

CNRS – EPOC 5805

AviExplore

User Guide

AviExplore User Guide

V2.0

Introduction

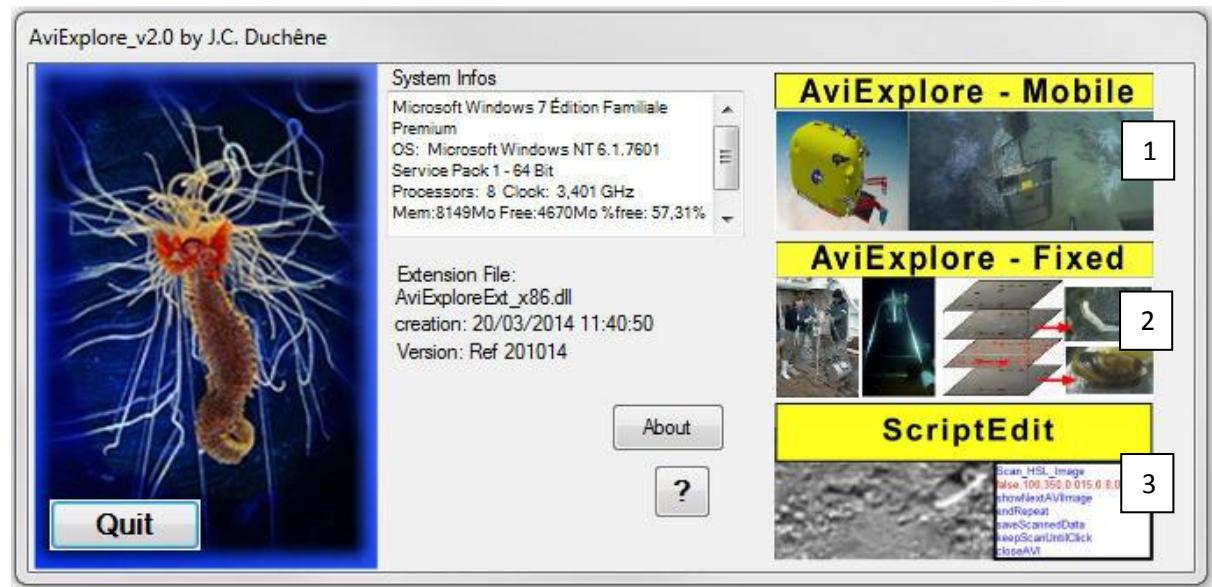
This program was developed to analyze larval tracks in video films. The core routines concerned loading of images in film and making image analysis on these images, then building output files exporting tracks characteristics. The present version, developed within Jerico Work package 10.1 (DEVELOPMENTS OF NEW TOOLS AND STRATEGIES FOR THE MONITORING OF KEY BIOLOGICAL COMPARTMENTS AND PROCESSES), is somewhat more complex and many modules and treatments have been added.

The program is made of 3 main modules and a service module.

1 - **AviExplore – Mobile** is developed to perform analysis on moving environment (films realized in situ with ROVs or other mobile devices). It has two sections: the first devoted to real-time acquisition, the second using recorded videos as input.

2 – **AviExplore - Fixed** is developed to analyze substrates such as colonizing plates or portions of surfaces. The two sections, as for module 1, concern analysis using real-time input and recorded video input.

3 – **ScriptEdit** is a common module provided to develop the scripts used in the various parts of the AviExplore modules. It can be used alone to treat and export sets of data on images.



Summary

AviExplore User Guide	1
1 - Program architecture	3
2 - Common features.....	5
2.1 - Pre-requisites	5
2.2 - Internal structures.....	5
2.3 - External companion files	5
2.4 - Date, Time and Positioning	6
2.5 - Time and image acquisition.....	9
2.6 – DateTime-Position file manager	10
2.7 – Experiment Params window.....	11
3 - Start Window.....	11
4 - ScriptEdit Module.....	13
4.1 - Scripts features.....	13
4.2 - Loading films or images.....	13
4.3 - Moving inside a film	14
4.4 - Extracting images from films.....	15
4.5 - Mask and Regions of Interest (ROIs) (4).....	15
4.6 - Film Editing tools (6).....	17
4.7 - Editing scripts	19
4.8 - Getting scripts from a database	25
4.9 - Particular case: scripts for real time calculations.....	28
5 – Jerico-Mobile Module.....	28
5.1 Input selector	30
5.2 – Real Time Input (Cam Input).....	30
5.3 – AVI Input.....	32
6 – Jerico-Fixed Module	37
6.1 - Selecting the input mode	39
6.2 Real Time Input.....	40
6.3 AVI Input.....	44
7 – Script Functions Reference	49
7.1 - Filters.....	49
7.2 - Morphology	61
7.3 - AVI related functions.....	65

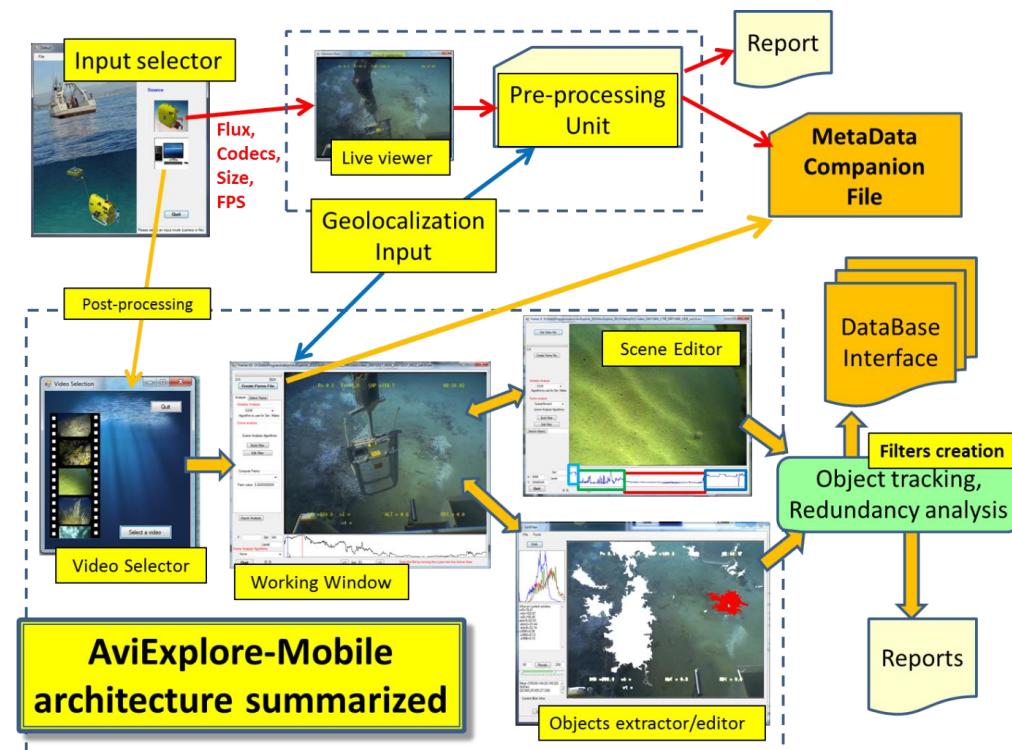
7.4 - LOG and Companion files related functions.....	68
7.5 – Data File management	69
7.6 – Loops.....	70
7.7 – Masks.....	71
7.8 – AdditiveMasks	73
7.9 - Blobs related functions	75
7.10 - ROIs related functions	77
7.11 – Scroll Tables.....	81

1 - Program architecture

1 – AviExplore modules

AviExplore is a program allowing visualization and computing on images using a set of functions assembled into scripts. These functions can be used in Real Time context (or Pre-processing context: RT compatible functions) and in Film context (or Post-processing context: all functions are then available). Modules for fixed images analysis are more concerned with activity detection and measure.

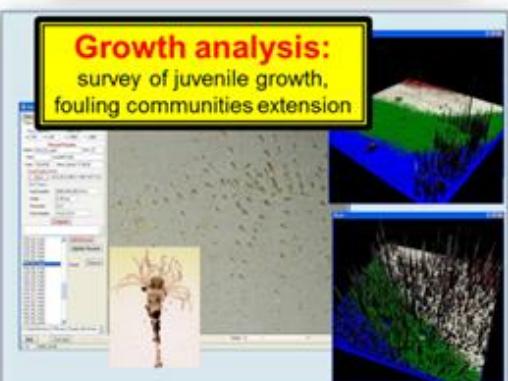
AviExplore – Mobile module



AviExplore - Fixed module

**Video imaging
by fixed cameras**

Growth analysis:
survey of juvenile growth,
fouling communities extension



Recruitment analysis:
shape analysis on juveniles, early
competition, substrate coverage



Fouling analysis: control
of anti-fouling efficiency



ScriptEdit module

This module is a service module. It was devoted to script writing and testing, but can be used to measure and export data and features from analyzed videos. Its main purpose is to develop scripts to be used directly or in other modules.

2 - Common features

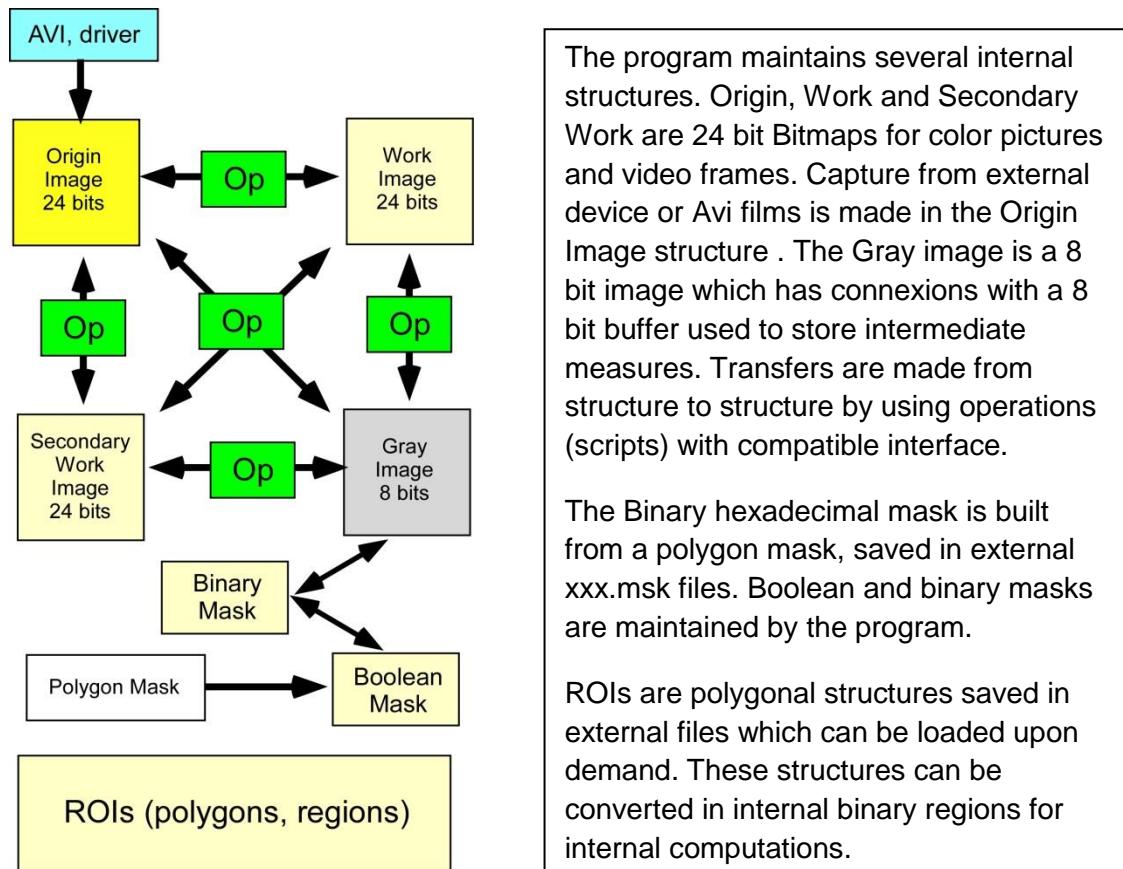
AviExplore uses common features in order to standardize the image analysis and processes.

2.1 - Pre-requisites

Images are captured from video devices such as cameras through a standard interface. Images also come from films. The AVI standard is used and the compression format selected is Microsoft **MPEG4 Version 2**. The corresponding codec must be installed on the computer. This is a pre-requisite. Another pre-requisite is the presence of **Microsoft Framework v4.5** which is available on the Microsoft site.

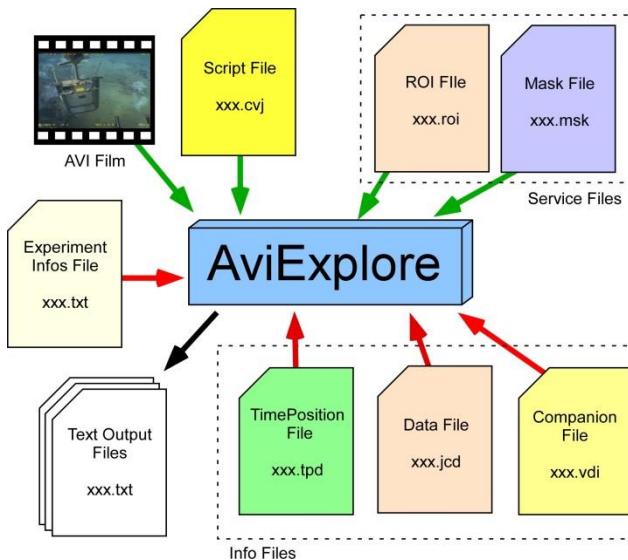
2.2 - Internal structures

The program uses several internal structures to store the information. Understanding of these structures is useful for writing scripts to extract information from the images.



2.3 - External companion files

The processing uses external files to gather information on the images. This information is used to draw recurrent structures such as mask or ROIs and is used by the scripting language.



The program create, use and maintains several type of files :

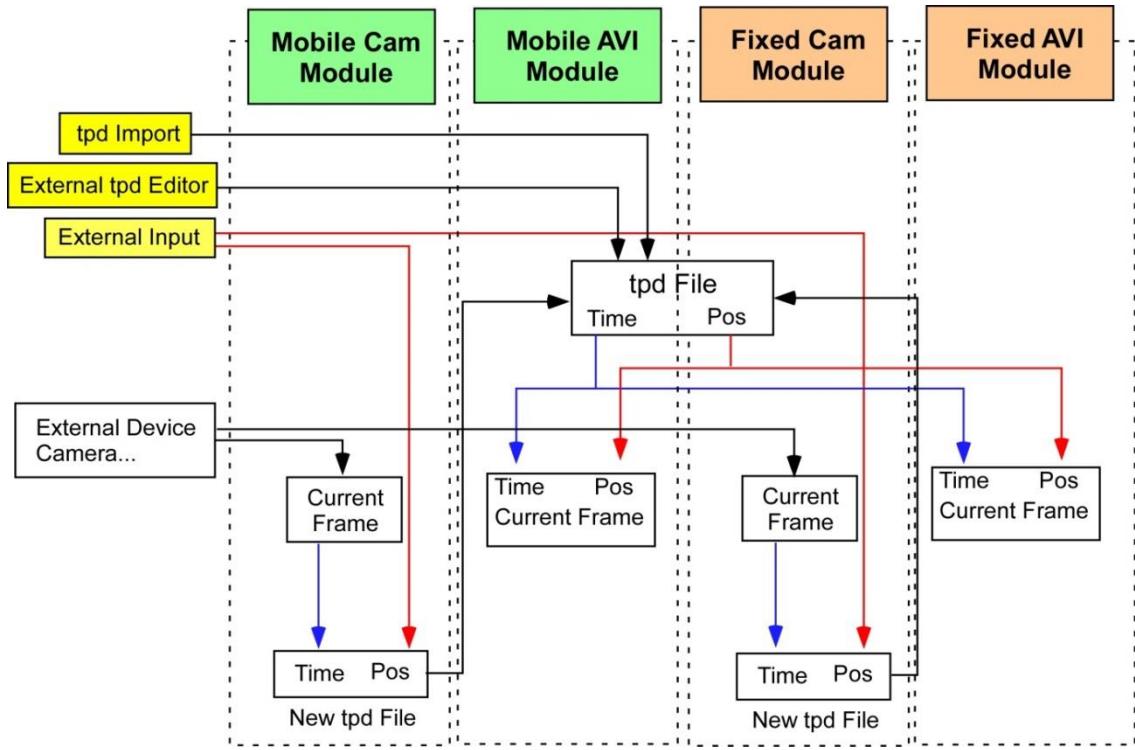
- The AVI input file (main input)
- Script files to process images (processing)
- ROI files to store working areas (service)
- Mask file to get fast mask loading (service) (files opened upon request)
- Data and companion files to store intermediate results, TPD file for date and position, Experiment Infos file for parms and info loading (files opened at AVI loading)
- Text outputs with usually TAB separated fields to allow direct import in spreadsheet

2.4 - Date, Time and Positioning

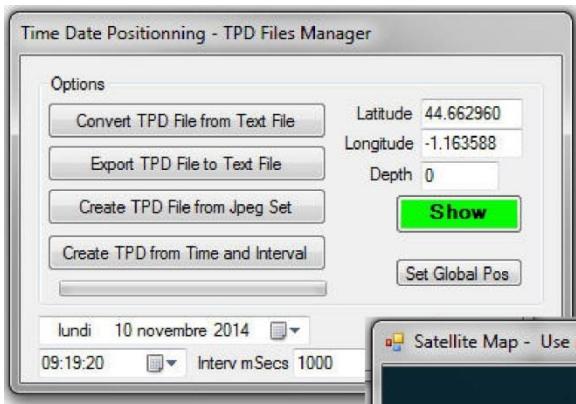
Images used in the modules may be localized in time and space. Files are created that gather information on the film, including relative and absolute time and geographic position (latitude and longitude) for each individual frame in the film. This information will be collected from a standard interface. For the moment it uses an input text file, filled from data collected from external devices (such as GPS). Some films do not have such information. A module allows creating a reconstructed Date-Position File which will be loaded transparently with the analyzed film.

When a film is created with the Camera modules, it creates an associate .tpd file gathering information on location (at the moment the current global position (latitude, longitude, depth) and Date-Time of current image. In this case, real-time acquisition the current time is collected on the PC clock.

When creating Data files, if a TPD file exists, then existing time for each frame is collected in the TPD file. If TPD file does not exist creating the TPDfile uses the current Time from the computer clock and add the interval in millisecond used to capture the frames.

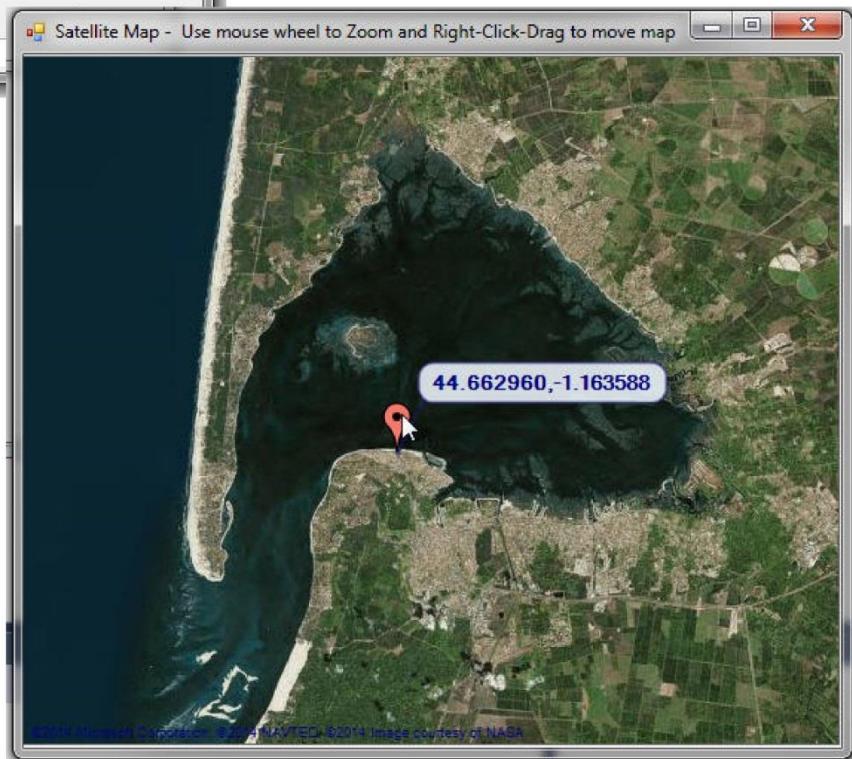


If a xxx.tpd file is present with the Avi film then it is loaded transparently and provides date, time and position for each frame in the analyze.

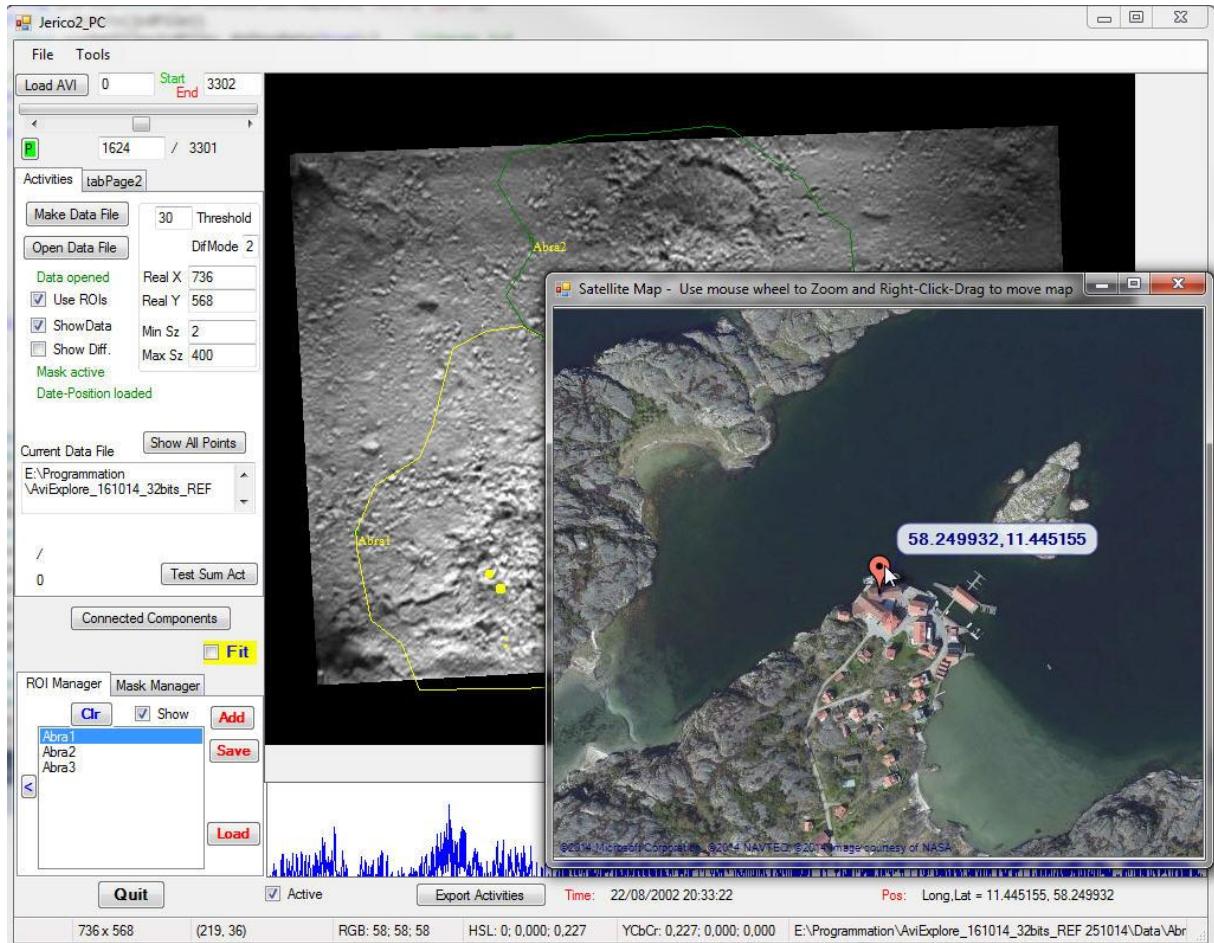


The TPD Manager window allows creation, conversion of Time Position Data Files. The current position coordinate can be entered and saved as Current Global Position

The « Show » button opens a satellite view showing the selected Global Position saved in the TPD associated file. An Internet connexion must be present

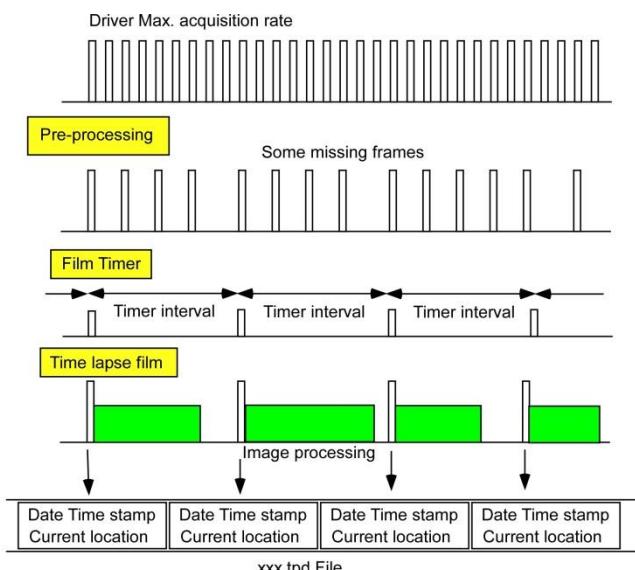


If you give only the start time and position of the measure then it is still possible to see where the experiment took place.

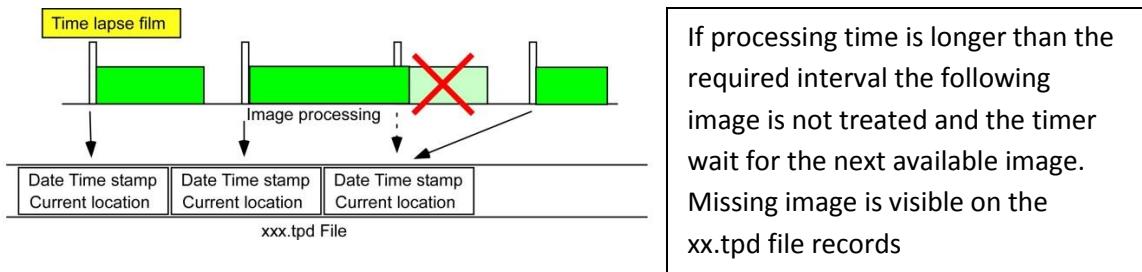


2.5 - Time and image acquisition

The real time acquisition is a time-lapse acquisition. This means that some pictures are grabbed from an image flux and these images are processed.



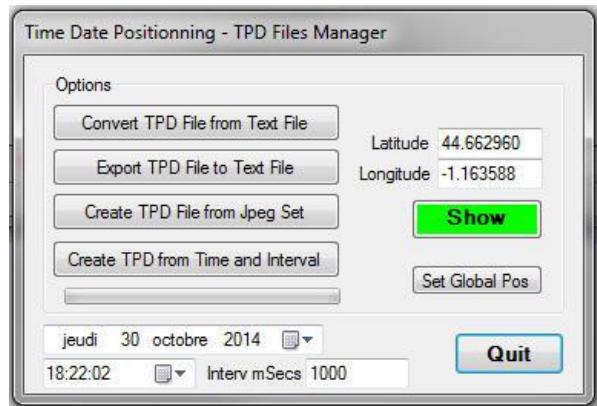
The driver sends images at the maximum rate. Images are pre-processed in the viewer and some images are not grabbed. When a film is created a timer is used to collect images at relevant intervals. Image processing time may vary with image complexity. This is why interval must be chosen wide enough to process the most complex images. Time information on each image is stored in the xx.tpd file.



