
THE PATTERNS CANNOT BE RENAMED.

Step	Action	Notes/Pictures
1	To create a new pattern, press in the Train Params menu and select **Add Pattern,** then repeat all the procedures listed at paragraph 8.3.2.	

The following is an example of 3 patterns, with the relevant control histograms.

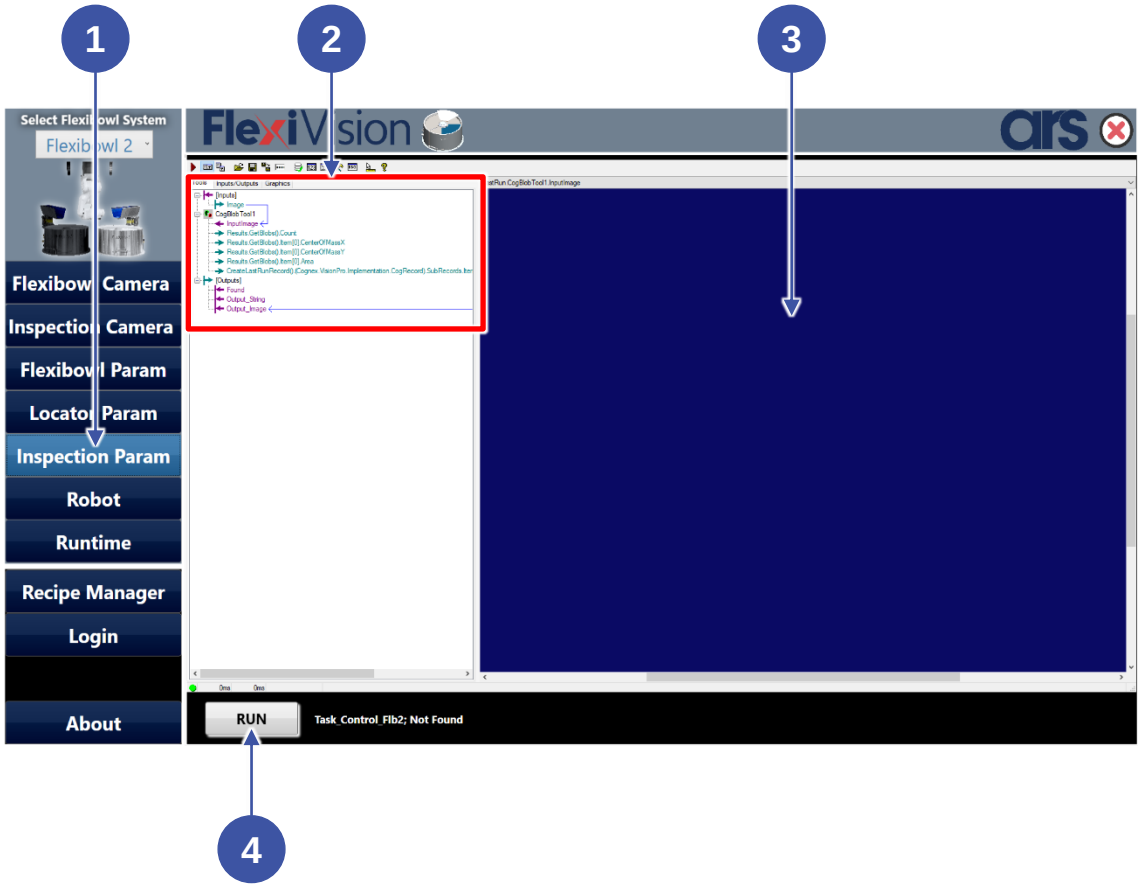


PAGE INTENTIONALLY LEFT BLANK

9 INSPECTION

9.1 Inspection Param page

By pressing LOCATOR PARAM in the OPERATIONS menu, the following page opens:



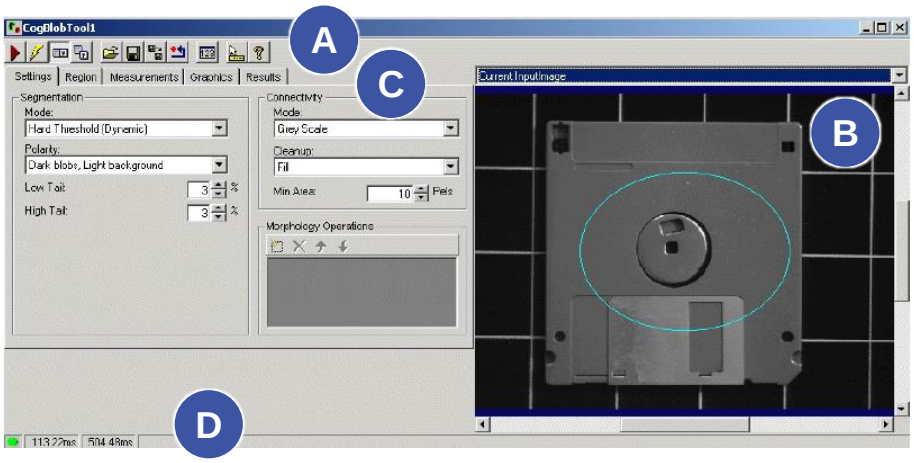
Position	Element/section	Description
1	OPERATION MENU	Includes all the control and operation procedures.
2	BlobEdit Control TOOL	
3	LAST INPUT IMAGE	
4	RUN key	

9.2 Blob Edit Control

NOTE
The FLEXIVISION standard configuration includes one blobtool.
It is possible to add tools (as described at Chapter 11).

The following is an extract from the Cognex User Manual: for further and complete details, please refer to the whole Cognex User Manual.

The Blob edit control provides a graphical user interface to the most commonly used features of the **CogBlobTool** and related Blob interfaces. The Blob tool allows you to search for blobs, which are any two-dimensional closed shapes in your input image. Using the Blob edit control, you can specify the segmentation, connectivity, and morphology parameters you want when the tool runs as well as the type of measurements that you want the tool to make.
You can examine the results of the search on the edit control's Results tab; you can also view the search results overlaid on the search image.



The Blob edit control includes the following components:

- A row of control buttons at the top left (**A**).
- A tool display window (**B**) that can display the Blob tool image buffers. Right click the tool display to bring up menu options that include zooming in or out of the image or showing a pixel or subpixel grid.
- A set of tabs organized by function (**C**). These functions include parameter settings to run the tool, the region within the input image to which the search is constrained, the measurements you want, and the search results. Pressing the Control + Tab keys scrolls through the set of tabs.
- A status bar at the bottom left of the control (**D**). A green circle indicates that the tool ran successfully; red means the tool ran unsuccessfully. The status bar also displays the time to run the tool and any error codes or messages. The first time that the status bar displays is the raw tool execution time. The second includes the time needed to update the edit control. Controls only update when they are visible.

This topic contains the following sections.

- Control Buttons
- Blob Edit Control Buffers
- Settings Tab
- Region Tab
- Graphics Tab
- Results Tab

9.2.1 Control Buttons

Button	Description	Function
	Run	Runs the Blob tool. You must first specify the segmentation, connectivity, and morphology parameters. You may constrain the blob search to a search region within the input image.
	Electric mode	Toggles electric mode. When selected, the tool runs automatically if certain parameters have changed. When the edit control is in electric mode, these parameters are indicated by electric bolt icons.
	Local image display	Opens or closes the local tool display window. This window has a selection box that you use to specify the image buffer you want to view.
	Floating image display	Opens one or more floating tool display windows, providing an additional tool display window. As with the local tool display window, you can specify the image buffer to view.
	Open	Loads a VisionPro persistence (.vpp) file, which contains a set of saved properties for this vision tool object type. Loading a persistence file for another object type throws an error and the load is unsuccessful. For more information about VisionPro persistence features, see the topic Persistence in VisionPro.
	Save	Saves the current properties of the underlying tool to a VisionPro persistence file. You have the option to save either the entire tool or the tool without its images or results.
	Save As	Saves the current properties of the underlying tool to a new VisionPro persistence file.

Button	Description	Function
	Reset	Resets the underlying tool to a default state. This tool gives you a choice between resetting to the default-constructed state, which is appropriate when you are using it in a Visual Studio.NET application, and its template-initialized state, which is appropriate for **QuickBuild** applications.
	Results	Opens a new, separate results window, allowing you to view run results without turning to the Results tab.
	Show ToolTips	Enable or disable the display of tooltips for individual items in the edit control.
	Help	Opens this help topic.

9.2.2 Blob Edit Control Buffers

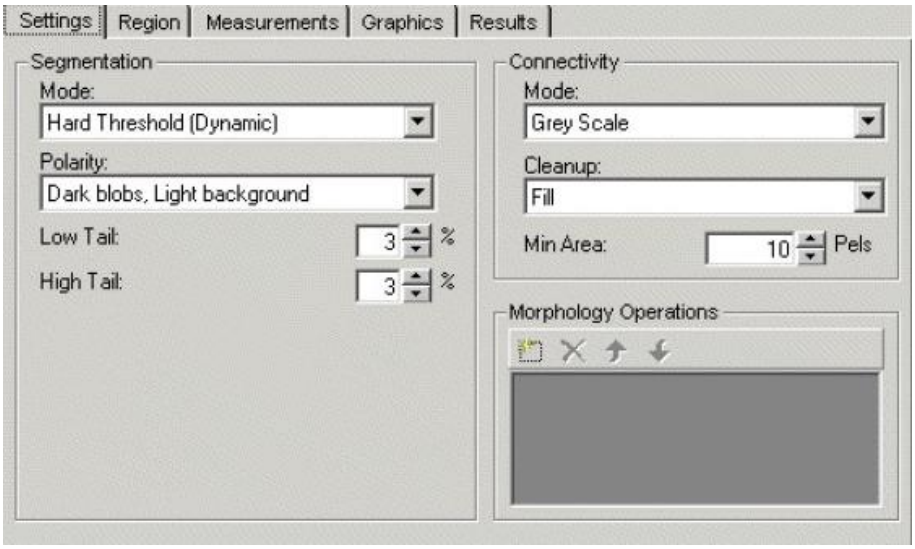
The Blob edit control has the following image buffers. One buffer exposes the underlying Blob tool's InputImage property. All buffers can be shown in both the local and floating tool display windows. Use the Graphics tab to select the blob analysis results that you can display in these buffers. Some of these buffers do not appear unless you select certain results to display. See the Graphics tab for more information.

- The Current.InputImage provides the input image to the Blob tool. This is the **InputImage** buffer.
- The LastRun.InputImage displays the image that the tool last analyzed.
- The Current.Histogram buffer displays a histogram of the current InputImage inside the region of interest, if any. It is generated by the blob tool when it creates its inspection records.
- The LastRun.Histogram buffer displays a histogram within the region of interest of the last image that the tool ran on.
- The LastRun.BlobImageUnfiltered buffer displays an image generated from the extracted blob data.
- The LastRun.BlobImage buffer displays an image generated from the extracted blob data, excluding blobs that were filtered out.

9.2.3 Settings Tab

This section contains the following subsections.

- Segmentation
- Connectivity
- Morphology Operations



Use the Settings tab to define the Segmentation, Connectivity, and Morphology parameters. When the edit control is in electric mode, electric bolt icons indicate parameters whose changes cause the tool to run automatically.

9.2.3.1 Segmentation

Blob analysis provides numerous methods of segmenting an image into object pixels and background pixels. The set of segmentation parameters that are displayed depends on the segmentation mode that is selected. See **CogBlobSegmentationParams** for a description of the parameters. The selection of the segmentation modes is equivalent to the following set of ICogBlobSegmentationParams methods:

- SetSegmentationNone
- SetSegmentationMap
- SetSegmentationHardFixedThreshold
- SetSegmentationHardRelativeThreshold
- SetSegmentationHardDynamicThreshold
- SetSegmentationSoftFixedThreshold
- SetSegmentationSoftRelativeThreshold
- SetSegmentationSubtractionImage

9.2.3.2 Connectivity

Connectivity analysis is the attempt to assemble object pixels (which have been determined by the segmentation criteria on this tab) into connected groups of object pixels or blobs. The connectivity parameters are part of the **CogBlob** interface.

- **ConnectivityMode** determines the connectivity mode used to connect object pixels into blobs.
- **ConnectivityCleanup** determines the cleanup method used after performing connectivity analysis of the blobs.
- **ConnectivityMinPixels** is the minimum size, in pixels, of features that will not be cleaned up during connectivity cleanup.

9.2.3.3 Morphology Operations

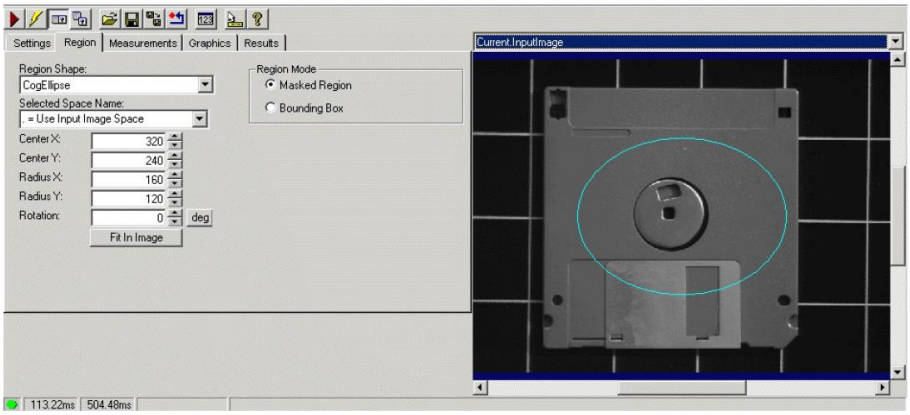
Morphological operations are applied to images to accentuate or minimize particular types of features within the image. The function buttons allow you to add one or more morphology operations to a **CogBlobMorphologyCollection**.
Use the function buttons on this tab to manage the scoring functions. They are described in the following table.

Button	Description	Function
	Add	Adds a new morphology operation. When you click this button, a menu appears with a list of available operations. Highlight the operation that you want and a new line appears in the list of functions.
	RemoveAt	Deletes the currently highlighted morphology operation in the list.
	Move	Moves the currently highlighted operation up in the list, and updates its index in the collection.
	Move	Moves the currently highlighted operation down the list, and updates its index in the collection.

9.2.4 Region Tab

This section contains the following subsections.

- Measurements Tab
- Measure Mode Table
- Sorting Parameters
- Extrema
- Computation Selection of Blob Properties

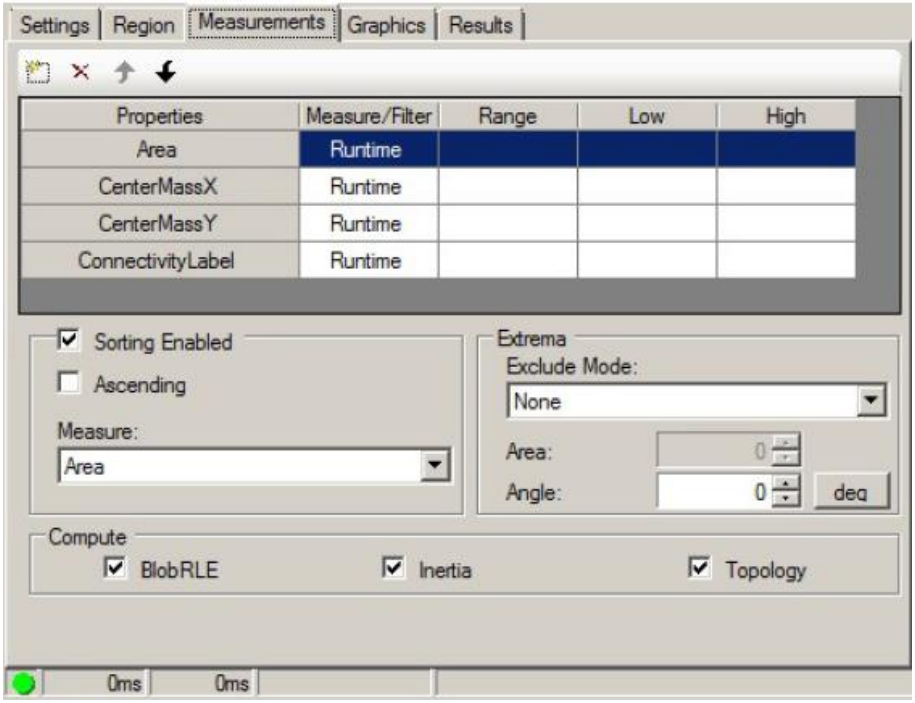


Use the Region tab to define the **Region**. This is the region on which the tool runs blob analysis. If no region is supplied, the tool performs blob analysis on the entire InputImage buffer.
Some of the parameters that specify the train region or train origin can be specified in either degrees (the default specification) or radians. The underlying tool keeps the values in radians but the edit control performs the conversion to degrees when appropriate.

Feature	Description
Region Shape	Select the shape of the input region. Selecting "None=Use entire image" means that the tool uses the entire input image. A Blob tool supports the following input region shapes: • CogCircle • CogEllipse • CogPolygon • CogRectangle • CogRectangleAffine The set of region-defining parameters that appear depend on the region shape you use. For more information on using a polygon as an input region, see the **Using Polygon Input Regions**.

