

PIKS Scientific
Image Processing Software
PIKSTool Manual
Version 3.2.1

Gerard A. Paquette

Table of Contents

1. Introduction.....	1
2. Basic PIKSTool	2
3. PIKSTool Image Display.....	3
4. PIKSTool Image Processing Usage Examples.....	4
4.1. Unsharp Mask Example.....	4
4.2. Histogram Equalization Example	6
4.3. Outlier Noise Removal Example	8
4.4. Noise Reduction by Filtering Example.....	11
4.5. Linear Edge Crisping on a Color Image Example	13
5. Additional Information	15
Appendix A. Glossary.....	16
Appendix B. PIKS Scientific Functions Accessible from PIKSTool Menus	17
Appendix C. PIKS Object Files	67
Appendix D. PIKS Scientific Elements Utilized Internally in PIKSTool	74
Appendix E. PIKS Scientific Elements Not Included in PIKSTool Version 3.2	75

1. Introduction

PIKSTool, version 3.2.1, accommodates the image processing elements that provide the 3-dimensional Euclidean space, time and spectra functionality of the ISO Programmer's Imaging Kernel System (PIKS) standard's Scientific Profile. The PIKS Application Programmer's Interface (API) derived from that standard contains the image processing operators, mechanisms, tools and utilities applicable for image processing problem solving. However, the PIKS API must be supported with extensive and cumbersome interconnecting software to achieve the desired results. PIKSTool offers an easy-to-use graphical user interface to the PIKS API that obviates the need for writing and building a program to solve image processing problems. PIKSTool internally provides these interconnections through easily accessed menus, pull down lists and dialogs presented on the computer display screen. Five examples are presented in Appendix A that demonstrate some image processing problem solutions possible by applying PIKSTool. These examples may be executed as training exercises with the PIKSTool downloadable from the pixelsoft.com website.

It is assumed that the reader of this document is acquainted with the terminology appearing in the PIKS standard.

Most of the PIKS elements are directly callable from the computer display but some are utilized internally in support of the image processing needs. PIKSTool supports successive processing of images through user-specified sequences of the variously identified functions (operators, mechanisms, tools, utilities and settings). Optional file storage and retrieval is initiated or accessed and utilized to save and restore any objects required or processed by PIKSTool.

The current version, PIKSTool 3.2.1, does not support composite identifier elements, index assignment, match point processing, asynchronous operation, event chaining or region-of-interest (ROI) processing (note that ROI control over image regions is supported but the manipulation of ROIs by specific operators, i.e., ROI processing, is not supported). Resampling is limited to nearest neighbour, 2 bilinear and 4 cubic convolution for image and ROI resampling in x and y dimensions and nearest neighbour resampling in z, t and b dimensions.

Non-standard extension tools and settings are also provided: *create_array* (numeric collection, neighborhood array or pixel values), *create_lookup_table*, *create_matrix*, cursor-identified *coordinate_display* and index-specified image plane *display_control*. 5D images are processed but only MON and 3-banded COLR images are displayed as such. VOL, TEM, SPE, VT, VS, TS, VTS and 4-banded color images are displayed as the image planes at z, t, b coordinates specified in Display Control and VC, TC and VTC are displayed as the 3D color images at z, t coordinates specified in Display Control. All displayed images, except those of the *image_constant* tool, are optimized over the displayable amplitudes of that image.

2. PIKSTool Operation

When initialized, PIKSTool provides a descriptive message box that, when clicked OK, will lead to an initial PIKSTool dialog box. The options provided by that dialog are File, Operators, Support, Settings and Help.

- Selecting the File option provides the following pull-down options:
 - Open Input Image – replaces any existing source image or ROI with the image whose header file is selected from the open dialog that follows. Note that operators do not require but accept any source image derived from execution of this option.
 - Save Destination Image – opens the save header dialog from which a destination image can be saved to file.
 - Open Input ROI – replaces any existing source image or ROI with the ROI selected from the open dialog that follows. Note that operators do not require but accept any source ROI derived from execution of this option.
 - Save Destination ROI – opens the save ROI dialog from which a destination ROI can be saved to file.
 - Exit is for the appropriate termination of PIKSTool.
- Selecting the Operators, Support or Settings options will provide pull-down lists of PIKSTool image processing functions described in more detail in the corresponding sections below.
- Selecting the Help option provides the identification of this version of PIKSTool under the subheading of “About PIKSTool...”.

Operators generally perform their designated process upon source images or ROIs. However the source image may be taken from file or be the existing destination image if one exists as well as be the existing source image. Most operators result in a destination image but some result in data presented in the System Messages. Some operators also result in a dialog from which the appropriate type of destination object can be specified for further object saving.

Regions of interest, ROI objects, can designate portions of images to be operated upon. ROI objects can be bound to source and/or destination images with the *bind_roi* mechanism, however, to make them effective, ROI control must be enabled by the *set_globals* mechanism. There are numerous tools available for creating ROI objects.

Support functions generally result in destination objects that are described in the System Messages and may be saved as the appropriately configured object files. Objects generated by these functions will be available for file storage according to the object type.

Settings allow the user to configure his usage of PIKSTool such as designation of the image or ROI plane to be displayed or enable the display of cursor-specified horizontal and vertical coordinates and contents.

3. PIKSTool Image Display

PIKSTool accesses disc-stored PIKS-formatted images as needed for the operations selected. Such images consist of a .header numeric file that describes the image and a .data information stream file (see Appendix D for details). Most PIKS operations require one source image that, after acquisition, will be displayed in the upper left of the display screen but some require a second source image that, after acquisition, will be displayed in the lower left of the display screen. The first and second destination images, if the latter results from the operation, are displayed to the right of the screen, first above second. When an image is complex, the imaginary portions are displayed in the same locations as would be occupied by second images.

Upon initiation of PIKSTool, the first source image (Src1) may be opened or it will be requested to satisfy the requirements of the first selected image processing operator. ROI operations function similarly. A second source image (Src2) will be requested only to satisfy the requirements of an image processing operation that needs two source images. The destination image(s) (Dst1 and Dst2) are defined and displayed after the first image processing operation completes successfully. All images are also described in the System Messages as they appear.

Multidimensional ROIs and monochrome images will be displayed as planes of that image as directed by the Display Control setting (see Appendix C, Settings). Three-banded colour images will be displayed as colour images at depth and time dimensions set with the Display Control. Except for images obtained from the *image_constant* utility, the displayed image will be optimized to the full amplitude display capability of the monitor.

One source and one destination image and one source and one destination ROI may coexist if and when created. Source and destination ROIs are displayed in corresponding source image or destination image space but ROIs extend a short distance below the corresponding image position. The most recently created ROI or image is predominantly displayed in the source or destination object space. When source or destination images and ROIs are simultaneously available, the user may select the alternate image or ROI by clicking on the underlapping portion of the object desired.