2.6 – DateTime-Position file manager

The ScriptEdit > Tools > Manage Date Time-Position Files menu brings you to the TPD manager. You find the same menu in other modules.

This window is useful to manage and transfer date position data. In the Option box you can enter latitude and longitude of the current experiment. The 'Set Global Pos' button makes this position available to other routines. On the left 4 buttons let you convert, export or create tpd files.



The first button, 'Convert TPD file from Text file' allows you to import **TPD** from text file, creates with an external program.

The 'Export TPD file to Text file' sends TPD data to a text file in readable format.

```
3303
10/25/2014 11:55:25.161
From Image Set with Root:E:\Data\Abra Kris\fo2208_3f\fo22083fp
08/22/2002 11:25:12.000 58.249932 11.445155 0
08/22/2002 11:25:32.000 58.249932 11.445155 0
08/22/2002 11:25:52.000 58.249932 11.445155 0
08/22/2002 11:26:12.000 58.249932 11.445155 0
08/22/2002 11:26:32.000 58.249932 11.445155 0
08/22/2002 11:26:52.000 58.249932 11.445155 0
08/22/2002 11:27:12.000 58.249932 11.445155 0
```

The textual format allows creating import files from third party programs

The 'Create TPD file from JPEG' set scans the JPEG files used to make a file and collects the creation date and time in the metadata parts, combines it with the position entered in the Position boxes to make a TPD file

The last button, 'Create TPD from Time and Interval' creates a TPD file based on:

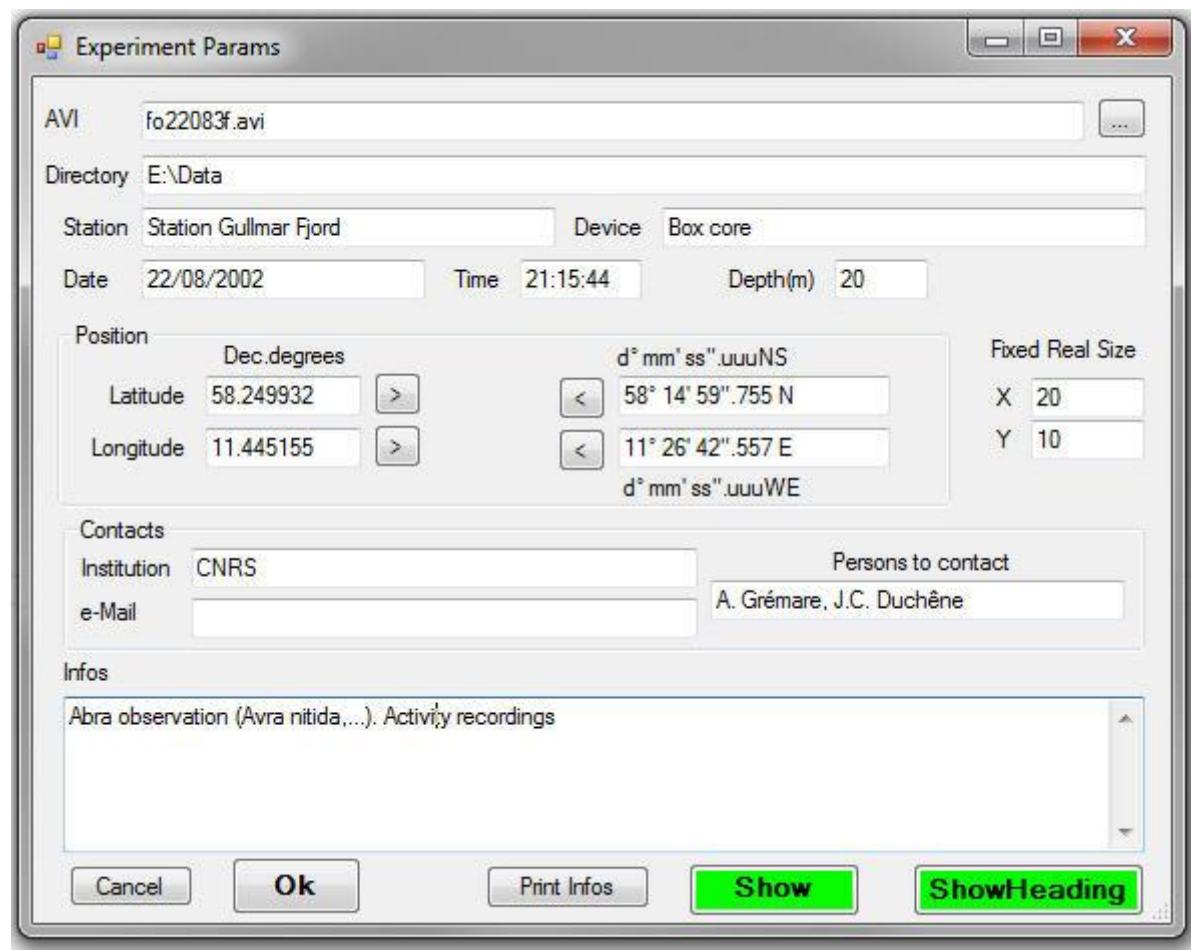
- Initial Date and Time collected from the two date and time pickers on the bottom left of the window,
- add to each frame the time interval , in milliseconds, you provide in the 'Interv.mSec' box

- and combines it to the position from Latitude and Longitude box.

2.7 – Experiment Params window

This window is created using menus in each module (Tools->Set Experiment parameters) and a button in AviExplore-Fixed Camera.

It brings a common window with information concerning the AVI.



Information on the experiment, location, time, devices are editable. The Ok button save the window to a text file (Avi_name.txt file). If present, this file will be loaded every time an avi is loaded, providing recorded information on the film. In the TPD file latitude and longitude are saved in decimal degrees. A converter allows viewing this in DMSs format. Two buttons allow showing satellite view with the first position or the single position ('Show' button) and a sequence of positions coming from the TPD companion file (Show Heading). Printing of the data is possible.

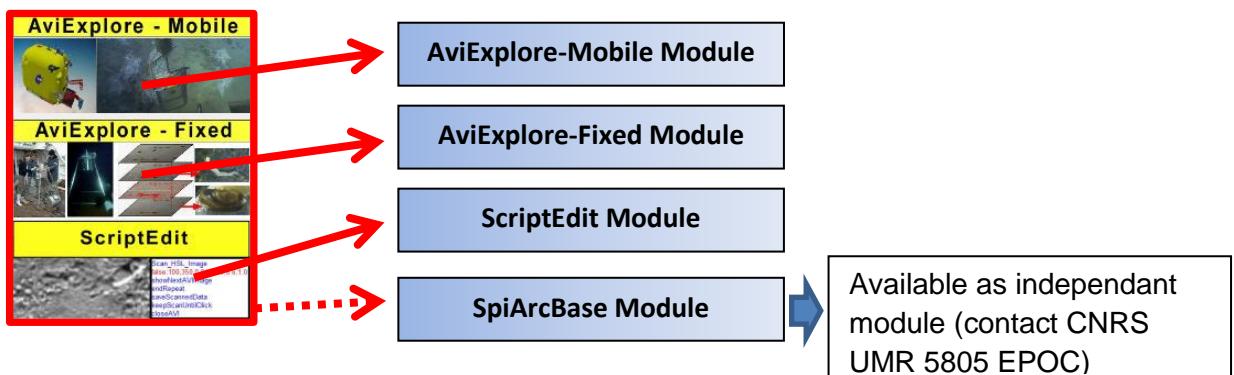
3 - Start Window

Opening AviExplore brings you to the Start Window. This window allows selection of the different modules of the program. There are 2 parts in this window.



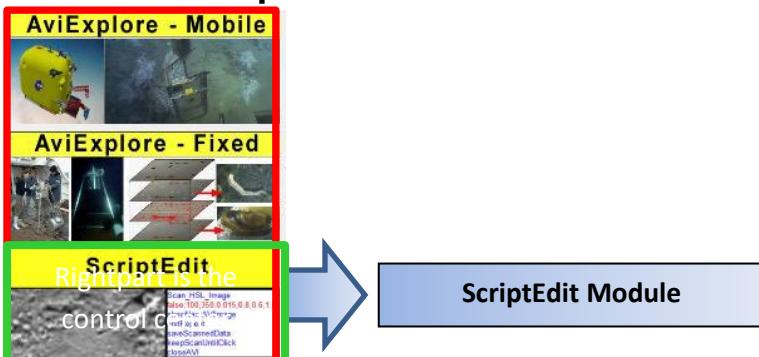
The center part is informative and provides System information concerning your computer. Consider that this program is very memory consuming as it deals with video and image processing. This means that at every moment sets of very large images can be stored in the RAM memory of the computer. Memory is also required for computing on the images. Information on the available memory of your system is provided in the upper box altogether with information on the speed of your system. Complex computations on large images can be very time consuming and directly infer on the ability of your machine to treat real time video fluxes. Below that box the availability and name of an extension file is shown. This file allows adding new functionalities to the program. It is written in C++ language. The extension file shown is a demo one and provides some test functions. Finally a button in the lower part of this center portion brings you to a Help file.

The right part is the control portion of the window. Moving your cursor over the pictures and clicking brings you either to the AviExplore – Mobile module, or the AviExplore – Fixed module, the SpiArcBase module and the ScriptEdit module. At the moment the SpiArcBase module is not included and is available for download as a separate program.



4 - ScriptEdit Module

4.1 - Scripts features



The ScriptEdit module is a tool allowing writing and testing scripts that will be used in the AviExplore modules. It can be used alone to get computed data from videos or images.

The interface proposes a window to display films and images



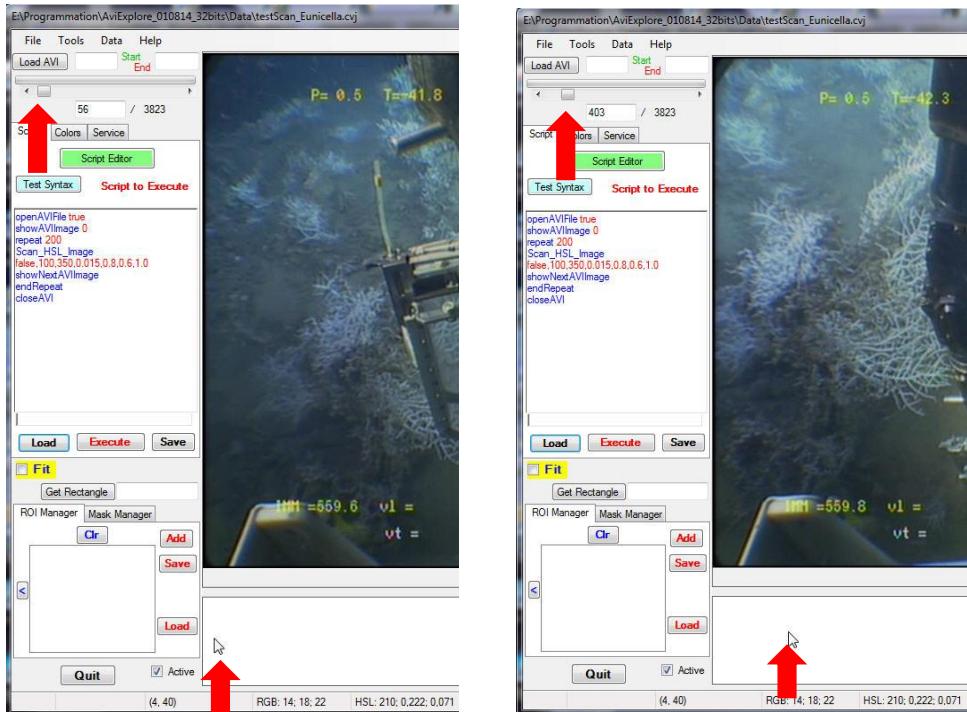
4.2 - Loading films or images

Films and images are loaded with the 'Load AVI' button or with the 'Load AVI' or 'Load JPEG' items from the File Menu. Images are shown in the viewer (1). Displacements inside a film is achieved with an active area (2) allowing fast moves or with the Scroll bar located ont the top left of the main window (3). Current image number is shown up there. It is possible to

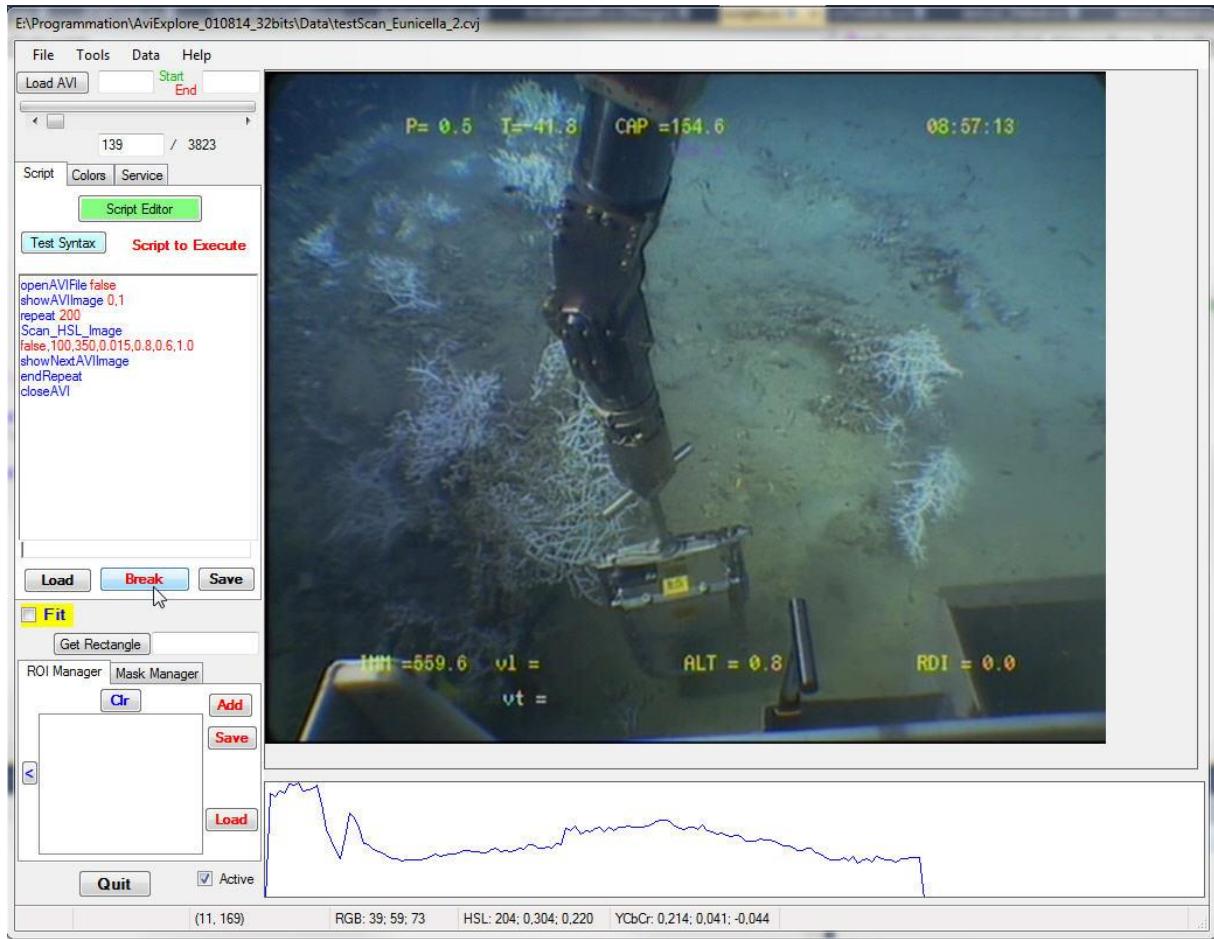
desactivate the active area by uncheking the checkbox located on the bottom left of this area.

4.3 - Moving inside a film

The active area can be used to move the film to an image, but in othercase it will display information on the results of computation on images.



Moving cursor over the active area allows fast motion within the video. Some functions use the active area to display Intermediate graphics, that can be exported to text files.



In this sample the script computes an index based on colorimetric analysis for each image in the film. The film moves on the Display Window and the corresponding computed index is shown on the Active Window below. The sample script uses a loop on the 200 first frames of the video.

4.4 - Extracting images from films

In the File menu the item “Save Jpeg” saves the current image as a Jpg file.

Extraction can also be done with script command ‘saveJpeg’, ‘saveJpegWithOverlay’ and ‘setJpegRootName’.



4.5 - Mask and Regions of Interest (ROIs) (4)

This is done within the Tabs ROI and Mask Managers.

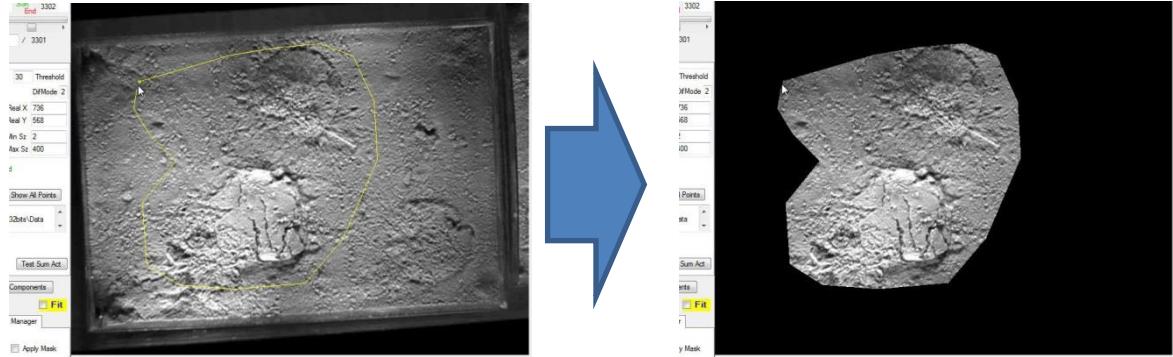
4.5.1 – Masking images (4)

The Mask Manager Tab gives access to boolean mask operations. The same interface will be found in the AviExplore – Fixed module. A mask is defined for a whole film. When applied it will mask areas in every pictures of a film.



Buttons allow operations on the mask:

- Draw Mask : you can draw the boolean mask on the window then save it as a xxx.msk file, reload it, erase it.

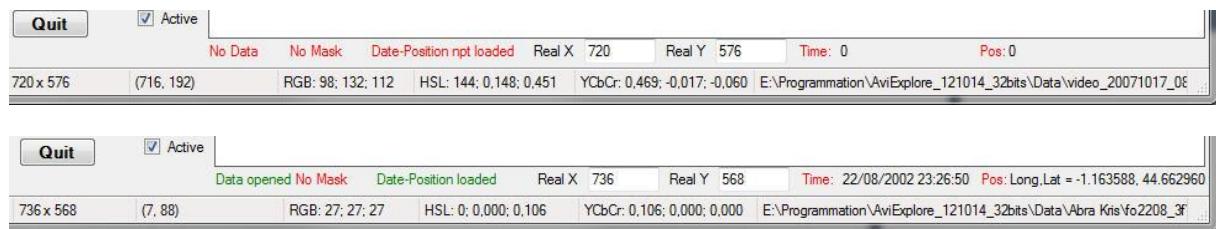


If the use Mask check button is checked the mask is applied at loading on the Origin Image. It sometimes needs to be applied in other occasion when working on objects. A switch `useMask` exists in these functions. It applies the mask afterward to the resulting computations. A temporary internal structure, called **binMask** can be created from Grey images. This can be found in the declaration of the script instruction.

4.5.2 - Regions of interest (4)

You can manage the Regions of Interest in the ROI Manager Tab (4). Regions of Interest (ROI) can be drawn, saved or loaded.

In the AviExplore – Fixed module, ROIs and mask may be added to a special companion data file created on demand (file type xxx.jcd). If the name of that data file is the same as the name of the avi film (i.e. ‘TestMeasure1.avi’ and ‘TestMeasure1.jcd’) then this companion file is automatically loaded when you open the Avi film. This is visible in the status portion of the window.

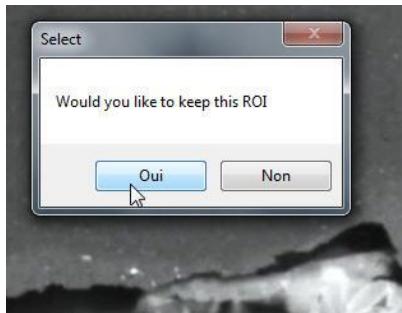


In this status area you can see on the lower part the size of the film in pixels, the position of the cursor in the window, with the RGB, HSL and YCbCr values of the corresponding pixel, the path of the current Avi File. On the upper band you can see the Data file status (opened or not), the mask (present or not), the Date-Position companion file (loaded or not), the Date and Time of the current frame of the film and the corresponding position (latitude and longitude) of that frame. The last values are provided by a Date-Time-Position insertion mechanism.

ROIs are mainly used in the AviExplore-Fixed portion of the program and you should see the corresponding section.

4.5.3 - Getting polygonal ROIs (4)

The 'Add' button allows editing or a ROI. You click on the picture to draw a polygonal area. The first point appears as a small yellow square. You double-click to close the polygon. This opens a series of dialog asking if you want to keep this ROI.

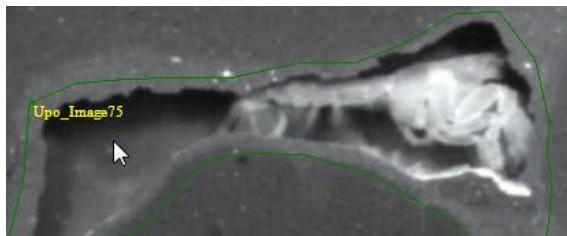


A double click closes the polygon



A name for the region is requested

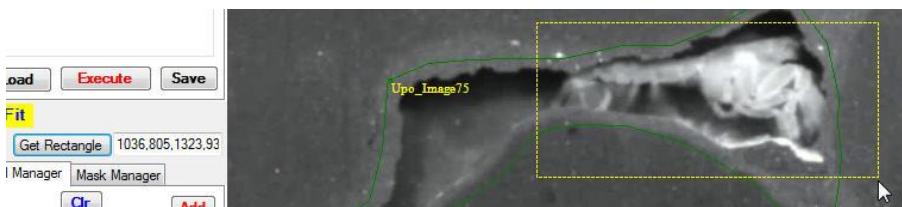
The ROI is added to an internal list and will be available for future computations. It is possible to store it in a RoiFile and reload it later.



4.5.4 - Getting rectangular ROIs (5)

A single rectangle is commonly used to restrict computations on an image to a particular area. Usually the functions request the typing of the rectangle parameters (X position, Y position, Width of ROI, Height of ROI). A helper is provided (5) allowing click and drag to draw a rectangle on screen, and get the positions in an editable text zone on the right of the 'GetRectangle' button. Text may be copied and pasted into the corresponding functions.

In AviExplore, rectangular areas are mainly used to crop the film (see the Film Editing Tools below).

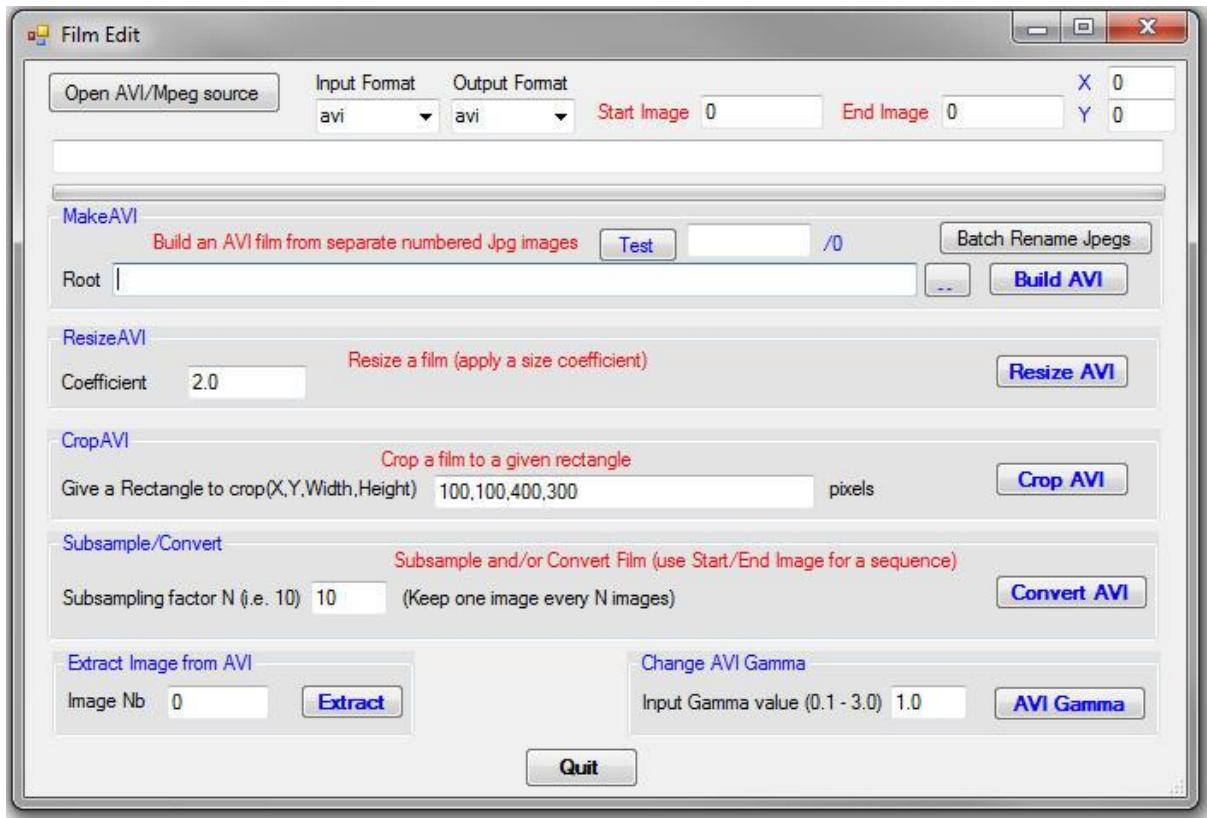


4.6 - Film Editing tools (6)

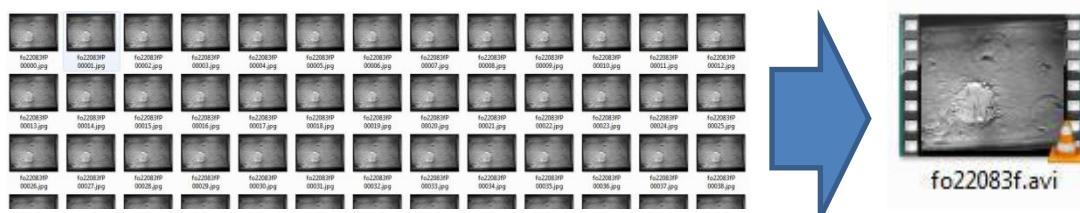
A set of video tools can be accessed through the Tools menu. Two menus are available (1) **Film Editing** and (2) **Make AVI from regions**.

a – Film Editing opens a window. It offers 6 possible treatments.

A common area on the top of the windows allows loading film, selecting an input and output format and start/end images of a sequence. Default gives the entire film.