Feature	Description
RegionMode	Defines the bounding box for the region. ▪ Pixel Aligned Bounding Box encloses the defined region within a rectangle. This means that the portions of the image that are outside the defined region but within the bounding rectangle are included. ▪ Pixel Aligned Bounding Box Adjust Mask encloses the defined region within a rectangle, but masks out the image portions that are outside the region but inside the enclosing rectangle. The result is that the defined region is closer to what you specify.
SelectedSpaceName	The coordinate space in which the region is interpreted.
Select Mode	Available when cogRectangle or cogRectangleAffine is the chosen Region Shape. Selects the set of parameters to use to define the rectangle. If cogRectangleAffine is chosen, note that the angle of rotation and skew can be specified in degrees or radians, although the underlying tool keeps the measurements in radians.
FitToImage	Centers the region of interest within the image. Adjust the shape's geometric properties so that its default size is based on the image and its SelectedSpaceName.

9.2.4.1 Measurements Tab



Use the Measurements tab to specify how measurements are made during blob analysis.

9.2.4.2 Measure Mode Table

At the top of this tab is a table of measures that you can add. For a description of the possible measures, see **Measure**. This table is similar in operation to the Morphology Operations table on the Settings tab. This collection of measure modes and this equivalent to the **CogBlobMeasureCollection** interface.
For each measure mode you select, you have a set of properties to specify. See the **CogBlobMeasure** interface for more information.

- **Mode** indicates how measure will be used during blob analysis.
- **FilterMode** is available only if you select Filter. Specifies how the filter range is interpreted -- whether blobs are within range specified or outside of it.
- **FilterRangeLow** is available only if you select Filter. Specifies low end of filter range.
- **FilterRangeHigh** is available only if you select Filter. Specifies high end of filter range.

9.2.4.3 Sorting Parameters

The sorting parameters allow you to sort the filtered collection of blobs, if you wish.

- **SortEnabled** enables the sorting of the filtered collection of blobs.
- **SortAscending** sorts the blobs in ascending order.
- **SortMeasure** is the blob measure that will be used for sorting the filtered collection of blobs.

9.2.4.4 Extrema

Extrema parameters define how the tool determines what pixels at the extreme portions of a blob are not actually part of the blob.

- **ExtremaExcludeMode** defines the mode used to exclude part of a features area when computing a bounding box or median point of a blob.
- **ExtremaExcludeArea** is the amount of area to exclude when computing a bounding box or median point. Depending on the exclude mode selected, this parameter can also be equivalent to the **ExtremaExcludeAreaPercent** or **ExtremaExcludeAreaPixels** property.
- **ExtremaAngle** is the angle that you want the bounding box or median point of a blob to be based on. Although you can specify this value in either degrees or radians, the underlying tool stores this value in radians.

9.2.4.5 Computation Selection of Blob Properties

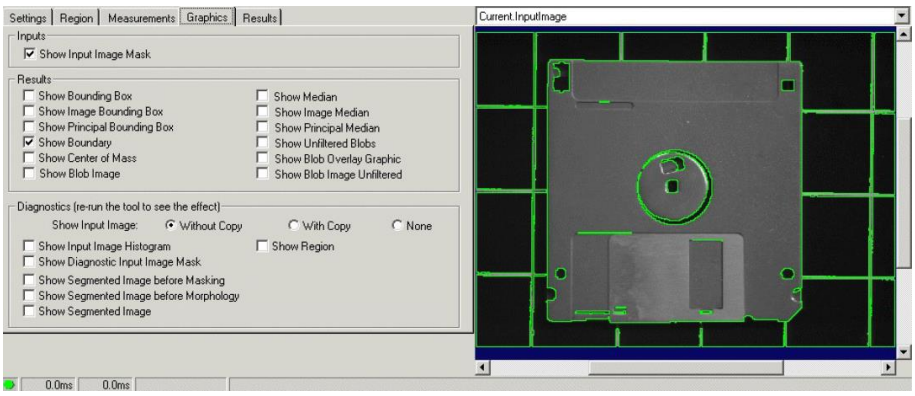
The computation parameters define which blob properties are computed and displayed as computation results. To improve the speed of the tool, you can disable the computation of specific blob results that you do not need. By default, all results are computed. To disable the computation of a specific blob result, uncheck the corresponding checkbox.

- **ComputeBlobRLE** enables saving run lengths of regions in the blob scene description. You must set this to true if additional blob measurements are required, such as region boundary, perimeter, or region image.
- **ComputeInertia** enables the computation of the inertia, principal inertia, angle, and elongation of the regions.
- **ComputeTopology** enables the computation of parent-child relationships between regions. If a requested measurement requires that a computation parameter be enabled but it is not, then the requested measurement is highlighted in red. For example, the CenterMassX and CenterMassY measurements require **ComputeInertia** be set to true.

9.2.5 Graphics Tab

This section contains the following subsections.

- Inputs
- Results
- Diagnostics



Use the Graphics tab to select what results to see from running the Blob tool. Some of the edit control image buffers appear only if certain types of results are selected. The selections on this tab are equivalent to the modes that can be set with the **LastRunRecordEnable** and **LastRunRecordDiagEnable** properties.

9.2.5.1 Inputs

Check Show Input Image Mask to show a graphic representing the run-time mask, if you supplied one. The graphic is shown on the Current.InputImage display.

9.2.5.2 Results

For more information about the possible results that you can select, see **LastRunRecordEnable**. The results appear in the LastRun.InputImage and, if you enable their creation, the LastRun.BlobImage and LastRun.BlobImageUnfiltered buffers.

- Show Bounding Box, Show Image Bounding Box, Show Principal Bounding Box -- displays the bounding boxes of each blob result in the LastRun buffers.
- Show Boundary -- outlines each blob result in the LastRun buffers.
- Show Center of Mass -- highlights the center of mass of each blob result in the LastRun buffers.
- Show Median, Show Image Median, Show Principal Median -- highlights the median of each blob result in the LastRun buffers.
- Show Unfiltered Blobs -- displays all unfiltered blobs and holes in the LastRun buffers.
- Show Blob Overlay Graphic -- displays translucent overlay graphics over blobs and holes in the image.
- Show Blob Image -- generates a synthetic image showing the blobs.
- Show Blob Image Unfiltered -- generates a synthetic image showing the blobs but ignoring the effect of any filters specified in the **Measurements** tab.

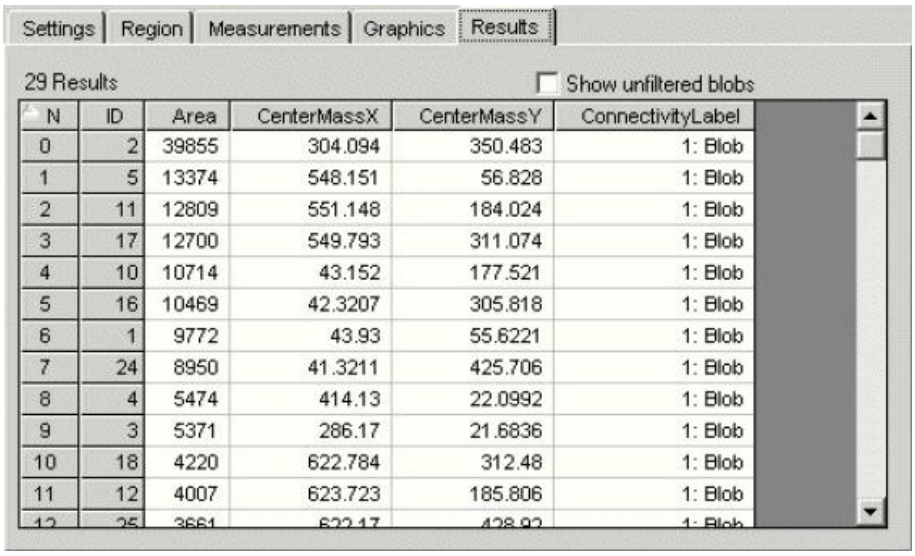
9.2.5.3 Diagnostics

For more information about the diagnostics values you can select, see **LastRunRecordDiagEnable**.

- Show Input Image Histogram -- allows tool to create LastRun.Histogram buffer. If this is not set, then the LastRun.Histogram is not available.
- Show Segmented Image before Masking -- includes a viewable version of the segmented image, before morphology and masking, in the
- LastRun.InputImage. The segmented image graphics appear only in the LastRun.InputImage buffer. If none of the segmented image options is selected, then this buffer is not created. See also **SaveSegmentedImageBeforeMasking**.
- Show Segmented Image before Morphology -- includes a viewable version of the segmented image, before morphology, in the LastRun.InputImage. See also **SaveSegmentedImageBeforeMorphology**.
- Show SegmentedImage -- includes a viewable version of the segmented image in the LastRun.InputImage. See also **SaveSegmentedImage**.
- Show Region displays the region of interest in the LastRun.InputImage, if it is available.
- Show Diagnostic Search Image Mask displays the run-time mask, if one was specified.

The Show Input Image option buttons let you specify whether a reference to the input image or a deep copy of the input image is displayed for the LastRun.InputImage. You can also specify that no image be displayed.

9.2.6 Results Tab

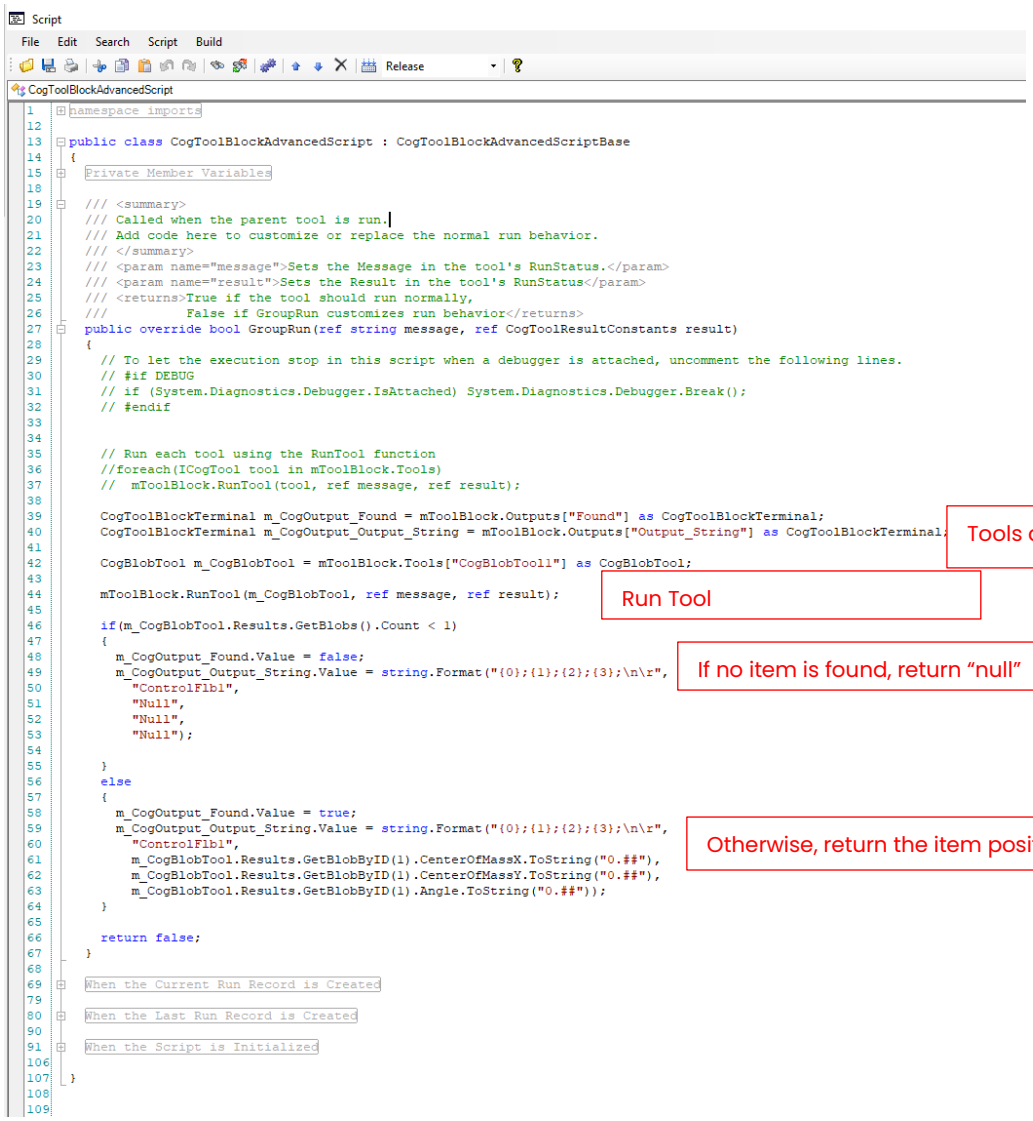


N	ID	Area	CenterMassX	CenterMassY	ConnectivityLabel
0	2	39855	304.094	350.483	1: Blob
1	5	13374	548.151	56.828	1: Blob
2	11	12809	551.148	184.024	1: Blob
3	17	12700	549.793	311.074	1: Blob
4	10	10714	43.152	177.521	1: Blob
5	16	10469	42.3207	305.818	1: Blob
6	1	9772	43.93	55.6221	1: Blob
7	24	8950	41.3211	425.706	1: Blob
8	4	5474	414.13	22.0992	1: Blob
9	3	5371	286.17	21.6836	1: Blob
10	18	4220	622.784	312.48	1: Blob
11	12	4007	623.723	185.806	1: Blob

This tab lists the collection of **CogBlobResult** from the blob analysis.

- **Count** is the index of this blob result in the blob result collection.
- **ID** is a unique ID that identifies this blob result in the blob result collection.
- **Area** is the area of the blob specified in units of the input image's selected space at run-time.
- **CenterOfMassX** is the x-coordinate of the center of mass of the blob specified in units of the input image's selected space at run-time.
- **CenterOfMassY** is the y-coordinate of the center of mass of the blob specified in units of the input image's selected space at run-time.
- **ConnectivityLabel** essentially indicates which result is a blob and which is a hole. This definition is dependent on the polarity setting.
- **GetBlobs** determines whether you include unfiltered blobs in the list.

9.3 Standard script

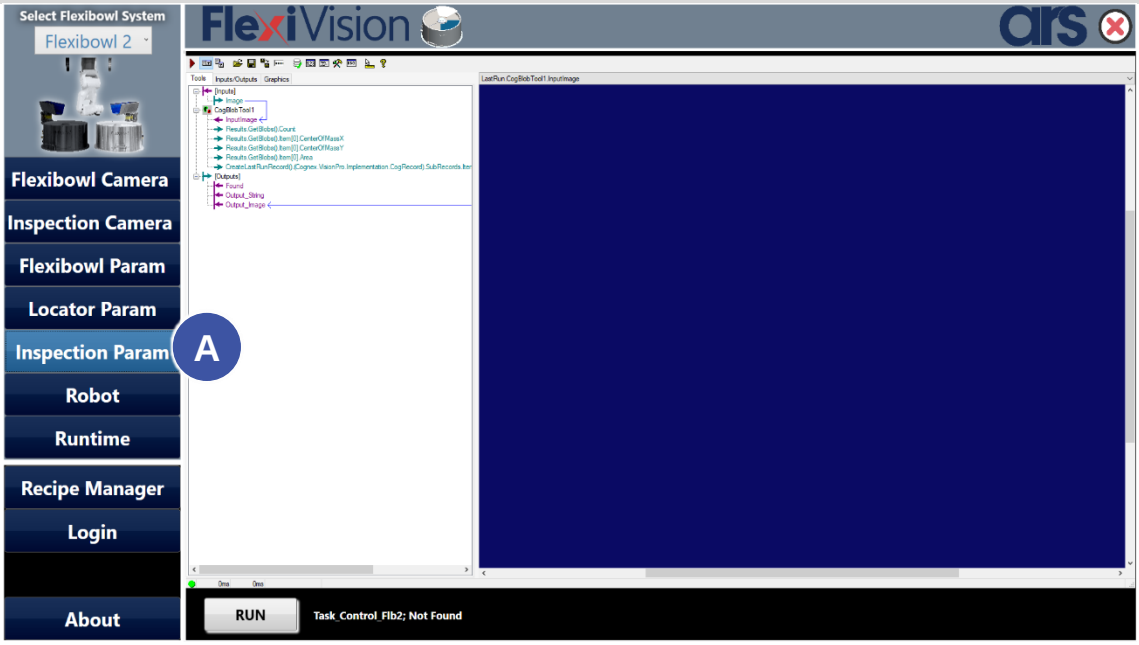


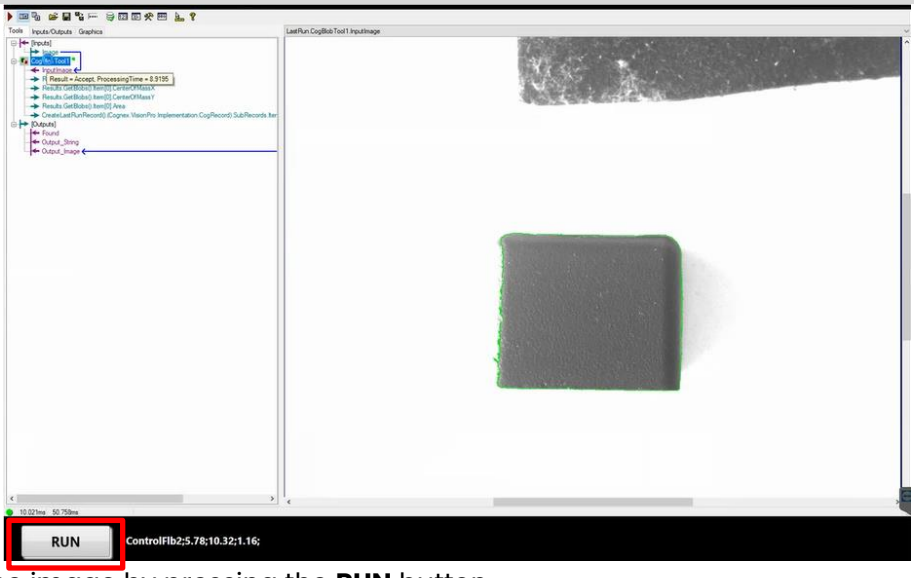
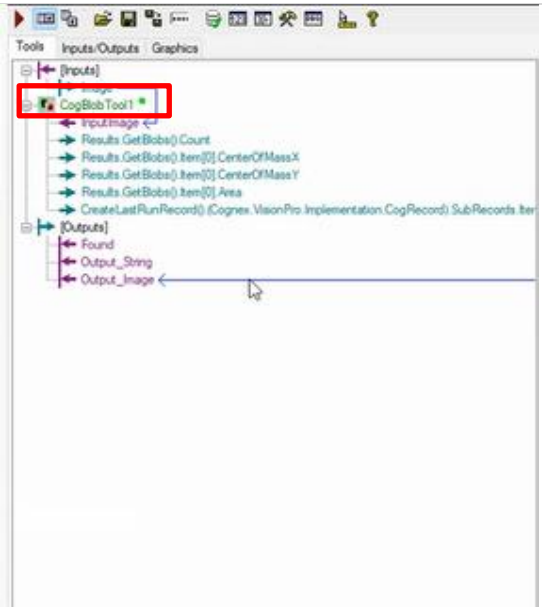
9.4 Inspection Param procedure - example

