

DIVA 3D

A software package for Video Processing,
Analysis and Visualization.

Version 4.0.0

Part I: Introduction and User's Guide



Alpha Tec Ltd

www.alphatecltd.com
alphatec@alphatecltd.com

Alpha Tec Ltd

Sales and distribution:

Alpha Tec Ltd
23 Papafi street
Thessaloniki 546 38
Greece

Tel, Fax: +30 2310 348019

Web:

www.alphatecltd.com

Email:

 alphatec@alphatecltd.com

 alphatec@otenet.gr

DIVA3D Manual

PART I: Introduction and User's guide

CONTENTS

1. Introduction.....	1
2. Hardware requirements	1
3. Installing and uninstalling DIVA3D	2
3.1 DIVA3D Demo	6
4. User's Guide	6
4.1 Video Data Representation.....	6
4.2 EIKONA3D Library features	8
4.3 DIVA3D GUI Description	11
4.3.1 Modules menu.....	14
4.3.2 Help menu	29

1. Introduction

DIVA3D for Windows is an integrated, powerful and flexible software package for digital video processing and analysis. It is based on an extensive library of routines (EIKONA3D library) related with the suitable manipulation of the specifically designed data structures and the multitude of the supported 3D processing, analysis and visualization algorithms. EIKONA3D Library can be used as a stand-alone Application Program Interface (API) for the development of custom video processing applications for Windows platforms using the C programming language. Though, the whole DIVA3D package provides in addition a suitable friendly and easy-to-use Graphical User Interface (GUI) that gives easy access to all library routines through a suitable menu structure. In addition, it guides the user to the selection of the suitable input/output parameters for each routine through carefully designed dialog boxes and it displays suitable error/result notification messages. DIVA3D GUI was developed for the Microsoft Windows environment and, thus, it has the look and feel of a usual Windows program. The present document discusses the hardware requirements of DIVA3D package, explains the utilized video data representation, provides an overview of DIVA3D library features and, finally, gives a brief guide for the usage of DIVA3D GUI.

2. Hardware requirements

DIVA3D for Windows requires a Pentium class or compatible processor PC running Microsoft Windows 95/98/ME/NT/2000/XP/2003 operating system. However, for better performance, a minimum machine configuration of a Pentium processor or compatible with at least 128 Mbytes RAM is highly recommended. Furthermore, in order to take advantage of the display capabilities of DIVA3D, a color monitor with a true color SVGA card is recommended.

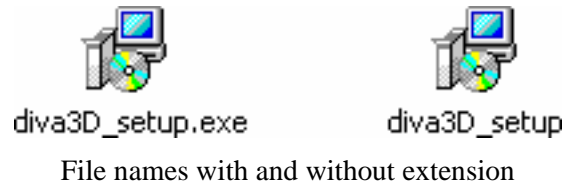
The minimum system requirements are:

- Entry-level PC, with an Intel Pentium processor or compatible.
- Microsoft Windows 95/98/Me/NT/2000/XP/2003
- 64 MB of RAM installed (128 or above recommended)
- 20 MB of available hard-disk space
- 24-bit display adapter card or above recommended

3. Installing and uninstalling DIVA3D

DIVA3D is distributed as one executable named *diva3D_setup.exe*.

Hint: The way file names are displayed may differ, depending on your computer settings. File extensions (3-letter suffixes indicating the file type) may be shown or hidden (e.g. *diva3D_setup.exe* or *diva3D_setup*) as shown in the following figure:



Before you start installation, please be sure that your computer system meets the requirements for installation. These are mentioned in section 2.

To install DIVA3D complete the following steps:

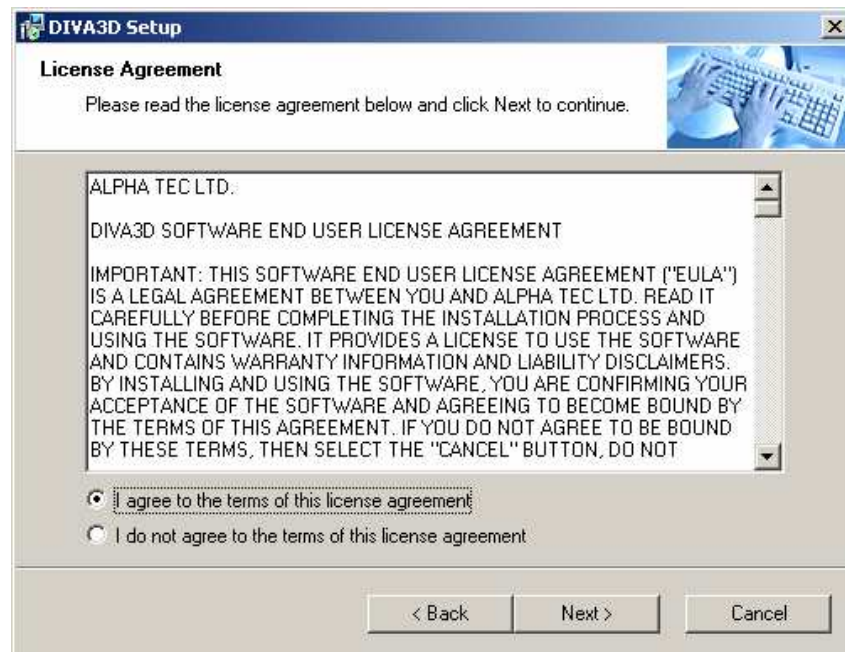
STEP 1: Run the DIVA3D installer

Double-click on the *diva3D_setup.exe* file to launch the installation. The following screen shot will appear:



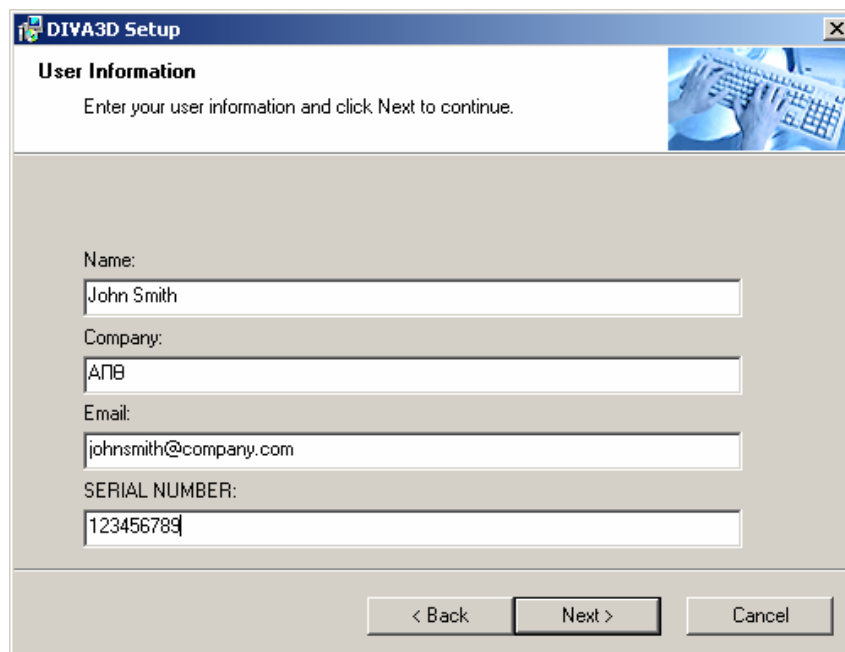
To proceed with installation, click on the **Next** button.

STEP 2: Read the DIVA3D license agreement



After reading the license agreement, you must click agree, in order to continue installing DIVA3D software.