MakeAvi takes a batch of numbered images and assemble it into a new AVI film. A root name can be selected or typed and the function search for this root with increasing numbers starting at 0.



Resize AVI uses a provided coefficient to resize a film into a new one.



Crop AVI uses a rectangle to create a new cropped film. The rectangle can be set by using the 'Get Rectangle' button, then copying and pasting the result

Subsample & Convert builds a new film taking one image every N frame, reducing the film size and limiting, depending on motion dynamics of the targets, the redundant images.

A conversion is possible, using the common Input and Output Formats (top of the window).

Extract image from AVI allows electing an image number and creating a independent Jpg image.

Change AVI Gamma creates a new film with modified Gama darker or brighter (Gamma value from 1.0 to 3.0).

b – ‘**Make AVI from regions**’ menu can use ROIs to rebuilt the same number of new films, cropped to the ROIs bounding boxes in a single pass. This function depends on available memory, as it is heavy resource consuming. The example below shows extraction of 20 films out of a source film to individualize larval settling activities under near UV light. Only 5 films are presented.

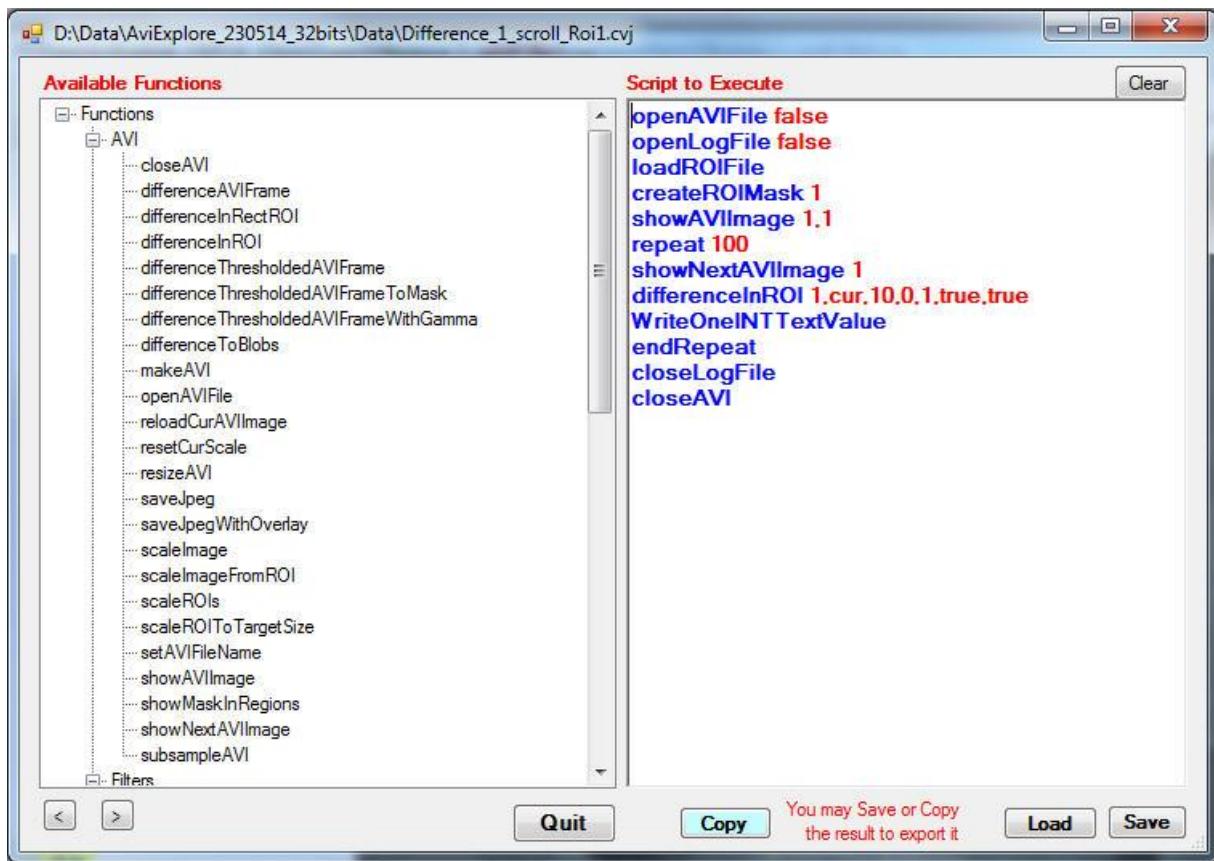


4.7 - Editing scripts

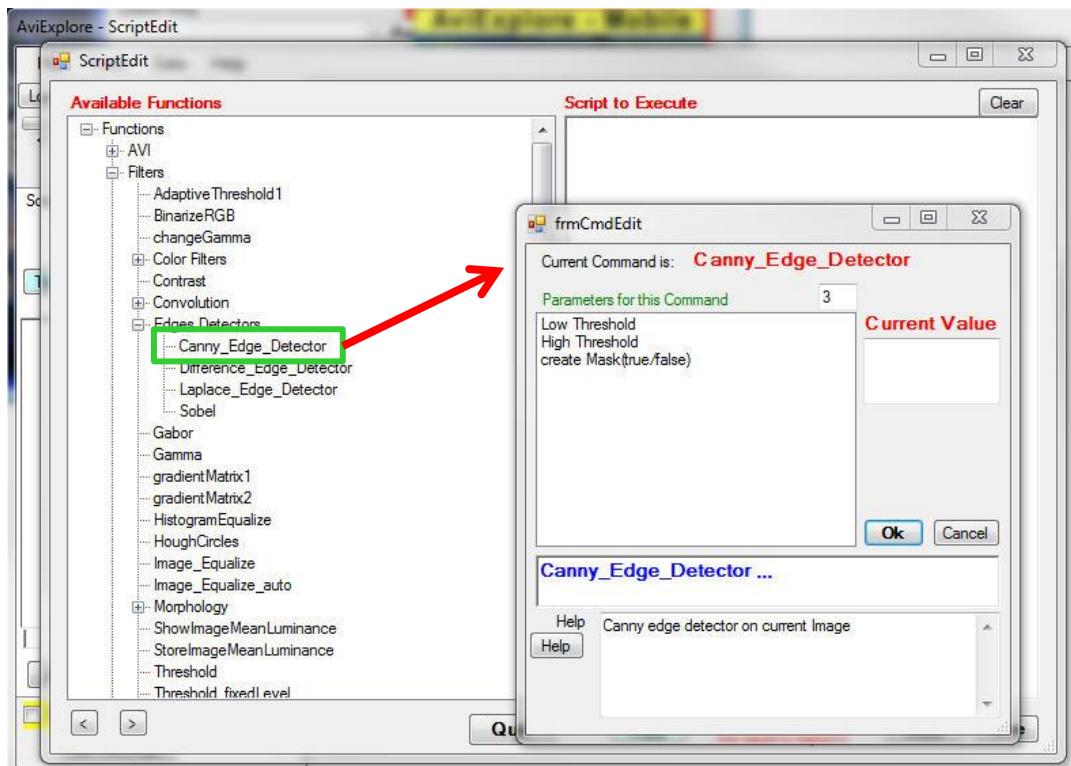
Script edition is achieved using the ‘Script Editor’ button (7), the results being placed in the script window (8). This window is an editable one and parameters may be changed to run different trials of a same set of functions.

4.7.1 - Script Editor button (7)

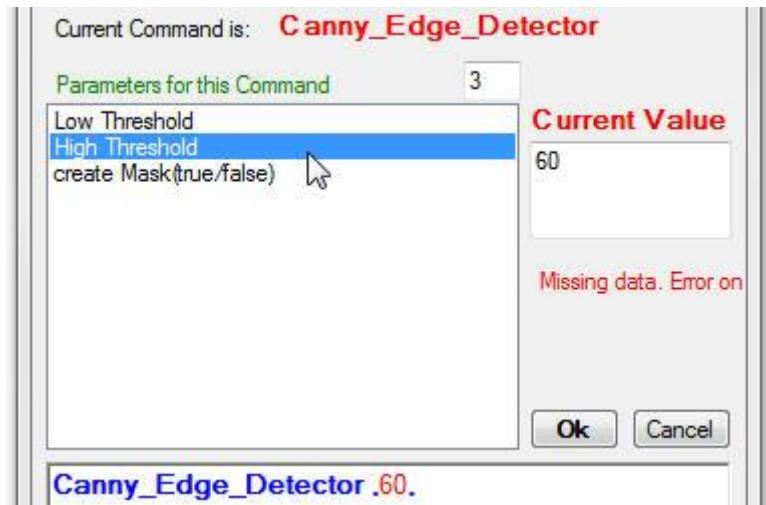
Pressing the ‘Script Editor’ button brings a window presenting the available functions allowing to enter these parameters in an easy way. Functions are visible on the left panel, result of the scripting process is visible on the right panel, ready to be transferred in the main window. Buttons on the lower part of this window allow saving the script, loading an existing script to edit, copying the current script to be pasted in the main window.



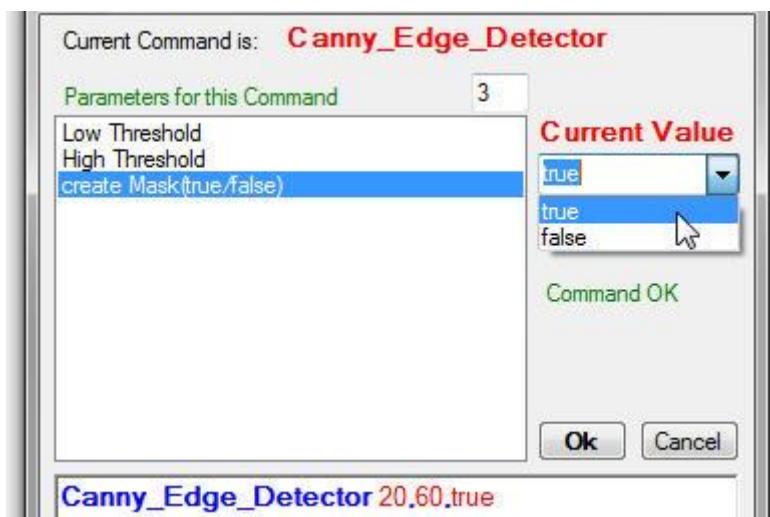
A double click on a function starts a Function Editor Window that simplifies the parameters setting. It uses a dictionary to ask for the correct parameters and control the validity of types values.



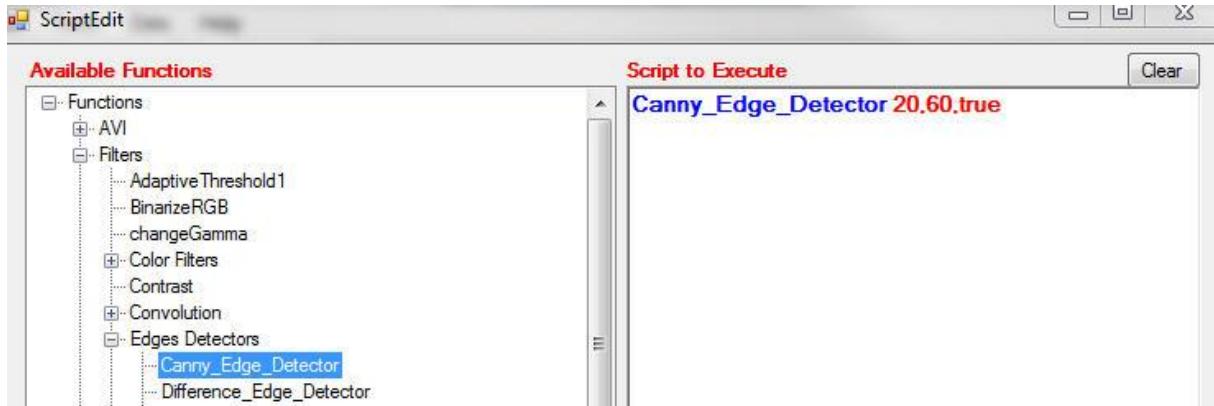
As an example: a double click on the line **Canny_Edge_Detector** opens the Helper window. It presents the required parameters for that function. If you click on a parameter line you can edit the value in the **Current Value** box. Value is typed automatically in the Function constructor below. A rough analysis of typed parameters indicates errors on parameters types or missing parameters. The value (in the sample window the high threshold for the Canny function) is inserted at the right position in the lower box taking care of the comma separation. This is the final function.



Some parameters, like booleans, require a standardized spelling. In this case the helper provides a selection menu and enters itself the value in the function box. When all the parameters are typed, with an assumed correct spelling, the function controller turns green with a **Command OK** indication.

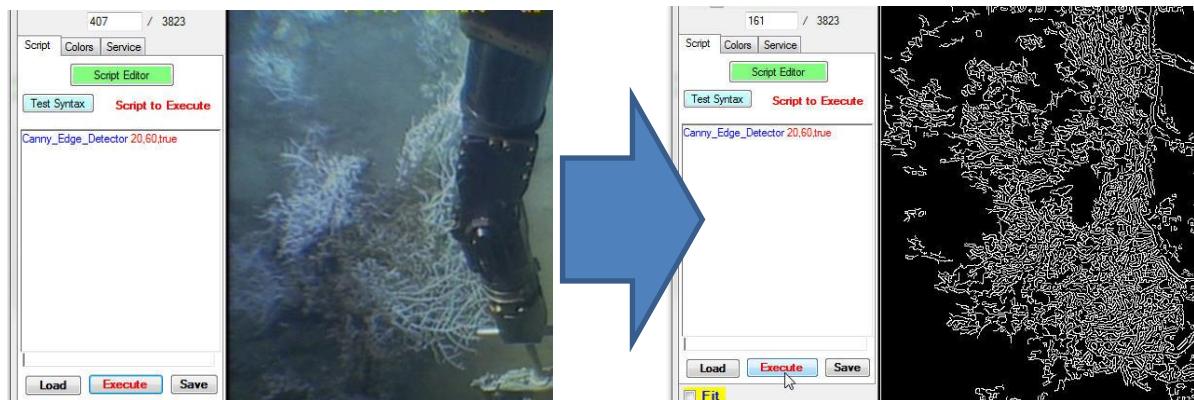


Clicking OK at that point copies the function at the cursor position in the calling Script window.



It is then possible to save the script for later usage or copy it to paste it in the ScriptEdit window for execution.

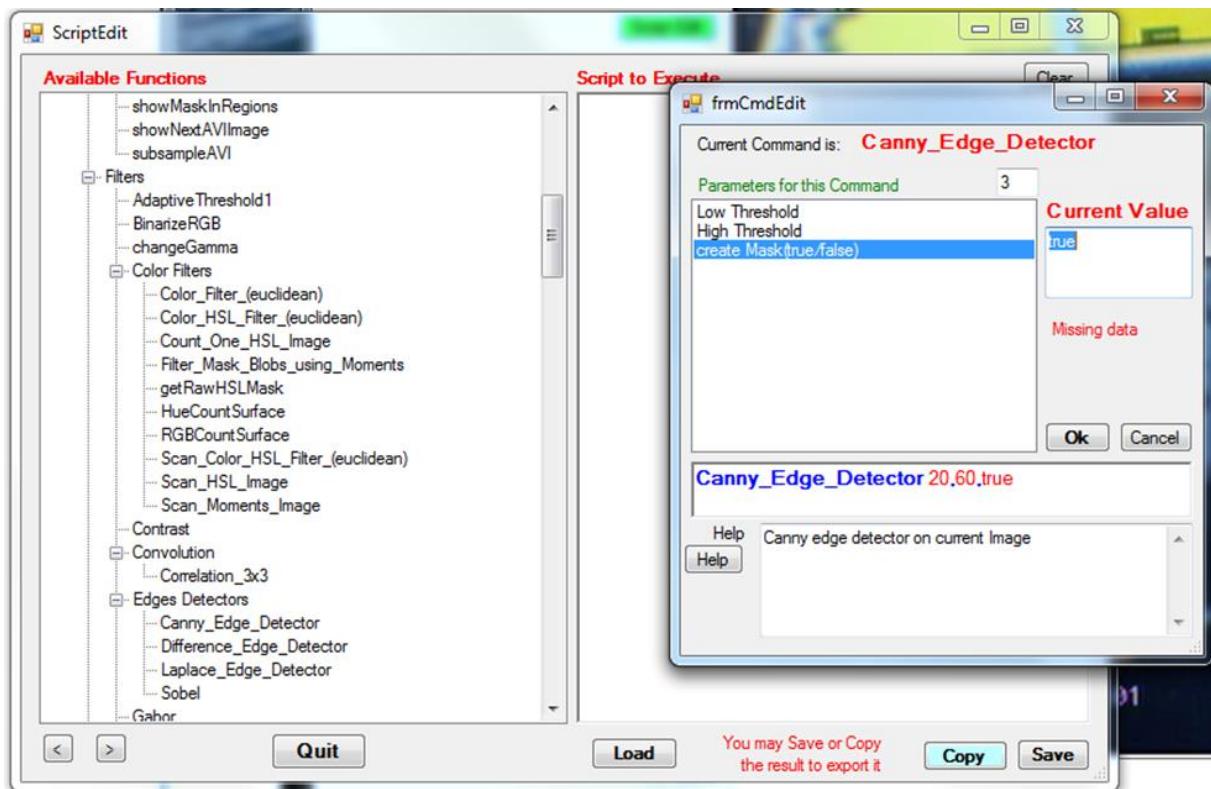
Execution is made on the current image of a film or on a jpg image loaded with a function.



For more information on the scripting process look at the Scripting chapter.

4.7.2 – Script Box (8)

The script box shows the current script. It is editable. An existing script can be loaded with the Load button, executed with the Execute button and the changes can be exported in a text file ending with '.cv'. A syntactic coloration allows to identify the keywords belonging to the current dictionary. Keywords are in blue when recognized, data are in red separated by commas.



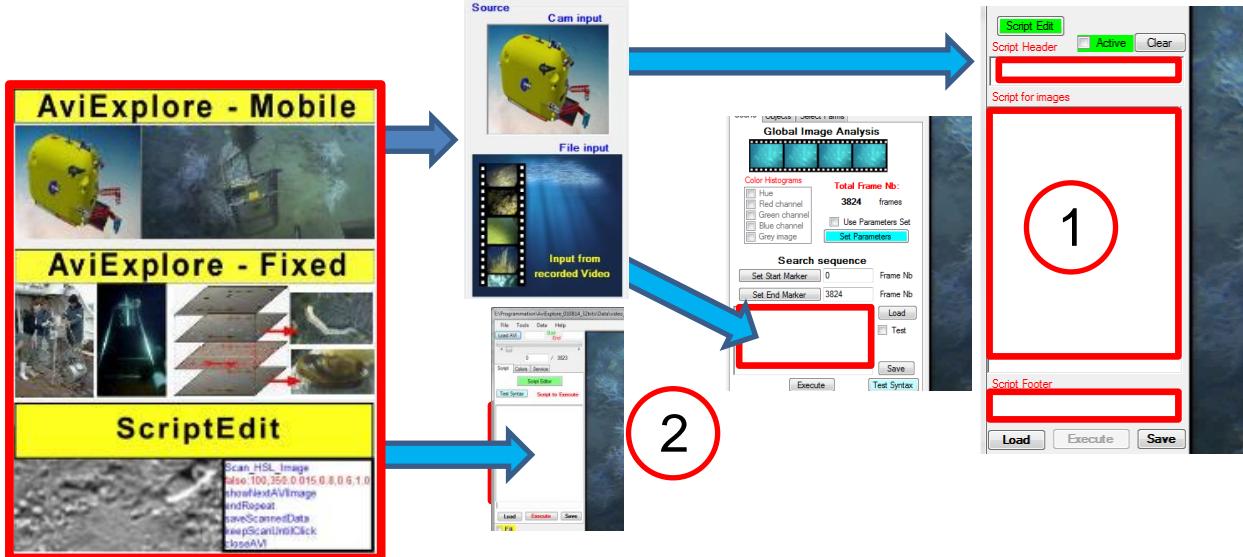
This program uses windows with a common interface in order to make it simple.

Modules of this program use scripts. The scripting language allows computation during the presence of a frame on the screen. The script is a succession of textual commands, saved in 'xxx.csv' files. Each instruction, belonging to a dictionary, has parameters that can be changed by users to perform different types of computation and image analysis or treatment.

A script can be used in different environments, i.e. general script writing (in the ScriptEdit window) or in mobile module (inside the real-time acquisition window or the post-treatment window) or finally within the fixed module (in the post-processing window). Each of these window use a different set of instructions. More, the evolutions of this program lead to modifications in the instruction set and in the parameters of certain function. This release can load scripts from older versions. A real-time window can load scripts written for fixed analysis. To circumvent the potential syntax problems, a special common instruction has been inserted in this release. This instruction MUST be the first instruction of a script. This is the 'Version' instruction (within the **Misc.** section). Parameters are 6M, 6F or 6A, standing for Version #6 of the scripting language and instruction set corresponding to real-time (M), post-treatment (F) or complete set (A). If this instruction is not found in the first position, this release will assume that this is an older or a non-compatible script version.

4.7.3 - Scripting Helper

Scripts may be edited or executed in ScriptEdit, AviExplore – Mobile and ScriptEdit windows.

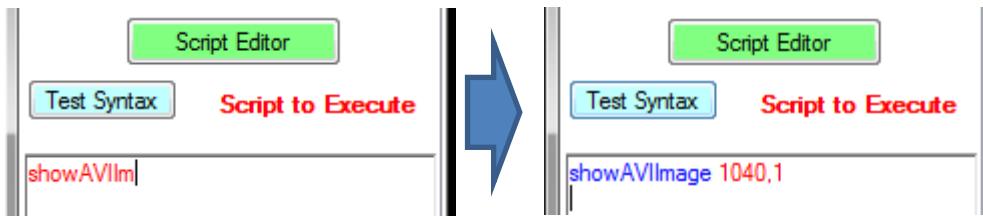


There are two distinct cases:

- 1) Real Time acquisition (Mobile with Cam Input). A restricted set of instructions can be used in this mode where loops may be very time consuming. Instructions with a global scope must be used in a Header zone and Footer zone, while instructions concerning individual image treatment must be used in the current zone. When instructions with heavy computational time are required, it is possible to change the acquisition timer to provide sufficient time for the routine to process and eventually export their results.
- 2) Post-treatment mode allows using the whole set of instruction. Long calculations may be activated. There will be an influence on the overall duration of the treatment.

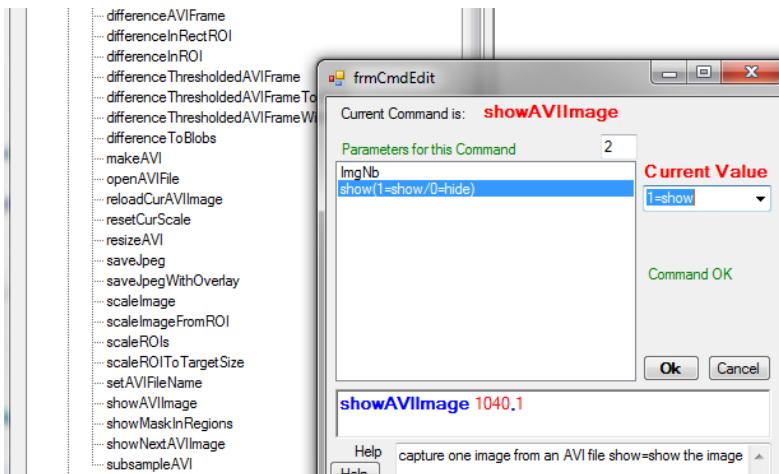
4.7.4 - Text editing

Script instructions may be obtained easily with the Script Editor Button (see [Script Editor](#)). Text may be typed directly in the script window. It is possible to make a rough test of the syntax with the Test Syntax button. Instructions have 3 possible colors: green for comments, dark blue for instruction and red for data and parameters. When you start writing the program uses an internal dictionary to look for instruction and colorizes in blue the known instruction (in case of mistyping, it remains red).

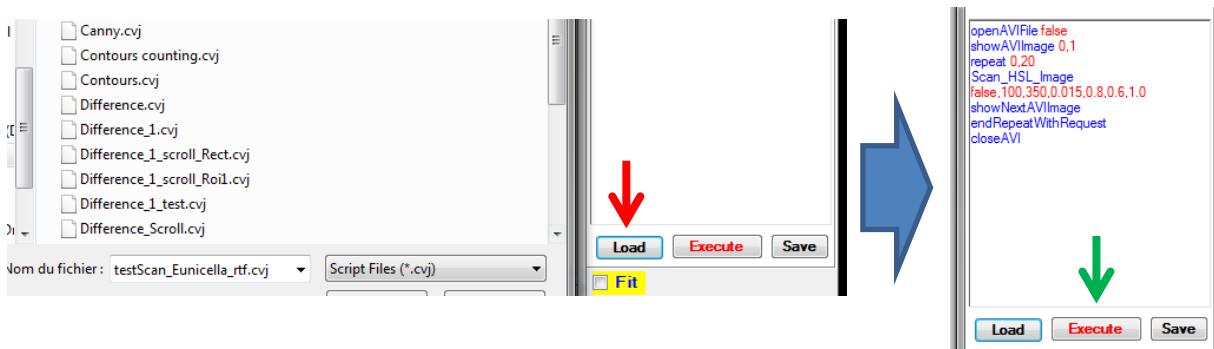


The screen capture shows an instruction showing the frame #1040 of the current film. The 1 on the left side of the text line indicates that this image will be drawn on the main picture window on the top right of the main window. Frames may be loaded and not shown for speed or other reasons.

The script Editor Button allows fast capture with correct syntax (see the corresponding section).



The script written may be saved for future use. It is written in RTF format. It is a 'xxx.cvj' file.



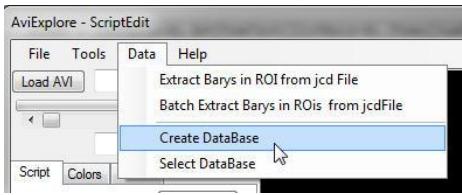
3 buttons allow Loading, Saving and Executing the scripts. The script present and visible in the script window is executed. It may be modified to test parameters when execution is not running.

Character size may be adjusted with the selection box over the script on the left.

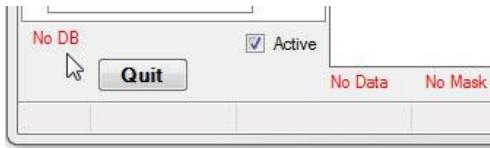


4.8 - Getting scripts from a database

You can create and feed or share a personal database containing scripts. The menu Data > Create DataBase brings you to a dialog to create an empty **mdb** file. This file is fully compatible with Microsoft Access Databases. You can edit it from the Access program, but normal operations can run without it.

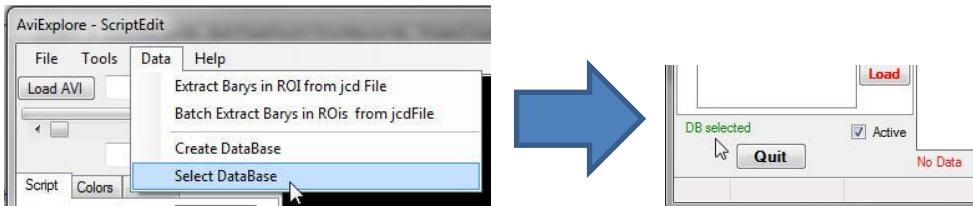


Once the DataBase is created you can activate it. An indicator is present in the lower part of the main window, showing '**No DB**' in red.