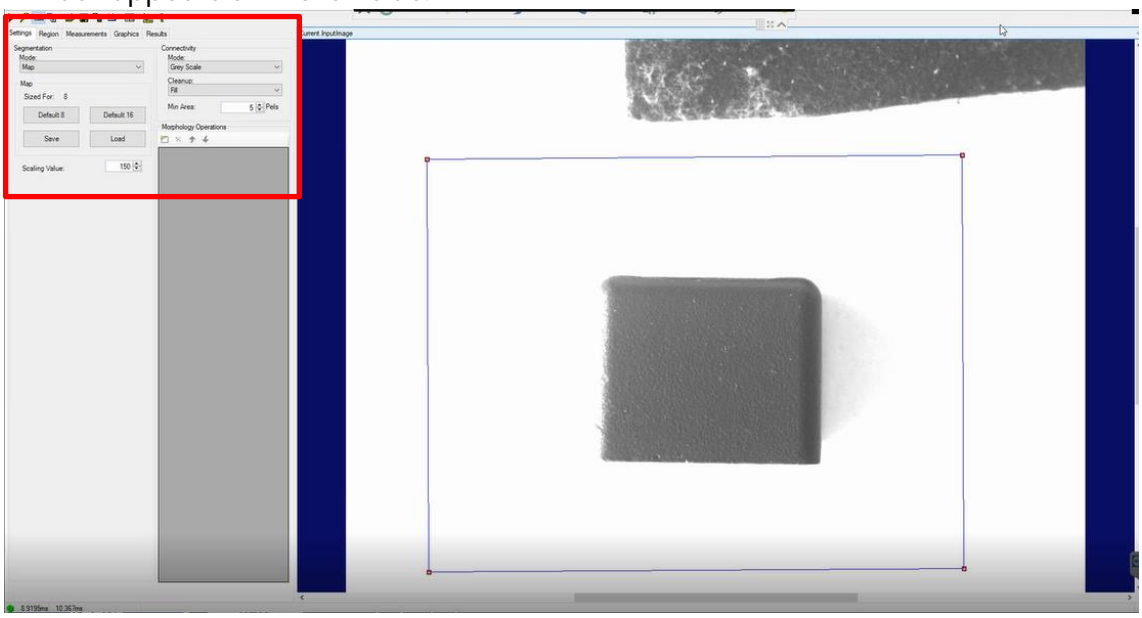
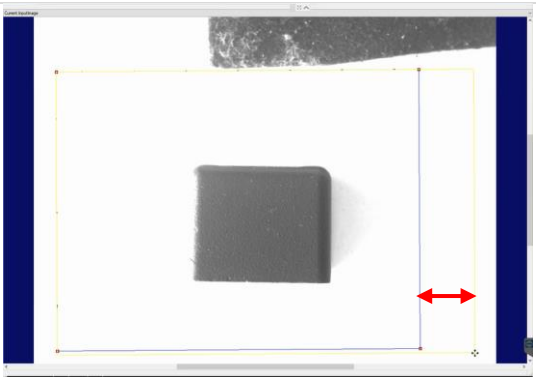
9.4.1 How to carry out the Inspection Param procedure

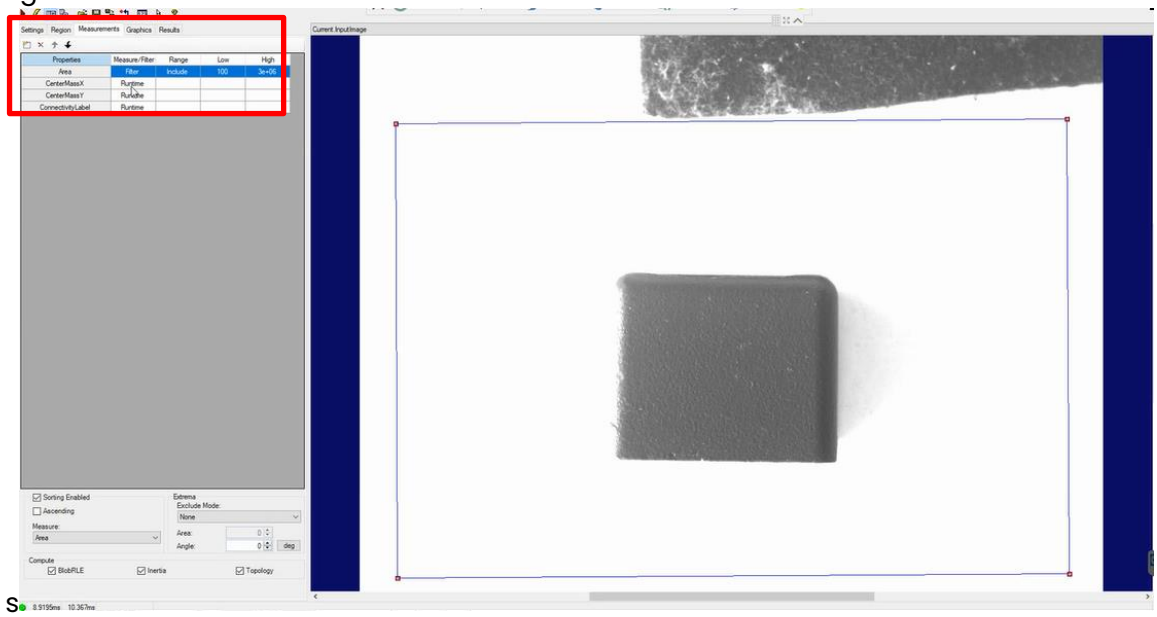
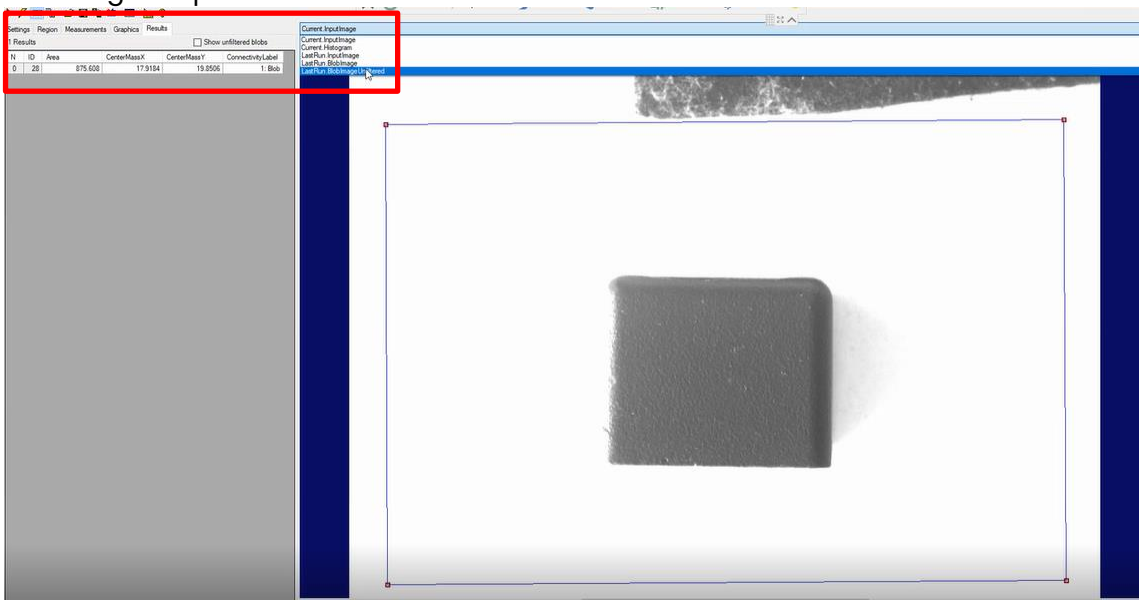
NOTE
This procedure can be carried out by the following users:
- **TECHNICIAN**
- **ARS**

NOTE
This procedure allows to carry out further inspections on the item picked up on the Flexibowl.

Step	Action	Notes/Pictures
1	 Click on *Inspection Param* (**A**) of the operations menu.	

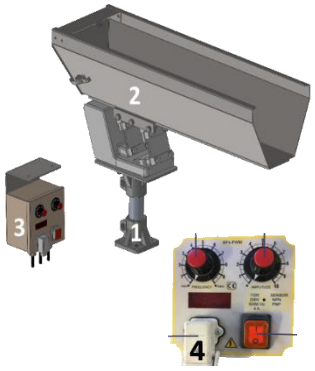
Step	Action	Notes/Pictures
2	 Update the image by pressing the **RUN** button.	
3	Doubleclick on **CogBlobTool1**.	

Step	Action	Notes/Pictures
4	A mask appears on the left side. 	
5	By dragging the vertexes, define the region for image acquisition	

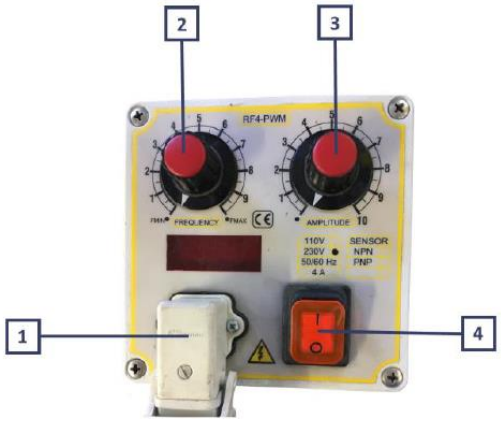
Step	Action	Notes/Pictures
6	From the Measurement menu, apply a filter which includes areas between the lower an higher set values. 	
7	Press 	
8	Enter the Results menu, select the LastRun Blob Image on the upper side of the window: the image is updated. 	

10 HOPPER/FLOW FEEDER

10.1 Installation

Step	Action	Notes/Pictures
1	Place the support (**1**) on the machine (use M8 screws to anchor it firmly).	
2	Place the vibrating base (**2**) on the support (**1**) fastening it firmly with the screws.	
3	Place the Controller (**3**) in a suitable place	
4	Connect the system to the power supply 220Vac +/- 5% (110Vac upon request) and connect the cable of the base to the outlet connector of the Controller (**4**).	

10.2 How to start the bulk feeder

Step	Action	Notes/Pictures
1	Connect the cable of the linear base to the outlet connector of the controller, then connect the vibrator to the outlet connector (**1**).	
2	Turn the frequency adjustment (**2**) and amplitude adjustment (**3**) knob of the controller to "•".	
3	Turn on the controller with the ON/OFF button (button at position 1 (**4**))	
4	Slowly turn the adjustment knobs (**2** and **3**).	

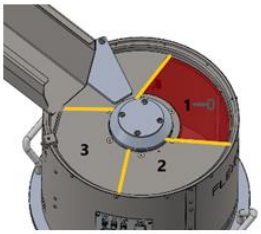
NOTE
Before bringing vibration to maximum (Amplitude Potentiometer (3)) it is recommended to look for the maximum possible amplitude using the Frequency potentiometer (2).

NOTE
Check the dedicated manual for electric connection and potential adjustments of the inner trimmers.

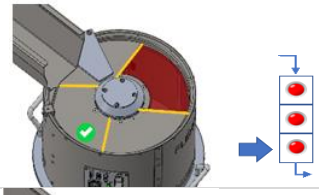
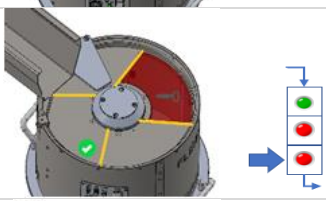
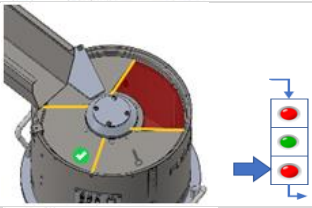
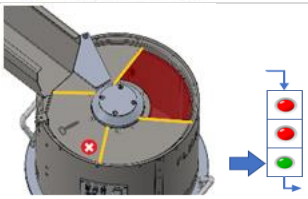
10.1 Bulk flow control

In the following, the procedure to have a constant part-flow on the FlexiBowl® (or equivalent) surface is described.

Example: let's assume a 60-degree forward movement for the disc and 3 steps necessary to shift from picking area to hopper dropping area.



A FIFO register (shift register) stores the histogram result acquired from the camera above the picking area. The status of the last register turns the hopper ON/OFF.

Time	Action	Notes/Pictures
T0	At time T0, no parts are left in the pick area Fifo array [**3**] is false, the hopper is activated. **Fifo array [1] is false.**	
T1	At time T1, there are parts left in the pick area, Fifo array [**3**] is false, the hopper is activated. **Fifo array [1] is true.**	
T2	At time T2, no parts left in the pick area, Fifo array [**3**] is false, the hopper is activated. **Fifo array [1] is false.**	
T3	At time T3, no parts left in the pick area, Fifo array [**3**] is true, the hopper isn't activated. **Fifo array [1] is false.**	

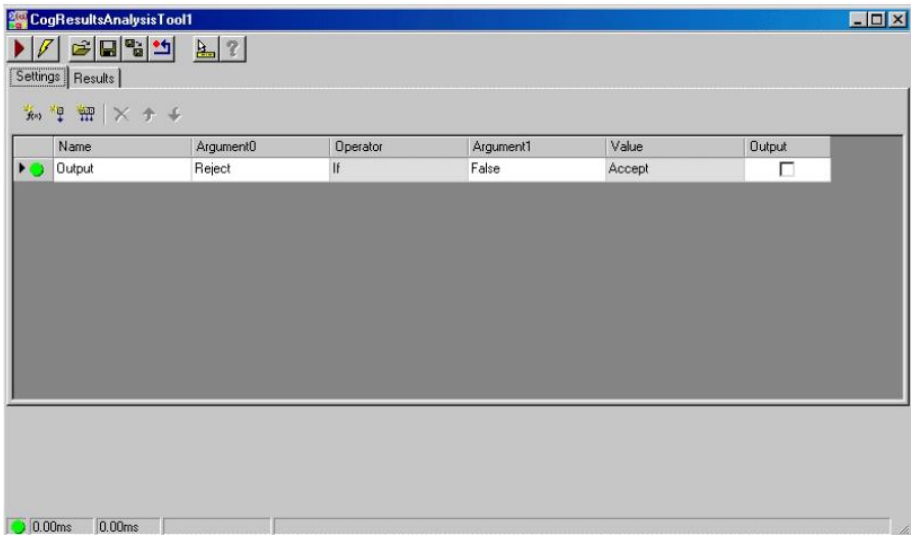
10.2 Results Analysis Edit Control

The following is an extract from the Cognex User Manual: for further and complete details, please refer to the whole Cognex User Manual.

This topic contains the following sections.