STEP 3: Enter user information

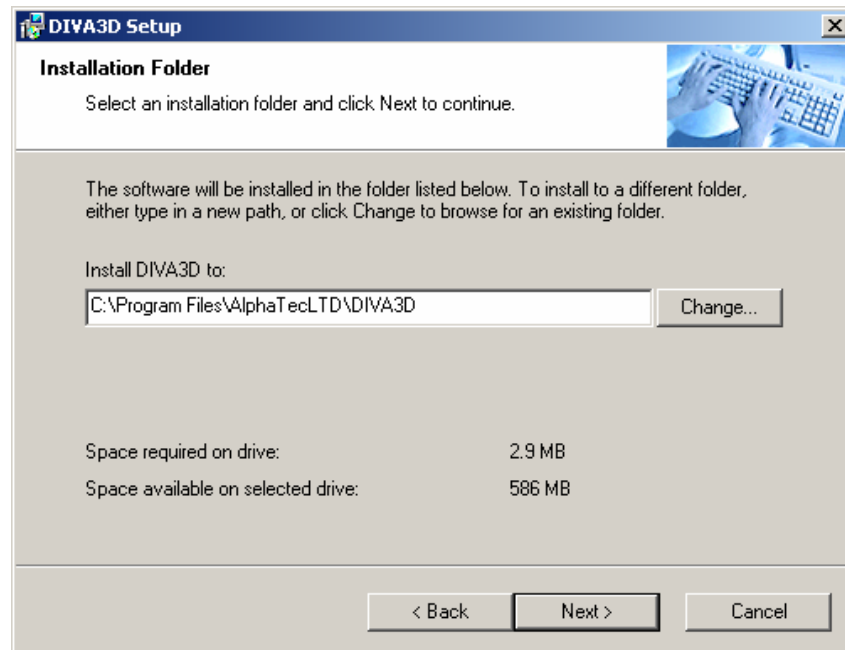


It is important to input your correct name, company name and email information that you used in your correspondence with Alpha Tec Ltd when you purchased your

software. If you give wrong input, the installation will be aborted. Remember, if you circulate DIVA3D in an unauthorized way, your personal data should circulate as well and you will be spoofed immediately.

Enter user information and click **Next** to continue.

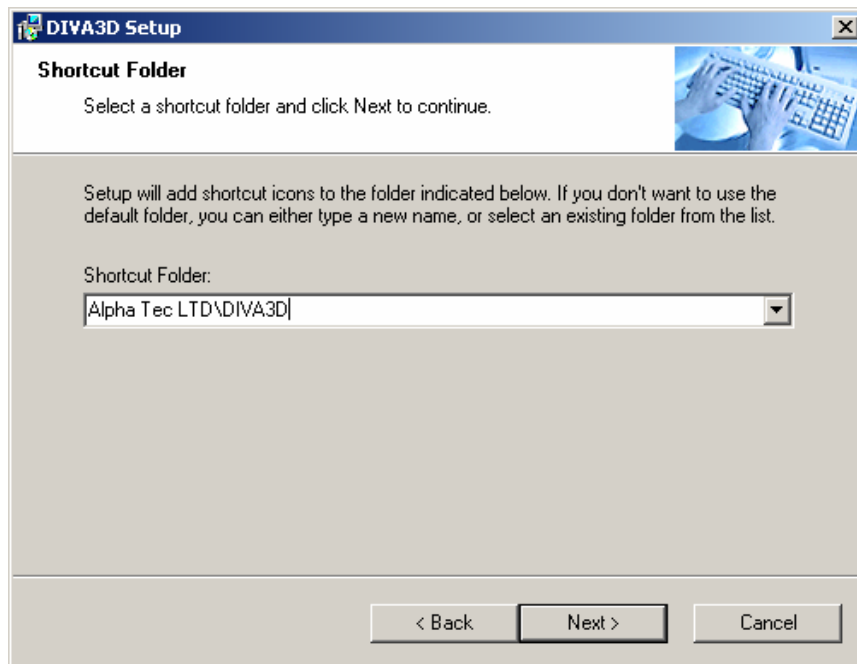
STEP 4: Select the installation folder



DIVA3D has a default installation directory. If you wish to change it, select the directory in install DIVA3D by clicking **Change**. When you have selected the directory, click **Next**.

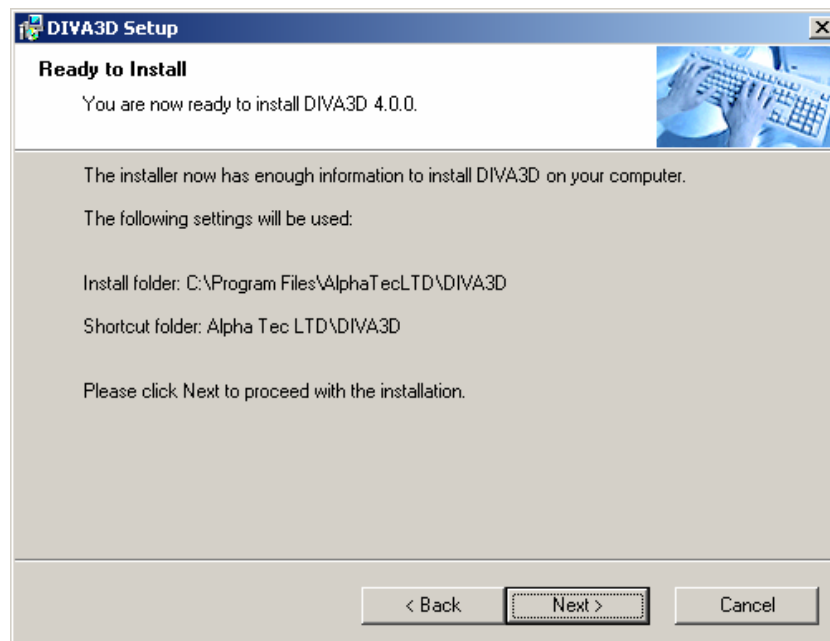
STEP 5: Select a shortcut folder

Next, the select shortcut folder window will appear



If you do not want to use the default folder, you can either type a new name, or select an existing folder from the list.

STEP 6: Ready to install



If you are ready to install click **Next**, otherwise click **Back** in case you want to make your changes.

After a successful installation the following window will appear:



By clicking **Finish**, DIVA3D is installed and is ready to use.

To uninstall DIVA3D complete the following steps:

- 1 Go to Start menu > Settings > Control Panel
- 2 Select Add / Remove Programs
- 3 Select Diva3D and click Change/Remove

or

go to the shortcut folder of DIVA3D by clicking Start>Programs>Alpha Tec LTD>Diva3D (this is the default shortcut folder) and click Uninstall Diva3D.

3.1 DIVA3D Demo

DIVA3D demo can be downloaded from www.alphatecltd.com. It is installed in the same way with the full DIVA3D version. It has full functionality, but can process video streams up to 1000 frames of any size, e.g. 512×512×15.

4. User's Guide

4.1 Video Data Representation

The representation of a video sequence is achieved effectively via the use of the Video Stream Data structure that keeps all the information required to define a video sequence. Attributes of this data structure are :

video length: Specifies, in number of frames, the length of the video.

video format: Specifies if the video stream's video format is PAL or NTSC.

rate: Specifies the number of frames displayed within a second during playback.

color format: Specifies the color model used to encode the color information inside the video stream. Typical color formats are RGB, CMY, HSI, HSV, HLS).

filename: Name of the video file containing the video sequence. This file is associated with the video stream buffer.

video stream buffer: The handling of digital video data is achieved with the use of the Video Stream Buffer (Volume) data structure. In both the program's GUI and documentation the terms Volume and Video Stream Buffer are used interchangeably.

Members of the Video Stream Buffer (Volume) data structure are:

- **pixel depth(data type):** The supported data types for a single video stream buffer element (voxel) are unsigned char (8 bit per voxel), int (32 bits per voxel), and float (64 bits per voxel). The first can also be used for storing binary data. According to their data type, video streams are divided in three categories: ucval video streams (which contain unsigned char data), ival video streams (which contain integer data) and fval video streams(which contain float data).
- **color type:** This specifies whether one or three channels should be used. One channel is used for binary or greyscale video stream and three channels for color video streams (e.g. RGB, CMY, HSI, HSV, HLS).
- **dimensions:** The dimensions of the three-dimensional array that stores the video stream buffer data. Supposing that the 3D data represent a number of two-dimensional frames, the dimensions are the number of frames and the number of rows and columns of each frame.
- **VOI position:** The user can specify a VOI (Volume Of Interest), that is a subpart of the video stream buffer , in order to restrict processing only in that part.
- **pointer to 3D data:** The pointer(s) to the three-dimensional array(s) in memory. In the case of use in a Windows 3.* application program, additional pointer to data handles are provided.
- **user field:** An additional string field is provided, which can be used from the user to store video stream buffer - related information, such as name, description, etc.

This data structure enables the uniform manipulation of all the supported 3D data combinations, that is binary/8-bit/integer/floating-point data in one/three channels. All library routines can be applied on either the whole video stream buffer or a user-specified VOI of the video stream buffer.

4.2 EIKONA3D Library features

The library routines can be divided in several categories. A brief description for each category is given below:

Memory Allocation/Deallocation routines : These include routines for allocating and deallocating memory space for keeping 3D data and related data structures that are to be processed. In addition, there are routines for initializing a video stream buffer structure and for performing video stream buffer integrity tests.