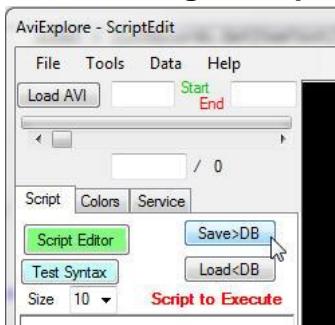
4.8.1 - Using a DataBase

To open and activate a DataBase you can use the menu item **Data > Select DataBase**. This asks for an existing database and makes it accessible for loading and writing operations.

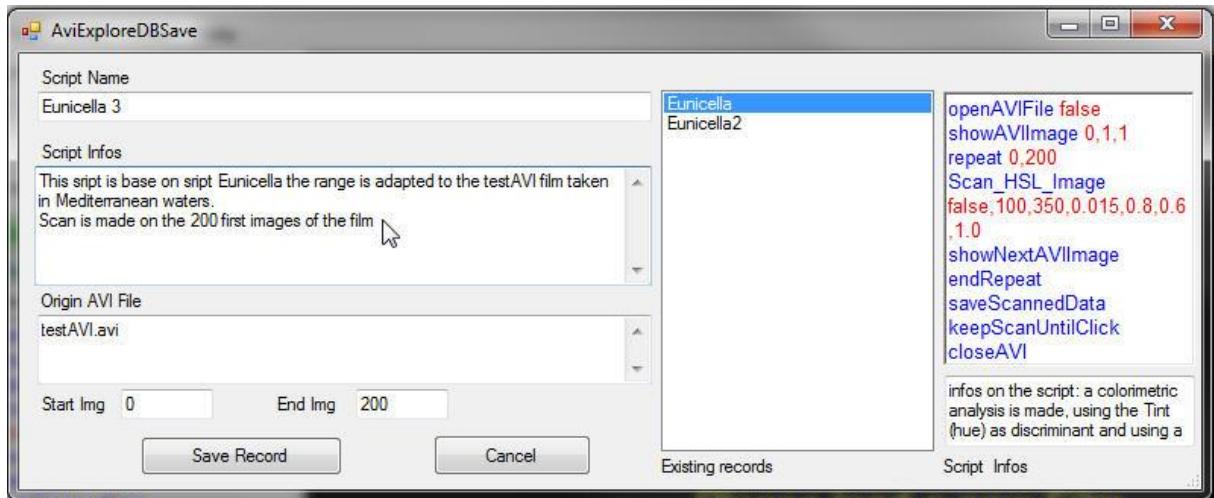


The DataBase indicator turns green. Now the 2 buttons '**Save>DB**' and '**Load<DB**' over the script panel are accessible.

4.8.2 - Saving a script as new record



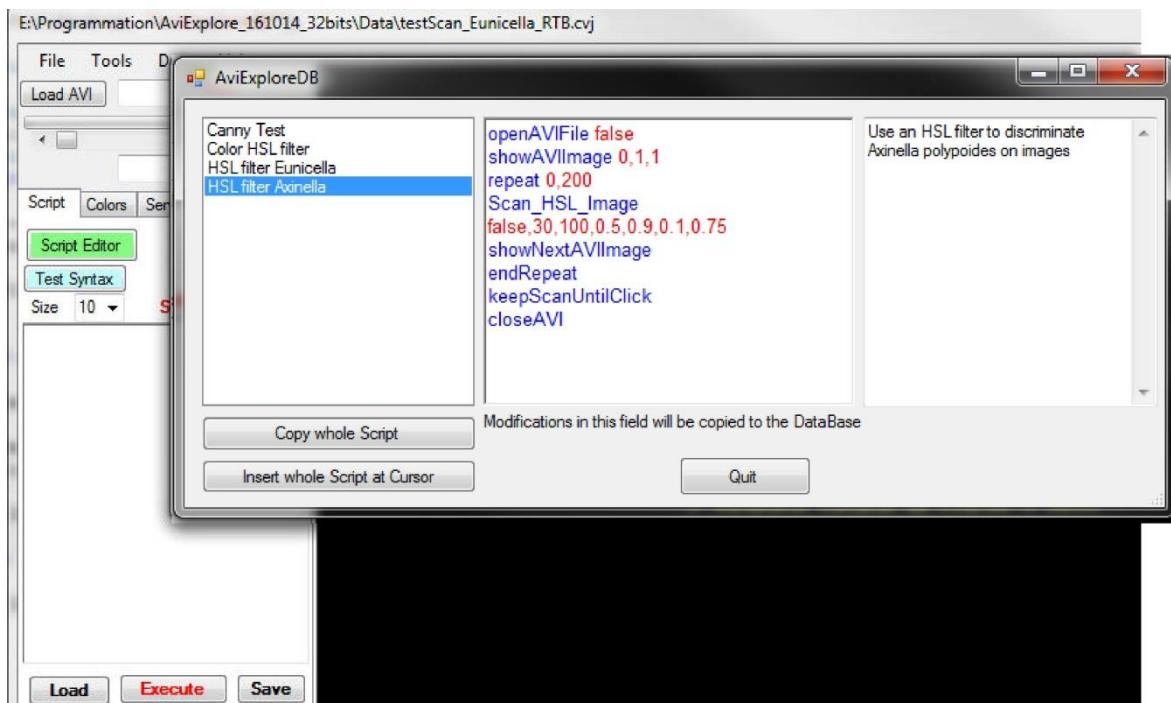
The database window allows creating a new entry in the personal database.



The name of the new script is entered. The scripts existing in the database are shown on the right panels for comparison. Clicking on existing record items shows the scripts and information stored in the database. Fields on the left concern the new script present in the main window. Information is typed as well as the source origin Avi file, and the start and end images uses to test the script.

4.8.3 - Getting scripts from the database

The 'load<DB' button gives access to a loading window.



You can select available scripts in the left list. A click in this list shows information on the script and information on that script. To copy the corresponding script in the main window 4 buttons are present. The two on the left let you copy the whole script replacing the whole window or inserting it at the caret position. The two on the right do the same operation but use internal dictionaries to remove instructions that cannot appear in real-time script like instruction to reate abi or log files, close it. Generally all instructions happening inside the

repeat / endrepeat loops are kept. These scripts are called image script because they are applied to single images. This is described in the next section.

4.9 - Particular case: scripts for real time calculations.



In the RT window a different environment allows semi-automated sorting of global functions. When a video streams brings many images at the timer pace, some function cannot be used.

An example of a script with automatic sorting:

```

Script Editor
Test Syntax
Size 10
Script to Execute
openAVIFile false
showAVIImage 0.1,1
repeat 0,200
Gamma 2.19
Contrast 2.1,Origin,Work,true
Scan_HSL_Image
false,100,350,0.015,0.8,0.6,1.0
showNextAVIImage
endRepeat
saveScannedData
keepScanUntilClick
closeAVI

```

Load Execute Save

The left capture shows a load in the ScriptEdit window. The script is complete. On the right side is an automatic loading for the Mobile Cam Input Window. The script is split between 3 boxes.

The upper one collects the Header functions, executed once (like openAviFile requesting a working AVI file).

The lower collects the global functions like the closeAVI command closing the active film.

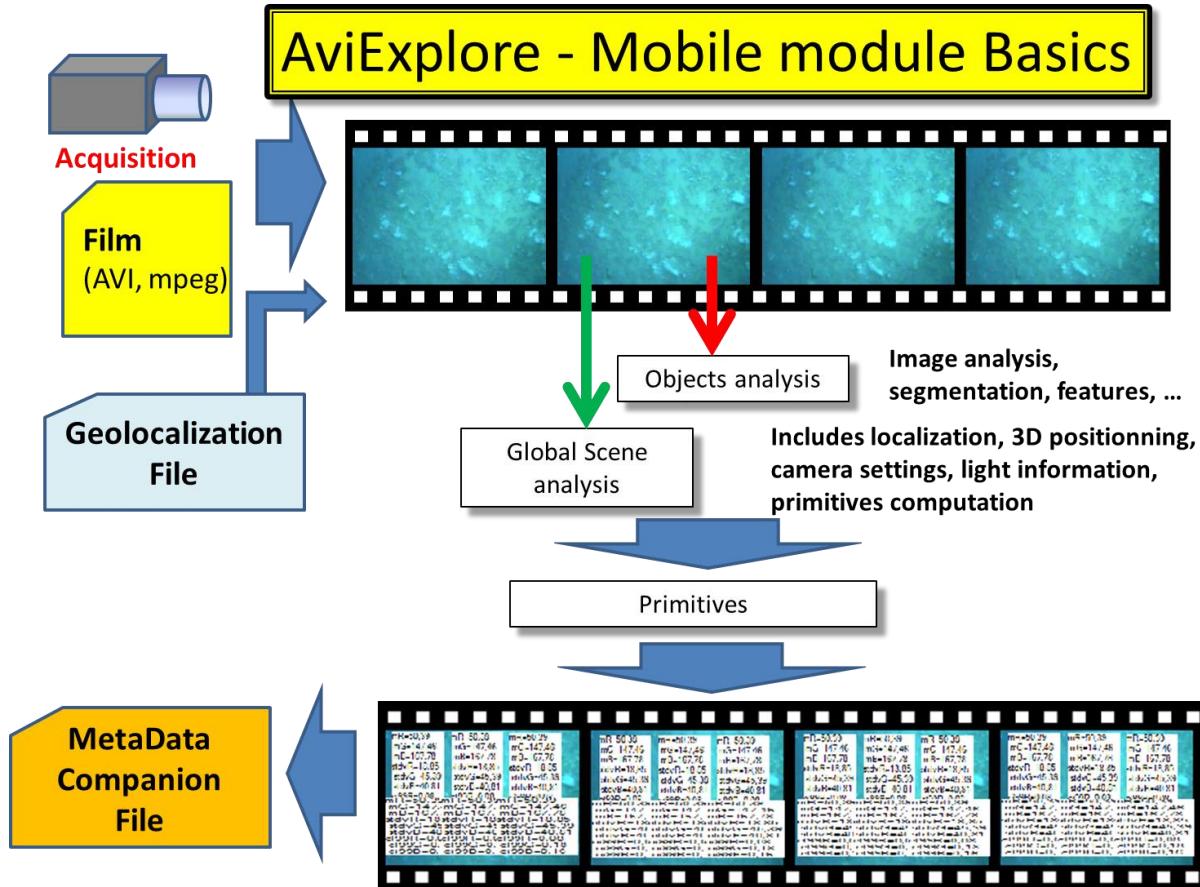
The center window filters the functions available for Real Time processing.

Script Edit
Script Header Active Clear
openAVIFile false
Script for images
Gamma 2.19
Contrast 2.1,Origin,Work,true

5 – Jerico-Mobile Module

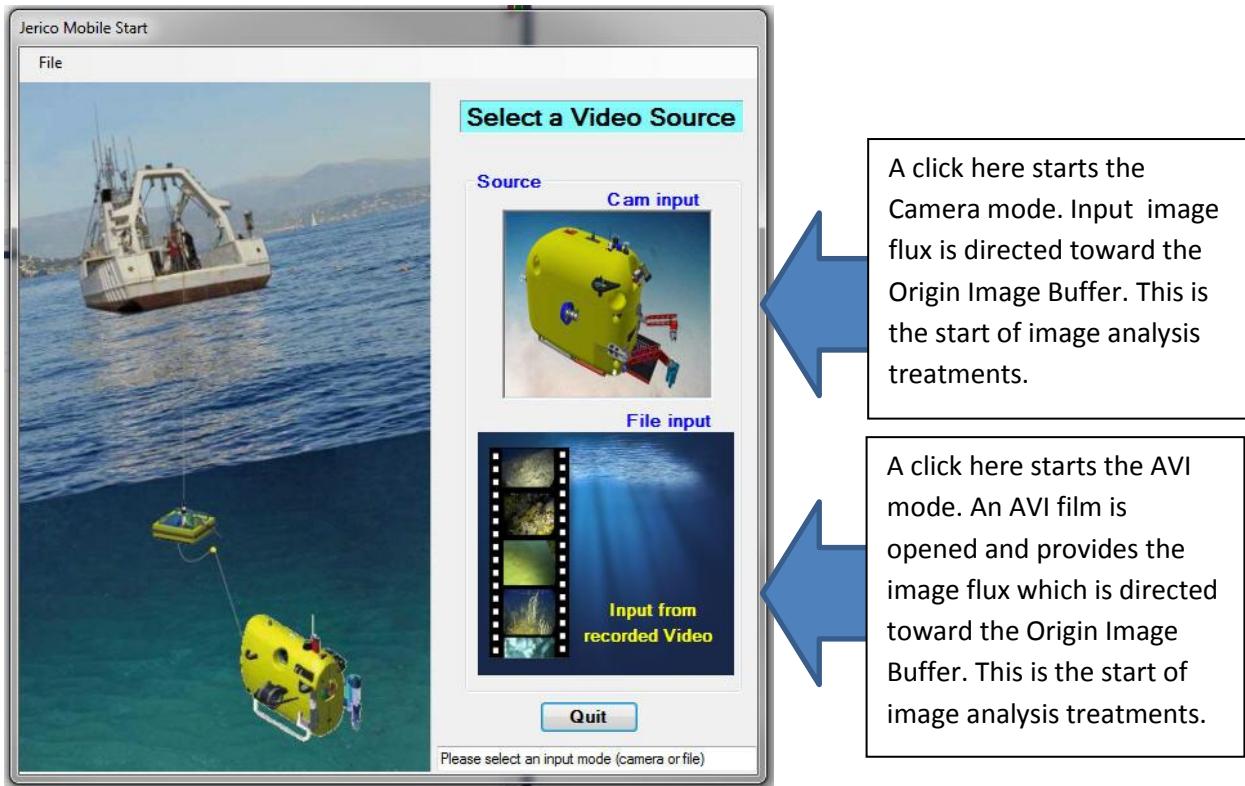
The AviExplore-Mobile module allows extraction of information on film taken with moving sensors and cameras.

The Camera input part of the module is dedicated to image acquisition with time-lapse recording. The interval between frames is used to compute simple features and produce a companion file containing global image parameters. If the interval between frames is increased then more computation can be done. If computation is longer than the time gap, then images may be lost.

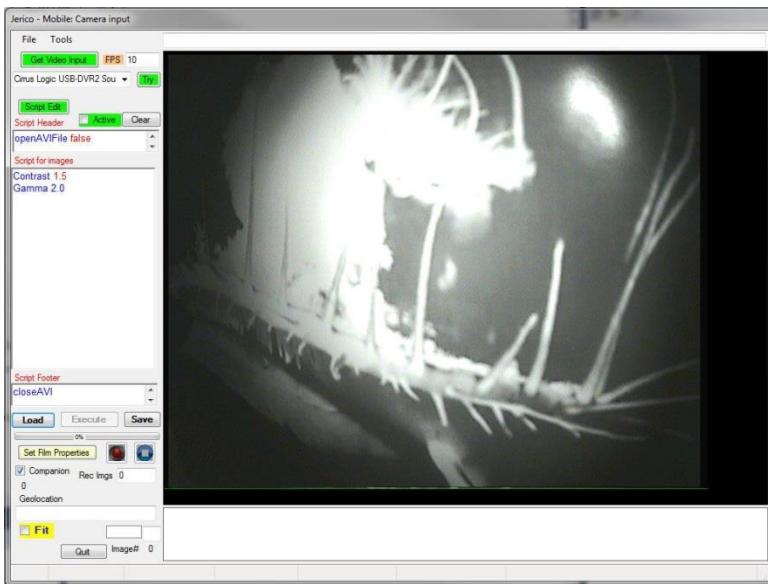


The AVI input part contains the same components but as long as time in not limiting, tools are present to analyze the data and select some interesting scenes. In this part, an AVI film produces the image flux and it is possible to analyze the similarity level between frames, using the companion files created. New companion file can be created, using different sets of measures. Different algorithms are used to compute the similarity level between a selected image of the film and the other images of the film.

5.1 Input selector

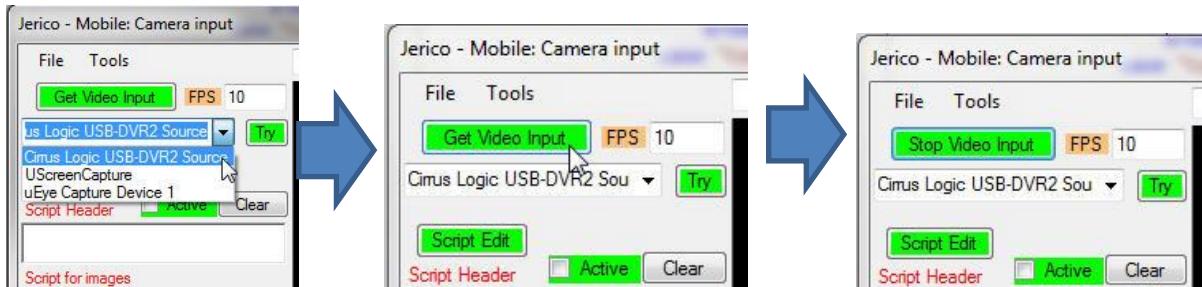


5.2 – Real Time Input (Cam Input)



5.2.1 – Video Input selector

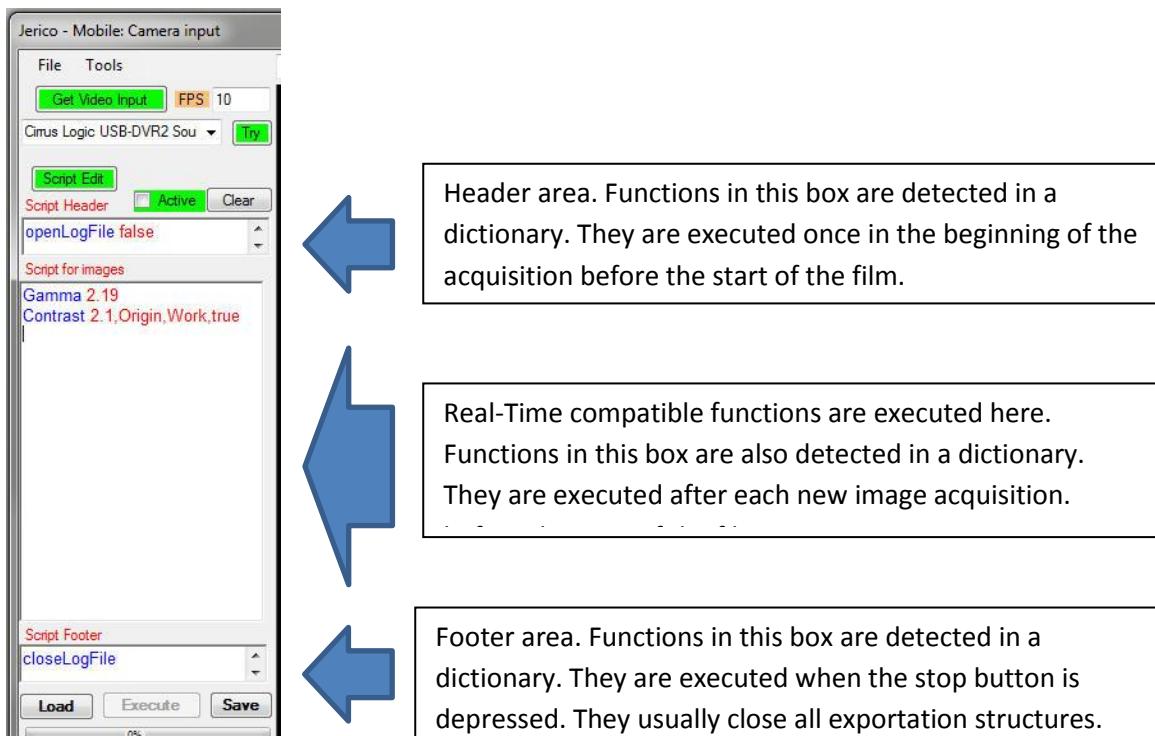
Available video inputs are detected in a scrolling device list on the top left of the window. A 'Try' button allows making a new scan of available devices.



The 'Get Video Input' brings you, if necessary, to a device dependent introduction window where you can set up your device. Then the real time image is displayed a full speed on the viewer. In the lower left part of the window an indicator gives an idea of the ratio of skipped images with a reset to 0 after 1000 images acquired.

The script portion in this module has been modified. Some commands are globals and must be used in the header or the footer part of the script. Other commands may be used within the main script portion.

5.2.2 - Real-Time Scripting



Function like openAVIFile, OpenLogFile, closeLogFile, closeAVI, ... can be used exclusively in the Header/Footer sections.

Some functions cannot be executed in this mode. This is the case for Repeat, EndRepeat, or showAVIImage, showNextAVIImage,...

Loops are not allowed because in real time mode the flux of images come from the camera. It cannot be reloaded. Each image is unique.

5.2.3 - Real-Time Recording

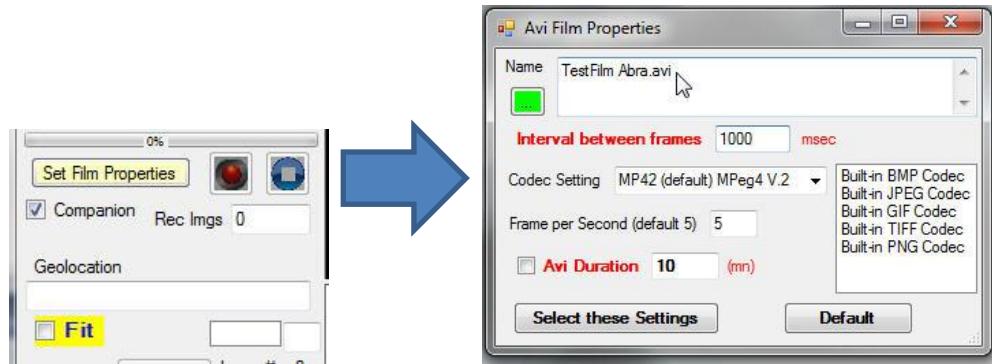
The recording takes place within a video session started with the ‘Get Video Input’ button.

When recording is finished, do not forget to end the session by clicking the ‘Stop Video Input’. This releases all the video acquisition structures.

If you check the ‘companion’ check then a companion file will be generated.

In the lower left part of the window is the recording buttons. Three operations are necessary:

- Set the film parameters, with the ‘Set Film Properties’ button.

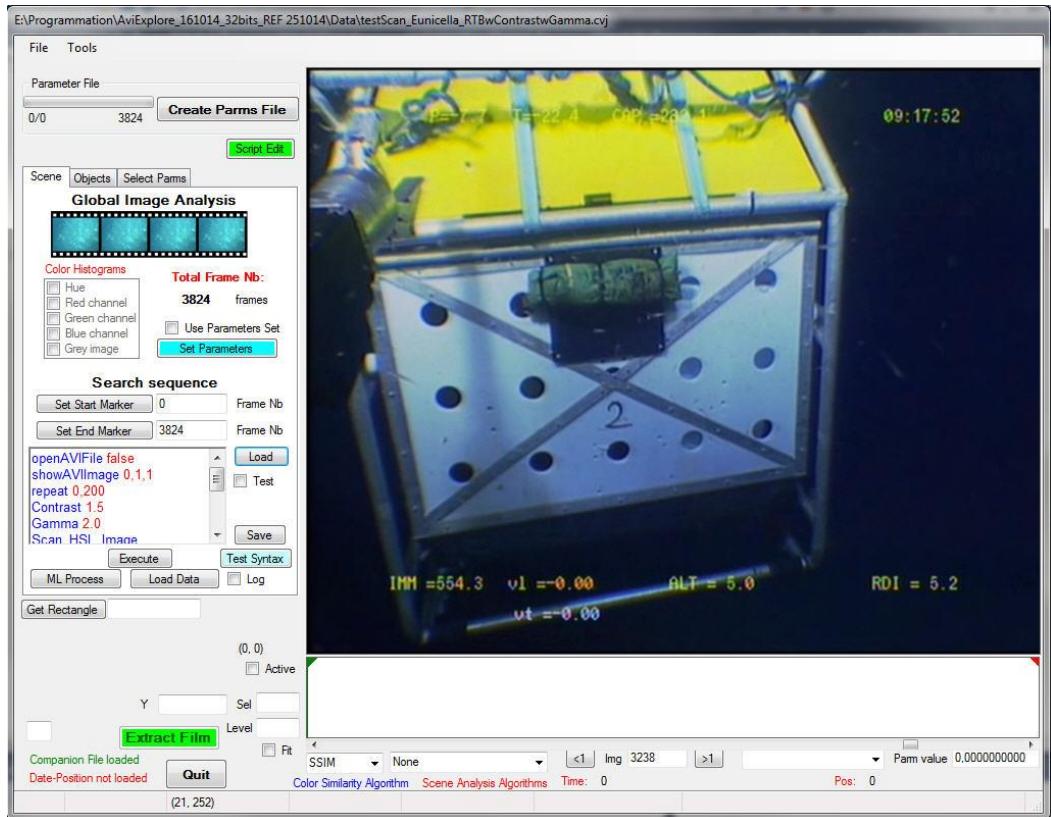


This brings you to a configuration window where you give a name to the future film, select its codec among the available ones (Mpeg 4 Version 2 is used in the program (MP42), its frame rate (default 5) and the AVI duration in minutes.