- Control Buttons
- Settings Tab
- Results Tab

The Results Analysis edit control provides a graphical user interface to the Results Analysis tool, which you use to define a set of expressions that will allow the most recent run of the tool group to give a passing, warn-level, or reject-level result. Using a Results Analysis tool, you can combine the results from one, several, or all the vision tools in a tool group and generate a value that can be used to judge whether the tool group generates a Warn or Reject status. VisionPro ultimately uses this Warn or Reject status to determine the value of the RunStatus property for the tool group. The following figure shows a Results Analysis edit control before any expressions have been created:



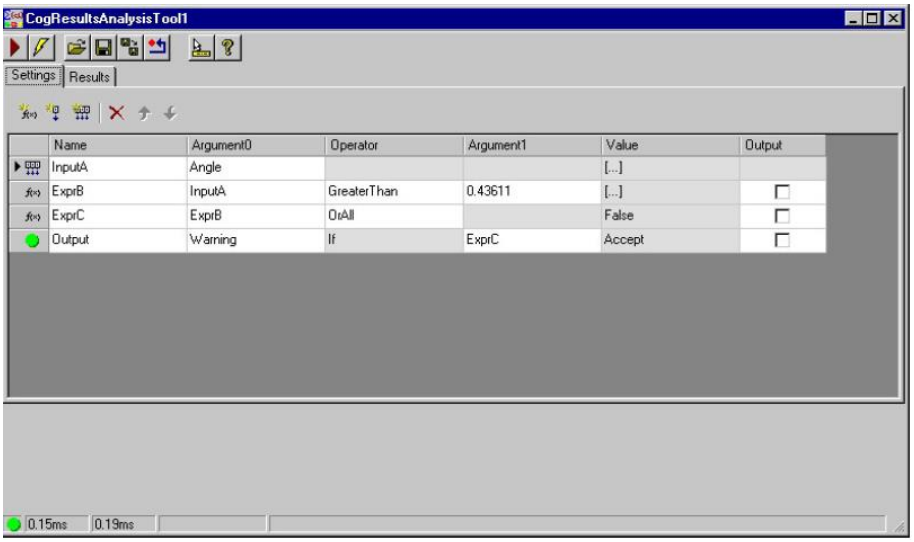
The edit control offers a Settings tab for defining the expressions that evaluate the results of other vision tools, and a Results tab for viewing the results of those expressions.

10.2.1 Control Buttons

Button	Description	Function
	Run	Reevaluate all the expressions currently defined for this Results Analysis tool and generate a new Accept, Warn, or Reject result.
	Electric mode	Toggle electric mode, where the Results Analysis tool executes automatically when you change the value of any argument to an expression.
	Open	Open a VisionPro persistence (.vpp) file that contains a set of saved properties for this vision tool object type. VisionPro reports an error if you try to open a .vpp file for another object type.
	Save	Save the current properties of the vision tool to a VisionPro persistence (.vpp) file. The edit control allows you to choose between saving the vision tool with or without its image buffers and tool results.
	Save As	Saves the current properties of the underlying tool to a new VisionPro persistence file.
	Reset	Reset the vision tool to its default state.
	Show ToolTips	Enable or disable the display of tooltips for individual items in the edit control.
	Help	Open this VisionPro online help file.

10.2.2 Settings Tab

Use the Settings tab to define the arguments and expressions you need to evaluate the results of other vision tools. The topic **Using Results Analysis** contains an example of defining the criteria for an application using the Settings tab. The following figure shows a Settings tab with one argument and two expressions:

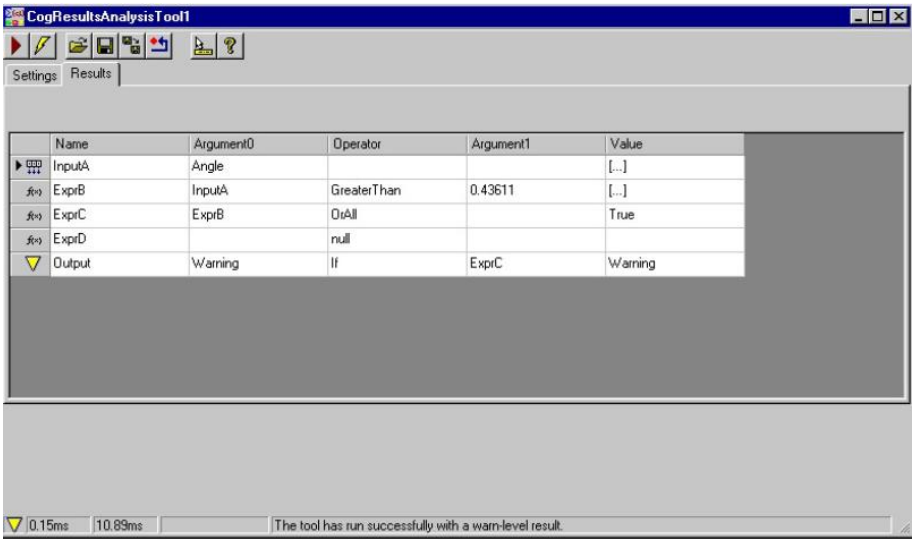


Use the following icons to add arguments and expressions to the Settings tab:

Button	Description	Function
	Add a new expression	Add a new expression.
	Add an expression	Add an expression.
	Add a value	Add a new value input, which can be any numeric, string, or boolean result from a vision tool in your application.
	Add a vector input	Add a vector input, which is the entire array of values in a collection.
	Delete	Delete the currently selected input or expression from the Settings tab.
	Move up	Move the currently selected input or expression up within the Settings tab.
	Move down	Move the currently selected input or expression down within the Settings tab.

10.2.3 Results Tab

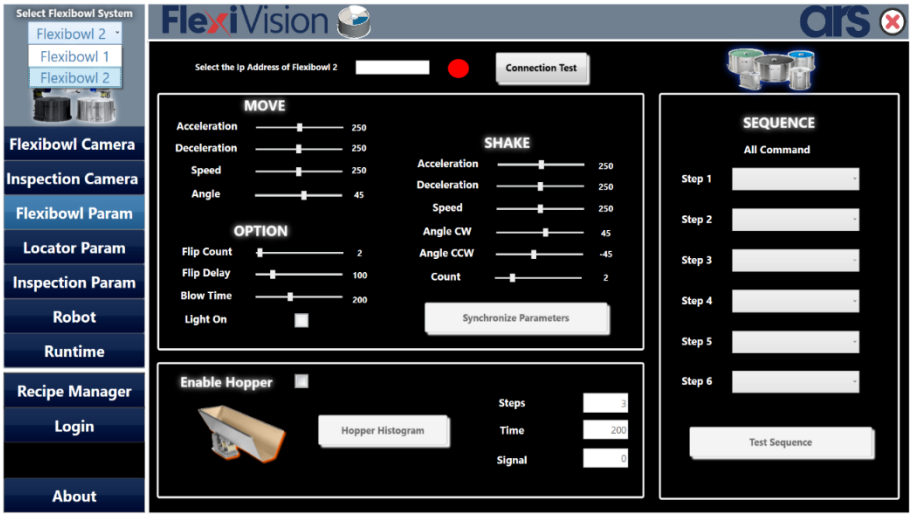
Use the Results tab to view the value of any input value or the result of any expression. The following figure shows an example Results tab after each the Results Analysis tool has executed:

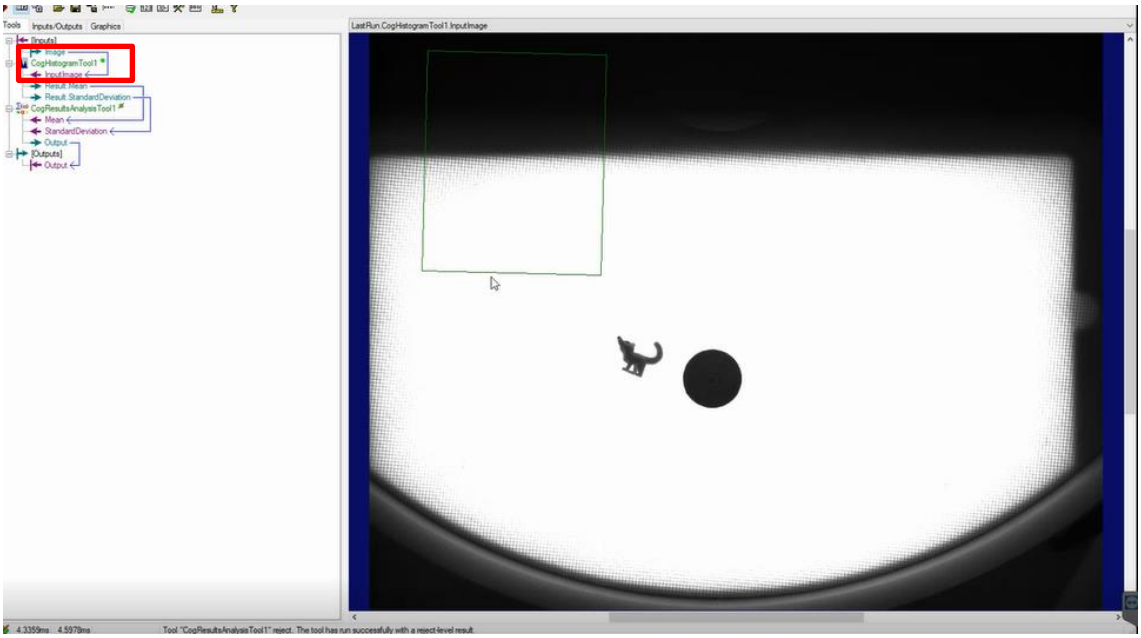
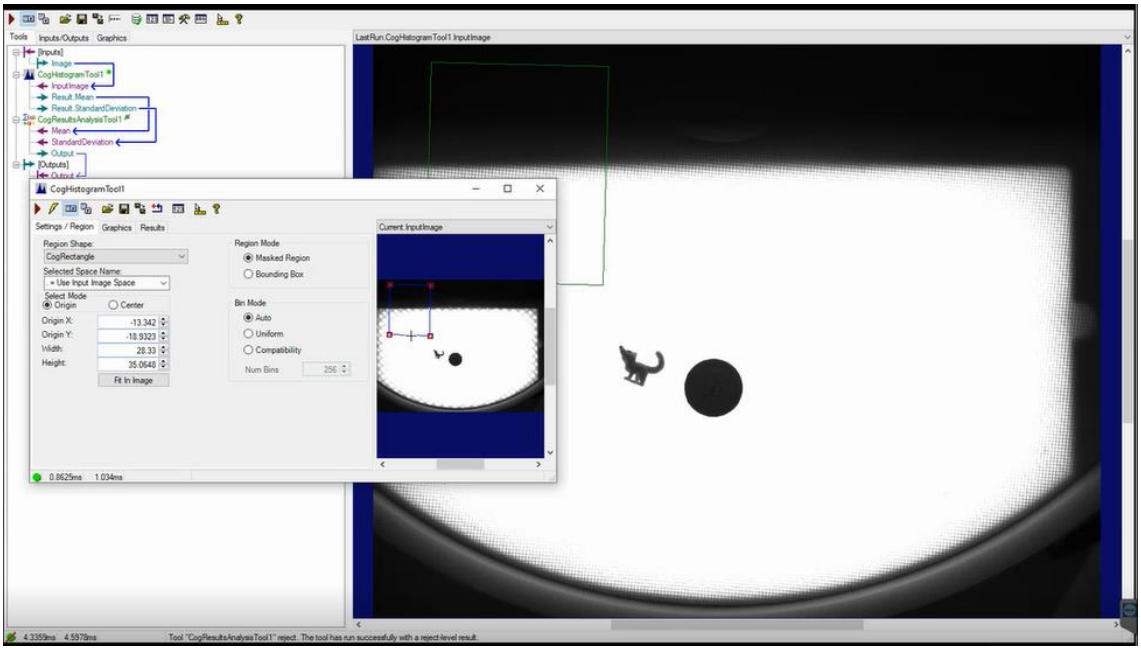


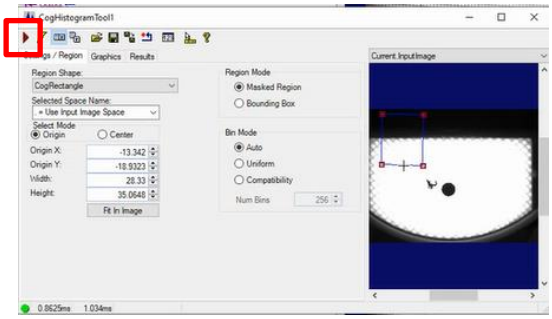
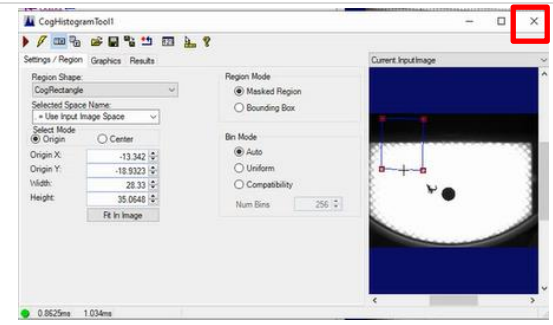
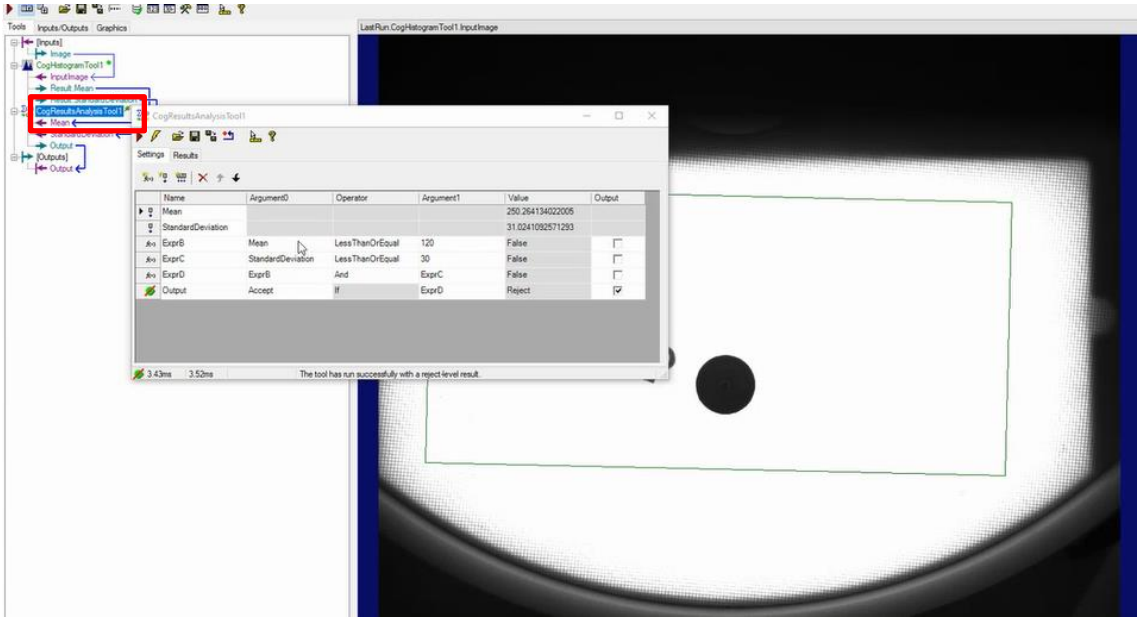
The tab updates the contents of each input value and expression after each execution of the QuickBuild job.

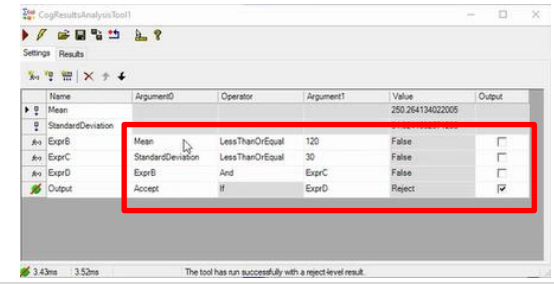
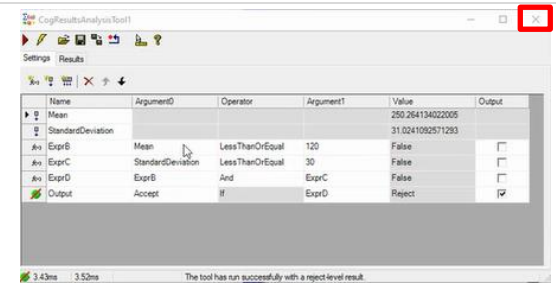
10.3 HOPPER hystograms

NOTE
Hopper parameters are set in the FLEXIBOWL PARAM page (see chapter 5).