Input/Output routines: These include routines for reading and writing 3D data to disk. Routines are provided for reading and writing 3D data as a sequence of 2D frames in raw, TIFF, Targa, BMP, JPEG, GIF, PBM, PGM and PPM formats, or as one file in raw format. In the first case, each frame is stored in a different file (in the case of raw format separate files are created for each channel), so that it can also be opened as a 2D image, whereas the filenames are composed of a base-name, sequential numbering and an extension (in the case of raw format a different extension is required for each channel). In the second case, all frames are stored in one file in a frame-row-column (z-y-x) sequence. An additional routine is provided for dumping data in ASCII format, so that they can be previewed by any ASCII editor.

Basic Operation routines: These include routines for performing several basic operations on video stream buffer data. These operations include clearing of a VOI with a user-specified value, copying a VOI of a video stream buffer to a selected position in another video stream buffer, extracting a frame of a video stream buffer to a 2D image, loading a 2D image in a video stream buffer frame, converting video stream buffer between the different data types and performing arithmetic operations between video stream buffers, such as logical AND, OR, XOR operations.

Filtering routines: These include routines for performing 3D linear and non-linear filtering operations on a video stream buffer. There are routines for histogram calculation, histogram equalization, moving-average filtering, median filtering (standard median, running median, multistage median and weighted median are supported), min/max filtering, filtering with user-specified 3D L-filters, adaptive filtering, and generalized filtering with user-specified 3D FIR filters via convolution or FFT. Finally, there are routines for video stream buffer interpolation and decimation.

Edge Detection routines: These include routines for 3D edge (surface) detection. Several techniques for edge detection are supported, including 3D extension of Sobel masks, 3D Kirsch edge templates, 3D Laplace operator, 3D range operator, edge detection through 3D Hough Transform.

Region Segmentation routines: These include routines for 3D region segmentation. Several common segmentation techniques are supported, which include thresholding, region growing, region splitting, region merging and region split-merging. Also, region labelling and object counting are supported.

Measurement routines: These include routines for 3D measurements. Surface and video stream buffer measurements are supported, as well as calculation of the bounding box in a video stream buffer and finding of the minimum and maximum voxel values.

Transform routines: These include routines for performing several transforms on the video stream buffer data. The supported transforms are the FFT (Fast Fourier Transform) and the Inverse FFT, the DCT (Discrete Cosine Transform) and the Inverse DCT. In addition, there are routines for performing convolution, for calculating the auto-correlation of a video stream buffer, for calculating the correlation between two video stream buffer, for 3D Power Spectrum Estimation using an extension of the Blackman-Tukey method, for transforming a complex video stream buffer between real/imaginary and magnitude/phase representation and, finally, for transforming a spectrum to 3D image for visualization.

Morphology routines: These include routines for performing 3D morphological operations on binary objects represented by binary video stream buffers. There are routines for performing the basic morphological operations that is Minkowski addition and Minkowski subtraction, dilation, erosion, opening and closing, using any user-defined structuring elements. In addition, there are more complex routines for performing three-dimensional morphological shape decomposition using one structuring element or a set of structuring elements.

Rearrangement routines: These include routines for rearranging the data of a video stream buffer. There are routines for rotating a video stream buffer around its center by ± 90 degrees around any axon or, generally, around any selected point by any specified angles. Also, there are routines for mirroring the video stream buffer data around its center or around specific axons and planes.

Resizing routines: These include routines for resizing a video stream with different ways. There are routines for uniform zooming and decimation (with the same integer factor for all dimensions), for 3D interpolation/decimation (Nearest Neighbour or trilinear) with any factor for each dimension, for linear frame interpolation/decimation, and for shape-based binary frame interpolation. Also, there is a routine for adding zero frames between the existing ones, which can be useful for visualizing contour data.

Visualization routines: A simple video stream buffer visualization way is the display of the frames as images, which can easily performed by an application program. The library provides routines for more complicated visualization of a video stream buffer. There is a routine for producing a frame gallery image for simultaneous preview of all frames of a video stream buffer. Routines are provided for sectioning a video stream buffer with a plane of any orientation and visualizing the section, and for visualizing a video stream buffer using parallel projection (normal, average, or maximum projections are supported) under any user-specified angles. Also, there are routines for producing a 3D surface representation of a video stream buffer for surface rendering and routines for video stream rendering (with the options of using or not depth and/or color information).

Video Stream input output (streamio) library: Streamio is a library interacting with the EIKONA3D library. In this library the concept of the video stream data structure is defined and it contains high level routines that act like an interface between video stream buffer, video file and the EIKONA3D library functions. All EIKONA3D Library functions apply only to the contents of a video stream buffer. Since loading a whole video sequence to a video stream buffer is extremely memory consuming, these functions had to be extended to be used to a video sequence without having to load the whole sequence to a video stream buffer. This problem is solved with the use of streamio library: access of the video data inside a stream is not implemented directly but indirectly using streamio library routines.

4.3 DIVA3D GUI Description

The various operations of DIVA3D package are organized into the hierarchical menu structure of a friendly GUI. For the majority of operations, a similar procedure is followed: the user selects an operation from the menu, gives the input and/or output video stream and specifies all other possible required parameters. Before proceeding to the description of each operation, we shall describe the handling of the most common dialog boxes:

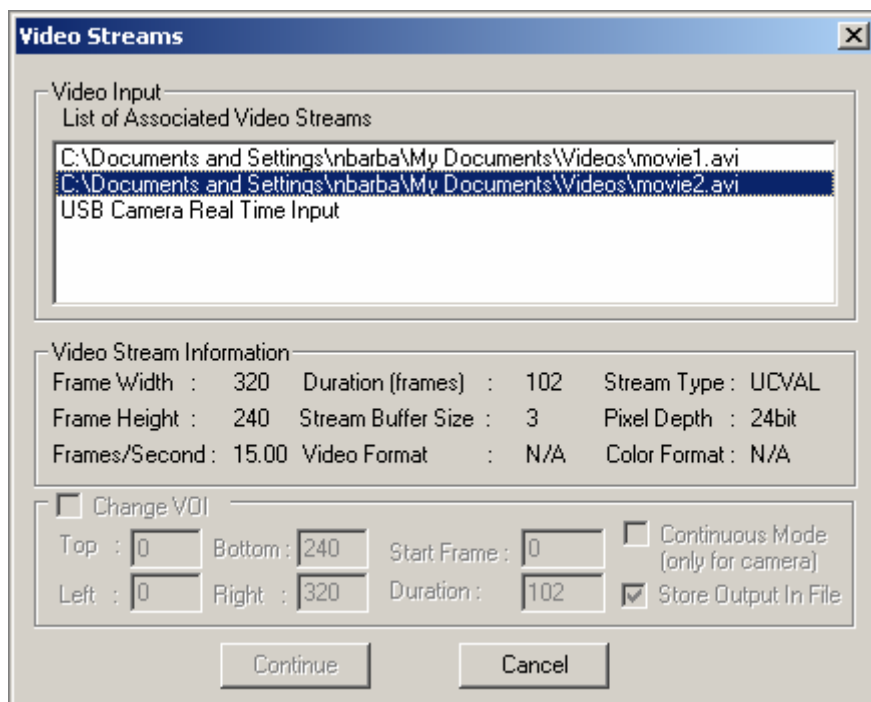


Figure 1. Video Streams Dialog Box

Video Streams Dialog Box: This dialog box prompts the user to select a video stream to destroy it, to view its attributes or to use it as an input to an operation. Video streams are referred to by their associated file filename while the user can use the *Video Stream Information* field to view all the information retrieved from the selected video stream. All operations can be applied to a subpart of the video stream referred to as VOI (Volume Of Interest), which can be specified by enabling the tag *Change VOI* and giving the desired values in the fields *Top VOI* (minimum frame row index), *Bottom VOI* (maximum frame row index), *Left VOI* (minimum frame column index), *Right VOI* (maximum frame column index), *Start Frame* (minimum frame time index), *Duration* (number of frames to process). The user can choose to store the output video stream of the process to a file or not with the *Store Output in File* checkbox. In case the input video stream is from a camera the user can select, using the *Continuous Mode* checkbox, if the performed operation will be executed until *Abort Process*, on Progress dialog box, is pressed.