- Start the film with the button. The film will be closed automatically when the requested duration is achieved.
- Stop the film before normal duration by using the

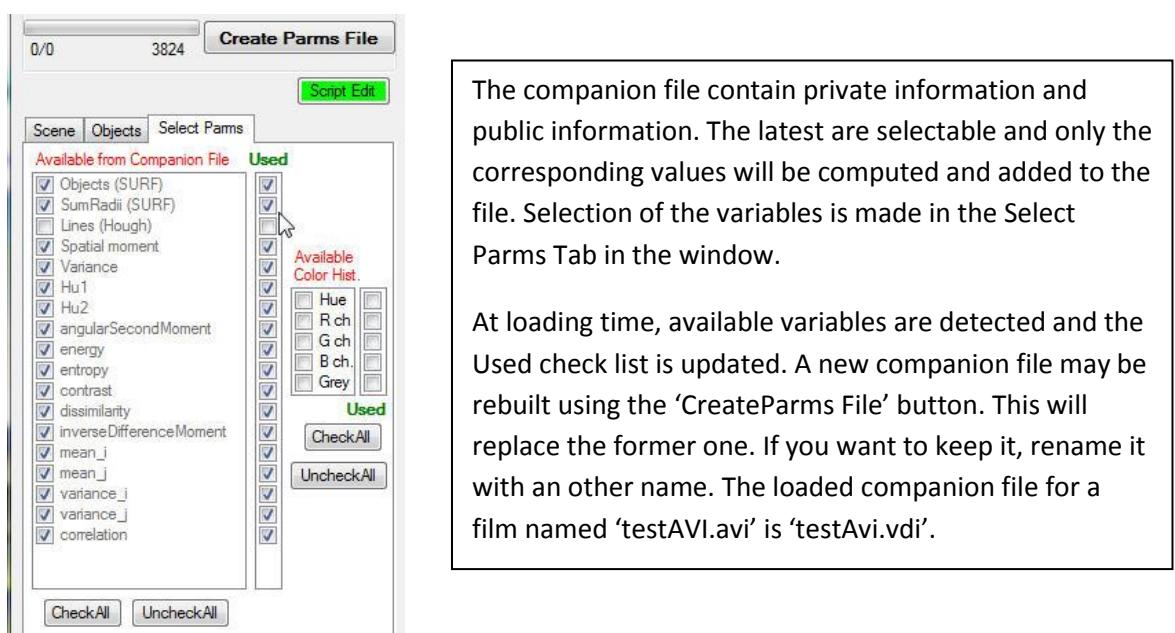
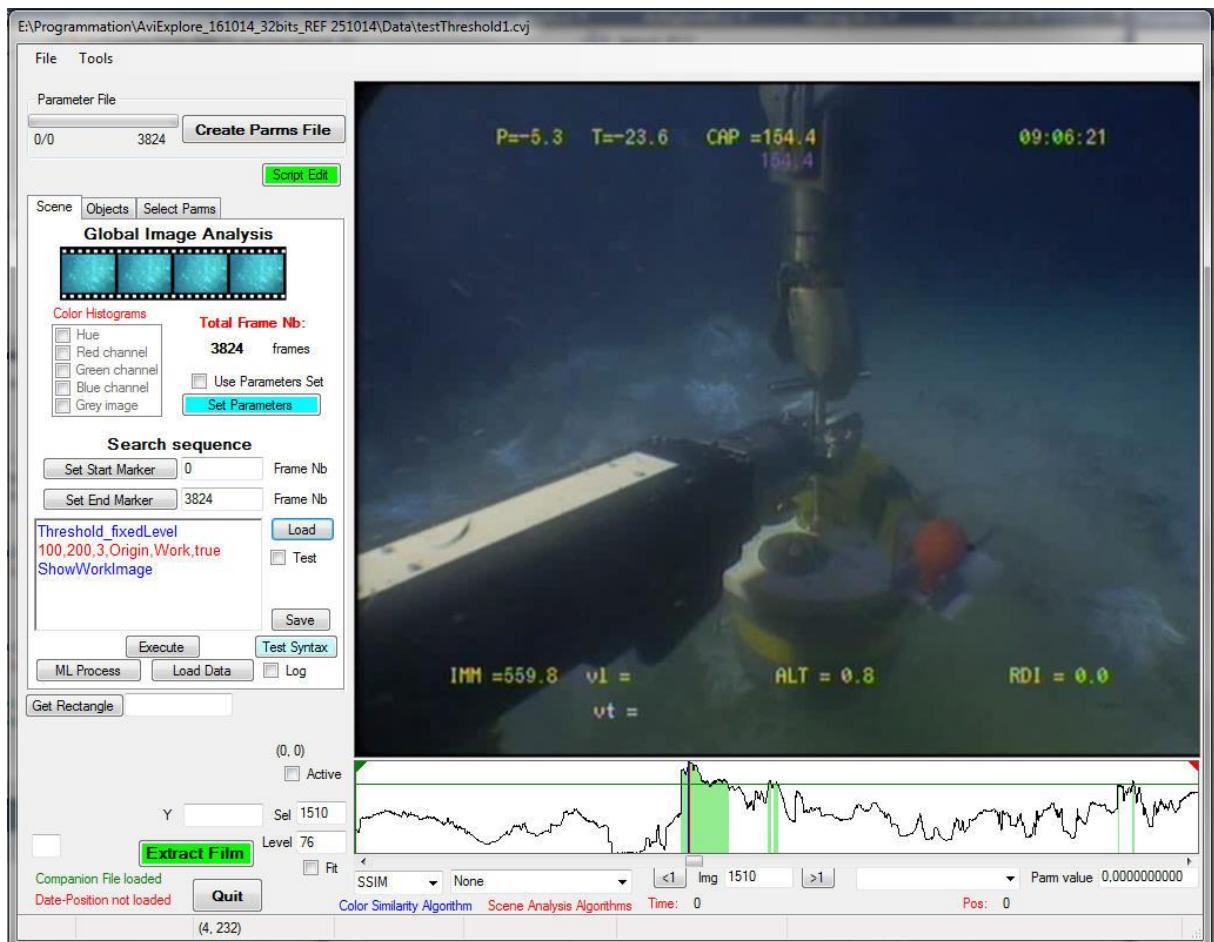
5.3 – AVI Input

The AVI input of Jerico-Mobile module is devoted to global frame analysis. A companion parameter file is used, which contains computed information on every frame of the film.



5.3.1 – Managing the Companion File

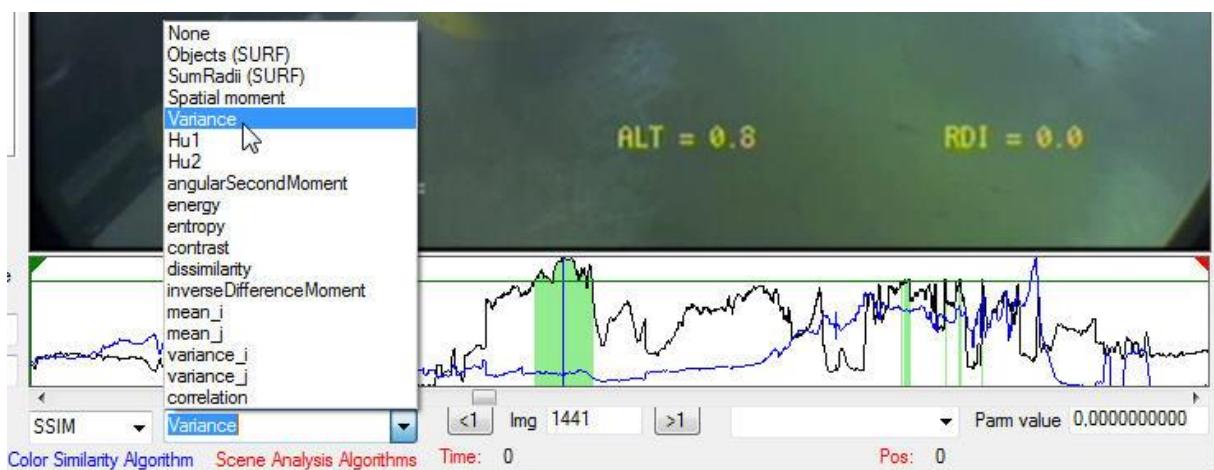
The window is opened on a selected AVI video. When loading the AVI the program searches for a companion file (a xxx.vdi file). If it is not found in the data folder (in the same folder as the file) a message is sent. A red indicator is visible in the lower left part of the window. The companion file can be created with the '**CreateParms File**' button in the window. If it is present, then it is loaded transparently. The indicator turns green. This file contains information selected when it was saved. The configuration of the companion file is selectable at creation time.



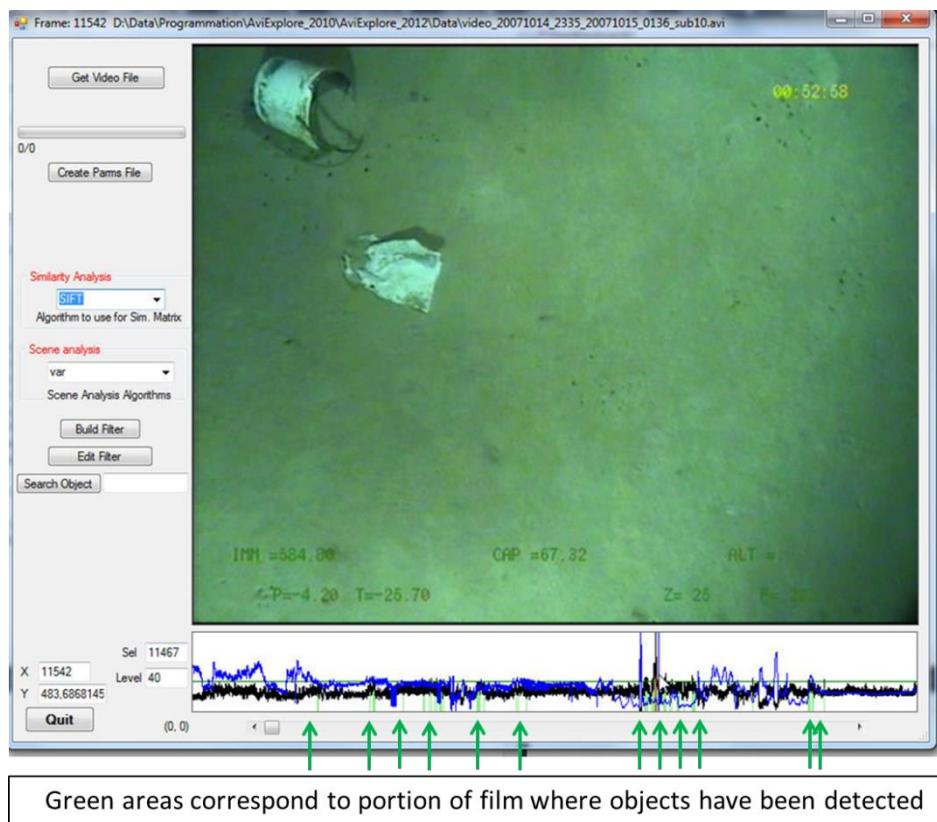
The lower part of the window contains the active window. Moving the mouse over or clicking the window shows or starts computations. Two scroll list below it give access to the algorithm used for similarity computation. Next to it is a scroll list with the parameters from the companion file. A click in this window selects the current image and computes similarities with all other images from the film. This is shown on the black curve visible in the active window. As a remaining the scene analysis scroll list allows displaying the various parameters stored for the film images in the companion file (the blue curve). These data may be exported as text file by a menu in the Tools Menu.



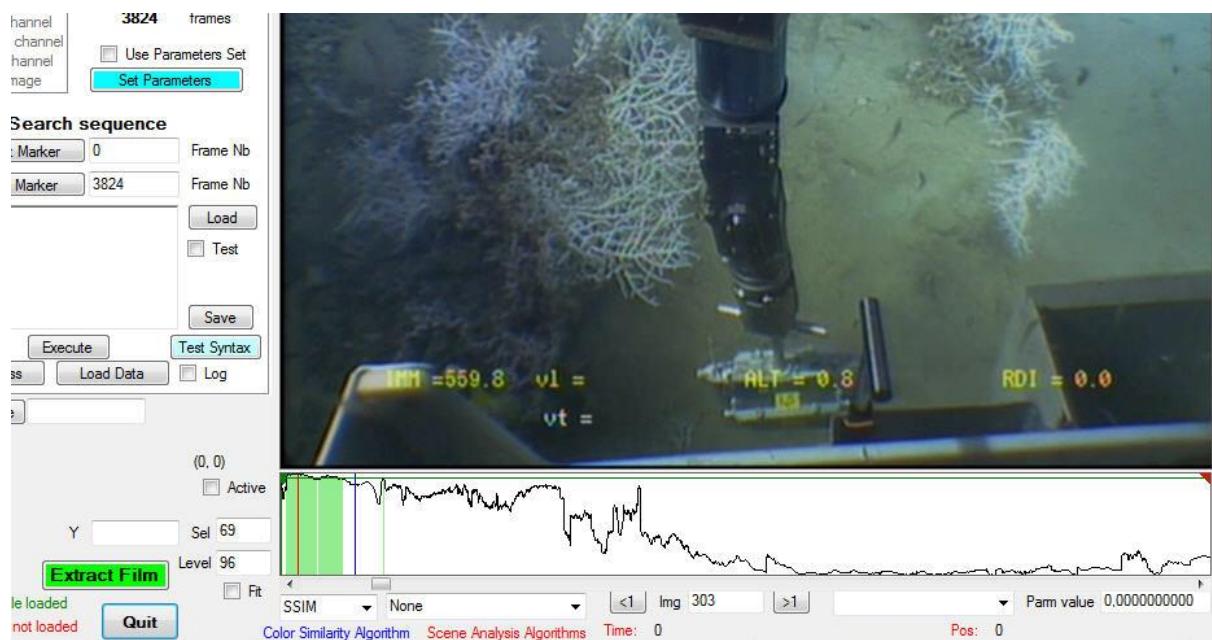
This example shows the SSIM similarity for image 1220, indicated by the vertical blue line. A right-click on the active window positions a green line for similarity level. The value of this selected level is indicated on the text box on the left of active window. All images having a similarity higher than the target level are shown in green colour. A click on the 'Extract Film' button allows creation of a new film created from the original film containing only the images with green colour. A Time-Position file is created with geolocalisation information corresponding to the saved images. An information file is also created with film, image and parameters info. This file is reduced version of the original film with selected frames.



Selection of one of the parameter displays a blue line showing the parameter computed for every frames.



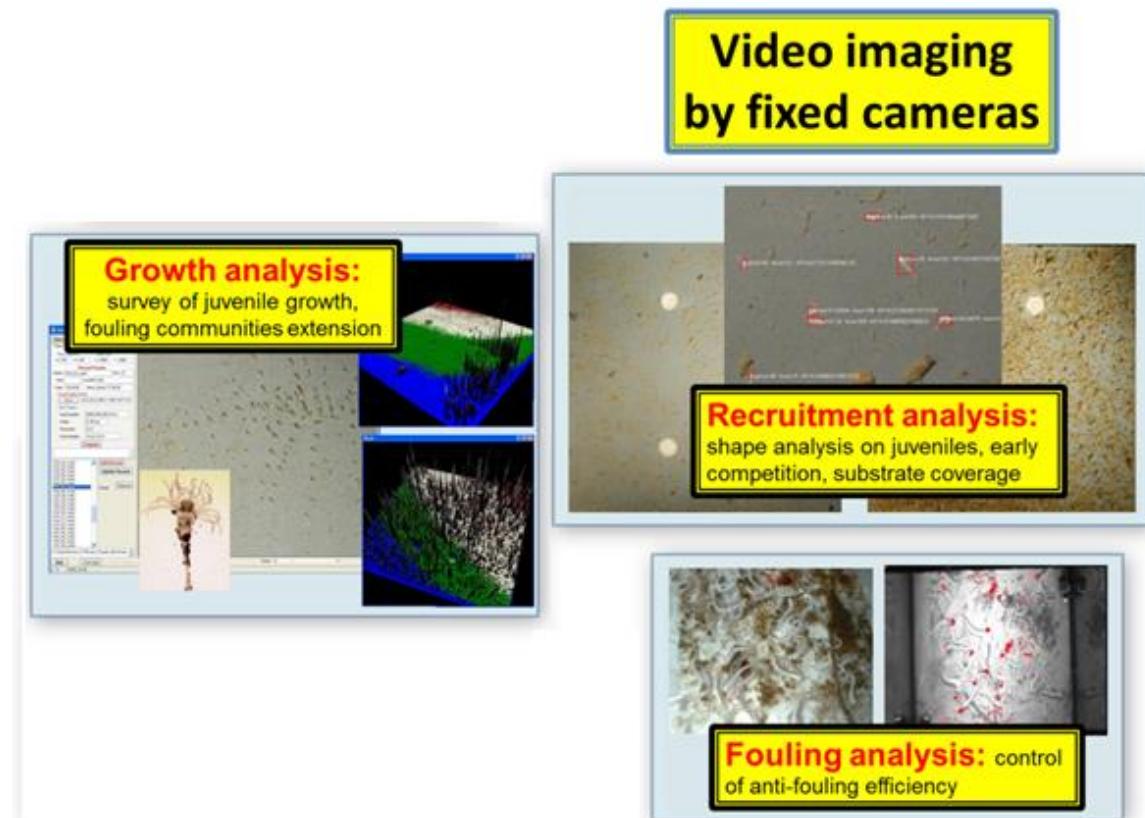
It is possible to extract the selected images to a new contracted film containing only relevant images to be analyzed further on.

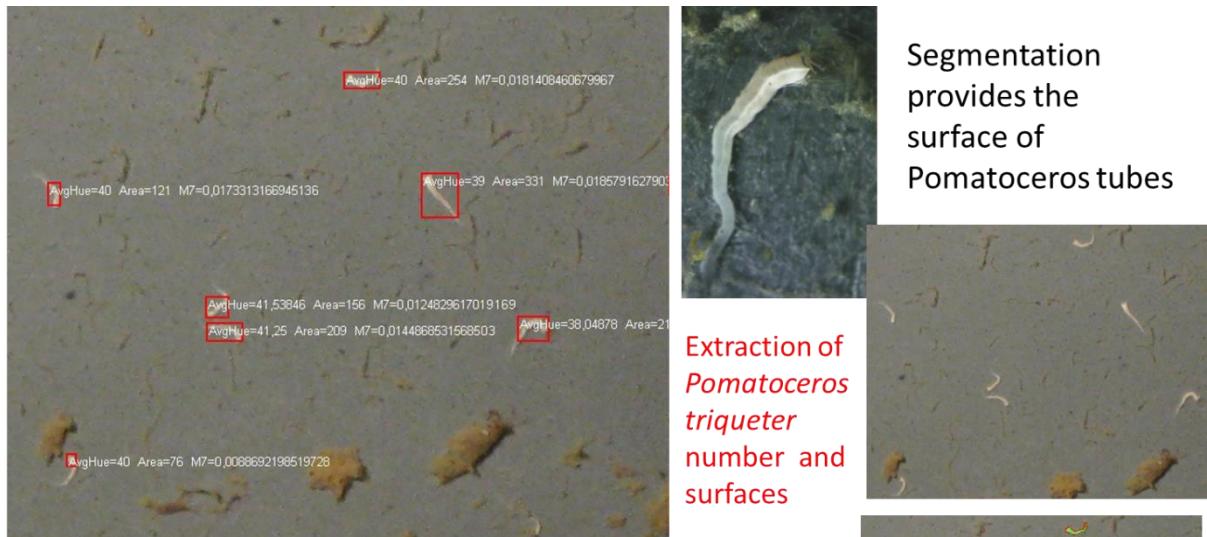
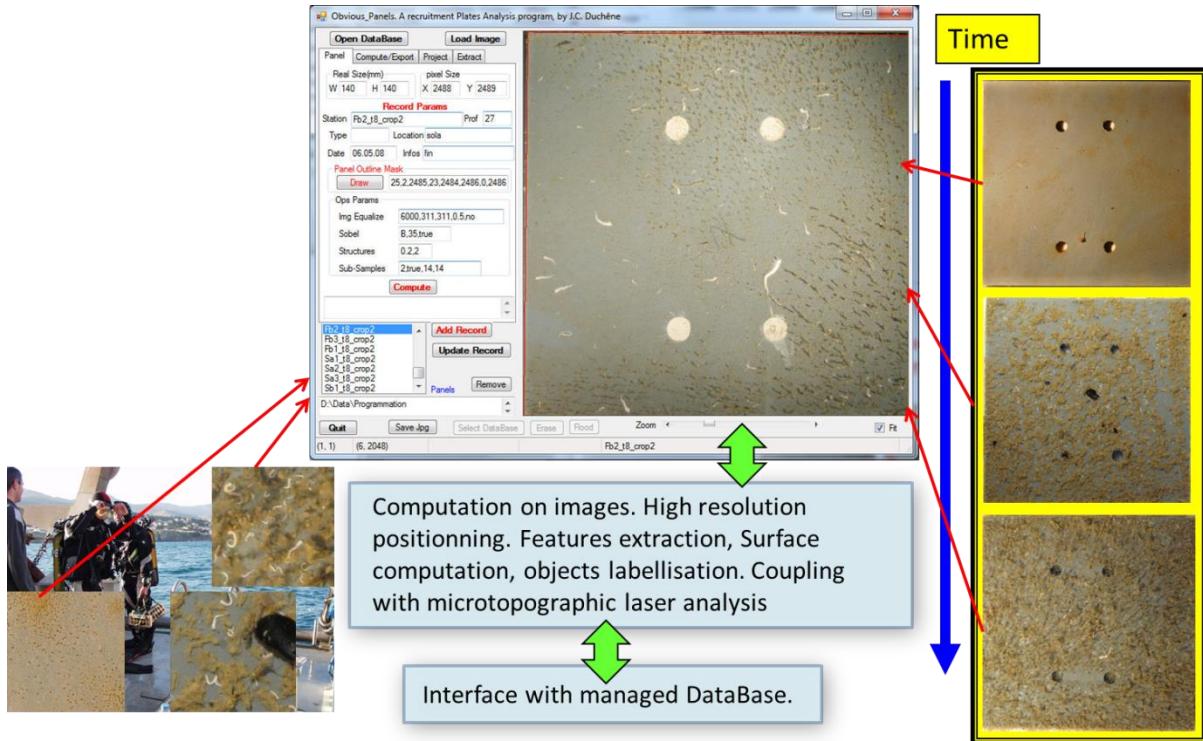


It is possible to load, modify and save scripts in the window for testing purpose and computations.

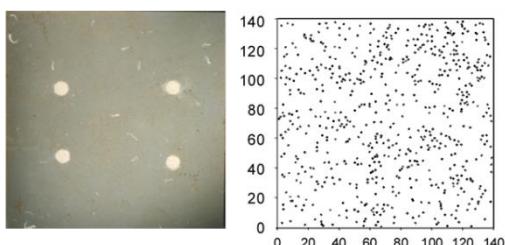
The public parameters of a Companion File can be exported in a textual format with a menu in the Tools menu.

6 – Jerico-Fixed Module

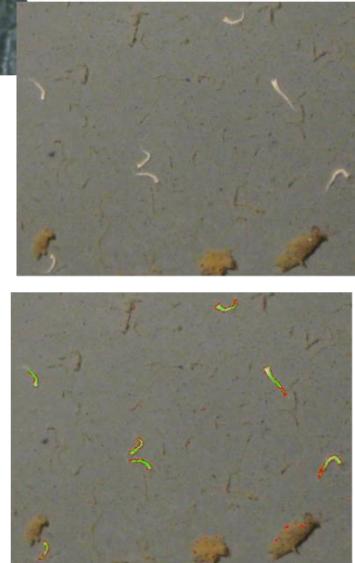


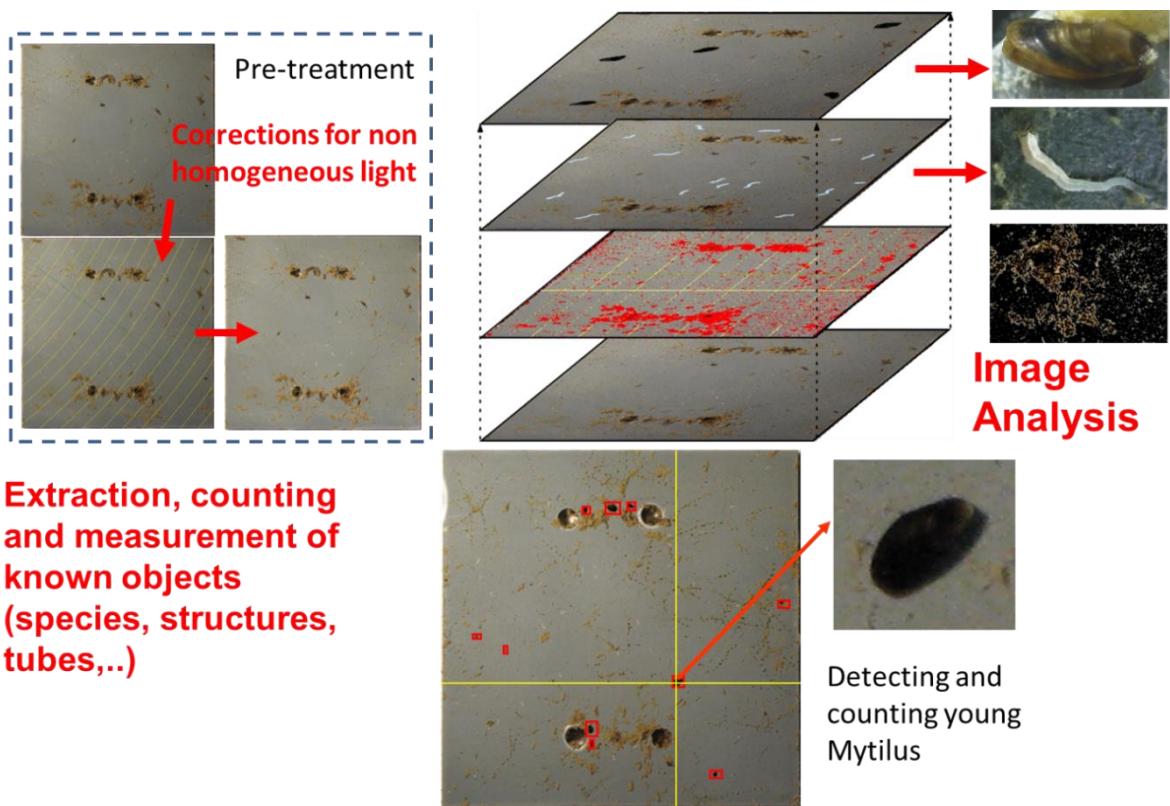


A training phase (user driven) allows to use a specific recognition set of routines to extract and count individuals.