4. PIKSTool Image Processing Usage Examples

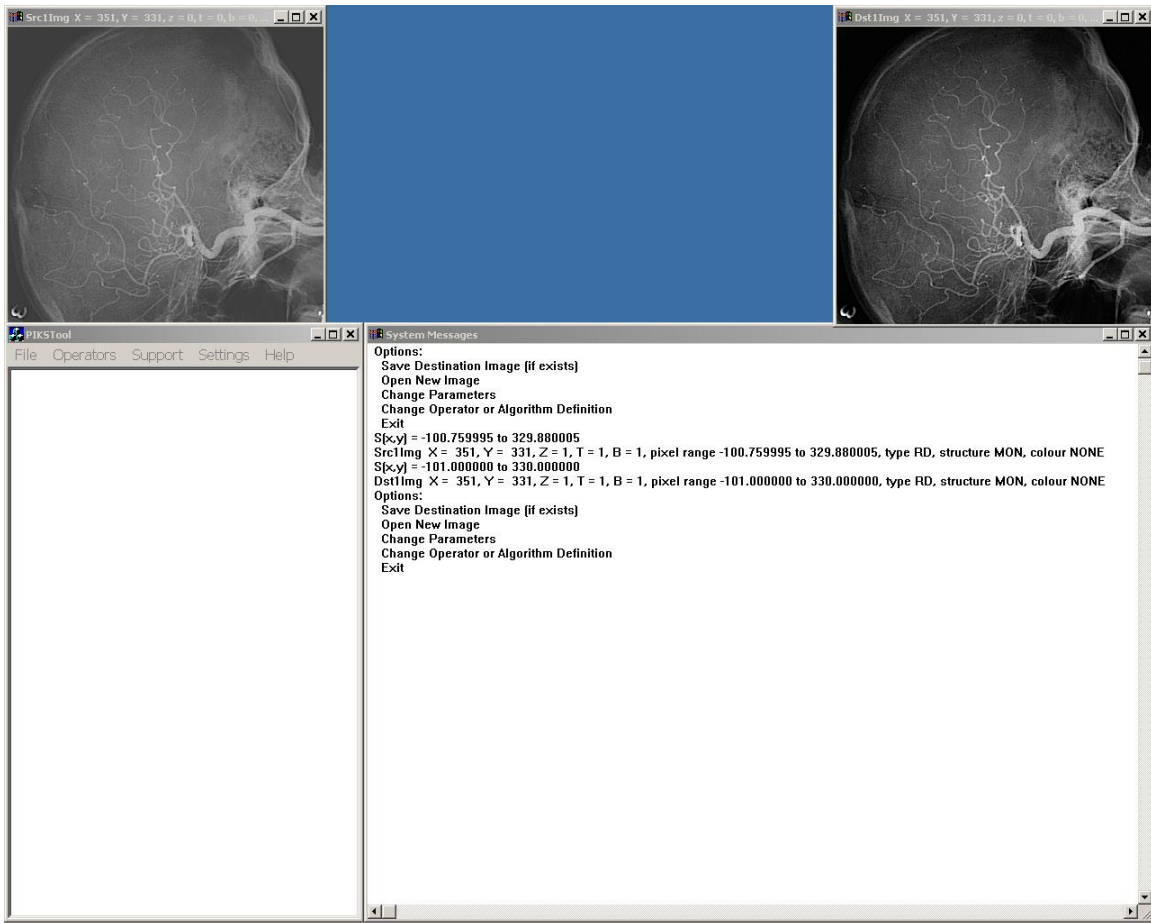
Five examples are presented to demonstrate the ease of problem solution without writing and compiling a program. Example solutions are available through the application of the PIKSTool graphical user interface. These examples also serve to demonstrate typical operations of PIKSTool as learning exercises.

All of the examples assume the user has initiated execution of PIKSTool and clicked OK on the introductory window. The PIKSTool dialog window will subsequently appear in the lower left of the display screen and will serve as the starting point for all of the example scenarios in this section.

To leave PIKSTool operation after any of the examples, click File in the PIKSTool dialog window and click Exit.

4.1. Unsharp Mask Example

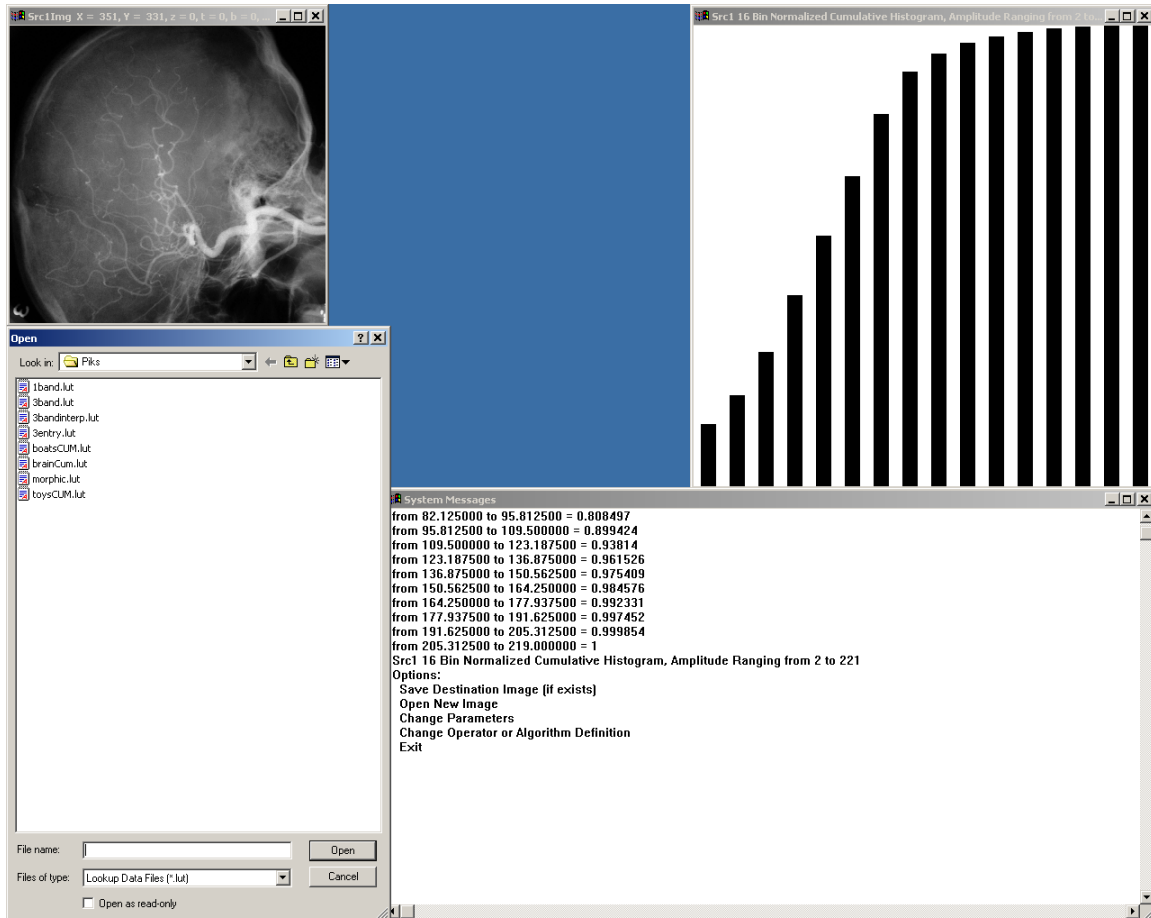
1. Click File, click “Open Input Image” and double click the “brain.header” image name.
2. Click OK the “Select an Operation” message.
3. Click Operators, select Enhancement and click *unsharp_mask*.
4. In the “Unsharp Mask” dialog, click “From Repository”, leave the “First weight factor”, “Second weight factor” and “Bias factor” unchanged at default values (3, -2, 0, respectively) and click OK.
5. In the “Repository Impulse Response Kernels” pull down window, click on the arrow to the right of the view window and scroll to the “088 UNIFORM 5 x 5”. Click on that object and click OK.
6. Click Operators, select Point and click *window_level*.
7. In the “Window Level” dialog, enter the Above Window and Below Window Clip Values as the higher and lower pixel range values shown for the Dst1Img in the System Messages. Leave the Width and Level default values unchanged (255 and 127) and click OK.
8. In the “Select Source Image” dialog, leave the default of “Destination, if exists” unchanged and click OK.