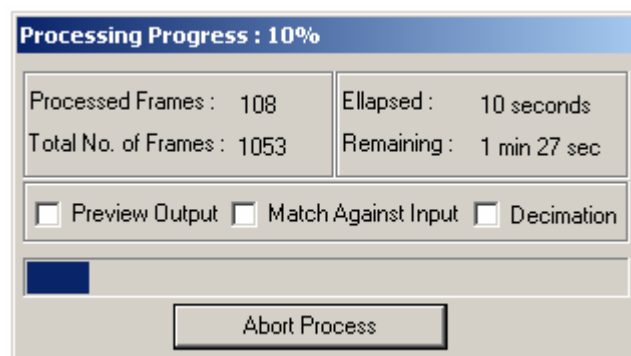


Figure 2. Progress Dialog Box

CProgress Dialog Box: This dialog box appears whenever an operation takes place. It informs the user about the process progress with the progress bar, about the number of frames processed / to be processed and about the remaining and elapsing time. The checkboxes are for the user to preview the output video stream while the process is taking place, to match it against the input video stream and to decimate the previewing windows of both input and output video streams.

Stream Buffer Operations: Some of DIVA3D operations apply only to the contents of the video stream buffer and not to the whole video stream. For such cases, together with the dialog box to define any parameters for the operation, the following field will

appear informing the user that the current operation cannot be applied to the whole video stream.

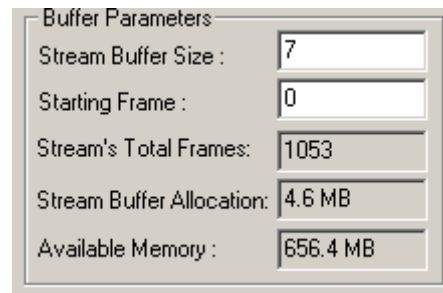









Figure 3. Dialog Box portion for video buffer operations.

Using this field the user can select how many frames will be processed and which, of the frames of the video stream, will be the starting frame of the process. There are also additional information about the memory allocation estimated for the performed operation.

DIVA3D Player: Whenever a video stream is created, DIVA3D displays this stream using the DIVA3D Player. The playback of the video stream is independent of the size of the video buffer size.



Figure 4. A view of DIVA3D Player window.

The user can control the movie playback by using the buttons :  to play the entire movie,  to step one frame back ,  to step one frame forward ,  to stops playback,  to set the lower process limit for all the future operations for that video stream,  to set the upper limit ,  to play only the portion of the movie the user selected with the two limit buttons. Note that the portion of the movie which is

selected with the two limit buttons is highlighted. By right clicking on the movie window a popup menu appears containing options for playing/stopping the movie, selecting the view size, setting the movie's volume level setting the playback speed. Finally by right clicking on the DIVA3D Player window a menu appears where the user can select to go to the beginning of the movie, at the end of the movie or to display coordinates. After selecting display coordinates the user can click to any point of the movie window and its spatial coordinates (X,Y,T) along with the selected point's luminosities (R,G,B) will appear in the DIVA3D Player window title.

The main window of DIVA3D has a menu bar consisting of the following drop-down menus: **File**, **Operations**, **Visualization**, **Modules** and **Help**. **File**, **Operations** and **Visualization** menus are deactivated since they refer to DIVA3D Student Edition and are not explained in this manual. (For more information on these please refer to DIVA3D Student Edition manual.).

The menu options of Modules and Help menus are described in the following subsections.

4.3.1 Modules menu

This menu provides a flexible expansion mechanism for incorporating additional modules in DIVA3D by the developer or even by any user. Each such module adds a sub-menu under the **Modules** menu and should be developed in the form of a DLL (Dynamic-Link Library) that follows certain rules, so that DIVA3D can determine all necessary information for it, such as the name and the structure of the sub-menu. The **Modules** menu is built dynamically when DIVA3D starts, after examining all the DLLs that exist in DIVA3D directory. Bellow are described the submenus that implement most of the functions of EIKONA3D Library, equivalent to the **File** , **Operations** , **Visualization** deactivated menus.. All of the modules described bellow must exist in DIVA3D's working folder for DIVA3D to function normally.

File IO Submenu: This menu includes file I/O operations which is described below. A view of this menu can be seen in Figure 5.

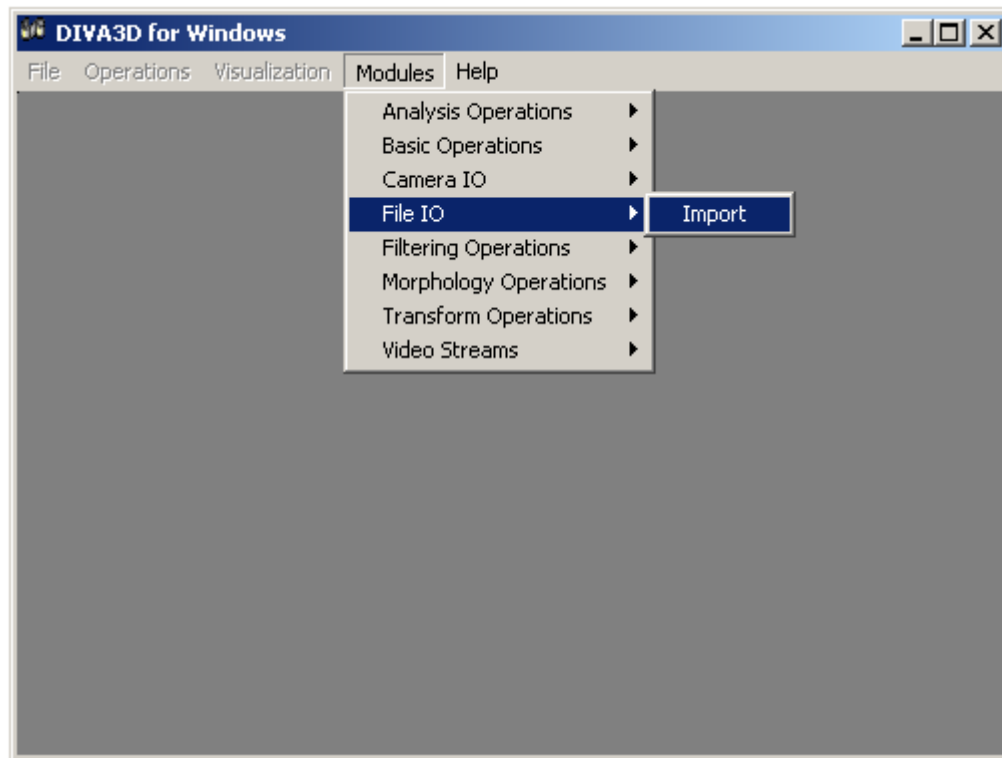


Figure 5. A view of the File IO submenu.

Import: This menu item enables the user to import a video file into DIVA3D. It displays a typical Windows Open File dialog box where the user can select the file he/she wishes to import. When a file is selected, DIVA3D creates a video stream structure with the attributes retrieved from the video file and displays it using the DIVA3D Media Player. The imported video stream, and all its attributes, can be viewed via the *Video Streams* dialog box and can be used as an input to a DIVA3D operation. The supported file extensions for importing video streams are :

- AVI files (*.avi): These are the common Audio Video Interleaved files containing ucval video streams, where each pixel's depth is 8 bit (unsigned character). These video streams are handled by the aviio module (aviio.dll).
- FVAL files (*.fval): Files containing fval video streams where each pixel's depth is 64 bits (float value). These video streams are handled by the fvalio module (fvalio.dll).
- IVAL files (*.ival): Files containing ival video streams where each pixel's depth is 32 bits (integer value). These video streams are handled by the ivalio module (ivalio.dll).

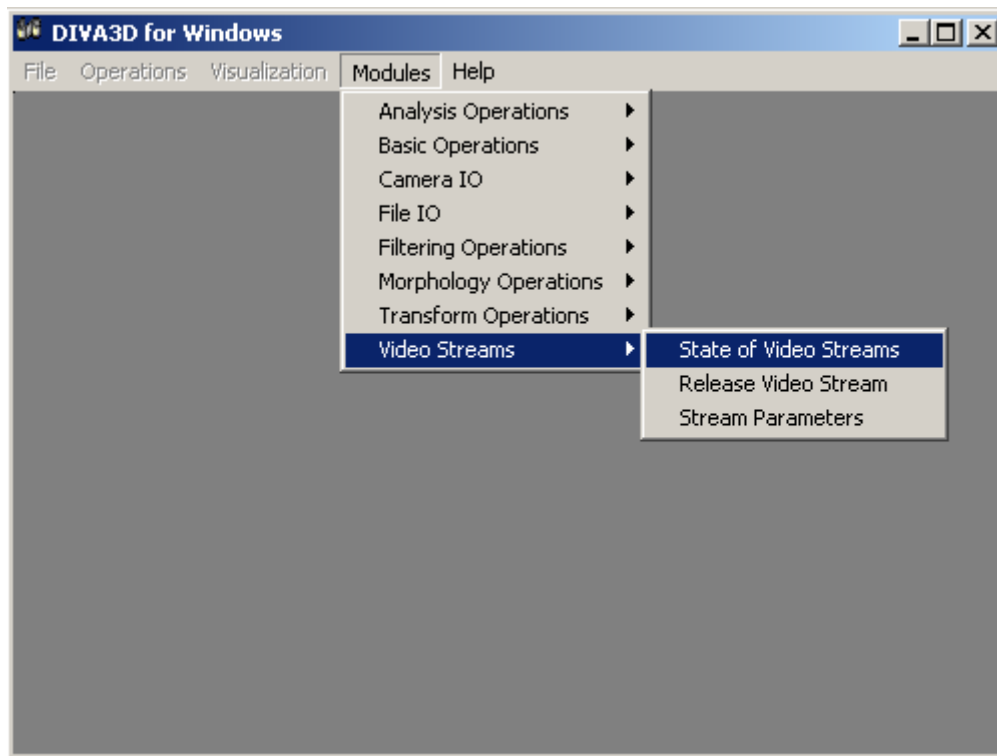


Figure 6: A view of the Video Stream submenu.