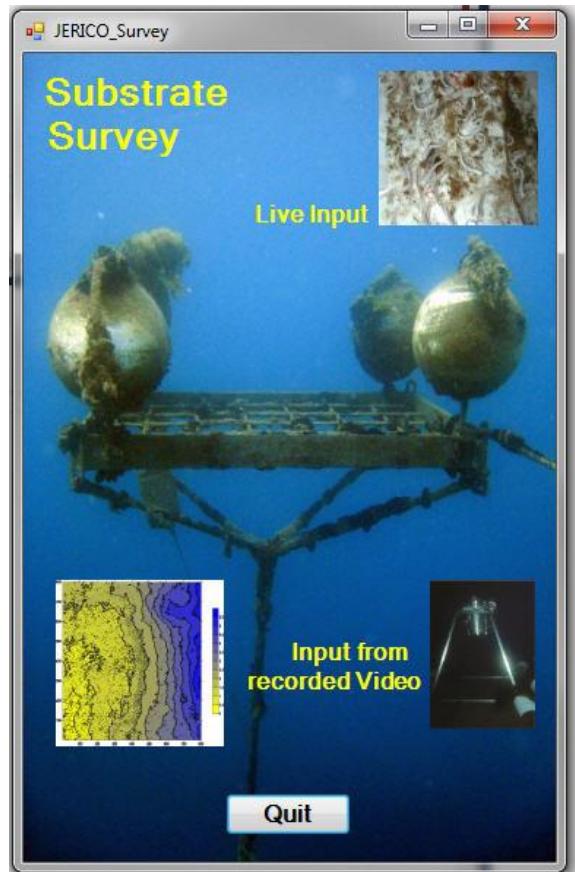


Pomatoceros counting on a 140 x 140mm panel





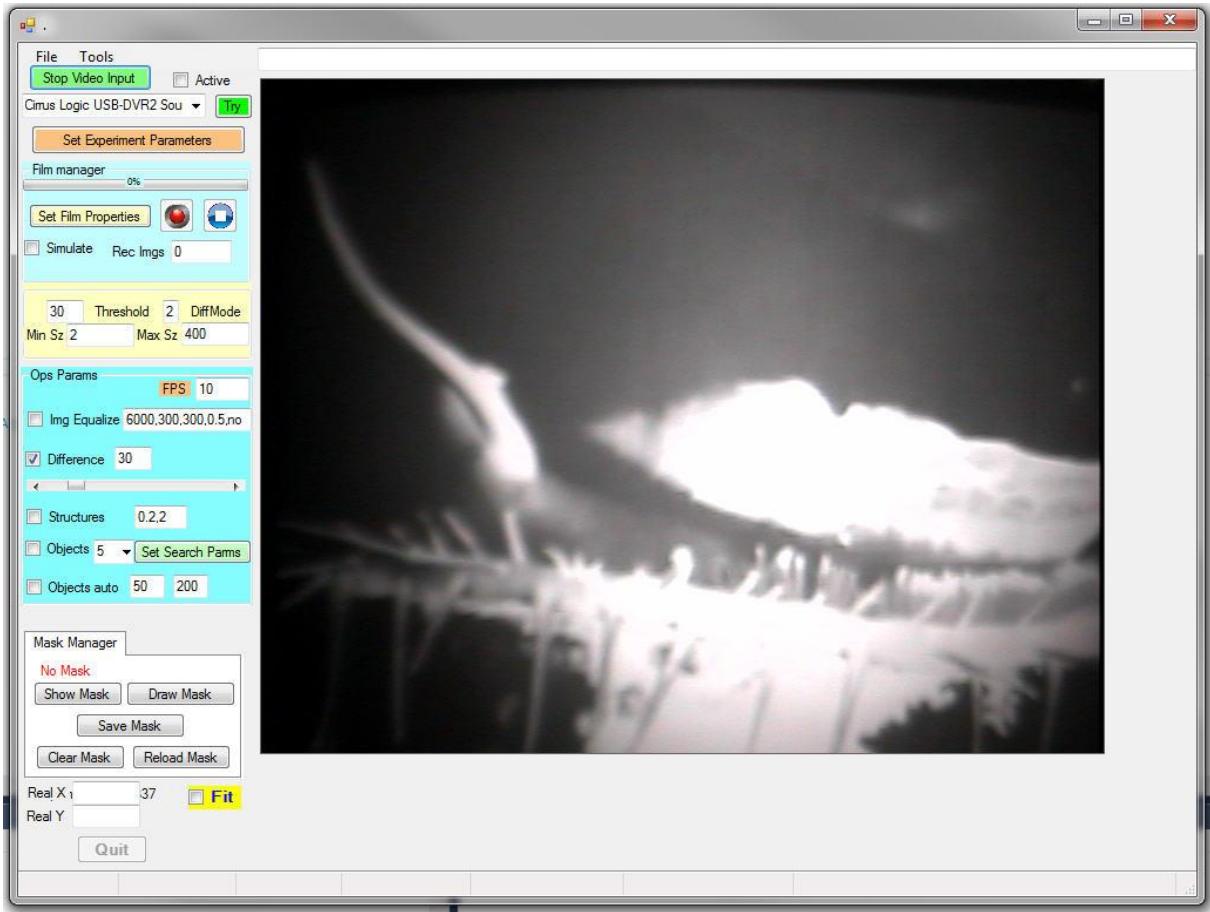
6.1 - Selecting the input mode



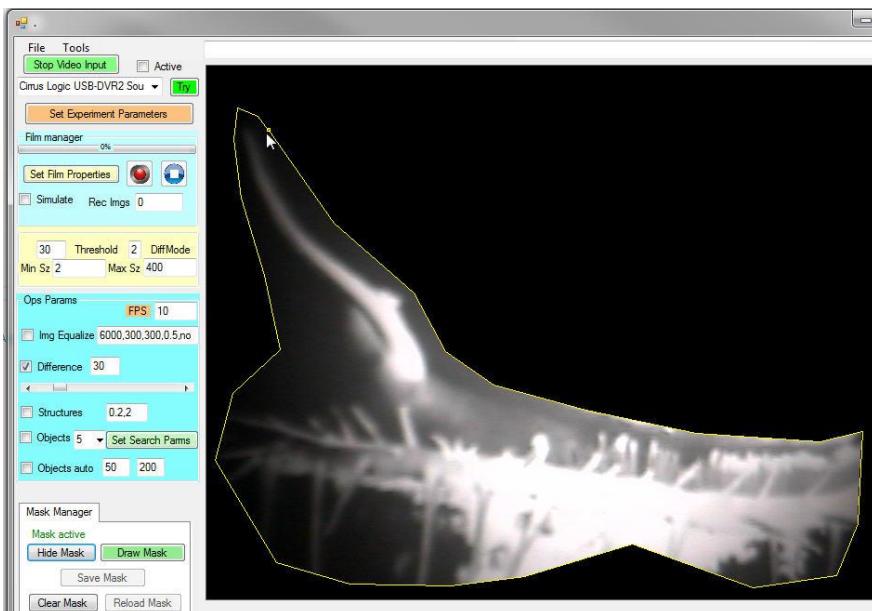
A click here starts the Camera mode. Input image flux is directed toward the Origin Image Buffer. This is the start of image analysis treatments.

A click here starts the AVI mode. An AVI film is opened and provides the image flux which is directed toward the Origin Image Buffer. This is the start of image analysis treatments.

6.2 Real Time Input



The real time window allows connecting and recording video from sensors. The flux is obtained from drivers and devices connected to the computer. It is possible to mask part of the image.



6.2.1 - Video Input selector



An input selector allows to choose the input device amongst the available ones. Then the video session is opened with the '**Get Video Input**' button. It will remain opened until the same button is depressed to finish the session. During this time images are acquired at maximum rate on the driver and displayed on the viewer. When this session is opened it is possible to start film recording.

When the Window is connected to the video flux, then it is possible to record images. During the record, the program creates and feed a data file (xxx.jcd file) and a Time-position file (xxx.tpd file) with current time and position. These files can be edited later.



Video input section (input selection, starting and stopping the video flux). The try button scans the video entries available on the computer.

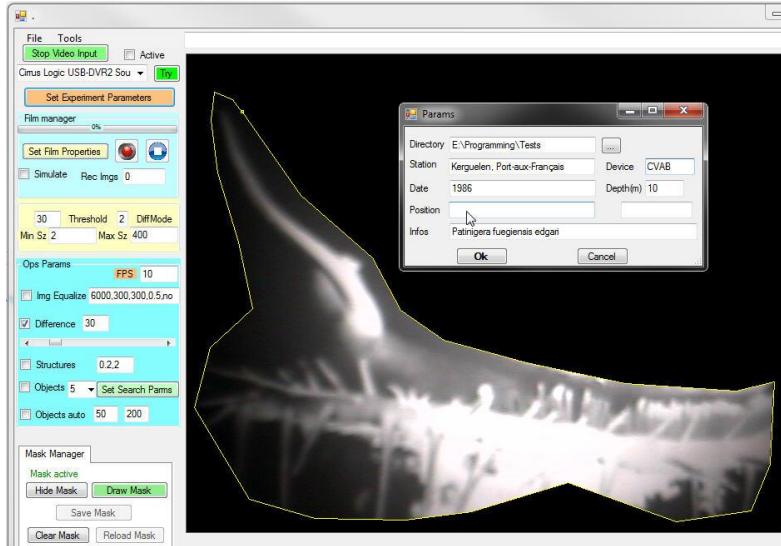
Setting the experiment parameters (infos)

Recording section. Enter name of the created video, frame interval, film duration,...

Parameters for the Data file created and completed during the recording.

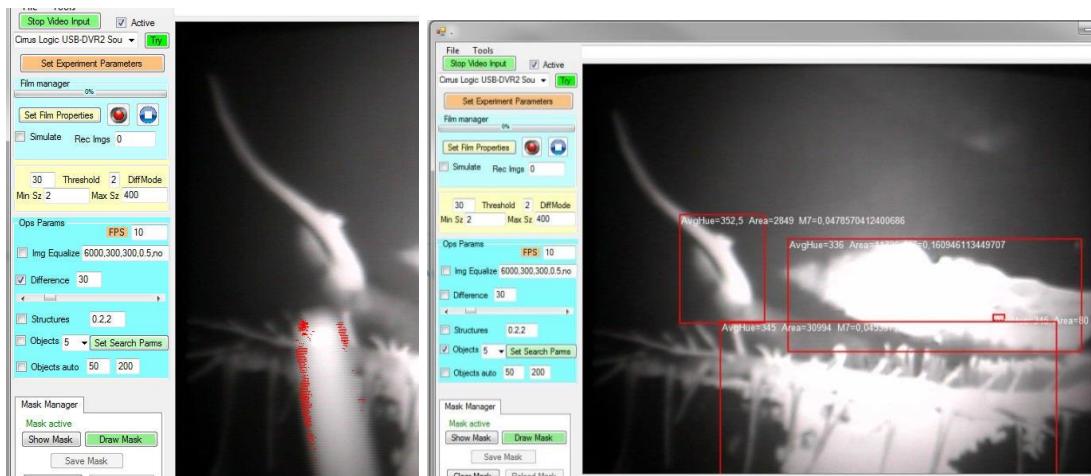
Applying filters to the video flux. This is activated by the 'active' checkbox on the top of the window, next to the Start/Stop Video Input button. 5 filters can be tested. These filter can be replaced by scripts in the scriptEdit module. Image equalize corrects the illumination of the frame related to a side projector by changing the gamma in some parts of the image. Difference, Structures and Objects have their counterpart in the scripts.

When the check box ‘**Active**’, next to the ‘**Get Video Input**’ button is checked then filters visible in the ‘**Ops params**’ may be applied. Check boxes allow selecting combinations of the filters. Some filters are long to process and the time interval between image acquisition must be set long enough for the process to achieve.



parameters

Setting the experiment



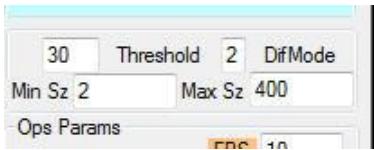
Applying the difference filter.

Searching for objects in the image

6.2.3 – Data File and Date Time-Position File recording

During the film acquisition a data file is created and a TPD file is created. These files have the same name as the recorded AVI file but have .jcd and .tpd endings.

The Data file records activities detected in the images and stores objects information as well as Date Time of the image acquisition.



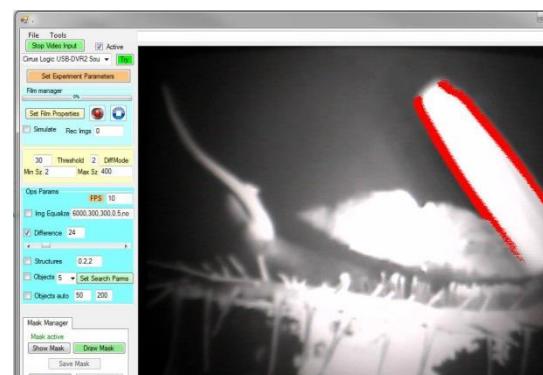
It uses parameters you enter in the Data box, with the Threshold value, the DifMode value, the min and max size of objects and the Real X and RealY size you enter (near the Quit button).

Threshold here is used to distinguish difference between pixels of succeeding images superior to a given value. The lowest the threshold the more pixel will be collected. Blobs, or pixels aggregates, are computed and objects are created from this scan. The DifMode indicate the way analysis is done:

DifMode=1 -> absolute value of difference;

DifMode=2 -> only positive values;

DifMode=3 -> only negative values.



Effect of a moving object with absolute difference

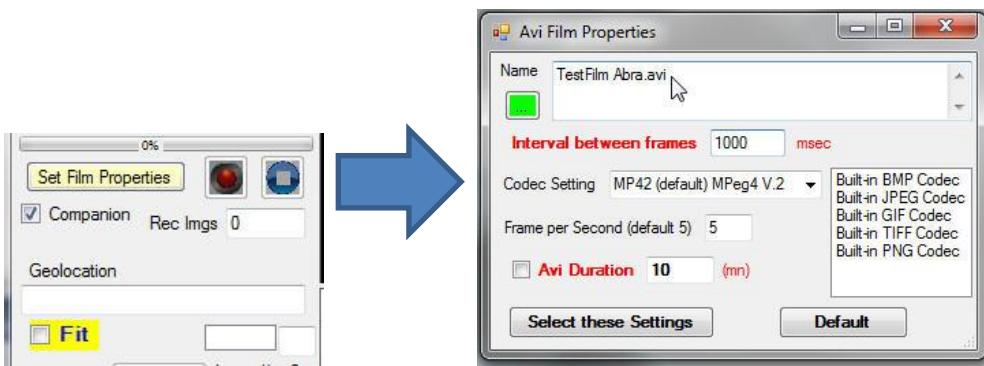
In this real-time version, comparison is made with the preceding image. In the Avi module, it is possible to select between preceding image, image separated by a given gap or a fixed reference image. A filter is made using minimum and maximum blob surface, to eliminate very small surfaces (likely to belong to image noise) or very large surfaces (that cannot correspond to the searched biological target for example).

TPD file stores the current image time and position (Latitude-Longitude) collected from the Date Time-Position manager (accessible in the Tools menu).

6.2.4 - Real-Time Recording

Real Time recording is made, during an opened video session, in the Film manager box.

Set the film parameters, with the 'Set Film Properties' button.



This brings you to a configuration window where you give a name to the future film, select its codec among the available ones (Mpeg 4 Version 2 is used in the program (MP42), its frame rate (default 5) and the AVI duration in minutes.



Start the film with the button. The film will be closed automatically when the requested duration is achieved.



Stop the film before normal duration by using the button

6.2.5 - Mask manager

This box lets you draw, save, load, clear and show a mask that will be applied on the image captured on the external device. This lets you eliminate unwanted parts of the image such as the sides of an immersed plate.

6.3 AVI Input

This module allows survey of activity on surfaces by using films taken with a fixed camera.



The module allows creating Datafiles with different configurations ('**Make Data File**' button), opening created datafiles. The Tools menu has a menu to analyze trajectories, Export activity Intensities, manage TPD files and Show the recorded current Global Position on a satellite map.

Check boxes control the use of ROIs, the data display or the display of in between images difference. It is possible to display the whole set of detection across the film to detect slowly moving objects or detect area where animals concentrate their activity.

6.3.1 - Activity measures

6.3.1.1 - Building a DataFile

Activity measure allows estimating local activities from images scanning. A data file can be created, incorporating time, position information as well as activities found in images by image analysis.

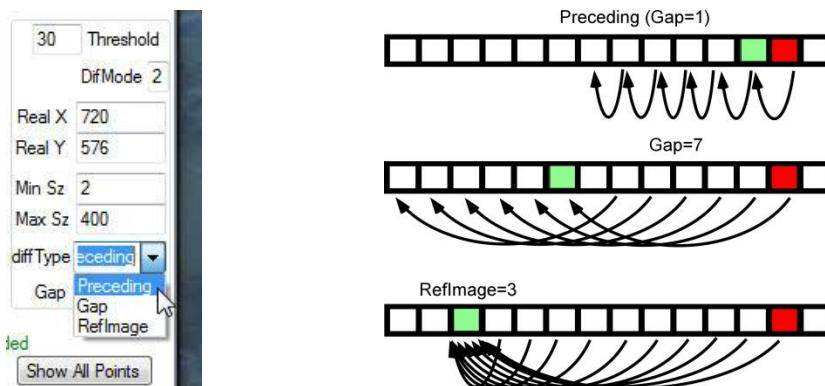
When you build an AVI film from Jpeg images, in the ScriptEdit window, or load a new film in the Jerico-Fixed AVI module there is no corresponding Data file present. You can use the 'Make Data File' button in the window to build a first version of the data file. It will scan and analyse the current film, using the Threshold value and the DifMode value.

Threshold here is used to distinguish difference between pixels of succeeding images superior to a given value. The lowest the threshold the more pixel will be collected. Blobs, or pixels aggregates, are computed and objects are created from this scan. The DifMode indicate the way analysis is done:

DifMode=1 -> absolute value of difference;

DifMode=2 -> only positive values;

DifMode=3 -> only negative values.



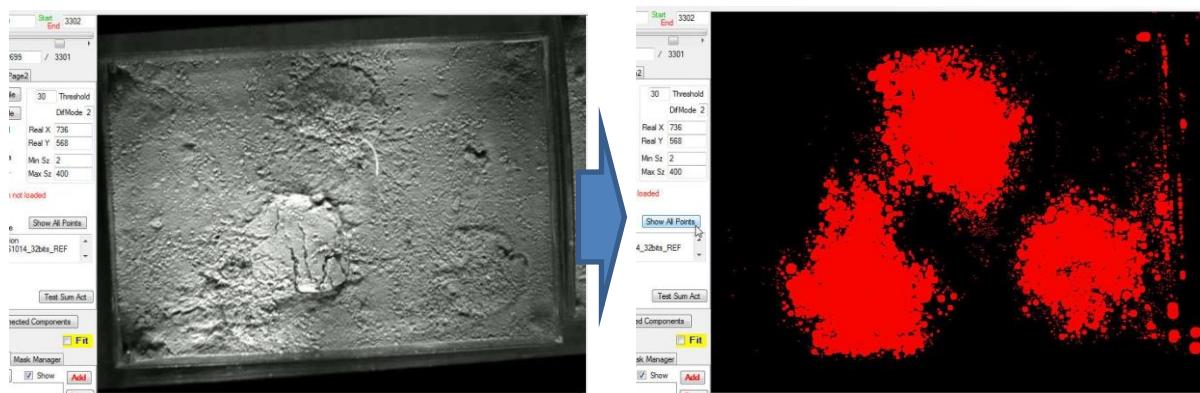
A filter is made using minimum and maximum blob surface, to eliminate very small surfaces (likely to belong to image noise) or very large surfaces (that cannot correspond to the searched biological target for example).

The difference type can be: Preceding, Gap or RefImage. If Gap is selected then you must fill the Box Gap, providing the interval between images to compare.

You can create many data files with different parameters.

If a file has the same name as the film you open (ending with a .jcd type) it will be loaded automatically with the film. You can rename your data files to have the one you prefer loaded with the film. If it is loaded an indicator turns green on the upper part of the Activities panel, indicating a ‘Data opened’ status.

When no ROI is present, objects labelling is null. This means that all the objects have a 0 label. If you click on the ‘**Show all Points**’ button then you will see an additive scan of all detected activities on the window. Points are shown in red if the ‘useROI’ check box is left unchecked. If this checkbox is checked but there is no ROI, points are shown in blue.

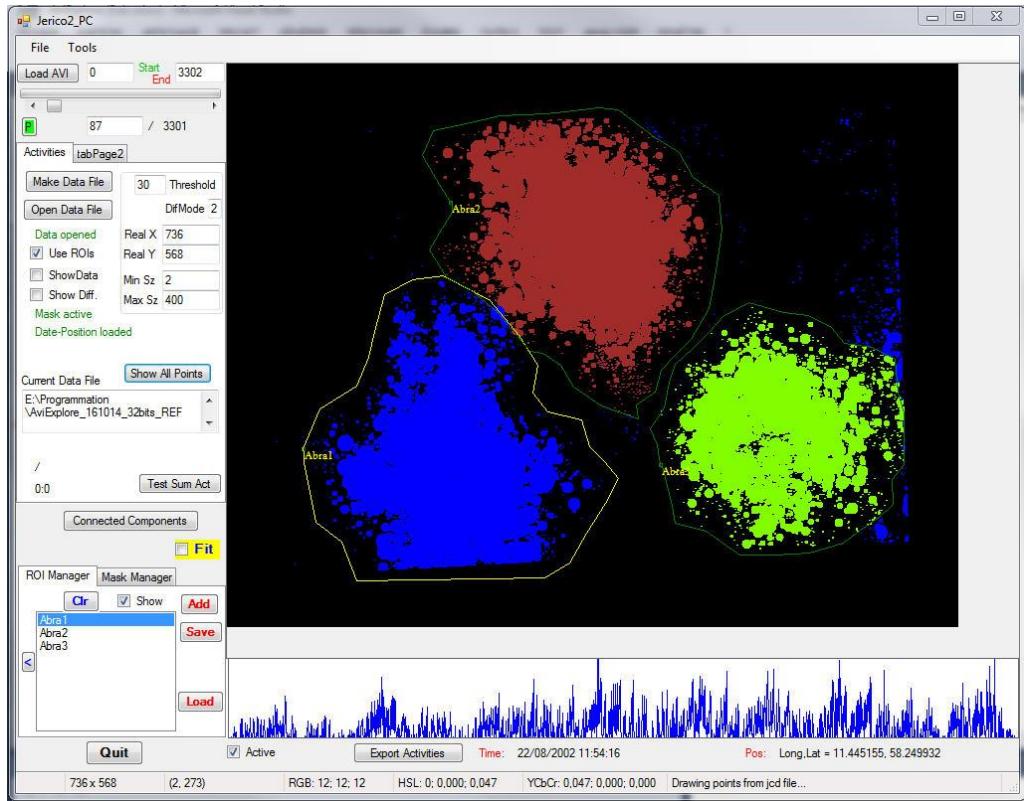


You can use the mask manager to load a mask (file type .msk) and use it to mask the image. Mask file can be created by drawing a mask in the mask manager, then saving it to a .msk file. This mask can be reloaded later.

If a mask is present during the creation of a data file it is saved inside the data file. So it is automatically reloaded with the data file loading. If the data file contains ROIs then clicking on ‘Show all Points’ button displays the objects belonging to each ROI with a different color (8 colors with a modulo operator).

When the data file is created the sum of all activities found in ROIs are stored in the data file. In the ROI manager, clicking on a ROI name shows the ROI outlined in yellow. Other ROIs are shown with a green outline.

6.3.1.2 - Exporting a DataFile



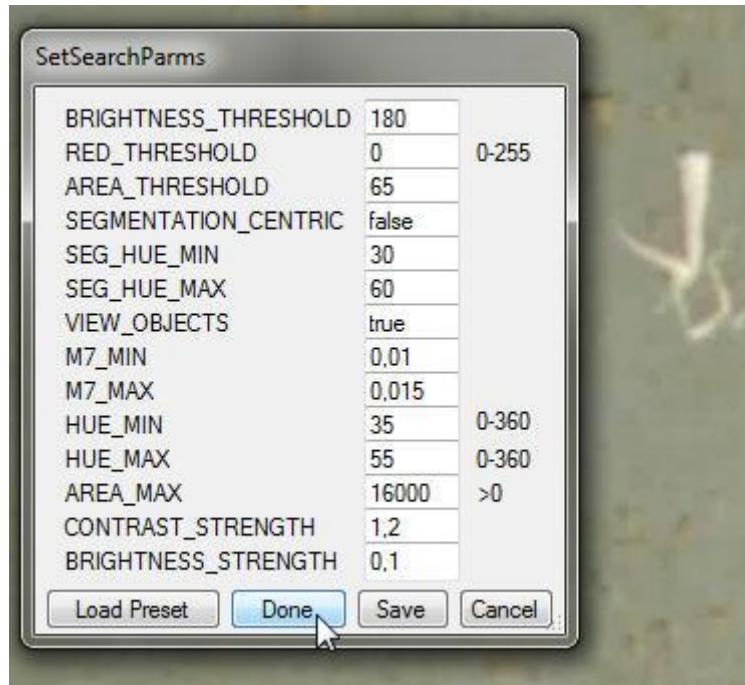
The intensities values correspond to the sum of surface activity found in each image inside a common ROI. It is stored in the data file and can be exported with the button 'Export Activities' on the lower part of the window or with the 'Tool> Export Intensities' menu.

6.3.2 – Extracting blobs information

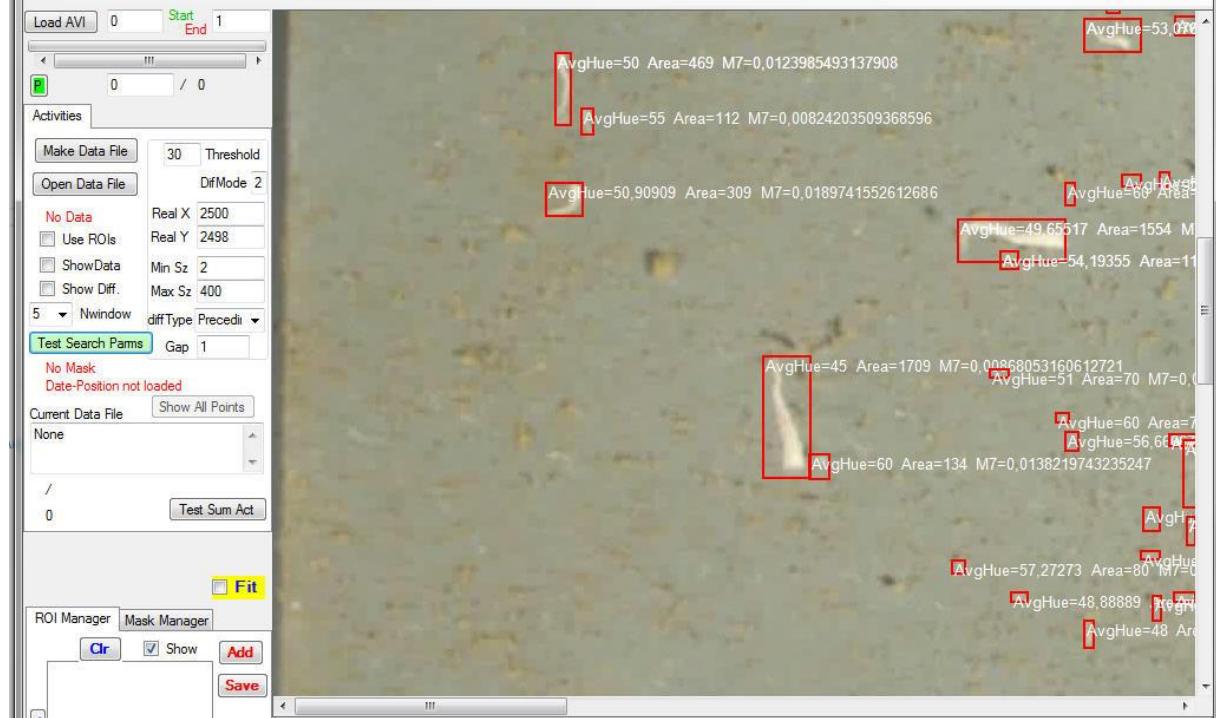
A button allows testing parameters for data extraction. For testing, data can be exported to text file.



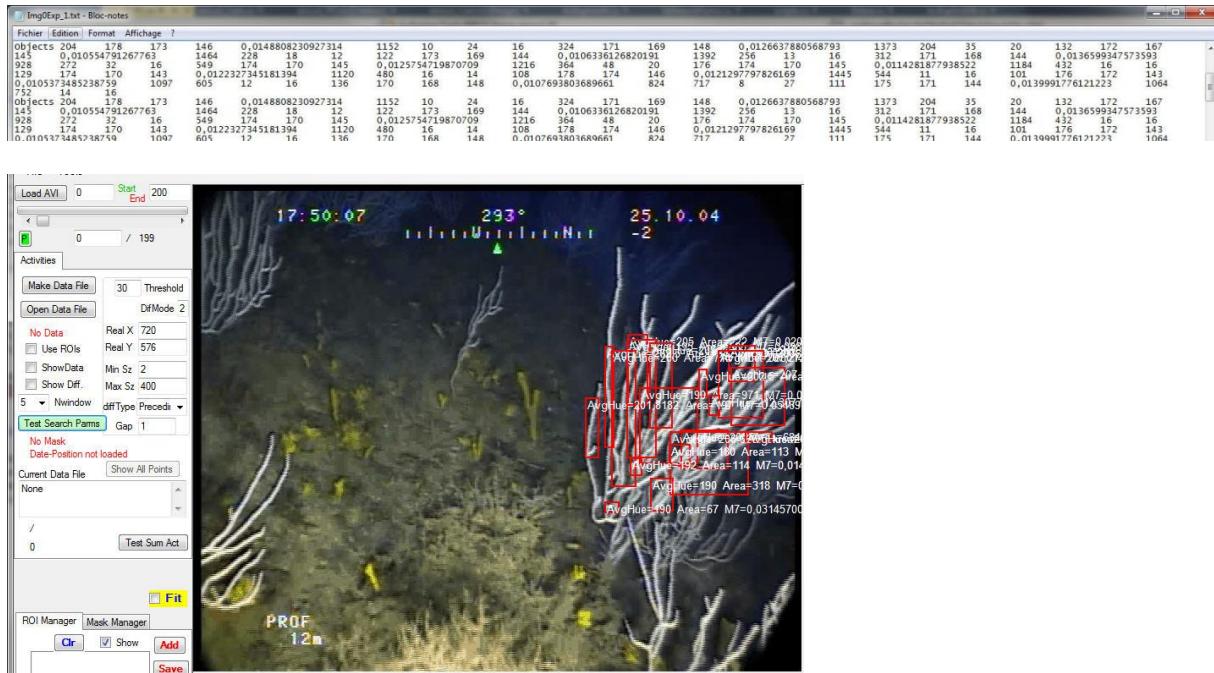
Results are visible on the main window. If export is requested data are sent to a text Log file.