Unsharp Mask Example Screen Display After Step 8

4.2. Histogram Equalization Example

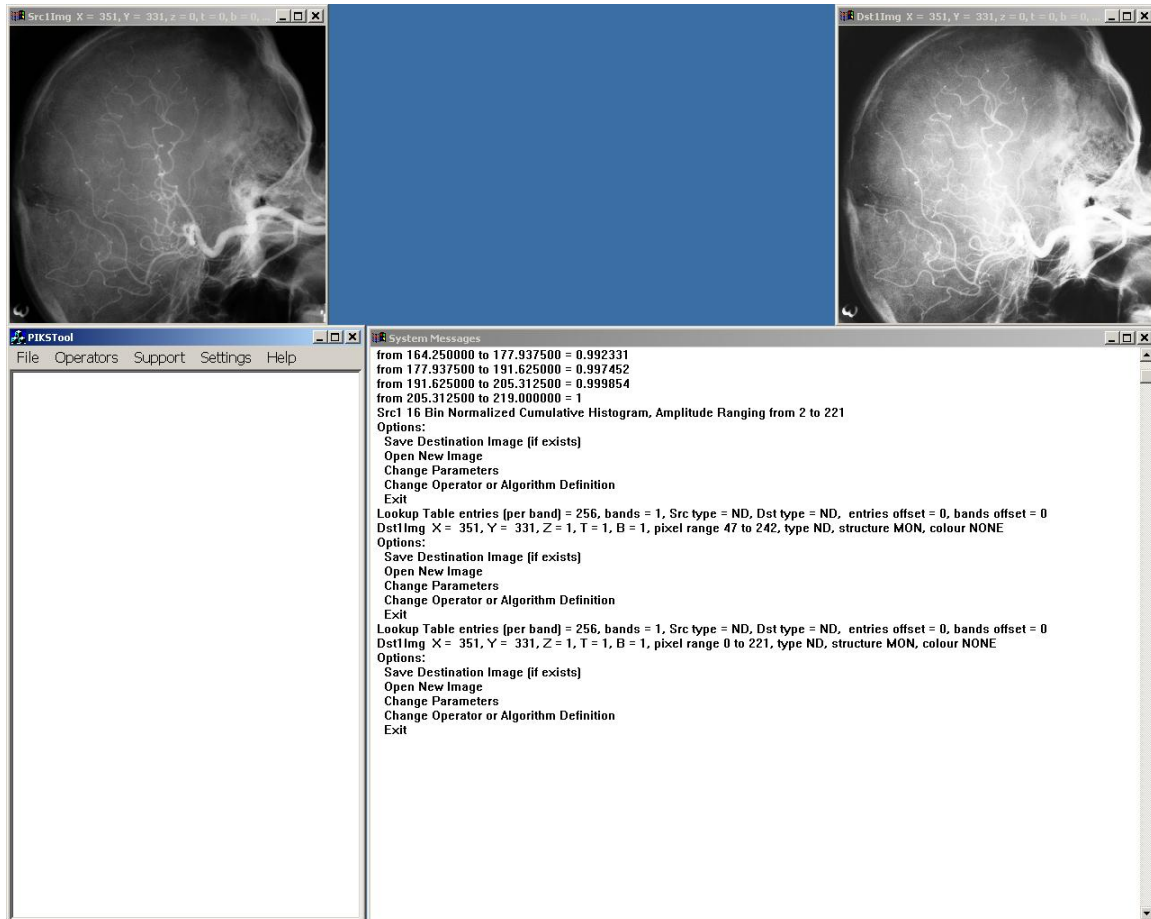
1. Click File, click “Open Input Image” and double click on the “brain.header” image name.
2. Click OK on the “Select an Operation” message.
3. Click Operators, select Analysis and click *histogram_1d*.
4. In the “Histogram 1d” dialog, leave the “Number of Histogram Bins (NP)” unchanged as the default value 16, click “Normalized Cumulative LUT (ND and SD images only)” and click OK.



Histogram Equalization Example Screen Display After Step 4

5. In the “Open” dialog now appearing, select or enter a name to be used for saving the lookup table and click Open.
6. Click Operators, select Point and click *lookup*.
7. In the “Lookup” dialog click 1D and OK.
8. In the “Open” dialog now appearing, double click on the name used to save the lookup table in step 5.