Video Stream Submenu: This menu includes video stream management operations which are described below. A view of this menu can be seen in Figure 6.

- **State of Video Streams:** Displays the *Video Streams* dialog box where the user can get information about the imported video streams and their attributes. By double clicking on a video stream a DIVA3D Media Player window is created where the user can view the selected video stream.
- **Release Video Stream:** Displays the *Video Streams* dialog box but with a *Release* button instead of *Continue*. By selecting a video stream and pressing Release, the memory occupied for the video stream is released and the video stream is destroyed. The user can select and destroy more than one video streams, using the *Release* button, before closing the dialog with the *Close* button.
- **Video Stream Parameters:** Displays the following dialog box where the user specifies parameters related with the video stream buffer data structure and its members.

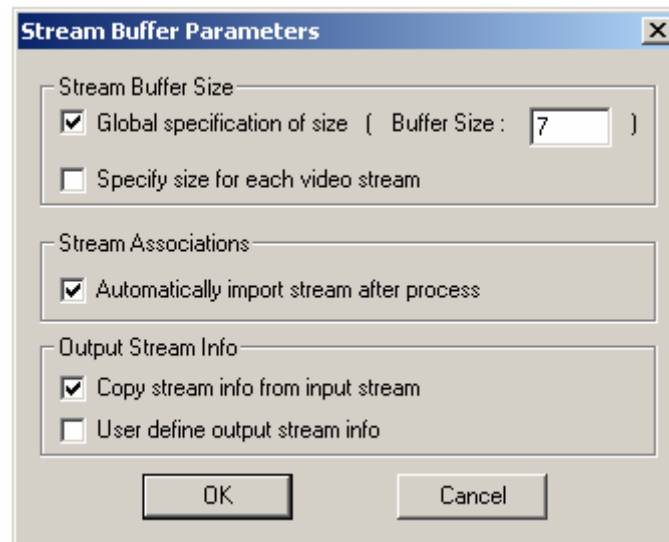


Figure 7. Stream Parameters Dialog Box

The *Stream Buffer Size* field is used to specify the size, in number of frames, of the video stream buffer of the video stream. It can be either defined once, and this definition will apply to all the video files imported afterwards, by checking the *Global specification of size* checkbox and defining the default video stream buffer size in the *Buffer Size* edit box, or it can be defined each time a video file is imported with the *Specify size for each video stream* checkbox.

The *Stream Associations* field defines if the output video stream will be automatically imported after an operation has finished.

The *Output Stream Info* field is used for defining the way of defining the output video stream data structure information fields. There are two options: The user can either select to copy, for each output video stream, the information retrieved from the correspondent input video stream or to define manually these information fields each time a video stream is created.

Basic Operations Submenu: This menu leads to the various routines of the EIKONA3D library for the 3D processing of video streams. In most cases, the functions require an input video stream, selected through the *Video Streams* dialog box, and possibly some related parameters, given through additional dialog boxes. In case the user selected to use an output file to store the outcome of the process, see *Video Streams* dialog box, he/she will be prompted with a typical *Save As...* dialog box to select a file to store the output video stream. Processing is performed only

within the volume of interest (VOI) that can be uniquely specified for each input video stream. The menu items are described in detail below and a view of the Basic Operations menu can be seen in Figure 8.

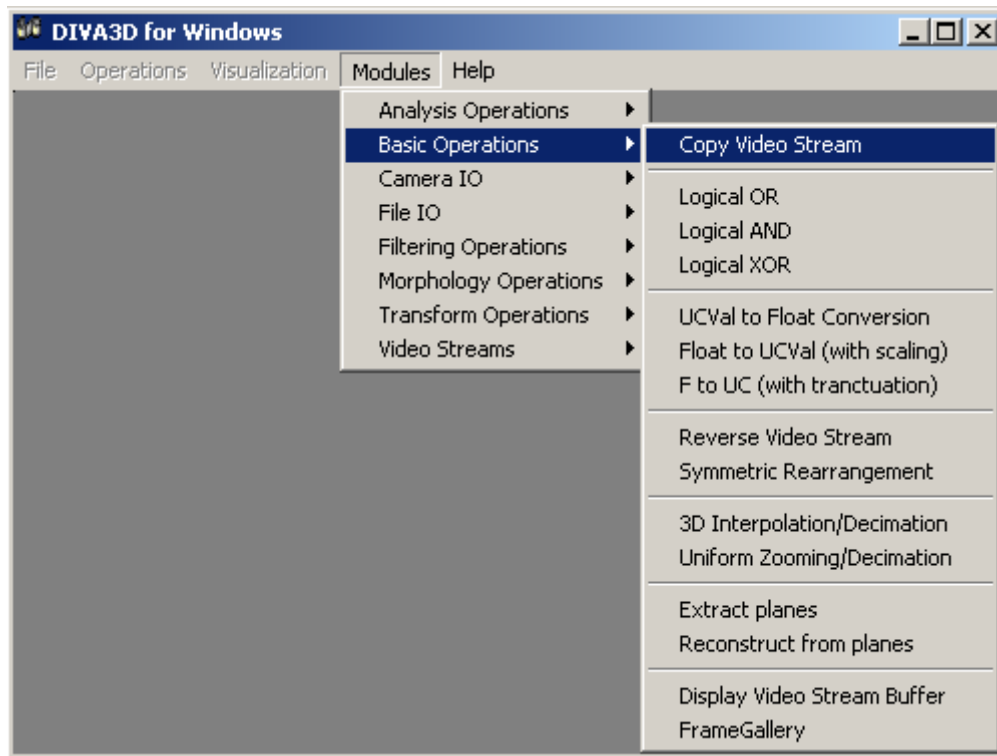


Figure 8. A view of the Basic Operations submenu.

- **Copy Video Stream:** Copies a VOI of a video stream to a newly created video stream. The user first selects the input video stream and VOI through the *Video Stream* dialog box and then a file to contain the output video stream.
- **Logical AND:** Performs the bit-wise AND logical operator between two video streams (for binary video streams only). The user first selects the two input video streams, through two successive *Video Streams* dialog boxes, and then a file to contain the output video stream.
- **Logical OR:** Performs the bit-wise OR logical operator between two video streams (for binary video streams). The same procedure as in Logical AND is followed.
- **Logical XOR:** Performs the bit-wise XOR logical operator between two video streams (for binary video streams). The same procedure as in Logical AND is followed.

- **UCVal to Float conversion:** Converts a video stream with unsigned char data type to a video stream with float data type. The user selects the input video stream and an *.fval file to contain the output video stream.
- **Float to UCVal conversion (with scaling):** Converts a video stream with float data type to a video stream with unsigned char data type using scaling (the input value range is scaled to the range 0...255). The user selects the input video stream, which should be an fval video stream, and a file to contain the output video stream.
- **Float to UCVal conversion (with truncation):** Converts a video stream with float data type to a video stream with unsigned char data type using truncation (input values less than 0.0 are set to 0 at output, whereas input values greater than 255.0 are set to 255 at output). The same procedure as in Float to UCVal conversion (with scaling) is followed.
- **Reverse video stream:** Creates a reversed copy of the input video stream. The user first selects the input video stream and a file to store the output video stream.
- **Symmetric:** Produces the symmetric of a video stream in respect to its centre. The user selects the input video stream and if selected a file to store the output video stream.