Step	Action	Notes/Pictures
1	Enter the FLEXIBOWL PARAM page 	

Step	Action	Notes/Pictures
2	Press **Hopper Histogram**	
3	This dialogue window opens. Doubleclick on **CogHistogramTool1.** 	
4	The hopper histogram shall control the Flexibowl region to check if there are items or not (that is if a sector is filled or if it is empty and has to be filled) 	

Step	Action	Notes/Pictures
5	Modify the region by the **Settings/Region** instruments and/or by dragging the vertexes.	NOTE The control histogram has to cover the widest possible area.
6	Press  to update the image.	
7	Close by .	
8	Doubleclick on **CogResultsAnalysisTool1.** 	

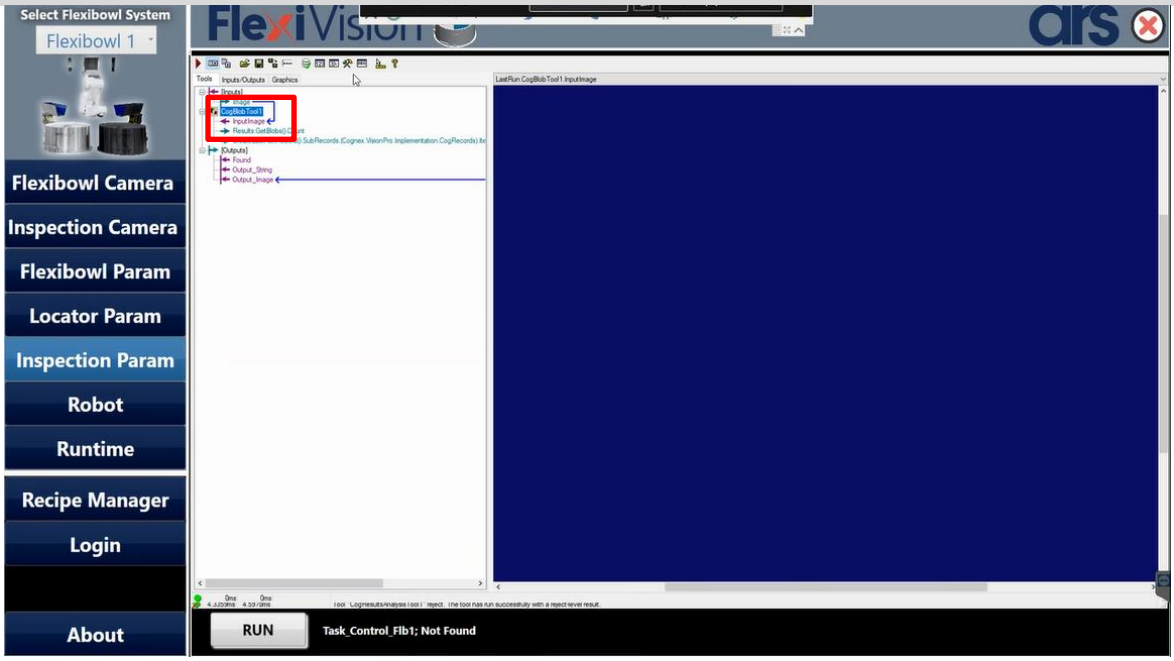
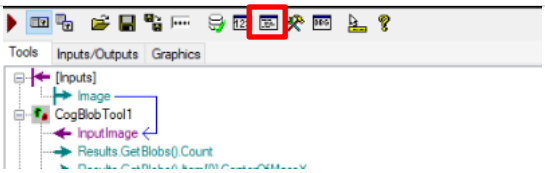
Step	Action	Notes/Pictures
9	Check the mean and standard deviation values (or their combination) and set the conditions to define a sector as “filled” or “empty”.	
10	Press  to update the image.	
11	Close by  .	

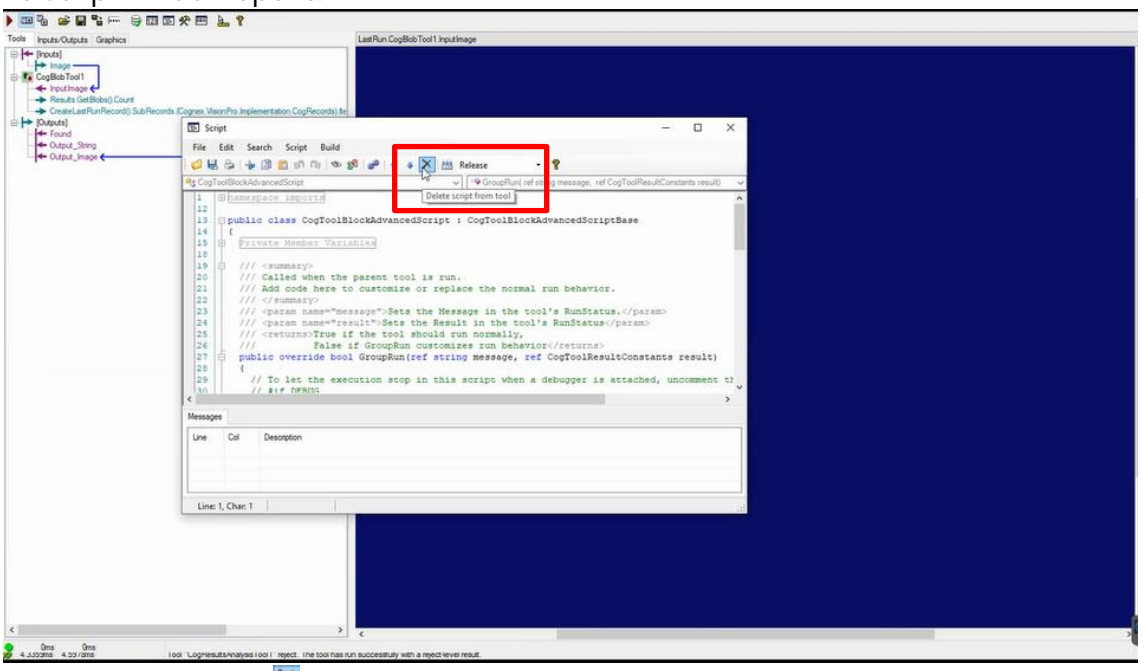
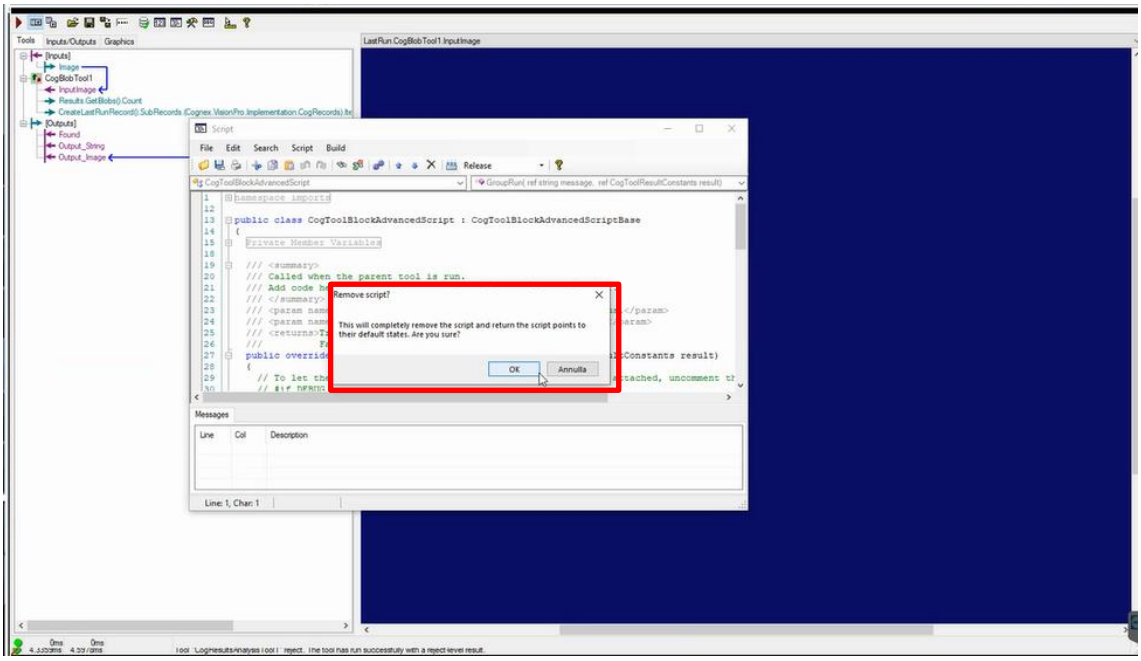
PAGE INTENTIONALLY LEFT BLANK

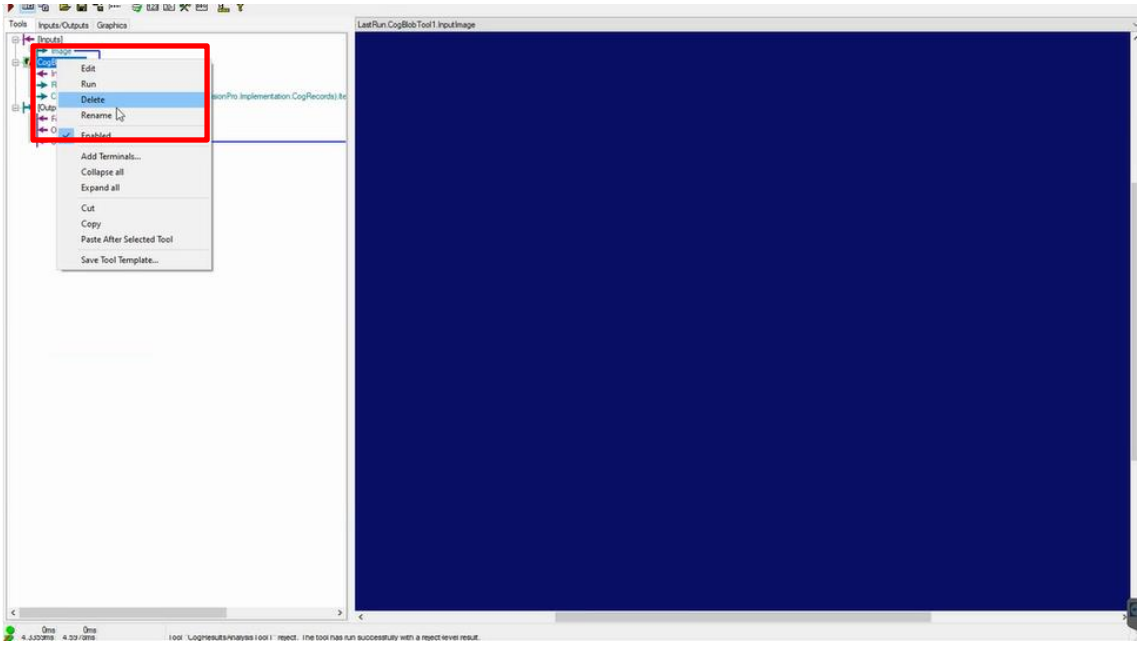
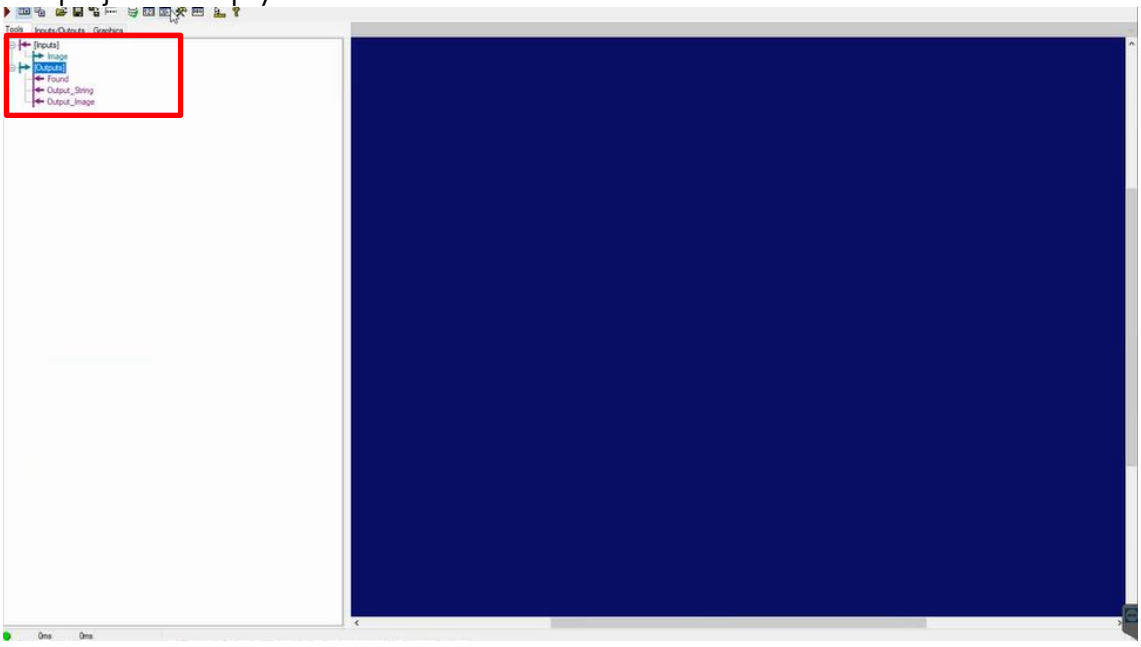
11 INSPECTION AND RECOGNITION TOOLS

NOTE
These procedures can be carried out by the following users:
- **TECHNICIAN**
- **ARS**

11.1 How to delete a tool

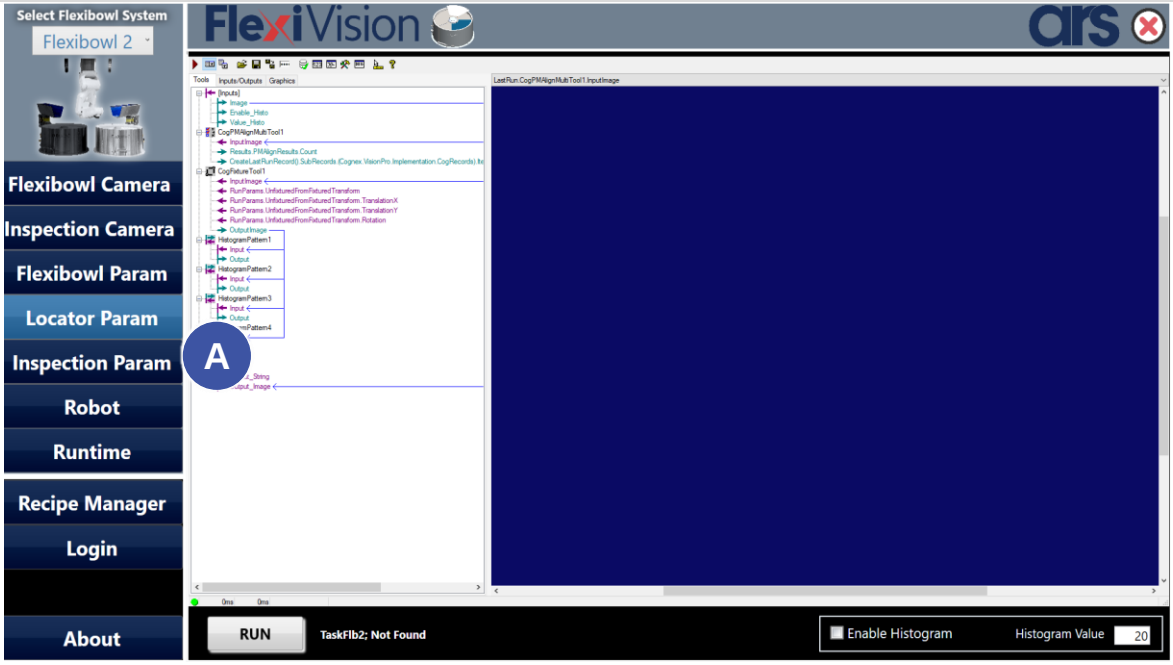
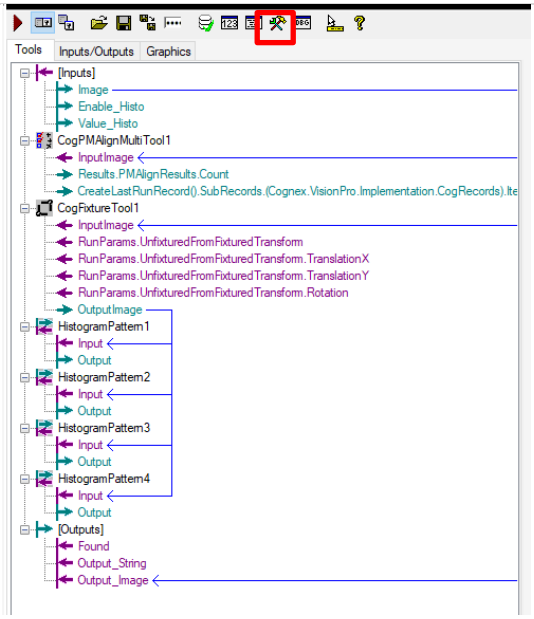
Step	Action	Notes/Pictures
1	 Select the desired tool.	
2	On the button bar, doubleclick on the **Script** icon .	

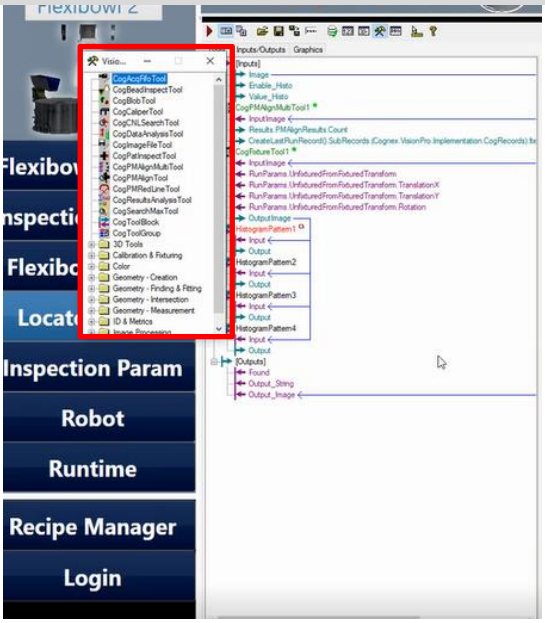
Step	Action	Notes/Pictures
3	The Script window opens.  Press the **Delete** icon to delete the script from the tool.	
4	Press OK: the script shall be removed completely and the script points shall return to their default status. 	