9. In the “Select Source Image” dialog, click “Source, if exists” and click OK.



Histogram Equalization Example Screen Display After Step 9

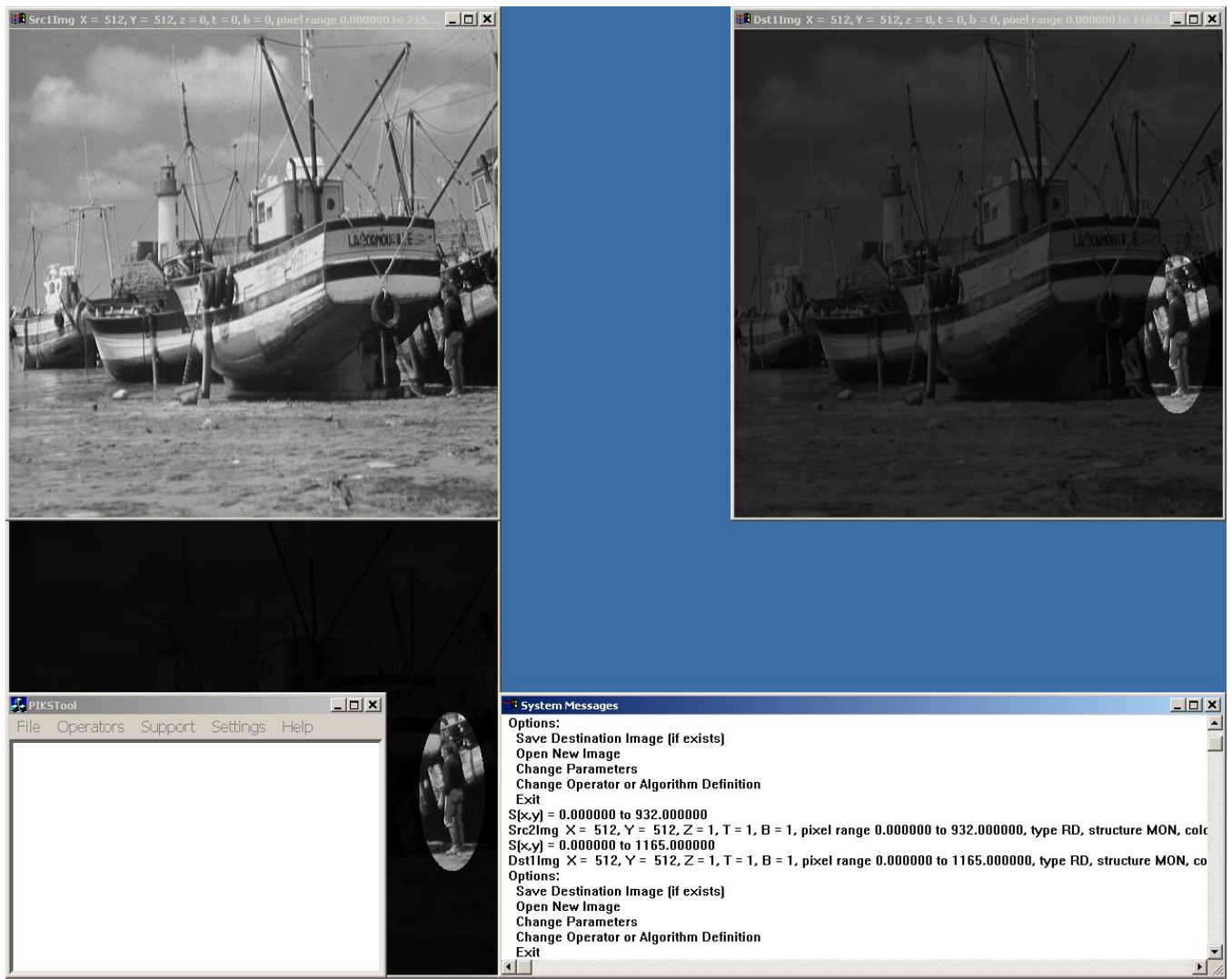
4.3 Highlighting Example

1. Click File, click “Open Input Image” and double click on “boats.header”.
2. Click OK on the “Select an Operation” message.

Note: The user may choose to skip to step 9 and use the suggested ellipse values.

3. Click Settings and click on “Coordinate Display”.
4. In the “Image/ROI Coordinates Access Mode” dialog, check the option therein and click “Accept”.
5. Move the cursor to some space to the left of the man standing behind the foreground boat and click the right mouse button.
6. Move the cursor to some space to the right of the man and click the right mouse button.
7. Move the cursor to some space above the man and click the right mouse button.
8. Move the cursor to some space below the man and click the right mouse button.
9. Click Support, select Utilities and click on *convert_image_datatype*.
10. In the “Convert Image Datatype” dialog, click “Float Real (RD)” and OK.
11. Click Support, select Tools and click on *roi_elliptical*.
12. In the “ROI Elliptical” dialog leave the “Overall Dimensions (NP)” unchanged (512, 512, 1, 1, 1), change the “First Index” “Centre” to a value halfway between the first two “Mouse coordinate” “x” values reported in the System Messages (approximately 465) and “Length” to the difference of the first two “x” values (approximately 70), change “Second Index” “Centre” to a value halfway between the third and fourth “Mouse coordinate” “y” values reported in the System Messages (approximately 320) and “Length” to the difference of the third and fourth “y” values (approximately 170), leave the remaining “Ellipse Dimensions (NP)” unchanged (all 0), click “FALSE” in “Internal Polarity”, click “2D” in “Mode”, leave the “Indices” unchanged (x, y, z, t, b) and click OK.
13. Click Support, select Mechanisms and click on *bind_roi*.
14. In the “Bind ROI” dialog, leave the default offsets unchanged (0, 0, 0, 0, 0) and click OK.
15. In the “Select Source ROI” dialog, leave the default of “Destination, if exists” unchanged and click OK.
16. In the “ROI Binding” dialog, click on “Destination” and OK.
17. Click Operators, select Point and click on *monadic_arithmetic*.

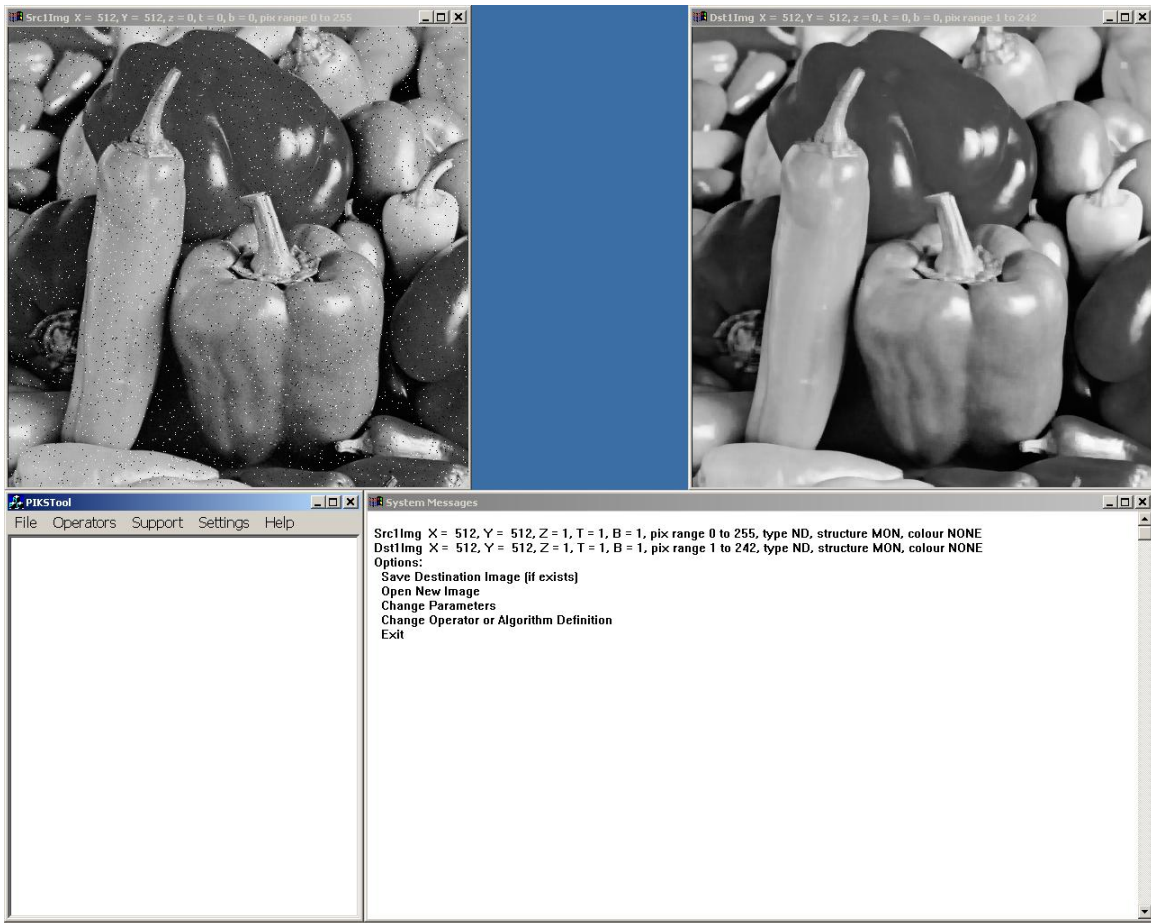
18. In the “Monadic Arithmetic Parameters” dialog, enter 0.25 as the “Value of Constant”, click “Multiplication by Constant” and OK
19. In the “Select Source Image” dialog, leave the “Destination, if exists” default unchanged and click OK.
20. Click Support, select Mechanisms and click on *set_globals*.
21. In the “Set Globals” dialog “ROI Control” option, click “Enabled” and OK.
22. Click Operators, select Point and click on *monadic_arithmetic*.
23. In the “Monadic Arithmetic Parameters” dialog, enter 4 as the “Value of Constant”, click “Multiplication by Constant” and OK.
24. In the “Select Source Image” dialog, click the “Source, if exists” option and OK.
25. Click Support, select Mechanisms and click on *set_globals*.
26. In the “Set Globals” dialog “ROI Control” option, click “Disabled” and OK.
27. Click Operators, select Ensemble and click on *dyadic_arithmetic*.
28. In the “Dyadic Arithmetic” dialog, click “Addition” and OK.
29. In the “Select Images to be Used” dialog, leave the defaults unchanged (“Second Source, (if required)” as “Existing Destination”, “Source” as “Existing Source” and “Destination” as “New as f(Source)”).