3D Interpolation/Decimation: Transforms a video stream using any specified interpolation/decimation factor for each dimension. The user first selects the input video stream and then sets the interpolation options through the *Interpolation Options* (Figure 8) dialog box. The user can choose to set interpolation factors to the three video dimensions (x,y,t) or to set the output video stream's frame dimensions and specify that the factors will be derived from the output video stream. There are two available methods for interpolation: Nearest Neighbour and Linear interpolation (the first is faster, whereas the second gives better results and is recommended). Finally, if selected, the user selects a file to store the output video stream.

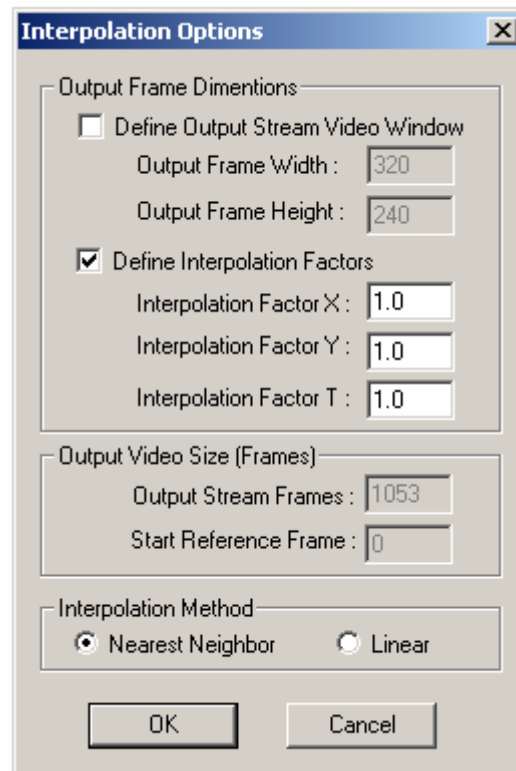


Figure 9. Interpolation Options Dialog Box

Note that because DIVA3D processes the video stream partially, the user may be informed that the video stream cannot be interpolated with this T factor . In such case the user should change the VOI (Volume Of Interest) of the video stream (e.g. select to process one frame less).

- **Uniform Zooming / Decimation:** Enlarges or decimates a video stream with the same specified zooming factor for all three dimensions. The user selects the input video stream, gives the factor through the *Zooming/Decimation* factor dialog box and a file to store the output video stream.
- **Extract planes:** Performs plane extraction on a selected 24 bit video stream. The user selects an input video stream, is prompted with the *Plane Extraction* dialog box to select which planes will be extracted and finally selects video files to store the output video streams.
- **Construct from Planes:** Combines three 8bit video streams to produce a 24bit video stream. User selects three 8bit input video streams and a file to store the output video stream. The three input video streams must have the same dimensions in both three axis (x,y,t).

- **Display Video Stream Buffer:** Display's the content's of a video stream buffer.
- **Frame Gallery :** This menu option is used for displaying the frames of a video stream buffer as a gallery image, that is one displaying all frames simultaneously one next to the other (usually in smaller size). The user first selects the input video stream. Then, he specifies the required parameters through the *Frame Gallery Display Options* dialog box, which are the frames per row in the output image and the interpolation method (Nearest Neighbour or linear) that will be used for the adaptation of the frame size. A default width of 100 pixels is used for the input frames in the output image. This operation applies only to the contents of the video stream buffer and not to the whole video stream

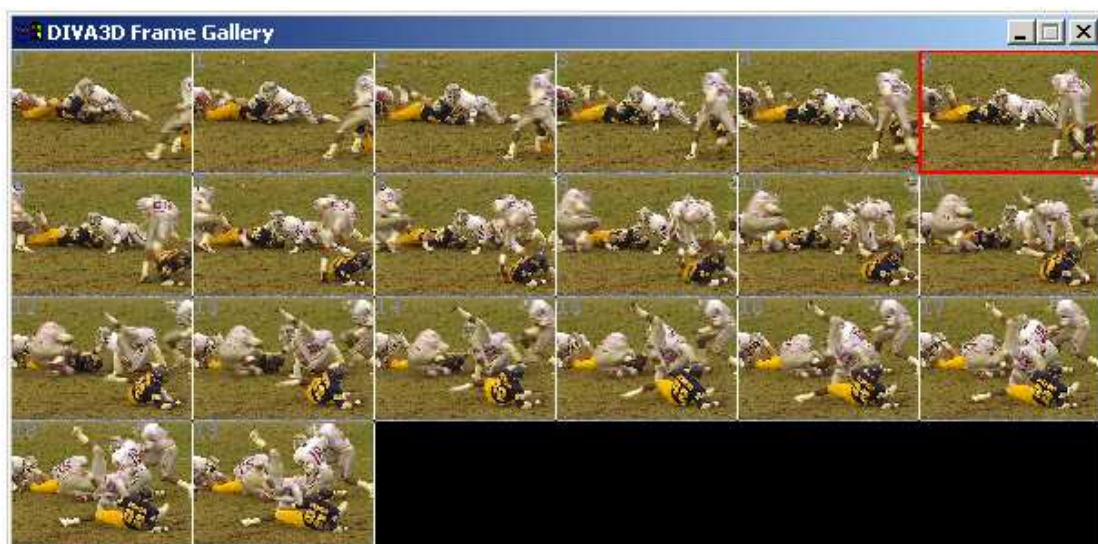


Figure 10 Output of the Frame Gallery Menu

Filtering Operations Submenu: This sub-menu (Figure 11) includes several 3D filtering operations, which are the following:

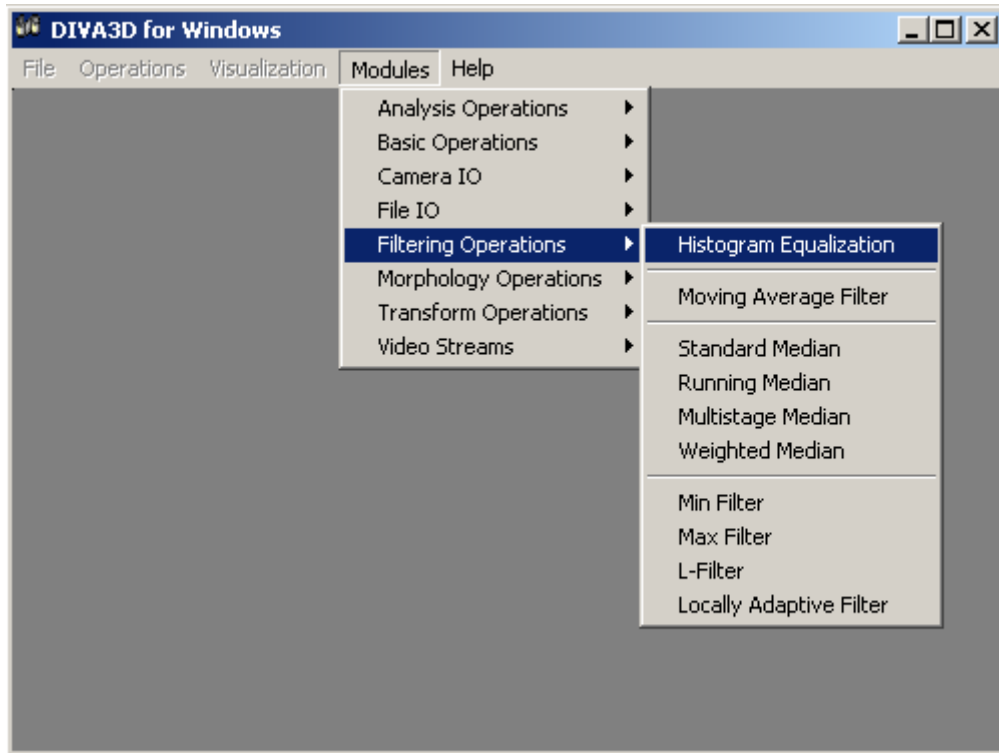


Figure 11. A view of the Filtering Operations Submenu

- **Histogram Equalization:** Performs histogram equalization of a video stream. The user selects the input video stream through the *Video Streams* dialog box and a file to store the output video stream.
- **Moving Average Filter:** Performs filtering of a video stream using the moving average filter. The user first selects the input video stream and then a dialog box appears for setting the dimensions of the 3D window mask through the *Filter Window Dimensions* dialog box. Finally the user selects a file to store the output video stream.