Step	Action	Notes/Pictures
5	Select the tool and delete it from the edit menu. 	
6	The project is empty. 	

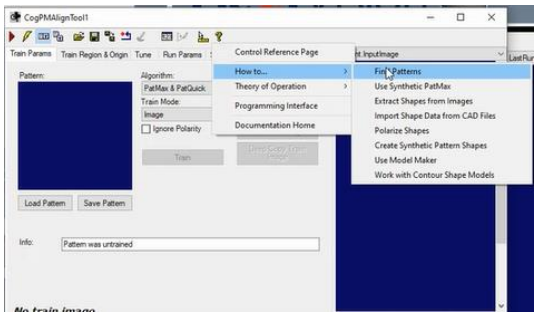
11.2 How to add or modify an inspection or recognition tool

11.2.1 How to add a tool

Step	Action	Notes/Pictures
1	Click on the required field (e.g. *Locator Param)* (**A**) of the operations menu.	
2	On the button bar, doubleclick on the **Show Tool Box** icon .	

Step	Action	Notes/Pictures
3	The side menu appears.	
4	Choose the desired tool.	
5	Drag its inside the program flow.	
6	Drag **image** to i**nput image** (to connect the image output to the tool input)	

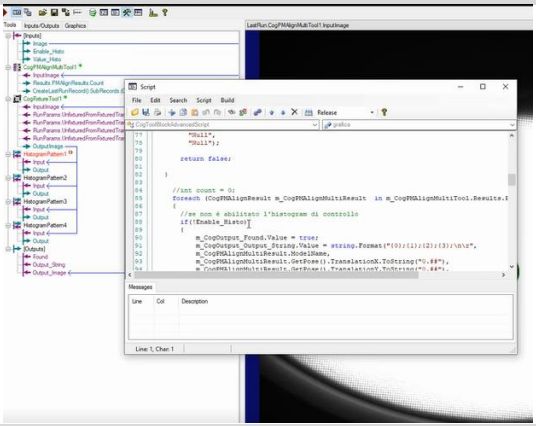
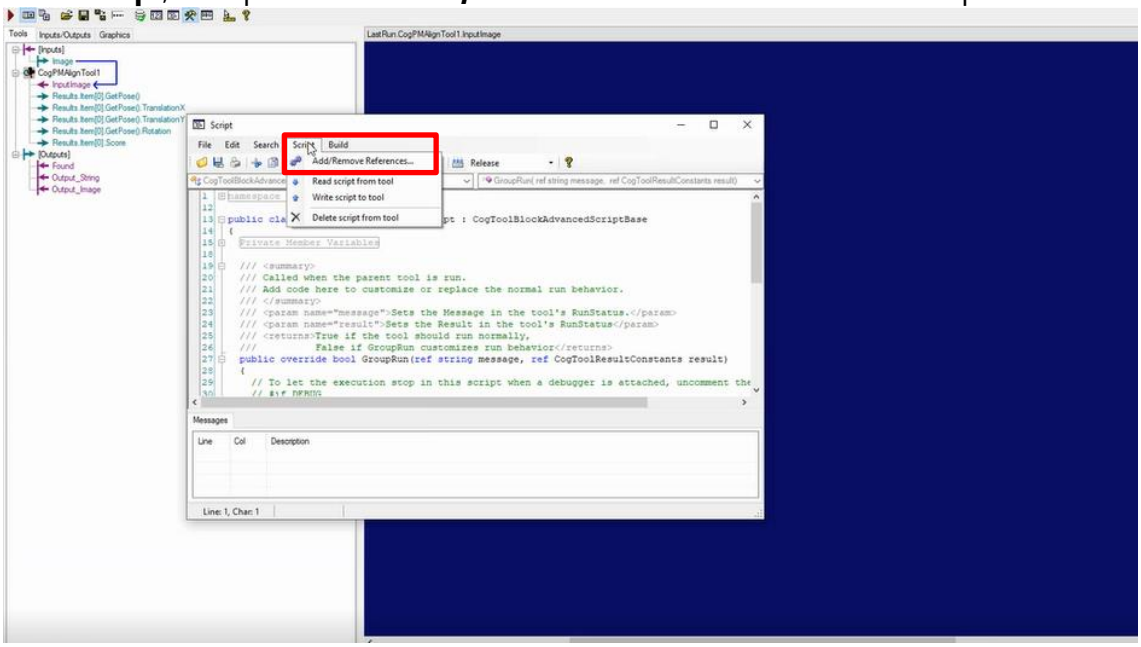
11.2.2 How to set the tool

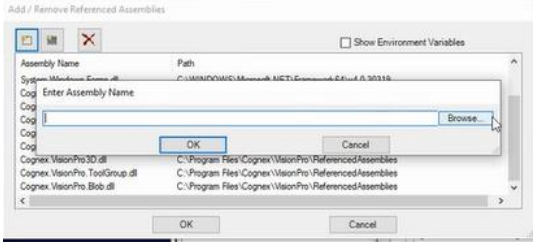
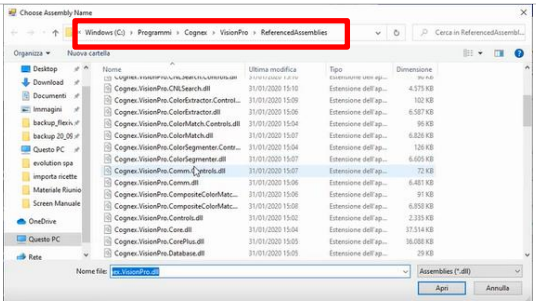
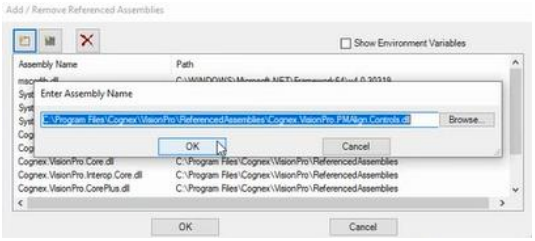
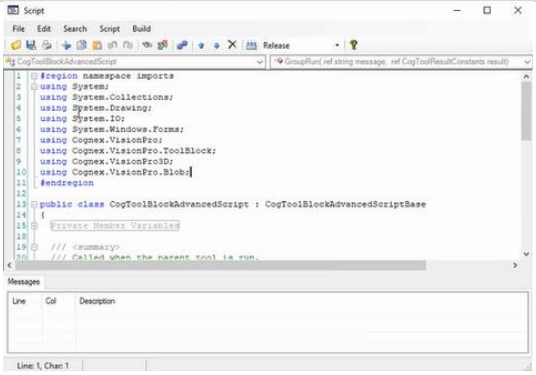
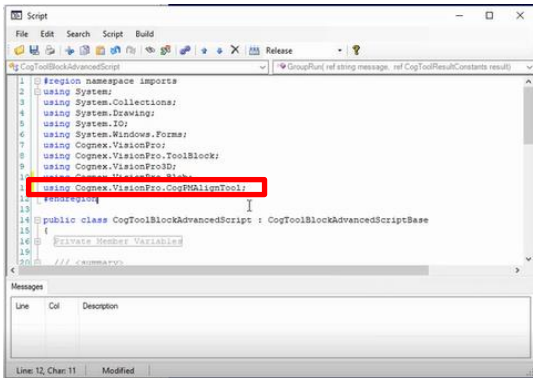
Step	Action	Notes/Pictures
1	Carry out the tool setting.	**NOTE** **For complete details about tools setting, refer to the relevant COGNEX procedure.** To enter the COGNEX instruction: - doubleclick on the desired tool; - press the **HELP** icon ; - select **HOW TO** on the side menu; - choose the instruction. 

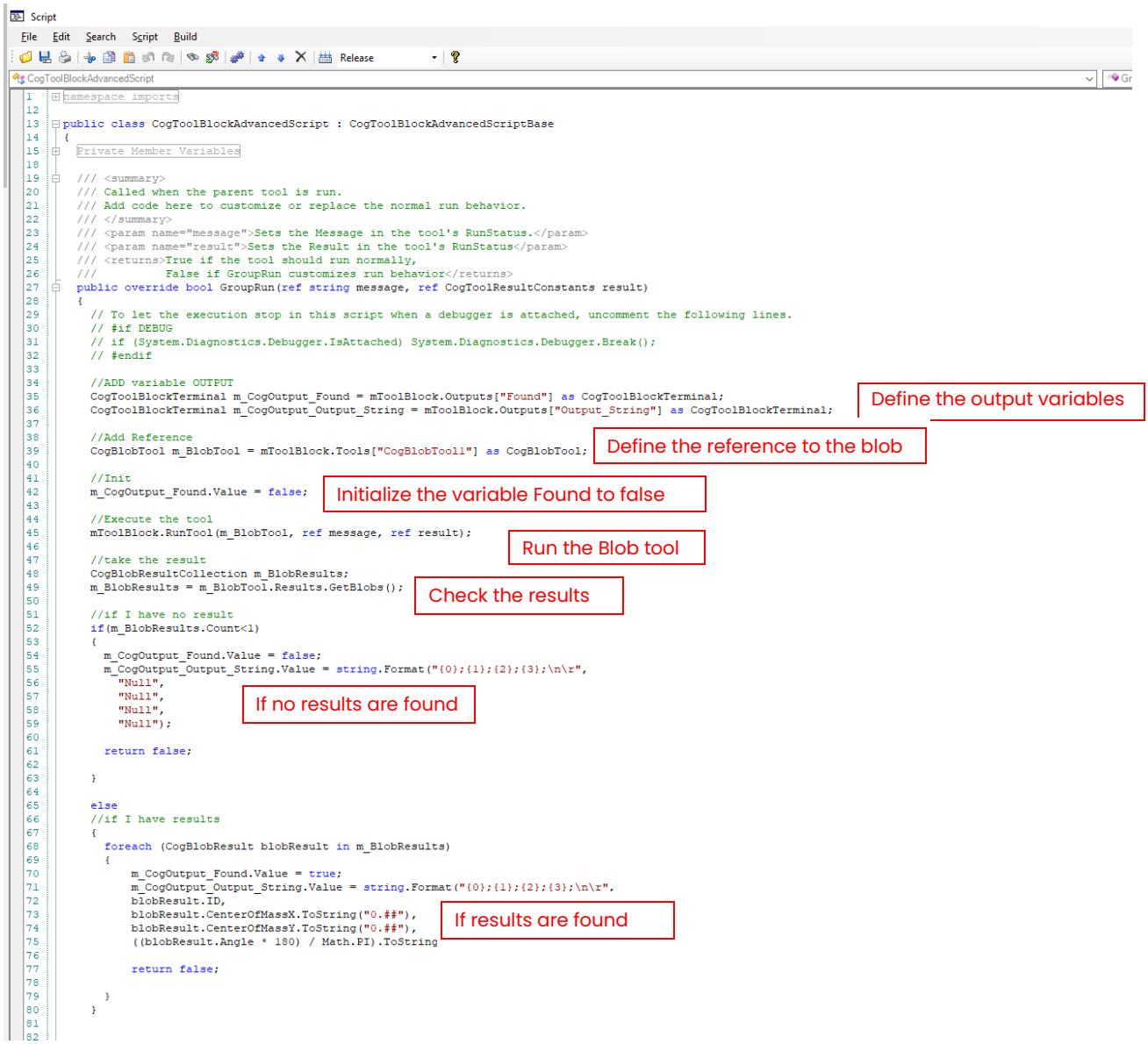
11.2.3 How to edit the script

Configure the tool by editing the script means replacing the variables related to the old tool with the variables related to the new one.
There are three output variables:

- **Found** (a bool to understand if the VisioTool is successfull or if it is necessary to turn the Flexibowl)
- **Output_String** (the output string which shall be sent to the robot)
- **Output_Image** (the tool output image which shall be shown in the RunTime window)**.**

Step	Action	Notes/Pictures
1	Once that tool is set, doubleclick on the **Script** icon of the button bar. This window opens.	
2	Select **Script**, then press the **Add/Remove References** icon from the top bar.	

Step	Action	Notes/Pictures
3	Press the  to enter the **Enter Assembly name** mask, then press **Browse**.	
4	Select the tool from the ReferenceAssemblies folder.	 
5	Update the script by adding the libraries related to the new tools.	 



11.2.4 How to set the control histograms

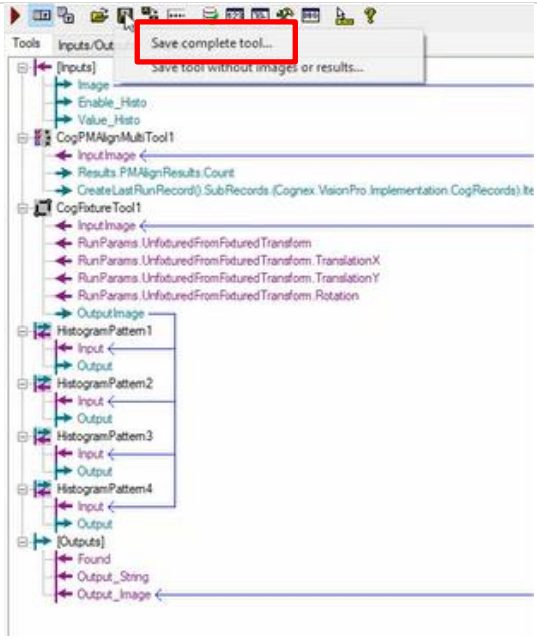
NOTE
It is necessary to add also a "CogFixtureTool", to enable the use of control histograms.
See Paragraph 8.3 for details.

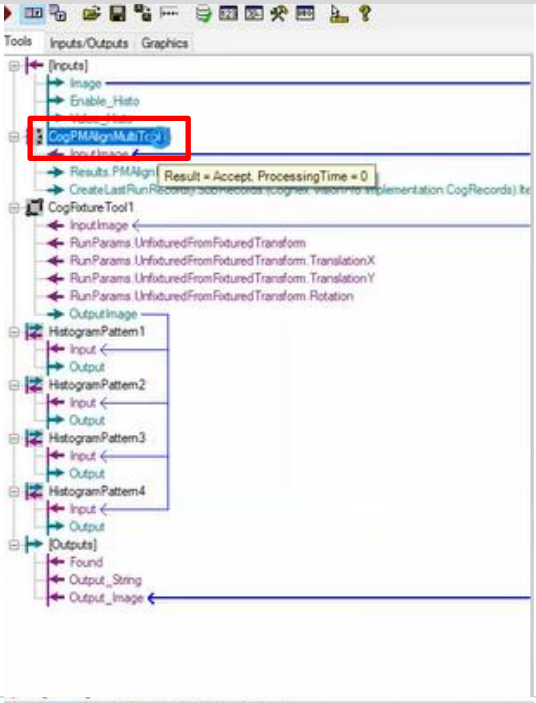
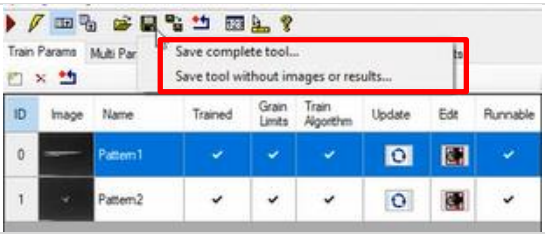
11.2.5 How to set the input image to the new tool

NOTE
See Chapter 12 - RUNTIME.

11.3 How to export tools

NOTE
This procedure can be useful for back up.

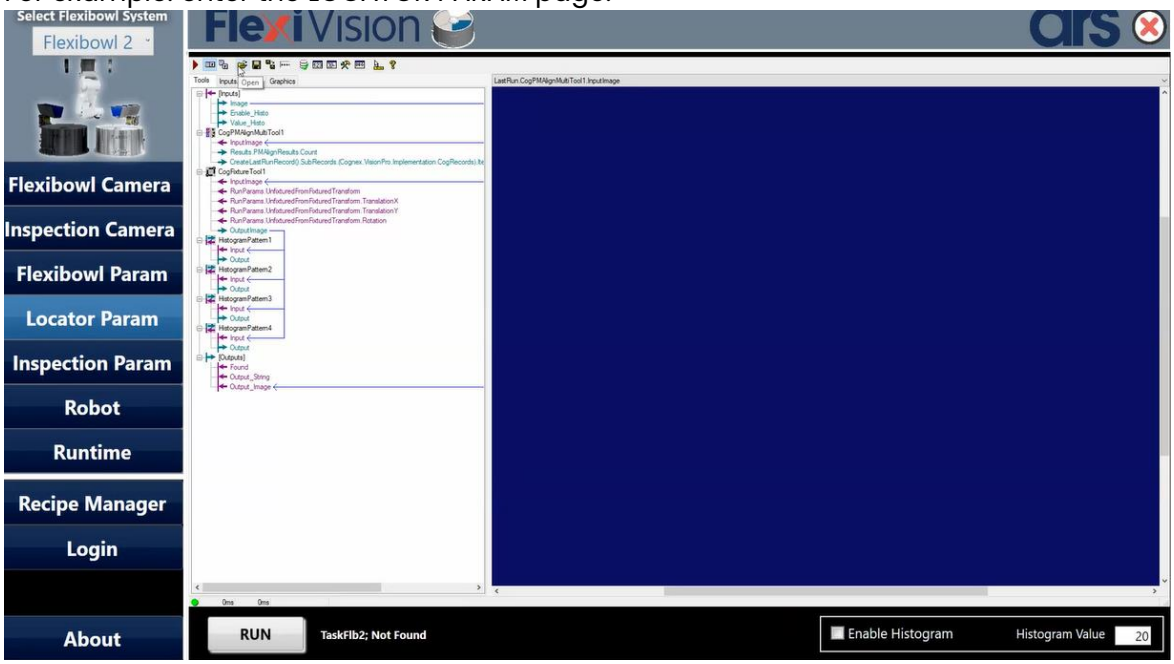
Step	Action	Notes/Pictures
1	Press the **SAVE** icon on the top bar.	
2	Select **SAVE COMPLETE TOOL**.	
3	Define a filename and a destination.	This file shall include all the general structure, shown above (tools and related script block).