Highlighting Example After Step 29

4.4. Noise Reduction by Filtering Example

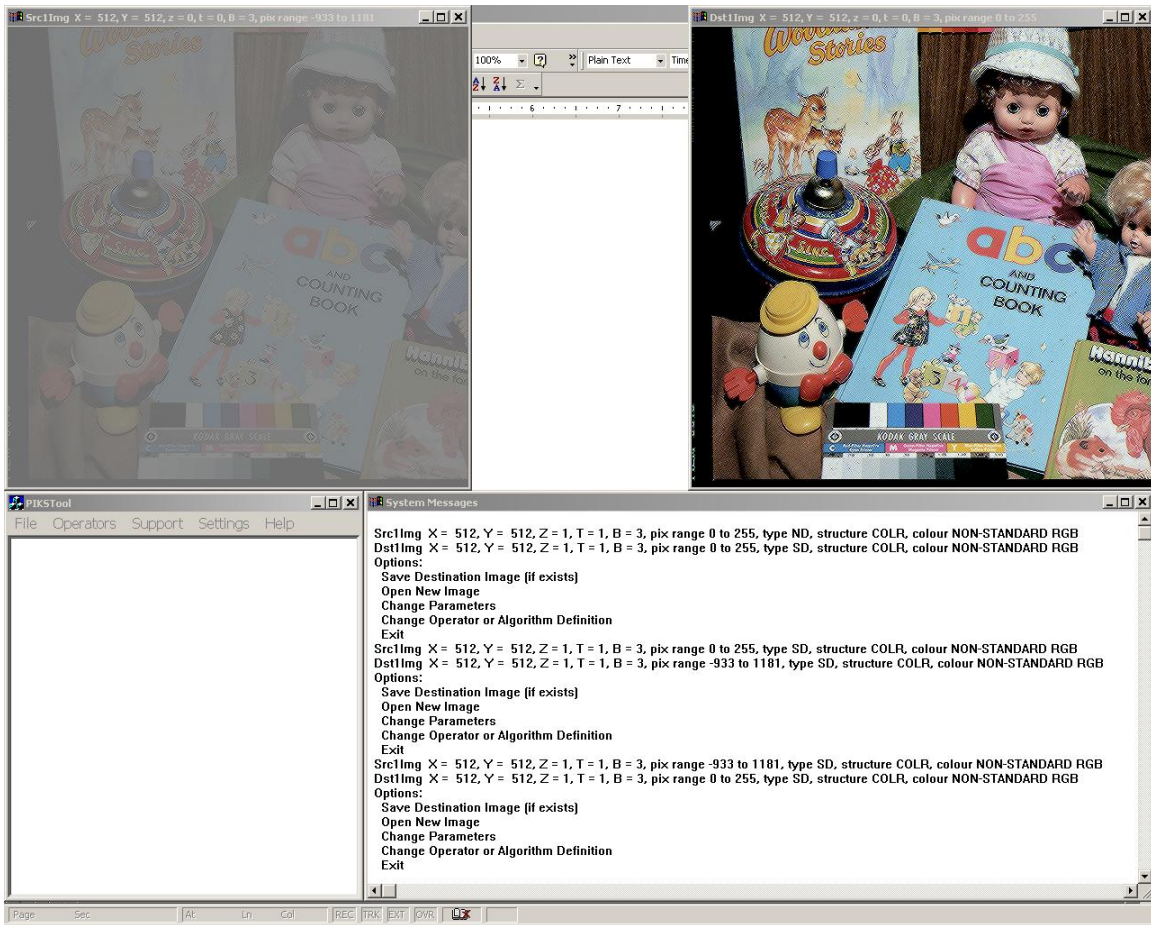
1. Click Support, select Tools, and click *create_array*.
2. In the “Select Array Type” dialog, click “Neighbourhood” and OK.
3. On the “Neighbourhood Array” dialog, click “BD”, click “Mask”, enter the “Size (NP)” “J” as 7 and the “Size (NP)” “K” as 7 and enter the “Key Position (SP)” “J” as 3 and the “Key Position (SP)” “K” as 3 and click OK.
4. In the “Neighbourhood Array Data” dialog, enter 1 for (Column, Row) indices of 3, 0; 3, 1; 3, 2, 0, 3; 1, 3; 2, 3; 3, 3; 4, 3; 5, 3; 6, 3; 3, 4; 3, 5 and 3, 6 and enter 0 for all other coordinates.
5. After all coordinates of the mask are entered, a dialog will appear to allow saving the mask. Enter the name of the mask, i.e., “7x7Cross” and click Save.
6. Click File, click “Open Input Image” and double click on “peppers_replacement_noise.header”.
7. Click Operators, select Filtering and click *filtering_median*.
8. In the “Filtering Median” dialog, click Key Pixel and OK.
9. In the subsequent request for “Mask Structuring Element File”, click the 7x7Cross.msk.



Noise Removal by Filtering Example Screen Display After Step 10

4.5. Linear Edge Crisping on a Color Image Example

1. Click File, click “Open Input Image” and double click on “toys.header”.
2. If the toys source image is now displayed, proceed to step 5.
3. In the “Image Structure Query” dialog, click “Colour Structure” and OK.
4. In the “Colour Image Parameters” dialog, click “non-standard RGB” and OK.
5. Click OK on the “Select an Operation” message.
6. Click Support, select Utilities and click *convert_image_datatype*.
7. In the “Convert Image Datatype” dialog, click “Signed Integer (SD)” and OK.
8. Click Operators, select Filtering and click *convolve_2d*.
9. In the Convolve_2D dialog, click the “From Repository” and “Enclosed Array” options and click OK.
10. In the “Repository Impulse Response Kernels” pull down, click on the arrow to the right of the view window and scroll to “010_High_Pass_3_3X3”. Click on that object and click OK.
11. In the “Select Source Image” dialog, leave the destination default unchanged and click OK.
12. Click Operators, select Point and click *window_level*.
13. In the Window Level dialog, enter 255 as the “Above Window” Clip Value, leave the default “Below Window” Clip Value as 0, leave the default Width and Level values unchanged (255, 127) and click OK.
14. In the “Select Source Image” dialog, leave the destination default unchanged and click OK.



Linear Edge Crisping Example Screen Display After Step 11

5. Additional Information

A glossary of terminology used in PIKSTool is provided in Appendix A.

A list of functional elements supported in this version of PIKSTool is presented in Appendix B.

Objects maintained as stored files with this version of PIKSTool are described in Appendix C.

PIKS Scientific elements inherently included but not menu-accessible within PIKSTool are listed in Appendix D.