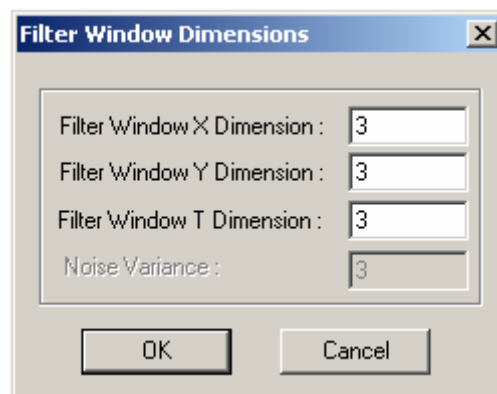


Figure 12. Filter Window Dimensions Dialog Box

- **Standard Median:** The standard 3D median filter. The same procedure as in Moving Average Filter is followed.
- **Running Median:** Performs filtering of a video stream using a faster implementation (running) of the median filter. The same procedure as in Moving Average Filter is followed.
- **Multistage Median:** Performs filtering of a video stream using the multistage implementation of the median filter. The same procedure as in Moving Average Filter is followed, with the difference that only one parameter is given for the 3D window size, as it can only be cubic.
- **Weighted Median:** Performs filtering of a video stream using the weighted median implementation of the median filter. The user first selects the input video stream. The user is prompted to select a *.par file containing the weights. The format of the *.par file is described in detail in page 27 of this manual.
- **Min Filter:** Performs filtering of a video stream using the minimum filter. The same procedure as in Moving Average Filter is followed.
- **Max Filter:** Performs filtering of a video stream using the maximum filter. The same procedure as in Moving Average Filter is followed.
- **Locally Adaptive Filter:** Performs locally adaptive filtering of a video stream that is filtering dependent on the local region content. The user selects the input video stream and a file to store the output video stream.

Analysis Operations Submenu: This sub-menu includes other sub-menus which are the following:

- **Histogram (pdf):** Calculates and displays the pdf of the histogram. First the user selects the input video stream through the *Video Streams* dialog box. After the histogram operation, the user is asked if he/she wishes to store the histogram to text files, one file for each plane.

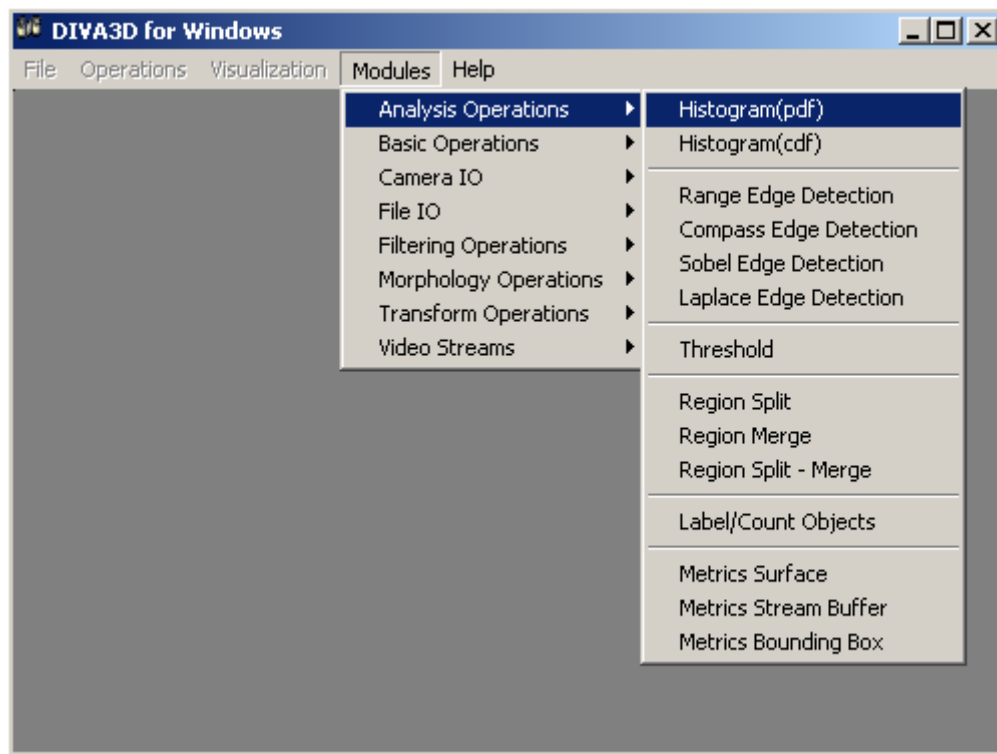


Figure 13. A view of the Analysis Operations Submenu

- **Histogram (cdf):** Calculates and displays the cdf of the histogram. The same procedure is followed like the pdf histogram calculation.
- **Range Edge Detection:** Performs edge detection using the range operator. The user first selects the input video stream and then defines the dimensions of the 3D window mask through the *Range 3D Window* dialog box. Finally the user should select a file to stored in the output stream.

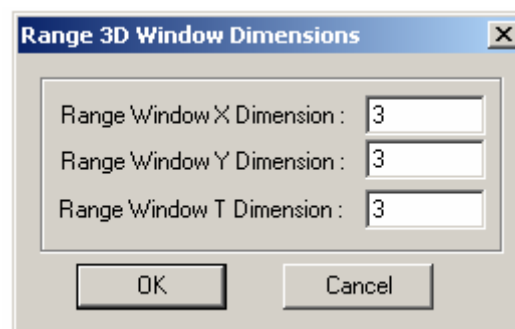


Figure 14. Range Window Dimensions Dialog Box

- **Compass Edge Detection:** Performs edge detection using the 3D Kirsh masks. The user selects the input video stream and specifies the direction of

the detected edge planes through the *Plane Direction (0,45,90,135)* dialog box.

- **Sobel:** Performs edge detection using the 3D Sobel masks. The user selects only input and output video streams.
- **Laplace:** Performs edge detection using the Laplace operator. The user selects only input and output video streams.
- **Threshold:** Performs region segmentation using simple thresholding. The user selects the input video stream, uses the *Threshold Parameters* dialog box (Figure 15) to define thresholding parameters and, if selected, selects a file to contain the output video stream.

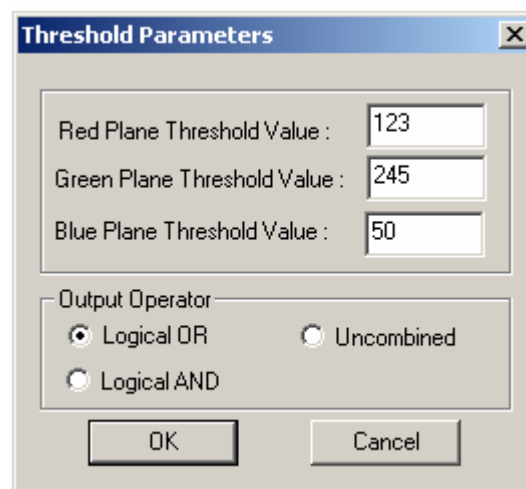


Figure 15. Threshold Parameters Dialog Box

If the input video stream is 24bit then the user must select three different threshold levels, one for each plane, and how the output stream is going to be composed: Either the three planes will be uncombined, producing a binary 24bit video stream, or the may be combined using the logical OR and AND operators to produce a binary 8bit video stream.

- **Region Merge:** Performs region segmentation using the region merging technique. The user selects the input video stream, defines the threshold value for region specification and the maximum desirable number of regions through the *Region Segmentation* dialog box. This operation applies only to the contents of the video stream buffer and not to the whole video stream.
- **Region Split:** Performs region segmentation using the region splitting technique. The user selects the input video stream, defines a threshold value

for region segmentation and selects a file to contain the output video stream. This operation applies only to the contents of the video stream buffer and not to the whole video stream.

- **Region Split-Merge:** Performs region segmentation using the region split-merge technique. The same procedure as in the Region Split operation is followed.
- **Label/Count objects:** Performs region (a region is considered as an object) counting and labelling. The user selects the input video stream and is notified for the number of objects found inside that video stream. This operation applies only to the contents of the video stream buffer and not to the whole video stream.
- **Surface:** Calculates the surface of the objects in a binary video stream (number of surface voxels). The user selects the input video stream and is notified for the measurement result. This operation applies only to the contents of the video stream buffer and not to the whole video stream.
- **Video Stream Buffer:** Calculates the video stream buffer of the objects in a binary video stream (number of voxels). The user selects the input video stream and is notified for the measurement result. This operation applies only to the contents of the video stream buffer and not to the whole video stream.
- **Bounding Box:** Finds the bounding box of a video stream buffer (the smallest VOI containing non-zero voxels). The user selects the input video stream and is notified for the measurement result. This operation applies only to the contents of the video stream buffer and not to the whole video stream.