Step	Action	Notes/Pictures
4	Save the parameters setting for each tool. As an example: Doubleclick on **CogPMAlignMultiTool** to open the Patterns mask.	
5	Select a Pattern and press the SAVE icon .	
6	Select SAVE COMPLETE TOOL.	
7	Define a filename (.vpp) and a destination.	
8	Repeat for each pattern (using different filenames).	

11.4 How to import tools

NOTE
This procedure can be useful in case of restoring data from a back up.

NOTE
This procedure can be carried out also for camera settings and calibration.

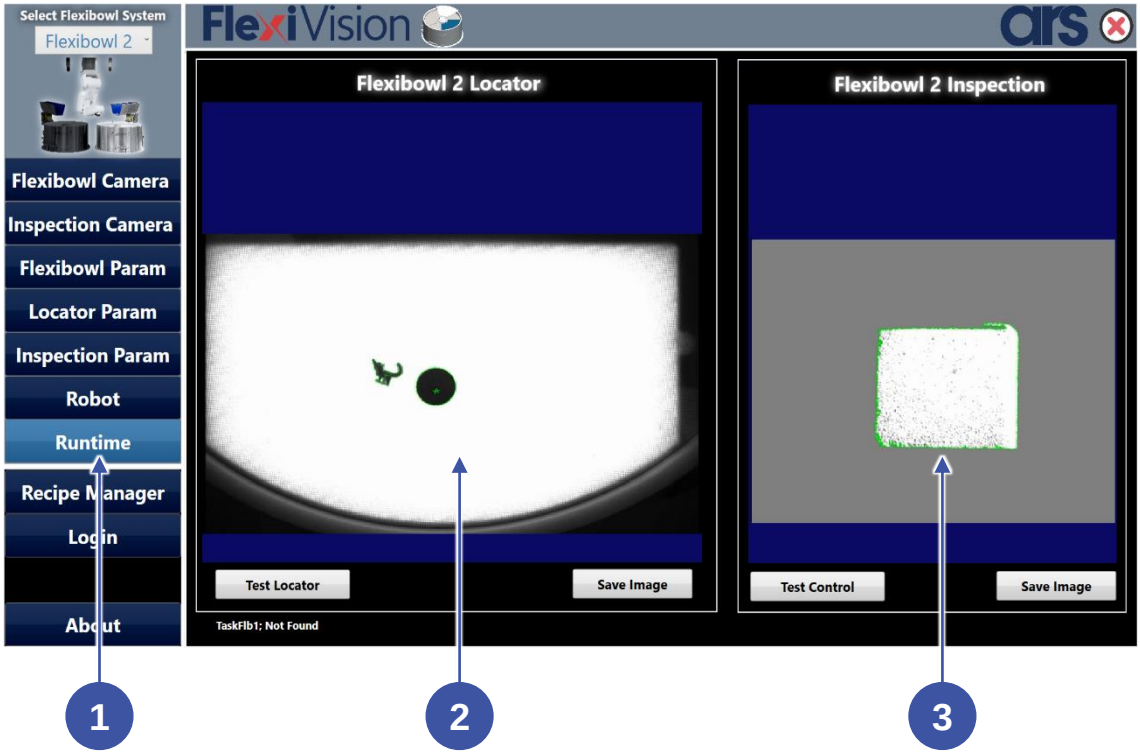
Step	Action	Notes/Pictures
1	For example: enter the LOCATOR PARAM page. 	
2	Press the OPEN icon on the top bar.	
3	Select the files related to the required tool.	
4	Doubleclick on **CogPMAlignMultiTool** to open the Patterns mask.	
5	Select a Pattern and press the OPEN icon .	
6	Repeat for each Pattern.	

12 RUNTIME

12.1 RUNTIME page

NOTE
Access to this page is allowed to the following users:
- **USER**
- **TECHNICIAN**
- **ARS**

By pressing RUNTIME in the OPERATIONS menu, the following page opens:



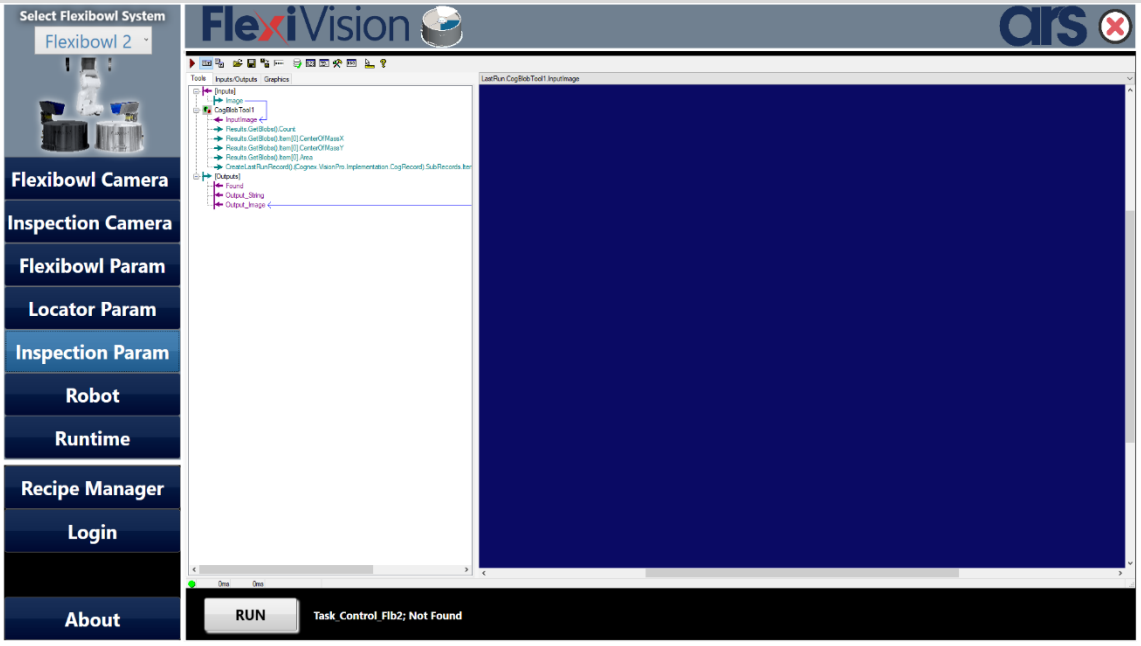
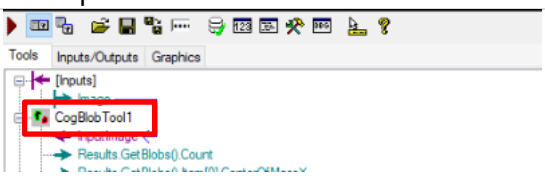
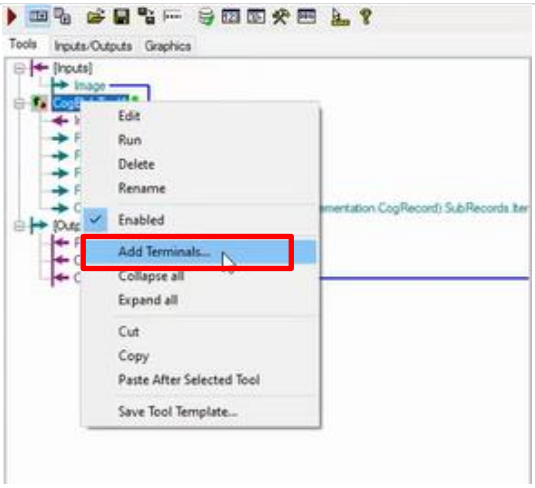
Position	Element/section	Description
1	OPERATION MENU	Includes all the control and operation procedures.
2	LOCATOR image	The locator image coming from the Toolbox is shown.
3	INSPECTION image	The ispection image coming from the Toolbox is shown.

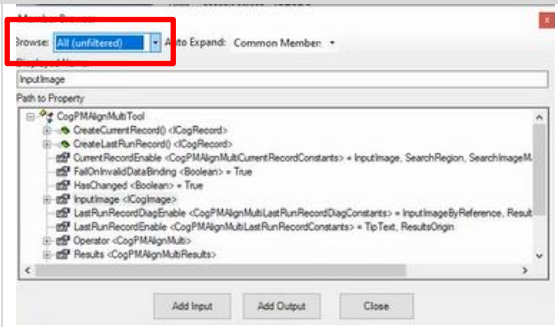
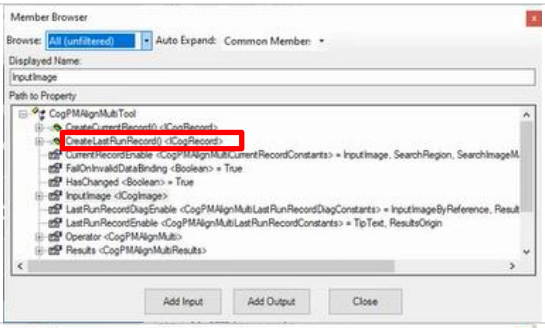
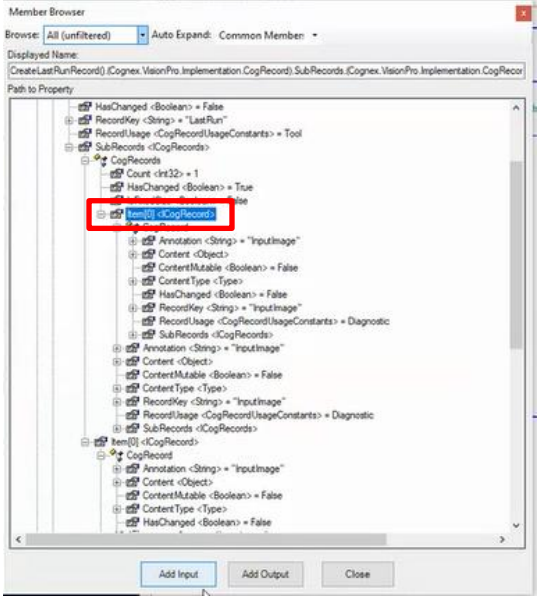
12.2 How to set the output image

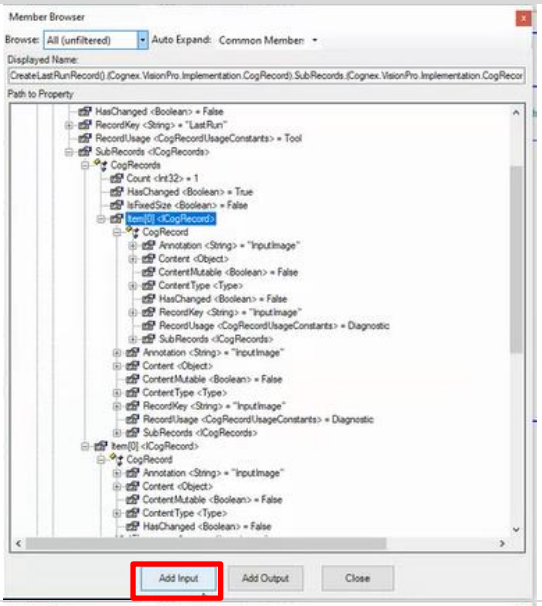
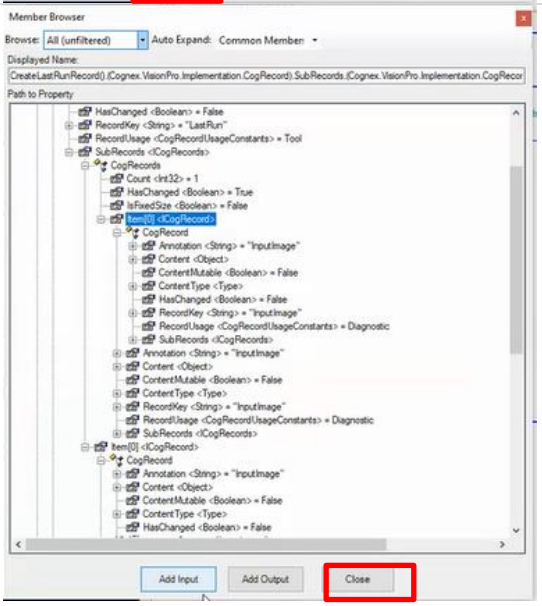
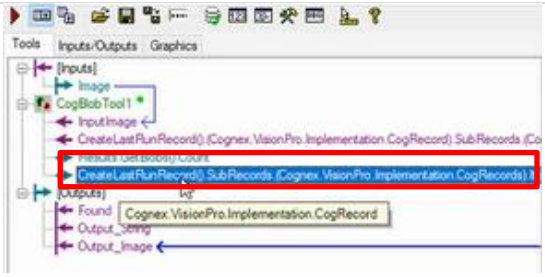
NOTE

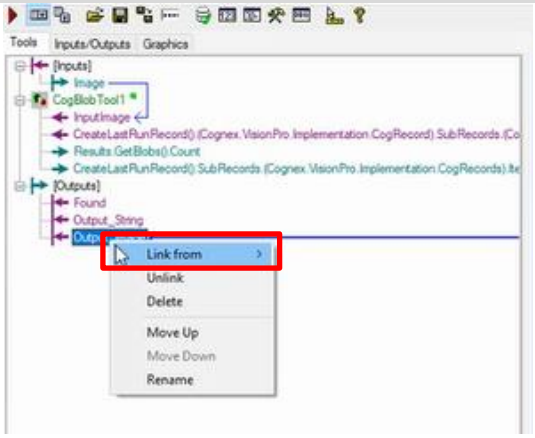
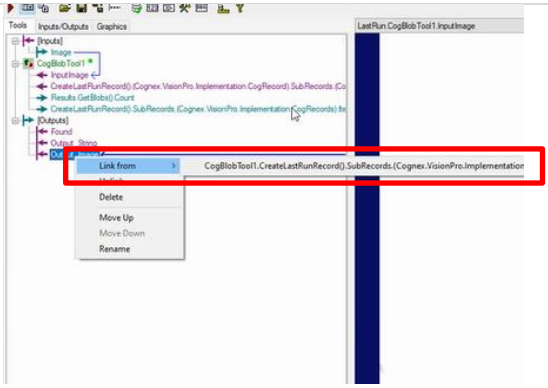
These procedures can be carried out by the following users:

- **TECHNICIAN**
- **ARS**

Step	Action	Notes/Pictures
1	 Click on **Inspection Param** or **Locator Param** of the operations menu.	
2	Select the tool we want to display as image (e.g. CogBlobTool1). Right click on **CogBlobTool1.**	
3	The following menu opens. Select **Add Terminal**.	

Step	Action	Notes/Pictures
4	Select **All (unfiltered)** in the Browse drop down menu.	
5	Select **CreateLastRunRecord**.	
6	Enter the SubRecords, find and select **Item(0).**	

Step	Action	Notes/Pictures
7	Press **Add Input**.	
8	Press **Close**.	
9	The created file is now visible in the structure.	

Step	Action	Notes/Pictures
10	Right click on **Output_Image** and select **Link from** in the menu.	
11	Select the required file and click.	
12	Go back to the RUNTIME page, to check if the image is displayed.	