PIKS Scientific elements not included within PIKSTool, version 3.2.1, are listed in Appendix E.

Appendix A. Glossary

band – spectral dimension of object display or file

centre – horizontal, vertical, depth, temporal or spectral centre of an ellipse ROI

coordinate – geometric position within object

data type external – BI = bit (images) or unsigned char, BP = TRUE/FALSE unsigned char, NP = long integer, SP = short integer, RP = float, CP = complex (real and imaginary) float

data type internal – BD = bit, ND = unsigned char, SD = short integer, RD = float, CD = complex float

depth – depth dimension of object display or file

destination – image or ROI displayed as Dst1 (image), Dst2 (image) or DstROI (ROI).

height – vertical dimension of object display or file

length – horizontal, vertical, depth, temporal or spectral length of an ellipse ROI

object – any structured PIKS data element that may contain from one to five dimensional representation

offset – zero coordinate-relative position within an object

polarity – background or pixel value of a ROI

ROI – region of interest

ROI control – ROI bound to an image will block that image where the ROI coordinate values are FALSE

ROI process – operators so designated will perform their functions upon bound ROIs as well as images themselves.

source – image or ROI displayed as Src1 (image), Src2 (image), Src1ROI (ROI) or Src2ROI (ROI).

time – temporal dimension of object display or file.

width – horizontal dimension of object display or file

Appendix B. PIKS Scientific Functions Accessible from PIKSTool Menus

OPERATORS:

3d_slice – Creates a monochrome image from a three-dimensional slice of a volume, temporal or spectral image.

Accessed from the Operators – 3D Specific option

The source image may have must have VOL, TEM or SPE structure and ND, SD or RD data type.

Dialog requests the horizontal, vertical and third dimension offsets, destination image sizes and offsets and the rotation angles.

The designated source image is processed to the displayed destination image of MON structure and RD data type.

accumulator – Accumulates amplitudes of pixels of an image.

Accessed from the Operators – Analysis option

The source image may have any structure and BD, ND, SD or RD data type.

Dialog requests the accumulation mode

The results are displayed in the System Messages.

adaptive_histogram_equalization – Performs image contrast stretching of each pixel plane by a spatially adaptive point transformation that is the cumulative histogram within a moving window over the pixel plane.

Accessed from the Operators – Enhancement option

The source image may have any structure and ND or SD data type.

Dialog requests the number of row and column histogram reference points and the window size

The source image is processed to the displayed destination image of the same structure and data type.

alpha_blend_constant – Performs an alpha blending combination of a pair of images using a constant alpha blend factor.

Accessed from the Operators – Ensemble option

The two source images may have any but the same structure and ND, SD or RD and same data type are utilized or requested.

Dialog requests the blend constant.

The source images are processed to the displayed destination image of the same structure and data type.

alpha_blend_variable – Performs an alpha blending combination of a pair of images using variable alpha blend factors.

Accessed from the Operators – Ensemble option

The two source images may have any but same structure and ND, SD or RD and same data type are utilized or requested. This is followed by requests for two alpha blend factor images.

No dialog

The source images are processed to the displayed destination image of the same structure and data type.

bit_shift – Performs bit shifting of an integer data type pixel image.

Accessed from the Operators – Point option

The source image may have any structure and ND or SD data type.

Dialog requests the number of bits to shift and the shift option

The designated source image is processed to the displayed destination image of the same structure and data type.

cartesian_to_polar – Performs conversion of the planes of an image in Cartesian coordinates to planes of an image in polar coordinates.

Accessed from the Operators – Geometric option

The source image may have any structure and BD, ND, SD or RD data type.

Dialog requests *rho_min*, *rho_max*, *theta_min* and *theta_max* parameters.

The designated source image is processed to the displayed destination image of the same structure and data type.

classifier_bayes – Performs Bayes quadratic Gaussian classification of a spectral image.

Accessed from the Operators – Classification option

The source image must be SPE structure and ND, SD or RD data type.

Dialog requests the number of classes followed by dialogs requesting the probability array, the mean vector array and the covariance matrix array.

The designated source image is processed to the displayed destination image of MON structure and SD data type.

classifier_nearest_neighbour – Performs nearest neighbour classification of a spectral image.

Accessed from the Operators – Classification option

The source image must be SPE structure and ND, SD or RD data type.

Dialog requests the number of classes and the number of patterns per class followed by dialog requesting the training pattern array.

The designated source image is processed to the displayed destination image of MON structure and SD data type.

colour_conversion_linear – Performs linear inter-band colour conversion of a trichromatic colour image between colour spaces.

Accessed from the Operators – Colour option

The source image must be COLR, VC, TC or VTC structure and SD or RD data type.

Dialog requests the destination type and the matrix source. If an internal matrix category is selected, only certain source to destination conversions are supported.

1. Colour conversions within “video” colour spaces, e.g., RGB to and from YIQ, assume gamma corrected (point nonlinear) RGB components.
2. Colour conversions within “colourimetry” colour spaces and between video and colourimetry colour spaces, e.g., CIE RGB to and from XYZ, assume point linear RGB components.
3. The *colour_conversion_linear* operator does not perform point transformations that gamma correct linear components or that linearize gamma correct components.
4. When source and designated images possess colour options supported by this operator, the appropriate conversion matrix can be called from the repository for performing the operation.

The designated source image is processed to the displayed destination image of the same structure and data type.

colour_conversion_nonlinear – Performs nonlinear intra-band colour conversion of a trichromatic colour image between RGB and nonlinear colour spaces.

Accessed from the Operators – Colour option

The source image must be COLR, VC, TC or VTC structure and RD data type but with limited colour options according to the conversion selected.

1. The conversion between RGB and IHS colour spaces is only defined for uncalibrated RGB (non-standard RGB).
2. The transformation between RGB and Yxy, Yuv, L*a*b* or L*u*v* is only defined for calibrated point linear RGB.

Dialog requests the conversion option. If the conversion involves L*a*b* or L*u*v*, a subsequent dialog requests the tristimulus values. If the conversion is to calibrated linear RGB, a subsequent dialog requests assignment of the colour option.

The designated source image is processed to the displayed destination image of the same structure and data type.

colour_conversion_subtractive – Performs a conversion of a colour image between additive uncalibrated RGB and subtractive uncalibrated CMY or CMYK colour spaces.

Accessed from the Operators – Colour option

The source image must be COLR, VC, TC or VTC structure and ND, SD or RD data type.

Dialog requests the conversion option, blackness factor (RGB to CMYK option only) and maximum integer pixel value. If converting from CMY or CMYK, the colour parameters will be requested.

The designated source image is processed to the displayed destination image of the same structure and data type.

colour_lookup_interpolated – Performs interpolated lookup-table manipulation of a colour image.

Accessed from the Operators – Colour option

The source image must be COLR, VC, TC or VTC structure and ND, SD or RD data type.

Dialog requests the number of bands to appear in the destination image, 3 or 4.

The designated source image is processed to the displayed destination image of generic structure, same colour and same data type.

complement – Performs logical complement of a Boolean or integer image.

Accessed from the Operators – Point option

The source image may have any structure and BD, ND or SD data type.

No dialog

The designated source image is complemented to the displayed destination image of the same structure and data type.

complement_roi – Performs logical complement of a ROI.