Transforms Operations Submenu: The operations in this sub menu apply only to the contents of the video stream buffer and not to the whole stream. This submenu includes the following operations:

- **FFT:** Performs the Fast Fourier Transform (FFT) of a video stream. The user selects the input video stream and two *.fval files to contain the real and imaginary parts of the complex fval output video stream.
- **IFFT:** Performs the Inverse Fast Fourier Transform (IFFT) of a video stream. The user selects the real and imaginary parts of the complex input fval video stream and a file to store the output video stream.

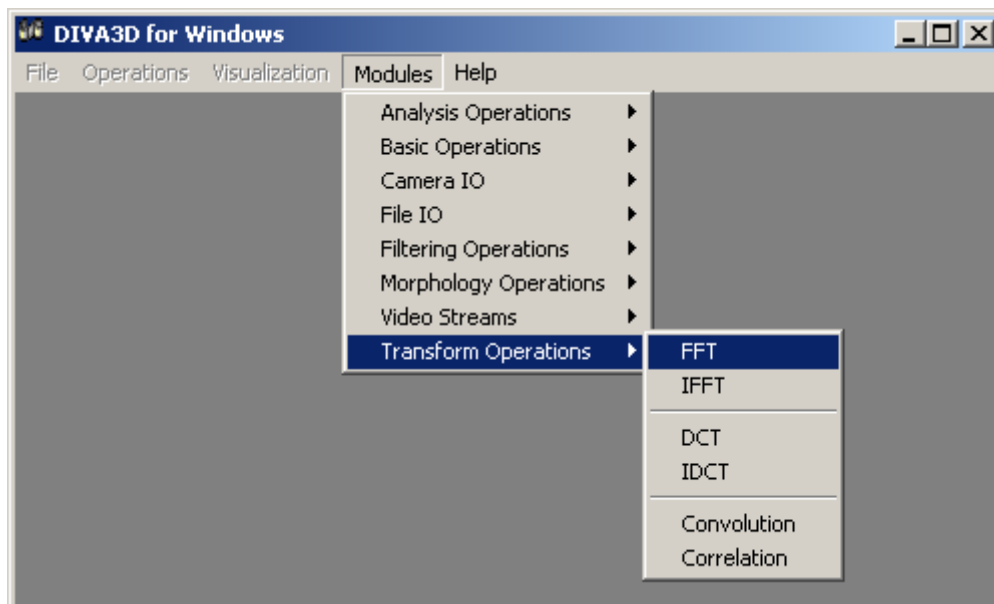


Figure 16. A view of the Transforms Submenu

- **DCT:** Performs the Discrete Cosine Transform (DCT) of a video stream buffer. The user selects an input video stream and a file, if selected, to store the output video stream.
- **IDCT:** Performs the Inverse Discrete Cosine Transform (IDCT) of a video stream. Same procedure is followed as in the DCT operation.
- **Convolution:** Calculates the convolution between two video streams. The user selects the two input video streams a file to store the output video stream.
- **Correlation:** Calculates the correlation matrix for two video stream buffers. The user selects the two input video streams output a file to contain the output video stream.

Morphology: This sub-menu includes the following 3D binary mathematical morphology operations, which require that a binary video stream buffer containing the 3D structuring element exists.

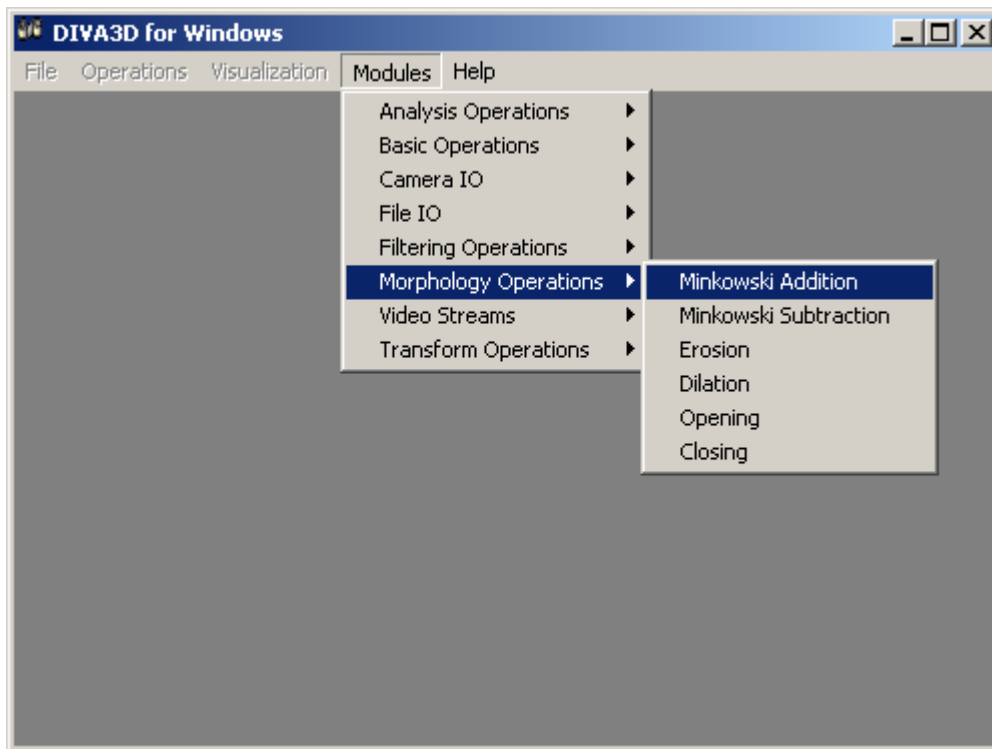


Figure 17. A view of the Morphology Submenu

Whenever a 3D structuring element is required, the user will be prompted with an *OpenFile* dialog box to select a *.par file containing the structuring element. If the *.par file follows a set of rules, DIVA3D will automatically create a binary video stream buffer containing the structuring element data from the file. These rules are: The overall number of the elements inside the file should be a cubic power of 2 (9,27 and so on) depending on the structuring element dimensions (3x3,5x5 and so on) and each number should be followed by a coma(,) operator.(Figure 18)

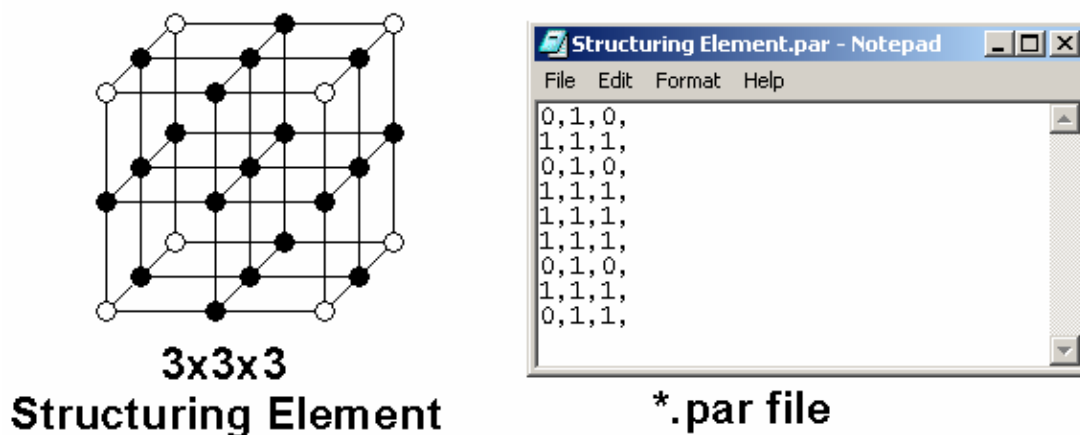


Figure 18. A 3x3x3 structuring element and its implementation on a par file

- **Minkowski addition:** Performs Minkowski addition of a binary video stream with a binary video stream buffer considered as the structuring element. The user selects the input video stream, a *.par file containing the structuring element for the operation and a file to store the output video stream.
- **Minkowski subtraction:** Performs Minkowski subtraction of a binary video stream buffer with another binary video stream buffer considered as the structuring element. The same procedure as in Minkowski addition is followed.
- **Dilation:** Performs dilation of a binary video stream buffer with another binary video stream buffer considered as the structuring element. The same procedure as in Minkowski addition is followed.
- **Erosion:** Performs erosion of a binary video stream buffer with another binary video stream buffer considered as the structuring element. The same procedure as in Minkowski addition is followed.
- **Opening:** Performs opening of a binary video stream buffer with another binary video stream buffer considered as the structuring element. The same procedure as in Minkowski addition is followed.
- **Closing:** Performs closing of a binary video stream buffer with another binary video stream buffer considered as the structuring element. The same procedure as in Minkowski addition is followed.

4.3.2 Help menu

This sub-menu provides the About... item which displays information about the program.