Some parameters are requested to select the exported objects. In this test condition, the main ones are M7 range and Hue range. Hue is a tint value obtained by switch from the usual RGB space to another colorimetric space (HSL : Hue, Saturation and Luminance). The hue value is a tint selected on a range from 0 to 359. The M7 value is the Hu invariant moment #7, characterizing the shape of the object.



When applied on the image, filtering and computations produce a number of objects that can be seen on the main window. If export have been request, a text file is produced showing the object parameters of each object in that image.



6.3.3 – Trajectometry tests

Testing trajectometry gives indications on the redundancy of measures. A tool is presented allowing computing trajectories on portion of films. In its present version, using the tool consist in providing 'Start Img' and 'End Img' of the sequence and using the 'Auto Track' button. Found trajectories are displayed in a list. Clicking on the trajectory in the window displays a table with values for the selected trajectory. It is then possible to copy it and paste it in a spreadsheet program.

7 – Script Functions Reference

Description of all available functions

7.1 - Filters

Threshold

Threshold an image, keeping values between 'Min' and 'Max'. You can select input and output image.

Params (5)

Min Pixel value

Max Pixel value

Source image (Origin,Work,SecW)

Dest image (Origin,Work,Gray,SecW)

Show image (true/false) you can hide the picture for speed reason or to avoid flicker in the viewer

Threshold_fixedLevel

Threshold an image, using a fixed threshold and various algorithms. A maximum output value can be specified. You can select input and output image (only outputs to Origin and Work images will be realized).

Params (6)

Threshold

Max output Pixel value

Threshold type (Binary, Inverted, Truncated, Otsu, To Zero, To Zero Inverted)

Source image (Origin,Work,SecW)

Dest image (Origin,Work,Gray,SecW)

Show image (true/false) you can hide the picture for speed reason or to avoid flicker in the viewer

AdaptiveThreshold1

Perform adaptive threshold on input image

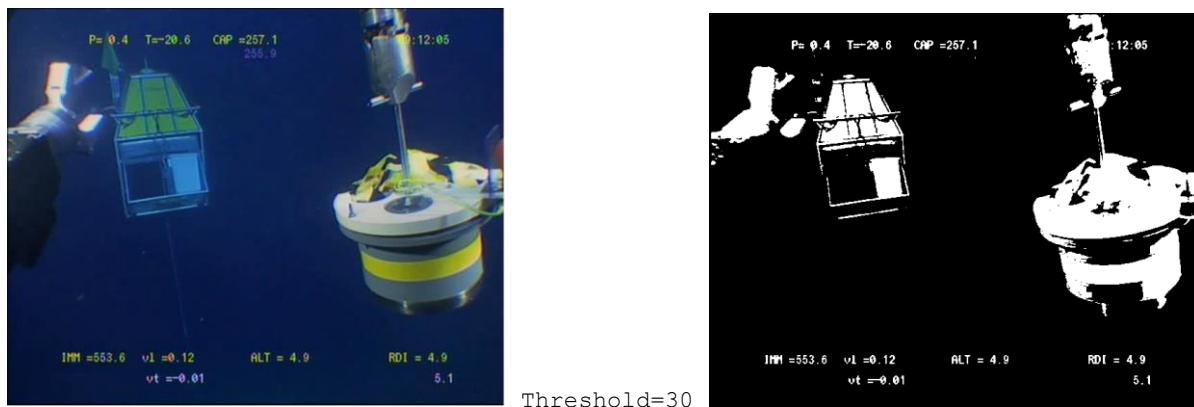
Params (4)

Threshold Threshold value (0-255)

Source image (Origin,Work,SecW)

Dest image (Origin,Work,Gray,SecW)

Show image (true/false) you can hide the picture for speed reason or to avoid flicker in the viewer



ThresholdToMask

Threshold an image, keeping values between 'Min' and 'Max'. Resulting pixels >0 are inserted in the boolean current mask. No image displayed. You can select input image.

Params (3)

Min Pixel value

Max Pixel value

Source image (Origin,Work,SecW)

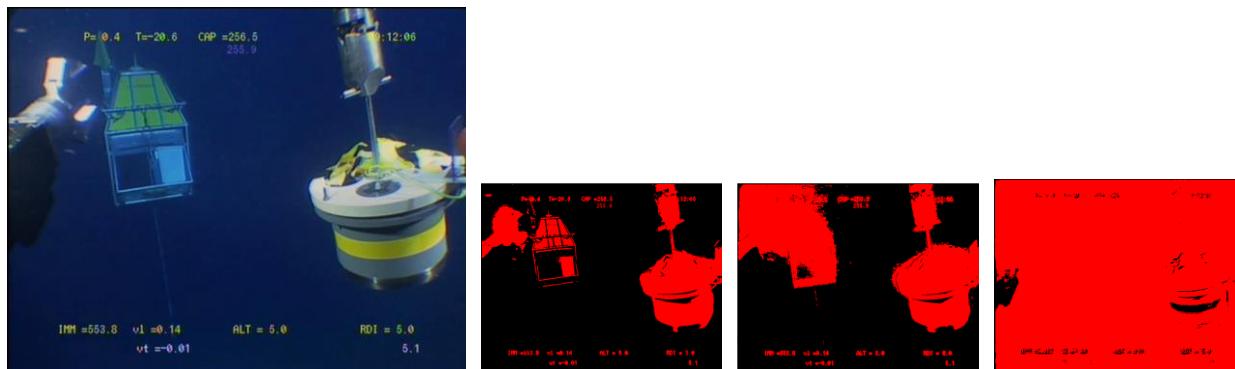
BinarizeRGB

Binarization of a colour image, using low and high threshold values for each channel R, G and B. Channel may be unselected individually. Boolean operator may be used on each color channel individually. A mask can be created with 3 different types (no mask, regionMask, boolean mask). You can select input and output image. Faster rate is obtained by using an origin image as input and a workImg as output.

The sum of every pixel found in every channel is memorized in the system. Call to function [WriteOneRGBCountTextValue](#) allows exporting these values in a Log text file previously opened with the command [openLogFile](#).

Params (15)

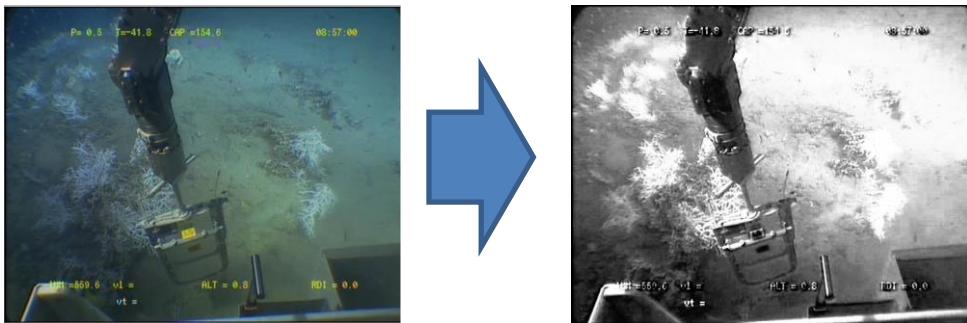
Threshold Low Red	Threshold value Red channel (0-255)
Threshold High Red	Threshold value Red channel (0-255)
Threshold Low Green	Threshold value Red channel (0-255)
Threshold High Green	Threshold value Red channel (0-255)
Threshold Low Blue	Threshold value Red channel (0-255)
Threshold High Blue	Threshold value Red channel (0-255)
Use Channel Red	Yes/not
Use Channel Green	Yes/not
Use Channel Blue	Yes/not
RGB boolen operator	AND/OR
Color	Color of detected pixels (0=red, 1=green, 2=blue, 3=white)
Create Mask	Yes/not a mask can be created during the process
Source image	(Origin,Work,SecW)
Dest image	(Origin,Work,SecW)
Show image	(true/false) you can hide the picture for speed reason or to avoid flicker in the viewer



Binarized image using R, G or B channel (thresholds 10 and 40). Mixtures are possible.

HistogramEqualize

Realize a histogram equalization on the image. No parameter. Use Origin Image, result in Work image.



Image_Equalize

Image equalize computes a correction, on Work Image, for a light coming from a side of the image, such as a projector. It applies a series of concentric gamma correction to the image, using a theoretic outside position, a step for the concentric correction and a ramp factor. Uses OrigImg, result in WorkImg.

Params (9)

<i>Origin distance</i>	distance of light source in pixel (i.e. 6000)
<i>Pos X0</i>	Reference Point 1 X Position (i.e. 300)
<i>Pos Y0</i>	Reference Point 1 Y Position (i.e. 300)
<i>Pos X1</i>	Reference Point 2 X Position (i.e. 0)
<i>Pos Y1</i>	Reference Point 2 Y Position (i.e. 0)
<i>Width of search zone</i>	(i.e. 350)
<i>Height of search zone</i>	(i.e. 350)
<i>Ramp coefficient</i>	for computation (i.e. 0.5)
<i>Memorize</i>	yes/no



Image_Equal_auto

Same as Image equalize with automatic parameters. Computes concentric gamma corrections on the image. Uses OrigImg, result in WorkImg.

Params (5)

<i>Origin distance</i>	distance of light source in pixel (i.e. 6000)
<i>Width of search zone</i>	(i.e. 350)
<i>Height of search zone</i>	(i.e. 350)
<i>Ramp coefficient</i>	for computation (i.e. 0.5)

Memorize yes/no

gradientMatrix1

Compute gradient on the image (8 points, sum of difference between center and points. Uses OrigImg, result in WorkImg.

Params (5)

Plane plane used. 0 -> Blue, 1 -> Green, 2 -> Red, 3 -> (Red+Green+Blue)/3
Distance from centre in pixels
Show pixels true/false

Gabor

Computes Gabor filter on the image, using OrigImg as input, result in WorkImg. Long computation...

Params (5)

Sigma value (i.e. 10.0)
Theta value (i.e. 0.0)
Lambda value (i.e. 100.0)
Psi value (i.e. 1.52)
gGamma value (i.e. 1.0)

Contrast

The contrast measure of the grey-level co-occurrence matrix can give a classification. It is related to the increase in the edges present in the image.

Params (4)

Contrast value between 0.0 and 3.0
Source image (Origin,Work,SecW)
Dest image (Origin,Work,SecW)
Show image (true/false) you can hide the picture for speed reason or to avoid flicker in the viewer

Gamma

Computes a new gamma value for the image (0.0-3.0)

Params (4)

Contrast value between 0.0 and 3.0
Source image (Origin,Work,SecW)
Dest image (Origin,Work,SecW)
Show image (true/false) you can hide the picture for speed reason or to avoid flicker in the viewer

changeGamma

Computes a new gamma value for the image (0.0-3.0), based on a reference brightness of another image, value stored in the global variables by the instruction **getImageMeanLuminance**. Result can be applied to selected image.

Params (1)

Dest image (Origin,Work,SecW)

getImageMeanLuminance

Computes the luminance or Origin Image and store the value in Global memory for later use (**changeGamma** for example)

Params (0)

ShowImageMeanLuminance

Computes the luminance or Origin Image and store the value in Global memory for later use (**changeGamma** for example). Displays the value in the Status line (Zoom box) and store it in the clipboard.

Params (0)

usePresetLuminance

Set the provided luminance value in the Global memory for later use (i.e. **changeGamma**)

Params (1)

Luminance value (0.0-255.0)

Correlation_3x3

3x3 correlation using the input matrix provided in the first 9 parameters.

Params (12)

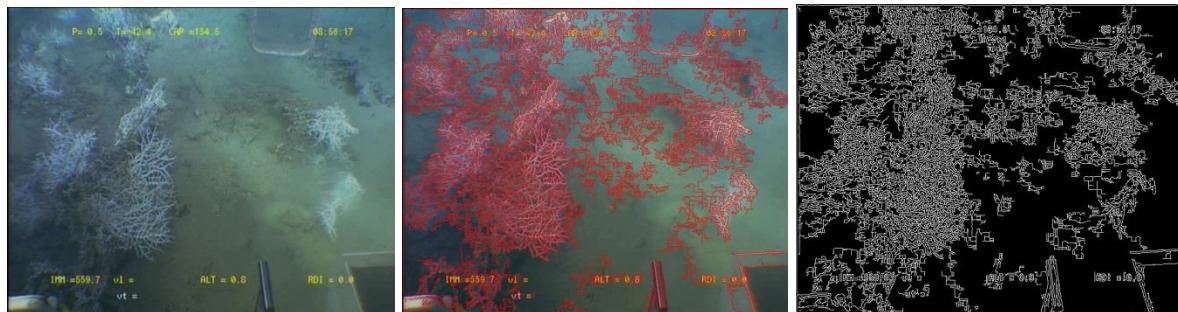
Matrix1 value (def=0)
Matrix2 value (def=-1)
Matrix3 value (def=0)
Matrix4 value (def=-15)
Matrix5 value (def=-15)
Matrix6 value (def=-1)
Matrix7 value (def=0)
Matrix8 value (def=-1)
Matrix9 value (def=0)
Source image (Origin,Work,SecW)
Dest image (Origin,Work,SecW)
Show image (true/false) you can hide the picture for speed reason or to avoid flicker in the viewer

Canny_Edge_Detector

Use Canny detector to find edges in the image. You can create a mask for future use, show the result window and may measure on it, or see the result as an overlay.

Params (6)

Low Threshold	(0-255)
High Threshold	(0-255)
Create Mask	(true/false) use the buffer to create a new mask
Source image	(Origin,Work,SecW)
Dest image	(Origin,Work,Gray,SecW)
Show image	(true/false) this let you display OrigImg with Canny result as red overlay



Difference_Edge_Detector

Use Difference Edge detector to find edges in the image.

Params (4)

usage	use in a masked image (=0, a mask must exist) or in image (=1)
Source image	(Origin,Work,SecW)
Dest image	(Origin,Work,SecW)
Show image	(true/false) this let you display the image

Laplace_Edge_Detector

Use Laplace Edge detector to find edges in the image.

Params (3)

Source image	(Origin,Work,SecW)
Dest image	(Origin,Work,SecW)
Show image	(true/false) this let you display the image

Sobel

Use Sobel Edge detector to find edges in the image. Two algorithms are used. A 0 value switches to the first, a non-zero value (1-255) switches to the second.

Params (4)

Threshold value	if 0 or null use first algorithm otherwise use threshold (1-255)
Source image	(Origin,Work,SecW)
Dest image	(Origin,Work,SecW)
Show image	(true/false) this let you display the image

Sobel->MaskBuf

Use Sobel Edge detector to find edges in the image. Result is placed in a Mask buffer filled with 0 or 255

Params (3)

Source Color	0 = (R+G+B)/3, 1=Red channel, 2=Green channel, 3=Blue channel)
Threshold value	(0-255)
Source image	(Origin,Work,SecW)

Segmentation_Merge

Segment the image using a merge algorithm

Params (8)

Type	0 = Merge, 1= Split/merge
Color	1 = Gray, 2 = Grey BT709, 3 = Tint
Threshold value	(0-255)
Regroup	true/false
Group value	integer
Source Image	(Origin,Work,SecW)
Dest Image	(Origin,Work,SecW)
Show image	(true/false) this let you display the image

Segmentation_autoMerge

Segment the image using an automerger algorithm

Params (8)

Mask Threshold	(0-255)
FillStructElemSize	(2-20)
Plane	0 = (R+G+B)/3, 1 = Red, 2 = Green, 3 = Blue
Seed	double
Std Dev on Seed	double
Source Image	(Origin,Work,SecW)
Dest Image	(Origin,Work,SecW)
Show image	(true/false) this let you display the image

Color_HSL_Filter_euclidean

Applies a filter in the HSL color space using Euclidean distances. Filtering is achieved using only the Hue value with a given range. A mask can be created.

Params (6)

<i>Hue value</i>	(0-360)
<i>Delta hue</i>	range around Hue value
<i>Mask</i>	0 = no mask, 1 = create mask
<i>Source Image</i>	(Origin,Work,SecW)
<i>Dest Image</i>	(Origin,Work,SecW)
<i>Show image</i>	(true/false) this let you display the image



Scan_Color_HSL_Filter_euclidean

Applies a filter in the HSL color space using Euclidean distances. Filtering is achieved using only the Hue value with a given range. A mask can be created. This function is used in a scanning context, within a `repeat/endrepeat` loop. This means that a result is drawn in the active window below the viewer, showing the results

Params (6)

<i>Hue value</i>	(0-360)
<i>Delta hue</i>	range around Hue value
<i>Mask</i>	0 = no mask, 1 = create mask
<i>Source Image</i>	(Origin,Work,SecW)
<i>Dest Image</i>	(Origin,Work,SecW)
<i>Show image</i>	(true/false) this let you display the image

Scan_HSL_Image

Applies a filter in the HSL color space. Filtering is achieved using Hue value range, a Saturation range and a Brightness range. Filter can be used inside ROIs. This function is used in a scanning context, within a `repeat/endRepeat` loop. This means that a result is drawn in the active window below the viewer, showing the results. If '`endRepeatWithRequest`' is used instead of '`endRepeat`' then program stops until a click is made in the Active window. This lets explore the curve drawn in the window and load at the corresponding images in the film.

Params (6)

<i>useROI</i>	true/false
<i>Hue min</i>	(0-359) hue
<i>Hue max</i>	(0-359)
<i>Sat min</i>	(0.0-1.0) saturation
<i>Sat max</i>	(0.0-1.0)
<i>Lum min</i>	(0.0-1.0) luminance, brightness

Lum max (0.0-1.0)

HueCountSurface

Count the pixels in an image between a low and high value. Value is memorized for possible export.

Params (2)

Hue min (0-359) hue
Hue max (0-359)

RGBCountSurface

Count the pixels in an image with a RGB range on selected channel and selected range. Value is memorized for possible export.

Params (6)

RChannel	true/false	Red channel used or not
GChannel	true/false	Green channel used or not
BChannel	true/false	Blue channel used or not
RThreshold	(0-255)	
GThreshold	(0-255)	
BThreshold	(0-255)	

Count_One_HSL_Image

Works only inside a **repeat/endRepeat** loop.

Analyze image replacing a target color in a given range and counting modified pixels. If low Hue is higher than high Hue the work is done on two areas (low to 359 and 0 to high).

Params (7)

useROI	true/false
Hue min	(0-359) hue
Hue max	(0-359)
Sat min	(0.0-1.0) saturation
Sat max	(0.0-1.0)
Lum min	(0.0-1.0) luminance, brightness
Lum max	(0.0-1.0)

saveScannedData

save the data obtained on the Active window after a scan (inside a repeat/endRepeat loop)

Params (0)

Scan_Moments_Image

Work only inside a **repeat/endRepeat** loop.

Compute moments for the current image

Params (0)

getImageMoment

Compute the Hu invariant moment for the current image. Double value is in the Global memory for export to Log file.

Params (0)

keepScanUntilClick

Allows ending a repeat/endRepeat loop before its regular ending by clicking on the active window.

Params (0)

Color_Filter_euclidean

Applies a color filter using Euclidean distances. Filtering is achieved using only the Hue value with a given range. A mask can be created. This function is used in a scanning context, wthin a **repeat/endrepeat** loop. This means that a result is drawn in the active window below the viewer, showing the results

Params (8)

Radius value	(0-400)
Centre Red	integer
Centre Green	integer
Centre Blue	integer
Mask	0 = no mask, 1 = create mask
Source Image	(Origin,Work,SecW)
Dest Image	(Origin,Work,SecW)
Show image	(true/false) this let you display the image

getRawHSLMask

Extract a mask in the binary Buffer. This mask contains objects isolated with Edge detector and HSL range values. The mask surface is memorized in the global integer.

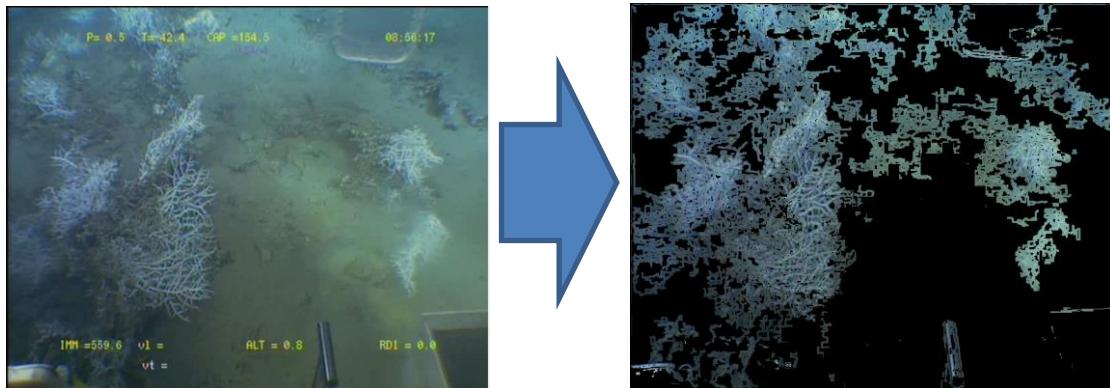
Params (8)

Canny Low	(0-255)
Canny High	(0-255)

<i>useROI</i>	true/false
<i>Hue min</i>	(0-359) hue
<i>Hue max</i>	(0-359)
<i>Sat min</i>	(0.0-1.0) saturation
<i>Sat max</i>	(0.0-1.0)
<i>Lum min</i>	(0.0-1.0) luminance, brightness
<i>Lum max</i>	(0.0-1.0)

Usage:

```
getRawHSLMask 10,60,false,150,220,0.1,0.9,0.1,0.9
```



ShowOriginImage

Shows the origin Image on the viewer. No parameter.

ShowWorkImage

Shows the Work Image on the viewer. This image is usually the result of operations on the origin image. No parameter.

ShowGrayOnOriginImage

Shows the Gray image (one channel) as an overlay over the Origin Image. Pixel values greater than 0 appear as red pixels. No parameter.

CopyOriginToWorkImage

Copy the Origin Image to the Work Image (24 bit color images). No parm.

CopyWorkToOriginImage

Copy the Work Image to the Origin Image (24 bit color images). No parm.

CopyWorkToSecWImage

Copy the Work Image to the Secondary Work Image (24 bit color images). No parm.

CopyOriginToSecWImage

Copy the Origin Image to the Secondary Work Image (24 bit color images). No parm.

CopySecWToOriginImage

Copy the Secondary Work Image to the Origin Image (24 bit color images). No parm.

CopyOriginToBinBuf

Copy the Origin Image to a hexadecimal temporary buffer. No parm.

CopyWorkToBinBuf

Copy the Work Image to a hexadecimal temporary buffer. No parm.

CopyBinBufToWork

Copy the one channel hexadecimal temporary buffer to a Work Image. R,G, and B values have the same BinBuf channel value. No parm.

7.2 - Morphology

KeyFeatures

Find key features in the image, using a FAST algorithm. Origin Image is used, result is in work Image. Feature can be displayed as circles whose diameter is related to the parameter. Sum of surfaces of these keyfeatures is available as a **memorized integer** for export to Log file. If images flux is too fast, statck overflow is possible and images are skipped.

Params (2)

Hessian	Hessian parameter (i.e. 500)
Show Features	Show features (true/false)

Contours

Search contours in the image. Origin Image is used, result is in work Image.

Params (2)

<i>Edge Low value</i>	Def=50
<i>Edge high value</i>	Def=200
<i>Max Level</i>	Def=0
<i>Line thickness</i>	Def=3

ORImage

Realize a boolean OR operation between two images.

Params (4)

<i>Source Image 1</i>	(Origin,Work,SecW,Gray)
<i>Source Image 2</i>	(Origin,Work,SecW,Gray)
<i>Dst Image</i>	(Origin,Work,SecW,Gray)
<i>Show Image</i>	(true/false) you can hide the picture for speed reason

ANDImage

Realize a boolean AND operation between two images.

Params (4)

<i>Source Image 1</i>	(Origin,Work,SecW,Gray)
<i>Source Image 2</i>	(Origin,Work,SecW,Gray)
<i>Dst Image</i>	(Origin,Work,SecW,Gray)
<i>Show Image</i>	(true/false) you can hide the picture for speed reason

XORImage

Realize a boolean XOR operation between two images.

Params (4)

<i>Source Image 1</i>	(Origin,Work,SecW,Gray)
<i>Source Image 2</i>	(Origin,Work,SecW,Gray)
<i>Dst Image</i>	(Origin,Work,SecW,Gray)
<i>Show Image</i>	(true/false) you can hide the picture for speed reason

NOTImage

Realize a boolean NOT operation from source image to destination image.

Params (4)

<i>Source Image 1</i>	(Origin,Work,SecW,Gray)
-----------------------	-------------------------

Source Image 2 (Origin,Work,SecW,Gray)
Dst Image (Origin,Work,SecW,Gray)
Show Image (true/false) you can hide the picture for speed reason

Dilate

Realize a dilatation morphologic operation from source image to destination image. Enter the type of structuring element (rect or cross) and the number of iterations.

Params (6)

Structuring type (rectangle,cross)
Number of iterations number of repeat of the dilatation operation (i.e. 1)
Source Image 1 (Origin,Work,SecW,Gray)
Source Image 2 (Origin,Work,SecW,Gray)
Dst Image (Origin,Work,SecW,Gray)
Show Image (true/false) you can hide the picture for speed reason

Erode

Realize an erosion morphologic operation from source image to destination image. Enter the type of structuring element (rect or cross) and the number of iterations.

Params (6)

Structuring type (rectangle,cross)
Number of iterations number of repeat of the erosion operation (i.e. 1)
Source Image 1 (Origin,Work,SecW,Gray)
Source Image 2 (Origin,Work,SecW,Gray)
Dst Image (Origin,Work,SecW,Gray)
Show Image (true/false) you can hide the picture for speed reason

Opening

Realize a Opening morphologic operation from source image to destination image. Enter the type of structuring element (rect or cross) and the number of iterations.

Params (6)

Structuring type (rectangle,cross)
Number of iterations number of repeat of the opening operation (i.e. 1)
Source Image 1 (Origin,Work,SecW,Gray)
Source Image 2 (Origin,Work,SecW,Gray)
Dst Image (Origin,Work,SecW,Gray)
Show Image (true/false) you can hide the picture for speed reason

Closing

Realize a Closing morphologic operation from source image to destination image. Enter the type of structuring element (rect or cross) and the number of iterations.

Params (6)

<i>Structuring type</i>	(rectangle,cross)
<i>Number of iterations</i>	number of repeat of the closing operation (i.e. 1)
<i>Source Image 1</i>	(Origin,Work,SecW,Gray)
<i>Source Image 2</i>	(Origin,Work,SecW,Gray)
<i>Dst Image</i>	(Origin,Work,SecW,Gray)
<i>Show Image</i>	(true/false) you can hide the picture for speed reason

Gradient

Realize a Gradient morphologic operation from source image to destination image. Enter the type of structuring element (rect or cross) and the number of iterations.

Params (6)

<i>Structuring type</i>	(rectangle,cross)
<i>Number of iterations</i>	number of repeat of the gradient operation (i.e. 1)
<i>Source Image 1</i>	(Origin,Work,SecW,Gray)
<i>Source Image 2</i>	(Origin,Work,SecW,Gray)
<i>Dst Image</i>	(Origin,Work,SecW,Gray)
<i>Show Image</i>	(true/false) you can hide the picture for speed reason

TopHat

Realize a TopHat morphologic operation from source image to destination image. Enter the type of structuring element (rect or cross) and the number of iterations.