Accessed from the Operators – Point option

The source ROI may have any structure

No dialog

The designated source ROI is complemented to the displayed generic destination ROI.

complex_composition – Performs conversion of a pair of real arithmetic data type images containing the real and imaginary or magnitude and phase parts of a complex image into a complex data type image.

Accessed from the Operators – Complex option

The source image pair may have any but same structure and RD data type

Dialog requests the conversion option

The designated source RD images are processed to the displayed CD destination image according to the option selected.

complex_conjugate – Computes complex conjugate of a complex image.

Accessed from the Operators – Complex option

The source image may have any structure and CD data type.

No dialog

The designated source CD image is processed to the displayed CD destination image.

complex_decomposition – Performs conversion of a complex data type image into a pair of real arithmetic data type images containing the real and imaginary or magnitude and phase parts of the complex image.

Accessed from the Operators – Complex option

The source image may have any structure and CD data type.

Dialog requests the conversion option

The designated source CD image is processed to the displayed RD destination images according to the option selected.

complex_magnitude – Computes magnitude or magnitude squared of a complex image.

Accessed from the Operators – Complex option

The source image may have any structure and CD data type.

Dialog requests the computation option

The designated source image is processed to the displayed destination image according to the option selected.

convolve_2d – Performs two-dimensional convolution with a general impulse response function array for several image boundary conditions.

Accessed from the Operators – Filtering option

The source may have any structure and ND, SD or RD data type.

Dialog requests the impulse response array source and the convolution option.

The designated source image is processed to the destination image.

convolve_5d – Performs five-dimensional convolution with a general impulse response function array for an enclosed array image boundary condition.

Accessed from the Operators – Filtering option

The source may have any structure and ND, SD or RD data type.

Dialog requests the source of the impulse response array.

The designated source image is processed to the displayed destination image.

cross_correlation – Computes the cross-correlation function array of a pixel plane of a pair of images.

Accessed from the Operators – Correlation option

The source may have any structure and ND, SD or RD data type.

Dialog requests the third, fourth and fifth indices and the number of columns and rows. Execution is followed by a dialog that permits saving of the .pwr static array

The designated source image is processed to the displayed destination array.

difference_measures – Computes squared difference measures between a pair of images.

Accessed from the Operators – Analysis option

The source and second source image may have any but same structure and SD or RD same data type.

Dialog requests the measure option

No destination image. The result is displayed in the System Messages

diffuse – Performs error diffusion half toning on an image using the Floyd-Steinberg method to reduce the number of amplitude levels per pixel.

Accessed from the Operators – Presentation option

The source image may be single- or three-banded structure and ND data type.

Dialog requests the level to be used at each displayed band. Current PIKS limits 3-banded levels to be between 1 and 6 and single-banded levels to be 64 or less.

The source image is processed to a single banded ND destination image.

dither – Performs ordered half toning on an image to reduce the number of amplitude levels per pixel.

Accessed from the Operators – Presentation option

The source image must be single- or three-banded structure and ND data type.

Dialog requests selection of the dither array followed by a request for the level to be used at each displayed band. Current PIKS limits 3-banded levels to be between 1 and 6.

The source image is processed to a single banded ND destination image.

draw_pixels – Inserts pixels specified by an external collection into an image.

Accessed from the Operators – Modification option

Requires destination image having any structure and any data type

Dialog requests selection of the pixel object

The specified pixels are entered into the displayed destination image.

draw_pixels_roi – Inserts pixels specified by an external collection into a ROI.

Accessed from the Operators – Modification option

Required destination ROI having any structure

Dialog requests selection of the pixel object

The specified pixels are entered into the displayed destination ROI

dyadic_arithmetic – Performs dyadic arithmetic combination of a pair of integer or real images.

Accessed from the Operators – Ensemble option

The source images may have any but same structure and ND, SD or RD and same data type.

Dialog requests selection of the combination option. A second source image of same structure and data type may also be requested if one is not displayed.

The source images are arithmetically combined, pixel-by-pixel, and the resulting combination is displayed as the same structure and data type destination image.

dyadic_complex – Performs dyadic arithmetic combination of a pair of complex images.

Accessed from the Operators – Ensemble option

The two source images having any but same structure and CD data type.

Dialog requests selection of the combination option. A second source image of same structure and data type may also be requested if one is not displayed.

The source images are logically combined, pixel-by-pixel, and the resulting combination is displayed as the same structure and data type destination image.

dyadic_logical – Performs dyadic logical combination of a pair of Boolean or integer images.

Accessed from the Operators – Ensemble option

The two source images having any but same structure and ND, SD or RD same data type

Dialog requests selection of the combination option. A second source image of same structure and data type may also be requested if one is not displayed.

The source images are logically combined, pixel-by-pixel, and the resulting combination is displayed as the same structure and data type destination image.

dyadic_logical_roi – Performs dyadic logical combination of a pair of ROIs.

Accessed from the Operators – Ensemble option

The two source ROIs may have any structure.

Dialog requests selection of the combination option. A second source ROI may also be requested if one is not displayed.

The source ROIs are logically combined, pixel-by-pixel, and the resulting combination is displayed as a generic destination ROI.

dyadic_predicate – Performs dyadic predicate combination of a pair of integer or real images.

Accessed from the Operators – Ensemble option

The two source images having any but same structure and same ND, SD or RD data type

Dialog requests the combination option. A second source image of same structure and data type may also be requested if one is not displayed.

The source image is compared, pixel-by-pixel, with a second source image. Pixels satisfying the comparison are set TRUE (1), FALSE (0) otherwise, in the displayed destination BD image.

edge_detection_orthogonal_gradient – Performs edge detection based upon the formation of the spatial orthogonal gradient of a pixel plane.

Accessed from the Operators – Edge Detection option

The source image may have any structure and ND, SD or RD data type.

Dialog requests sources of 2D horizontal and vertical gradient impulse arrays, threshold values, edge detection option, orientations of horizontal and vertical impulses and destination data type.

The designated source image is processed to the destination image of same structure and designated BD or RD data type.

edge_detection_second_derivative – Performs edge detection based upon the formation of the spatial second derivative of a pixel plane.

Accessed from the Operators – Edge Detection option

The source image may have any structure and ND, SD or RD data type.

Dialog requests source of 2D second derivative impulse array, crossing values, edge detection option and destination data type.

The designated source image is processed to the destination image of same structure and designated BD or RD data type.

edge_detection_template_gradient – Performs edge detection based upon the formation of the template gradient of a pixel plane.

Accessed from the Operators – Edge Detection option

The source image may have any structure and ND, SD or RD data type.

Dialog requests source of 2D template gradient impulse array, the number of arrays, threshold values, edge detection option and destination data type.

The designated source image is processed to the destination image of same structure and designated BD or RD data type.

erosion_dilation_boolean – Performs erosion or dilation on a Boolean image with a rectangular structuring element array.

Accessed from the Operators – Morphological option

The source image may have any structure and BD data type.

Dialog requests the number of iterations to apply, the type of operation and the processing option. This is followed by a request for the structuring element (.msk).

The designated source image is processed to the destination image of same structure and BD data type

erosion_dilation_boolean_roi – Performs erosion or dilation on a ROI with a rectangular structuring element array.

Accessed from the Operators – Morphological option

The source ROI may have any structure.

Dialog requests the number of iterations to apply, the type of operation and the processing option. This is followed by a request for structuring element (.msk).