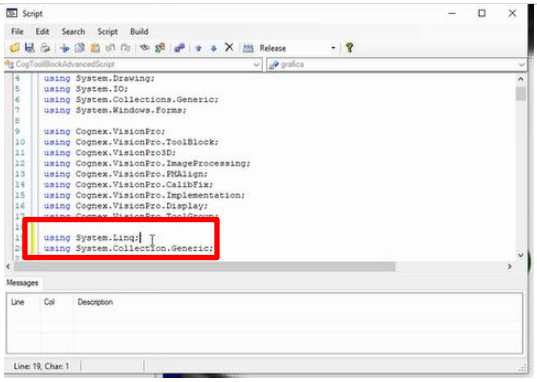
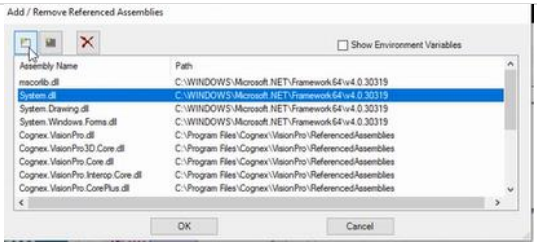
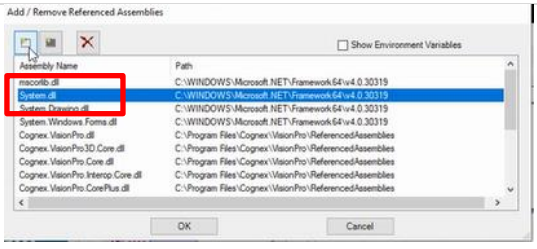
12.3 How to change items order

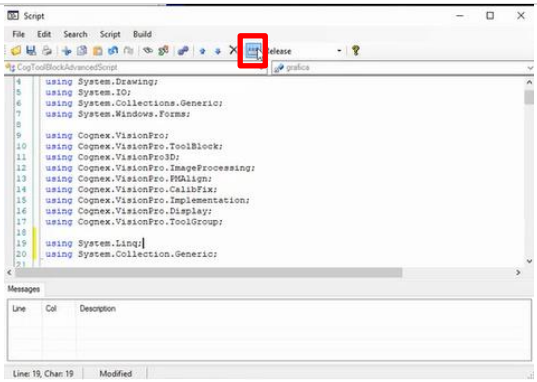
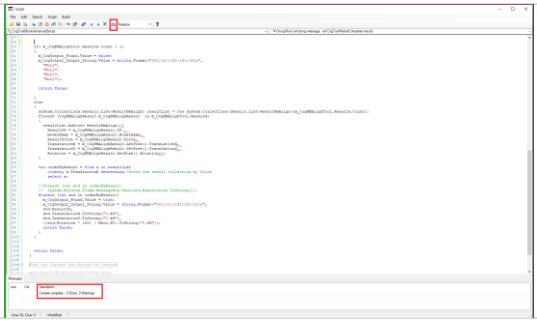
NOTE
These procedures can be carried out by the following users:

- **TECHNICIAN**
- **ARS**

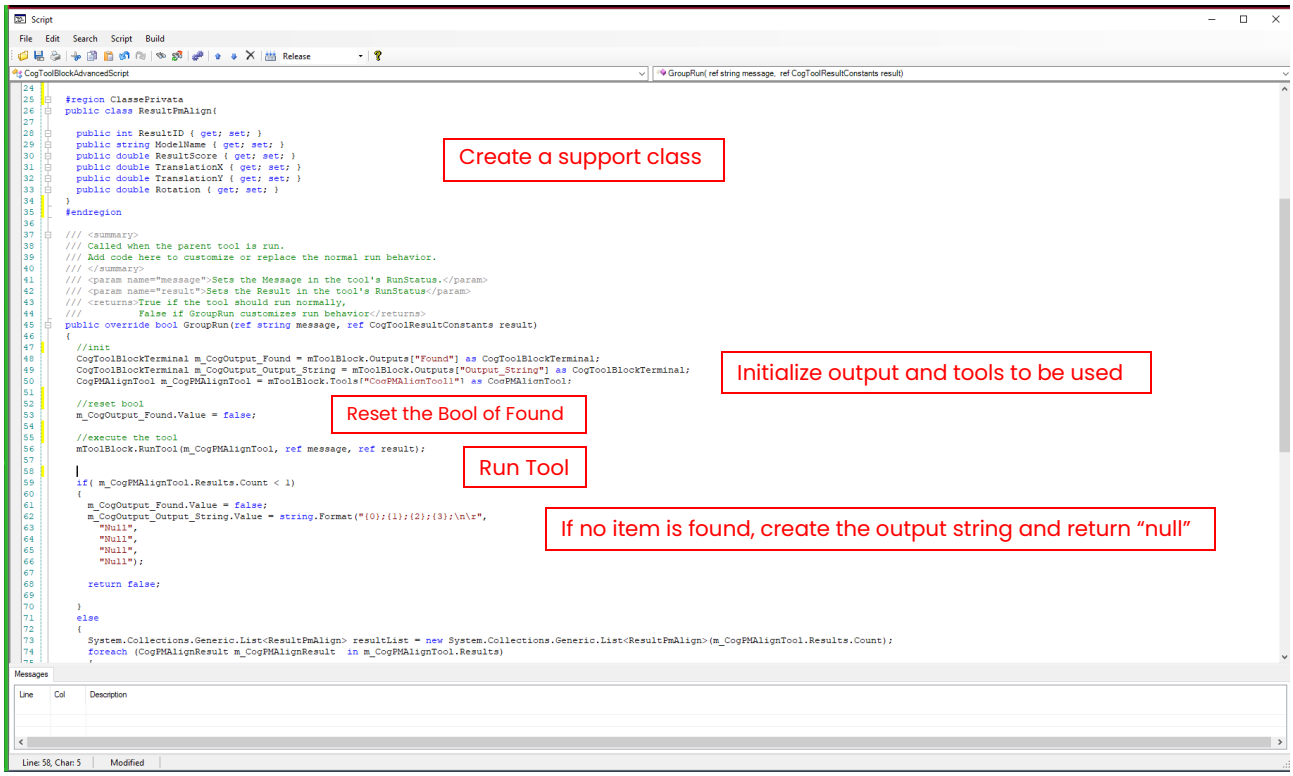
12.3.1 How to add usings and references

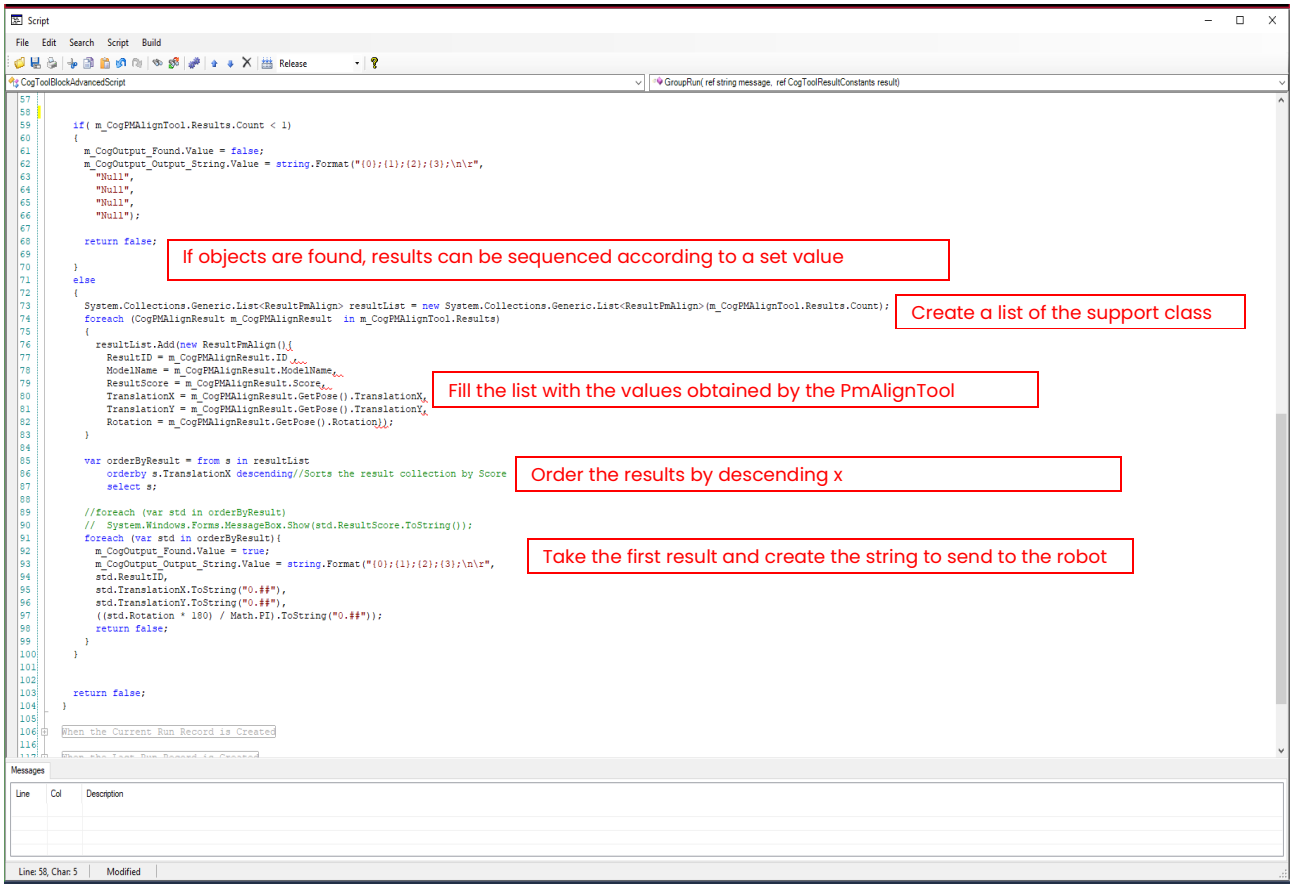
Step	Action	Notes/Pictures
1	Click on **Locator Param** of the operations menu.	

Step	Action	Notes/Pictures
2	On the button bar, doubleclick on the **Script** icon .	
3	The Script mask opens.	
4	Add the following two usings.	
5	Press the **Add/Remove References** icon from the top bar. This windows opens.	
6	Select **System.dll**.	
7	Press **ADD** icon .	

Step	Action	Notes/Pictures
8	Press **Browse.**	
9	FInd and select System.collections.dll, System.linq.dll,System.core.dll	
10	Modifify the string (see paragraph 12.3.2)	
11	Press the **Build Release**  icon to compile the script.	
12	If the script is correct, the following message appears.	
13	Repeat the procedure listed at paragraph 12.2.	

12.3.2 How to modify the string



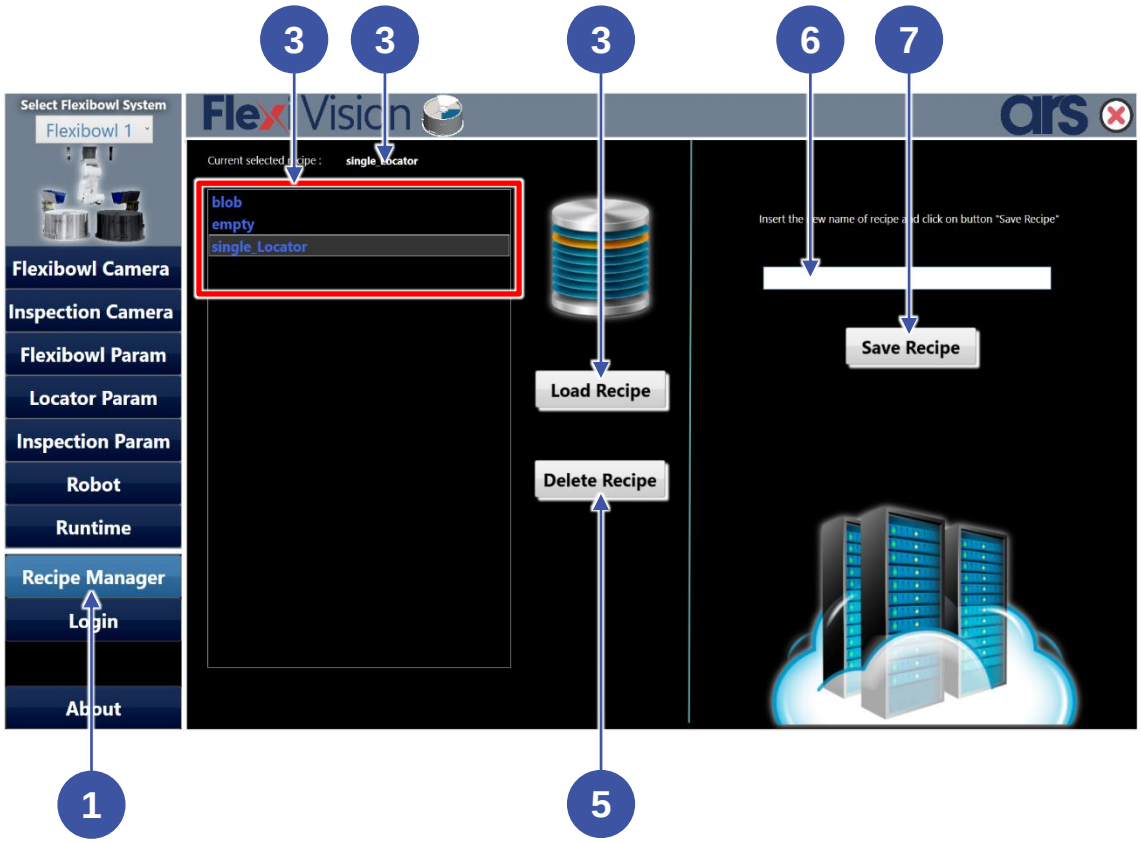

Script
File Edit Search Script Build
Release
CogToolBlockAdvancedScript
GroupRun(ref string message, ref CogToolResultConstants result)
if(m_CogPMAlignTool.Results.Count < 1)
{
m_CogOutput_Found.Value = false;
m_CogOutput_Output_String.Value = string.Format("{0};{1};{2};{3};\n\r",
"Null",
"Null",
"Null",
"Null");
return false;
}
else
{
System.Collections.Generic.List<ResultPmAlign> resultList = new System.Collections.Generic.List<ResultPmAlign>(m_CogPMAlignTool.Results.Count);
foreach (CogPMAlignResult m_CogPMAlignResult in m_CogPMAlignTool.Results)
{
resultList.Add(new ResultPmAlign(){
ResultID = m_CogPMAlignResult.ID,
ModelName = m_CogPMAlignResult.ModelName,
ResultScore = m_CogPMAlignResult.Score,
TranslationX = m_CogPMAlignResult.GetPose().TranslationX,
TranslationY = m_CogPMAlignResult.GetPose().TranslationY,
Rotation = m_CogPMAlignResult.GetPose().Rotation});
}
var orderByResult = from s in resultList
orderby s.TranslationX descending//Sorts the result collection by Score
select s;
//foreach (var std in orderByResult)
// System.Windows.Forms.MessageBox.Show(std.ResultScore.ToString());
foreach (var std in orderByResult){
m_CogOutput_Found.Value = true;
m_CogOutput_Output_String.Value = string.Format("{0};{1};{2};{3};\n\r",
std.ResultID,
std.TranslationX.ToString("0.##"),
std.TranslationY.ToString("0.##"),
((std.Rotation * 180) / Math.PI).ToString("0.##"));
return false;
}
}
return false;
}
When the Current Run Record is Created
Messages
Line Col Description
Line: 58, Char: 5
Modified
If objects are found, results can be sequenced according to a set value
Create a list of the support class
Fill the list with the values obtained by the PmAlignTool
Order the results by descending x
Take the first result and create the string to send to the robot

PAGE INTENTIONALLY LEFT BLANK

13 RECIPE MANAGEMENT

13.1 RECIPE MANAGER page

By pressing RECIPE MANAGER in the OPERATIONS menu, the following page opens:



Position	Element/section	Description
1	OPERATION MENU	Includes all the control and operation procedures.
2	CURRENT SELECTED RECIPE	Current selected recipe is shown.
3	RECIPE LIST	List of available recipes is shown.
4	LOAD RECIPE	Press to load a recipe selected in the list of available recipes.
5	DELETE RECIPE	Press to delete a recipe selected in the list of available recipes.
6	NEW NAME	Enter a new name for an existing recipe.
7	SAVE A NEW RECIPE	Press to save a new recipe.

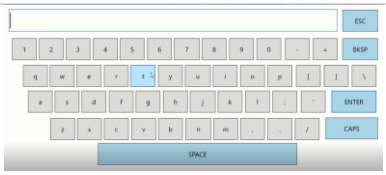
13.1.1 How to load an existing recipe

NOTE
This procedure can be carried out by the following users:
- **USER**
- **TECHNICIAN**
- **ARS**

Step	Action	Notes/Pictures
1	Enter the RECIPE MANAGER page.	
2	Select an existing recipe from the recipe list	The recipe name is surrounded by lines.
3	Press **LOAD RECIPE**	The key keeps coloured for a few seconds.
4	The recipe name appears on the current recipe name field.	

13.1.2 How to modify an existing recipe

NOTE
This procedure can be carried out by the following users:
- **TECHNICIAN**
- **ARS**

Step	Action	Notes/Pictures
1	Enter the RECIPE MANAGER page.	
2	Select an existing recipe from the recipe list.	
3	Load the existing recipe.	
4	Press on the NEW NAME field: a keyboard appears.	
5	Enter a new name for the selected recipe. If the name is already existing, a warning message appears.	
6	Press **SAVE RECIPE.**	
7	Modify the parameters, as required.	

13.1.3 How to create a new recipe

NOTE
This procedure can be carried out by the following users:
- **TECHNICIAN**
- **ARS**

Step	Action	Notes/Pictures
1	Enter the RECIPE MANAGER page.	
2	Select a NULL recipe from the recipe list.	
3	Set all the parameters by entering the pages of the Flexivision.	

Or, as an alternative, proceed as described at paragraph. 13.1.2.

13.1.4 How to delete an existing recipe

NOTE
This procedure can be carried out by the following users:
- **TECHNICIAN**
- **ARS**

Step	Action	Notes/Pictures
1	Enter the RECIPE MANAGER page.	
2	Select an existing recipe from the recipe list.	
3	Press **DELETE RECIPE.**	

PAGE INTENTIONALLY LEFT BLANK

14 HOW TO CONTACT US

14.1 ABOUT page

By pressing ABOUT page in the OPERATIONS menu, the following page opens:

ARS S.r.l.
Via G. Vico, 7 – 52100 Arezzo (AR) Italia
Tel. +39 0575 398611 - Fax +39 0575 398620
info@flexibowl.com - www. flexibowl.com