Params (6)

<i>Structuring type</i>	(rectangle,cross)
<i>Number of iterations</i>	number of repeat of the TopHat operation (i.e. 1)
<i>Source Image 1</i>	(Origin,Work,SecW,Gray)
<i>Source Image 2</i>	(Origin,Work,SecW,Gray)
<i>Dst Image</i>	(Origin,Work,SecW,Gray)
<i>Show Image</i>	(true/false) you can hide the picture for speed reason

BlackHat

Realize a BlackHat morphologic operation from source image to destination image. Enter the type of structuring element (rect or cross) and the number of iterations.

Params (6)

<i>Structuring type</i>	(rectangle,cross)
<i>Number of iterations</i>	number of repeat of the BlackHat operation (i.e. 1)
<i>Source Image 1</i>	(Origin,Work,SecW,Gray)
<i>Source Image 2</i>	(Origin,Work,SecW,Gray)
<i>Dst Image</i>	(Origin,Work,SecW,Gray)
<i>Show Image</i>	(true/false) you can hide the picture for speed reason

setRealXY

Set the real window X and Y values for surface computations

Params (2)

<i>Xmm</i>	Real horizontal size in millimetres
<i>Ymm</i>	Real vertical size in millimetres

setTimeIntervalSec

Set the time interval between images in seconds

Params (1)

<i>Interval</i>	time interval in seconds
-----------------	--------------------------

7.3 - AVI related functions

setAVIFileName

Used to enter an AVI file name manually. This name will be used by the function openAVIFile with file param set to true (existing file name).

Params (1)

<i>FileName</i>	Name of the file to be opened (i.e. <i>testAVI.avi</i>)
-----------------	--

openAVIFile

Open an AVI film (preferably encoded with an MPEG 4 V2 codec, for compatibility, but support other formats). If the parameter is set to 'false' an open file dialog allow to select a film from a disk. Otherwise an existing name must have been provided.

Params (1)

useFileName

false: open a dialog box to request a file name

filename: a regular file name (i.e. C:\film.avi)

ex.: openAVIFile false
 openAVIFile C:\film.avi

showAVIImage

Extracts an image from the currently opened film, stores it in OrigImg, optional copy to workImg, optional show on the viewer. Cannot be used in real time mode as pictures are collected on the input video device.

Params (3)

ImageNb: the number of image to be loaded. A test is made to control if the requested number is in the range of existing images of the film.
copy: if 1, copy of the extracted image to the working Image (workImg).
show: if 'true' the image is shown on the viewer. If 'false' the image is stored in memory for computations but not shown on the viewer.

reloadCurAVIImage

Reload the current image from the currently opened film, for a new set of processing. Image is stored in Origin Image buffer.

showNextAVIImage

Move the index of image to the next film image. Must be the last command before 'endRepeat' inside a 'repeat/endRepeat' or a 'repeat/endRepeatWithRequest' loop. Cannot be used in real-time mode as images are collected on the input video flux by the viewer or by the timer.

Params no parameter

saveJpeg

Save the displayed image to a Jpeg file, named with a root and the current image number. Request a number for compression quality (i.e. 100).

Params (2)

RootName: Root name used to build the Jpeg name, with the image number
Quality: compression quality (20-100)

saveJpegWithOverlay

Save the displayed image to a Jpeg file, named with a root and the current image number. Make a screen copy to integrate the overlays in the Jpeg. Request a number for compression quality (i.e. 100).

Params (2)

RootName: Root name used to build the Jpeg name, with the image number
Quality: compression quality (20-100)

SetJpegRootName

Provide a root name for the next Jpeg images created using this root and the image number. **This instruction resets the incremental Jpeg Number to 0.**

Params (1)

RootName: Root name used to build the Jpeg name, with the image number

savImageToRootJpeg

Save a Jpeg image, with a name build with a global root name provided by the instruction **SetJpegRootName**, and an incremental image number. This allows exporting sparse images numbered in a sequence. The instruction resets the incremental Jpeg number to 0. This allows generating a new sequence of images starting from 0. It is then possible to re-assemble these pictures into an AVI film using ScriptEdit menu '**Film Editing**' and selecting the '**MakeAvi**' box.

Can be used in a loop.

Params (2)

Source Image:	select the image to save (Origin,Work,Secondary Work)
Quality:	compression quality (20-100)

closeAVI

Close the current Avi. The last image remain as a snapshot on the viewer. Commands on the main AviExplore window are inactive (scroll, etc.).

Params no parameter

differenceAVIFrame

Realize a difference between current image and the preceding image. Absolute value of difference is shown. Difference uses current image in Origin Image and previous image in Secondary working image buffer. Difference is copied in Work Image for display.

Params (7)

isGray	work on color or gray image.
useRegionMask	compute only inside regions
Dst Image	(Origin,Work,SecW,binBuf)
lag	choose the lag type (Preceeding/Gap/ReflImage)
ReflImage	provides the gap for images difference (if lag=Gap -> gap, if lag=ReflImage ->number of reference image to use).
Show Image	(true/false) you can hide the picture for speed reason
DiffType	difference type (1=use absolute value/2=use positive value/3=use neg.Value)

differenceThresholdedAVIFrame

Realize a difference between current image and image separated by a gap. If lag='Preceeding,' use the preceding image, if lag='Gap', use ReflImage as a gap between the images (i.e. difference =currentImage-gap), if lag=Ref, use ReflImage as an absolute position in the film (i.e. ReflImage=0 will compare every image to image 0 in the film). No test is made about validity of ReflImage selected (must be in the range (0-total nb of film images). Eliminate all values under Thresold value. Absolute value of difference is shown. Difference uses current image in Origin Image and previous image in Secondary working image buffer. Difference is copied in Dest Image for display.

Params (8)

isGray	work on color or gray image.
---------------	------------------------------

useRegionMask	compute only inside regions
Threshold	(0-255) threshold the result of difference
Dst Image	(Origin,Work,SecW,binBuf)
lag	choose the lag type (Preceeding/Gap/ReflImage)
ReflImage	provides the gap for images difference (if lag=Gap -> gap, if lag=ReflImage ->number of reference image to use).
Show Image	(true/false) you can hide the picture for speed reason
diffType	difference type (1=use absolute value/2=use positive value/3=use neg. Value)

differenceInROI

Simple difference between two images with a ROI mask.

Params (7)

Existing ROI number	The number of an existing ROI
Input source	can be the image number or the text 'cur' for the currently used image
Threshold	0=auto otherwise (1-255)
Channel	0=gray, 1=Red, 2=Green, 3=Blue
Type of difference	1=abs, 2=positive, 3=negative
Show	show the difference true/false
Scroll	for use in a scrollable environment

differenceInRectROI

Simple difference between two images with a rectangular ROI mask.

Params (7)

Input source	can be the image number or the text 'cur' for the currently used image
Threshold	0=auto otherwise (1-255)
left	rectangle left coordinate
top	rectangle top coordinate
right	rectangle right coordinate
bottom	rectangle bottom coordinate
Show	show the difference true/false
Channel	0=gray, 1=Red, 2=Green, 3=Blue
Type of difference	1=abs, 2=positive, 3=negative
Show Dif	true/false show de differences as coloured pixels
Scroll	for use in a scrollable environment

7.4 - LOG and Companion files related functions

These function concern export of information computed on images

openLogFile

Open a Log file, which is a text file that can be read with text editors. This file will collect all the outputs from various function, in a textual form. Within a loop these function can compute parameters and store it inside the log file. If the parameter is set to 'false' an open file dialog allow to select a film from a disk.

Params (1)

useFileName

false: open a dialog box to request a file name
filename: a regular file name (i.e. C:\log.txt)

ex.: openLogFile false

openLogFile C:\log.txt

closeLogFile

This closes a currently active Log text file. A flush is made to store the remaining buffers in the output file.

Params no parameter

7.5 – Data File management

Warning: It is easier to use the ‘Make Data File’ button in the AviExplore-Fixed Avi module to create a datafile

CreateDataFile

Params (1)

Nb images in Film Data the size of source film ('cur' -> current size, or a number of frames

OpenDataFile

Opens a new Data File associated with a currently opened AVI. Film MUST be opened.

Params (0) No param.

SaveDataRecord

Params (2)

Min blob size	the minimum surface of blobs extracted
Max blob size	the maximum surface of blobs extracted
useMask	use an existing boolean mask (true/false)
useROIs	use existing ROIs (true/false)

CloseDataFile

Close an open JCD recording data file (on exit from a loop)

Params (1)
useROIs use existing ROIs (true/false)

CloseDataStream

Close an open JCD recording data stream (on exit from a loop)

Params (0)

ExportIntensities

Export the activity intensities detected in a data File

Params (0)

ExportIntensitiesTimeLabel

Export the activity intensities detected in a data File between two image number and for one selected label

Params (3)

<i>StartImage</i>	Start of sequence
<i>EndImage</i>	End of sequence
<i>LabelNb</i>	Number of an existing label

setDefaultPath

Use the path of the opened AVI file, or the path of the opened Data File to set the Default Path (avi= use currently opened avi directory, jcd= use currently opened jcd file, none= let you type the path from keyboard).

Params (3)

<i>Path text</i>	use avi or data directory (avi/jcd/none none to type in the path)
------------------	---

7.6 – Loops

repeat

Opens a simple loop to treat a sequence of images. In conjunction with showAVIIImage it is possible to position the start of the sequence, then repeat the treatment on the next 'Nb of repeat' images, loaded with a 'ShowNextAVIIImage' located within the loop.

Params (1)

Nb of repeat: the number of images in the current loop. A control is made to stop the loop if it reaches the end of the film.

Usage

showAVIIImage 100,1 repeat 100 ... do something on the first image showNextAVIIImage 1 ... do something on the first image endRepeat	'show frame 100 in the film, shows it on the viewer 'repeat 100 times 'load next image, shows it on the viewer 'end of the loop
---	--

endRepeat

Closes an open loop.

Params	no parameter
---------------	--------------

endRepeatWithRequest

Closes an open loop but wait for a click in the Active Window to proceed to the next instruction. This can be used in the '**Scan...**' instructions where a graphic is displayed in the active window. Moving the cursor over the curve et you see the images concerned by the displayed values.

Params no parameter

7.7 – Masks

ApplyMask

Apply a mak on an output selected image or buffer, using selected input.

Input masks can be:

- Boolean Mask with True/false values
- Binary hexadecimal mask (with values between 0 and 255)
- Gray0, using the Gray image as a mask erasing all the 0 pixels in the 8bit image
- GrayThresh, using the Gray image as a mask, erasing all the pixels below the Threshold value.

Destination images can be Origin, Work, Gray, Secundary Work images or the common binBuf structure (hexadecimal).

Params (3)

MaskSource the source mask (Mask/binMask/Gray0/GrayThresh)
Dest destination image or buffer (Origin/Work/Gray/SecW/binMask)
Threshold the Threshold value used only by the GrayThresh option

CopyMaskToBinaryMask

Copy the boolean buffer (true/false) to the hexadecimal mask buffer (0-255).

Params (0)

CopyBinaryMaskToMask

Copy the hexadecimal mask buffer (0-255) to the boolean buffer (true/false)

Params (0)

CreateBinaryMask

Create an empty binary binary mask.

Params (0)

AddToBinaryMask

Scan binary hexadecimal buffer and increment every binMask position where binBuf>0. If binBuf > 0 the position is set to 255, otherwise it is set to 0.

Params (0)

FilterBinaryBuffer

Maintain in the binary buffer all the blobs with surface comprised between Low and High limits, erase the others.

Params (3)

<i>MaskType</i>	the source mask type (0=default/1=regionMask/2=OutputMask/3=Additive mask)
<i>Low Limit</i>	low blob limit
<i>High Limit</i>	high blob limit

DilateMask

Dilate the selected binary mask

Params (1)

<i>MaskType</i>	the source mask type (binBuf/binMask)
-----------------	---------------------------------------

ErodeMask

Erode the selected binary mask

Params (1)

<i>MaskType</i>	the source mask type (binBuf/binMask)
-----------------	---------------------------------------

showMask

Draw the binary Mask (binMask or binBuf) on the top of a selected image. Number of concerned pixels is saved in integer global memory for possible export.

Params (3)

<i>dstImage</i>	select the image on which the mask pixels will be added
<i>SourceMask</i>	select binary Mask or binary Buffer (binMask/ binBuf)
<i>Color</i>	color of the pixel belonging to mask visible in the regions. Display color (0=white/1=red/2=green/3=blue)

fillBinBufHoles

Use the binary buffer created by many functions and fill the holes inside blobs using an opening process and iterations.

Params (1)

Nb iterations provide a number of iterations for the opening procedure.

showMaskInRegions

Work Image is used. Shows the part of the binary Mask (binMask or binBuf) present inside the regions obtained from ROIs. Number of concerned pixels is saved in integer global memory for possible export.

Params (2)

SourceMask select binary Mask or binary Buffer (binMask/ binBuf)

Color color of the pixel belonging to mask visible in the regions. Display color
(0=white/1=red/2=green/3=blue)

CountPixelsInMask

When a binary Mask exists, count all the pixel in the binMask. Remember the count in global integer.

Params (1)

MaskType Select the mask (boolean Mask/ binary mask).

7.8 – AdditiveMasks

CreateAdditiveBuffer

Create an integer buffer containing the number of hits concerning every single pixel across a film. The maximum number of hit is the total number of frames in the film. This indicate that something happened in a pixel at location x,y for every frame of the film. The fiman value range from 0 (pixel never impacted) to maximum frame number (pixel impacted in every frame of the film). Hits are detected using boolean operation on buffers or images.

This instruction is sed once in the header of the script, or outside loops. It resets all the values to 0.

Params (0)

AddToAdditiveBuffer

Scan binary hexadecimal buffer and increment every AdditiveBuffer position where binBuf>0 (using binaryBuffer as a mask where every non zero pixel incerement the corresponding buffer).

Params (0)

AddToAdditiveBufferWithThreshold

Scan binary hexadecimal buffer and increment every AdditiveBuffer position where binBuf>=Threshold value (using binaryBuffer as a mask where every non zero pixel incerement the corresponding buffer).

Params (2)

<i>Low Threshold</i>	the low threshold value (0-255)
<i>High Threshold</i>	the high threshold value(0-255)

CountPixelsInAdditiveBuffer

Get the total number of event counted in the buffer. Stores it in a global Integer (can be saved to Log with **writeOneIntTextValue**).

Params (0)

getHigherValueInAdditiveBuffer

Scans the buffer to find the higher location. Stores the value in a global Integer (can be saved to Log with **writeOneIntTextValue**). Stores position and value as a string (format: X,Y,value) that can be saved to Log with **writeOneTextString**.

Params (0)

ExportAdditiveBuffer

Export the matrix in TAB separated textual form to a text file. The file name is requested and this instruction cannot be placed in a loop or in the real time window.

Params (0)

ExportAdditiveBufferArea

Export the matrix in TAB separated textual form to a text file. Export concerns values located inside a provided restangle. The file name is requested and this instruction cannot be placed in a loop or in the real time window.

Params (5)

<i>XCorner</i>	left position
<i>YCorner</i>	top position
<i>Rectangle Width</i>	Width in pixels
<i>Rectangle Height</i>	Height in pixels
<i>Invert</i>	Invert the Y axis (true/false)

ShowAdditiveBuffer

Display the Additive Buffer with reduced color coding to show the high activity areas.

5 colors are kept:

- 0-40: gray
- 40-60: green
- 60-80: blue
- 80-100: red

>100: white

Origin image is used and the masked image is displayed in Work Image.

Params (0)

7.9 - Blobs related functions

differenceToBlobs

Realize a difference on image, separate existing blobs, compute information and eventually export it to text log file.

Params (1)

Export to LOG file A LOG file MUST be opened with **openLogFile** function

Usage: A gray image must be created. This can be done by using difference functions giving result in Work Image, then copying the Work Image in the Gray Image, then using this function.
Script example:

```
openLogFile false
openAVIFile false
OpenDataFile
showAVIImage 100,1,1
differenceInRectROI cur,15,100,100,500,500,true,1,1,true,false
CopyWorkToGray
differenceToBlobs yes
CloseDataFile
closeLogFile
closeAVI
```

clearThresholdedBlobs

Following a image difference that creates a Binary hexadecimal buffer with values between 0-255, this function erases blobs with gray value situated inside a range. **In real time only the current image can be used.**

Params (3)

Image	select the image to process (prec= preceeding image/ cur=current image/ next=next image)
Low Threshold	pixel threshold (0-255)
High Threshold	pixel threshold (0-255)

CountBlobs

Create a matrix, from Origin Image, using the colorimetric space HSL. Then count the blobs in that matrix. Output a histogram in internal structure. Histogram may be exported to log file if opened by using the exportToLog switch. Memorize in a global integer the number of blobs found.

Params (8)

<i>Hue Low</i>	Hue low value (0-359)
<i>Hue High</i>	Hue high value (0-359)
<i>Sat Low</i>	Saturation low value (0.0-1.0)
<i>Sat High</i>	Saturation high value (0.0-1.0)
<i>Bright Low</i>	Brightness low value (0.0-1.0)
<i>Bright High</i>	Brightness high value (0.0-1.0)
<i>minSurf</i>	minimum blob surface (i.e.1000)
<i>maxSurf</i>	maximum blob surface (i.e.30000)
<i>exportToLog</i>	if true and if Log is opened send the histogram to Log

Usage: **Script example**

CountBlobsInROI

Select a ROI then search and count the blobs inside. Blobs are stored in a string with the

format : <Number of blobs>
TAB <X barycenter> TAB <Y barycenter> (**n times**)
TAB <X barycenter> TAB <Y barycenter>...

Params (10)

<i>ROI number</i>	# of ROI
<i>Thresh Red</i>	Red Threshold (0-255)
<i>Thresh Green</i>	Green Threshold (0-255)
<i>Thresh Blue</i>	Blue Threshold (0-255)
<i>Check Red</i>	Switch Red Channel (0=no/1=yes)
<i>Check Green</i>	Switch Green Channel (0=no/1=yes)
<i>Check Blue</i>	Switch Blue Channel (0=no/1=yes)
<i>Blob Min Size</i>	Minimum blob size in pixels
<i>Blob Max Size</i>	Maximum blob size in pixels
<i>ShowInWorkImage</i>	allows drawing of blobs as circles in Work Image (0=no/1=yes)

CountROIBlobsInDifferenceBuffer

Count blobs in one ROIs from image difference buffer (result in stringRemember to export to Log file via **WriteOneTextString**). Can be used in real time mode to save the objects detected to Log.

Params (3)

<i>ROI number</i>	# of ROI
<i>Blob Min Size</i>	Minimum blob size in pixels
<i>Blob Max Size</i>	Maximum blob size in pixels

getBlobsObjectsMetrics

Search blobs in image difference buffer (in hexadecimal Binary buffer BinBuff, created by other routines such as difference routines). Use provided parameters to isolate relevant objects. Objects can be exported as text in the Log file.

Params (3)

ROI number # of ROI
Blob Min Size Minimum blob size in pixels
Blob Max Size Maximum blob size in pixels

7.10 - ROIs related functions

loadROIFile

Open a ROI file, a file containing descriptions of one or many Regions of Interest. These ROIs are polygons drawn in the ROI manager editor on the AviExplore main window and saved with the 'Save' button. It is an equivalent to the 'load' button in this ROI manager. It opens a open dialog to request the name of a ROI file. When the script is finished, the instruction clearROIs must be sent to clean the memory.

Params no parameter

clearROIs

When a set of ROIs has been loaded with 'loadROIFile' this function discards and cleans the internal buffers concerning the ROIs. This is the equivalent of the function 'Clr' in the ROI Manager window.

Params no parameter

createOneROIRegionMask

When a ROI file is opened and ROI list is present in the active configuration, this function allows creating an internal ROIMask, using one ROI in the ROI list. This mask can be used later to restrict computations on image to the requested area. Sum of all pixels belonging to the mask is saved in global integer memory.

Params (1)

ROInum: the number of the ROI used to create a mask. This number may be edited within the script window to change the currently used ROI and export subsequent results. The first line in the line has a number=1.

createROIsRegionMask

When a ROI file is opened and ROI list is present in the active configuration, this function allows creating an internal ROIMask, using all the available ROIs in the ROI list. This mask can be used later to restrict computations on image to the requested area. Sum of all pixels belonging to the mask is saved in global integer memory.

Params (0)

CountPixelsInRegionMask

When a region mask is created, Count all the pixels in this region mask.

Params (0)

resetROIsRegionMask

Reset the current hexadecimal ROI mask to 0.

Params (0)

clearImageOutsideROIs

When ROIs have been created or loaded, regions are usually created on the fly. This function allows setting to black all the pixels outside the ROIs, in the selected image.

Params (1)

Dest image select the image (Origin/Work/Secondary Work/Gray)

getROIsSurfaces

Find the number of pixels inside the ROIs. Remember the total number of pixel (to be exported to Log via **WriteOneIntTextValue**) and a text with the values for each existing regions (to be exported to Log via **writeOneTextString**).

Params (1)

createRegionBuf250

When a ROI file is opened and ROI list is present in the active configuration, this function allows creating an internal region table and a labelled buffer (hexadecimal) containing the label number of the ROIs. Maximum number is 250 to keep this buffer hexadecimal. This function is used with **CountPixelsInRegions** and **CountPixelsInRegionBuf**.

Params (0)

CountPixelsInRegionBuf

When a binary Mask exists, count all the pixel in the binMask in a table with number of pixels per region. Fill an internal integer table. *Works only if a region labelled buffer has been created previously* with instructions **CreateRegionBuf250**.

Params (1)

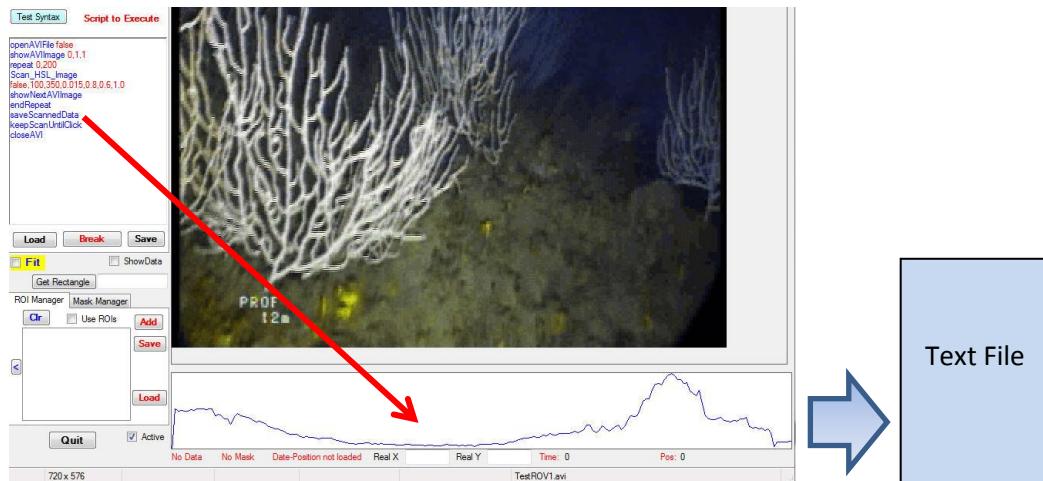
ExportToLog Export the internal table to Log file, if opened in a TAB separated format

KeepScanUntilClick

This command enters a waiting loop until a click is done on the scan window. This allows to move the mouse over the scan window to see the images corresponding to the computed values. Command must be send after an “endrepeat” command to keep the scanning window limits to the repeat instructions. When command is achieved, width of scan window is reset to the film length.

SaveScannedData

When a scan is achieved the values shown in the scan window may be exported to a text file using this command.



differenceInROI

Extract an image from the currently open film and shows it on the viewer.

Params (7)

<i>existing ROI number:</i>	(Default empty) type in the number of the ROI to use
<i>Input:</i>	(Default empty) type 'cur' to use the current image or a number to load another image
<i>Threshold:</i>	(Default empty) type 0=auto or a number (0-255) for the threshold
<i>Channel:</i>	Select a channel for computation (0=gray/1=Red/2=Green/3=Blue)
<i>typeDiff:</i>	Select the type of differentiation 1=absolute value (shows former and present positions) 2=difference set to positive value (shows present positions) 3=difference set to negative value (shows former positions)
<i>showDiff:</i>	Shows or hide the result (true/false)
<i>scroll:</i>	Use the scrolling window located under the viewer the display the 800 last values computed by the function (false/true)

writeOneINTTextValue

Some functions store a result internally for later use. This function allows sending this integer value to the currently opened Log file. A carriage return is added in the log file (single column output)

Params no parameter

writeOneINTTextValueNoCR

Some functions store a result internally for later use. This function allows sending this integer value to the currently opened Log file. No carriage return is added. This allows to use Tab function to perform

various computations on a same image and output TAB separated values to the currently opened Log file (multi column output when loaded in a spreadsheet program)

Params no parameter

writeOneDOUBLETextValue

Some functions store a result internally for later use. This function allows sending this double value to the currently opened Log file. A carriage return is added in the log file (single column output)

Params no parameter

writeOneDOUBLETextValueNoCR

Some functions store a result internally for later use. This function allows sending this double value to the currently opened Log file. No carriage return is added. This allows to use Tab function to perform various computations on a same image and output TAB separated values to the currently opened Log file (multi column output when loaded in a spreadsheet program)

Params no parameter

writeOneRGBCountTextValue

Some functions count pixels in the 3 channels RGB and store this result internally for later use. This function allows sending the 3 integer values of the count in Red, Green and Blue channel to the currently opened Log file. A carriage return is added in the log file (single column output)

Params no parameter

writeOneRGBCountTextValueNoCR

Some functions count pixels in the 3 channels RGB and store this result internally for later use. This function allows sending the 3 integer values of the count in Red, Green and Blue channel to the currently opened Log file. This function allows to insert this output in a TAB separated string sent to the currently opened Log file (multi column output when loaded in a spreadsheet program)

Params no parameter

writeOneTextString

Write a remembered text string to the currently opened log File. String ending with carriage return

Params no parameter

writeOneTextStringNoCR

Write a remembered text string to the currently opened log File. No carriage return.

Params no parameter

writeCR

Send a Carriage Return to the log file allowing to create a new line after compound string made with ...noCR functions.

Params no parameter

writeTAB

Sent a TAB character to the currently opened Log file.

Params no parameter

WriteCurrentDateTime

Collect the current Date time information from the system and send it to the currently opened Log file.

Params no parameter

7.11 – Scroll Tables

AddIntToScrollTable

Add an integer value, taken from the global memory process, to a scroll table. This displays the last 800 values treated inside a Repeat/endRepeat loop, or the last 800 values processed from camera input flux. This is a rolling display. The instruction sets a flag allowing the display of the scroll table on the active Window.

Params (0)

AddRGBToScrollTable

Add a channel (Red, Green or Blue) of anRGB value, taken from the global memory process, to a scroll table. Instruction like BinarizeRGB or RGBCountSurface fill this value. This displays the last 800 values treated inside a Repeat/endRepeat loop, or the last 800 values processed from camera input flux. This is a rolling display. The instruction sets a flag allowing the display of the scroll table on the active Window.

Params (1)

Channel Sets the channel to be displayed (1 = red, 2 = green, 3 = blue)

ResetScrollTable

Create a new table, erasing all previous values.

Params (0)