The designated source ROI is processed to a destination ROI of generic structure

erosion_dilation_grey – Performs erosion or dilation on a grey scale image with a rectangular structuring element array.

Accessed from the Operators – Morphological option

The source image may have any structure and ND, SD or RD data type

Dialog requests the number of iterations to apply, the type of operation and the processing option. This is followed by a request for structuring element (.msk).

The designated source image is processed to the destination image of same structure and data type

error_function_scaling – Performs Gaussian error function amplitude scaling of an image.

Accessed from the Operators – Point option

The source image may have any structure and RD data type.

Dialog requests the Gaussian spread factor.

The designated source image is processed to the displayed destination image with same structure and RD data type.

extrema – Detects the smallest and largest pixels of an image and records their values in an external array.

Accessed from the Operators – Analysis option

The source image may have any structure and ND, SD or RD data type

Dialog requests the mode

No destination image. The designated mode extrema data extracted from the source image is displayed in the System Messages

false_colour – Creates a multispectral or colour image from a multispectral or colour image by a linear transformation across bands.

Accessed from the Operators – Enhancement option

The source image having any spectral (SPE, VS, TS or VTS) or colour structure (COLR, VC, TC or VTC) and ND, SD or RD data type

Dialog requests specification of colour or non-colour destination image.

The designated source image is processed to the displayed destination image with structure as specified and the same data type as the source image.

fill_region – Performs filling of a region, defined by a contour, in a pixel plane of a Boolean image whereby FALSE state pixels within the region are set to the TRUE state and pixels outside the region are left unchanged.

Accessed from the Operators – Morphological option

Source image having any structure and BD data type

Dialog requests first and second index by cursor or by specified position and third, fourth and fifth indices

Designated source image is processed to the displayed destination image of the same structure and BD data type, images are described in the System Messages

fill_region_roi – Performs filling of a region, defined by a contour, in a pixel plane of a ROI whereby FALSE state pixels within the region are set to the TRUE state and pixels outside the region are left unchanged.

Accessed from the Operators – Morphological option

The source ROI may have any structure.

Dialog requests first and second indices by cursor or by specified position and third, fourth and fifth indices

The designated source ROI is processed to the displayed destination ROI.

filtering_homomorphic – Performs two-dimensional homomorphic filtering on an image using Fourier domain processing.

Accessed from the Operators – Filtering option

The source image may have any structure and ND, SD or RD data type.

Dialog requests CD 2D filter transfer function static array.

The designated source image is processed to the displayed destination image of the same structure and RD data type.

filtering_linear – Performs two-dimensional linear filtering on an image using Fourier domain processing.

Accessed from the Operators – Filtering option

The source image may have any structure and ND, SD or RD data type.

Dialog requests CD 2D filter transfer function static array.

The designated source image is processed to the displayed destination image of the same structure and RD data type.

filtering_median – Computes median of pixel planes of an image within a moving window mask.

Accessed from the Operators – Filtering option

The source image may have any structure and ND, SD or RD data type.

Dialog requests the processing option followed by dialog requesting choice of BD 2D mask array.

The designated source image is processed to the displayed destination image of the same structure and data type.

filtering_pseudomedian – Computes pseudomedian of pixel planes of an image within a moving window plus-shaped mask.

Accessed from the Operators – Filtering option

The source image may have any structure and ND, SD or RD data type.

Dialog requests mask width and height.

The designated source image is processed to the displayed destination image of the same structure and data type.

filtering_rank_order – Computes rank order sort value of pixel planes of an image within a moving window mask. Computed pixel is the r-th largest amplitude pixel among pixels in the window mask.

Accessed from the Operators – Filtering option

The source image may have any structure and ND, SD or RD data type.

Dialog requests rank value and processing option followed by a dialog requesting BD 2D mask array selection

The designated source image is processed to the displayed destination image of the same structure and data type.

flip_spin_transpose – Performs flip, spin or transpose manipulations on pixel planes of an image.

Accessed from the Operators – Geometric option

The source image may have any structure and any data type.

Dialog requests selection of the flip, spin, or transpose option.

The designated source image is manipulated to the displayed destination image of the same structure and data type.

flip_spin_transpose_roi – Performs flip, spin or transpose manipulation on pixel planes of a ROI.

Accessed from the Operators – Geometric option

The source ROI may have any structure.

Dialog requests selection of the flip, spin, or transpose option

The designated source ROI is manipulated to the displayed destination ROI of generic structure.

gamma_correction – Applies gamma correction or inverse gamma correction to an image.

Accessed from the Operators – Point option

The source image may have any structure and ND, SD or RD data type.

Dialog requests the four constant parameters, the breakpoint and the correction option.

The designated source image is processed to the same structure and RD destination image display.

histogram_1d – Generates a one-dimensional histogram of an image.

Accessed from the Operators – Analysis option.

The source image may have any structure and BD, SD or RD data type.

Dialog requests output object type as either histogram or Normalized Cumulative LUT.

- The histogram may be required to have specified bounds between the values given or be derived within the extrema of the specified image. This operation is followed by a dialog to optionally save the resulting .hst file.
- The normalized cumulative histogram is displayed according to the number of bins specified but the LUT will range over the data type of the image specified. This option is followed by dialog to optionally save the resulting .lut file.

The source image is processed to an RD histogram or lookup table object as directed by the choice of histogram option.

histogram_2d – Generates a two-dimensional histogram (co-occurrence matrix) of a monochrome image.

Accessed from the Operators – Analysis option

The source image must be MON structure and ND, SD or RD data type.

Dialog requests the number of array bins, the minimum and maximum amplitudes and the horizontal and vertical offsets. A static array storage dialog request follows.

The source image is transformed to an RD static array that is displayed and optionally saved as an .hst2d file designated in the storage dialog.

histogram_modification – Performs image contrast stretching of each pixel in a pixel plane by a nonlinear point transformation using a lookup table transformation. The transformation is chosen such that the histogram of the resultant pixel plane has a desired mathematical form.

Accessed from the Operators – Enhancement option

The source image may have any structure and ND, SD or RD data type.

Dialog requests the lower and upper output values, exponential spread factor, lookup table offset and histogram option.

The source image is processed to a displayed destination RD image.

histogram_scaling – Performs window-level amplitude scaling of an image in which a specified fraction of high amplitude and low amplitude pixels are clipped to specified values and intermediate value pixels are linearly rescaled in amplitude.

Accessed from the Operators – Point option

The source image may have any structure and ND, SD or RD data type.

Dialog requests the lower and upper output values, lower and upper clip fractions and the number of histogram bins.

The source image is processed to a displayed destination image of the same data type.

histogram_shape_1d – Computes shape descriptors of a one-dimensional histogram.

Accessed from the Operators – Shape option

The source 1D histogram is to be RD data type.

Dialog appears for retrieval of the histogram (.hst) file.

Source histogram 1D (.hst) file is utilized in calculating the *histogram_shape_1d* values displayed in the System Messages.

histogram_shape_2d – Computes shape descriptors of a two-dimensional histogram (co-variance matrix).

Accessed from the Operators – Shape option

Source 2D histogram static array is to be RD data type.

Dialog appears for retrieval of the histogram (.hst2d) file.

Source histogram 2D (.hst2d) file is utilized in calculating the *histogram_shape_2d* values displayed in the System Messages.

hit_or_miss – Performs morphological operations on a Boolean image using a hit or miss transformation over a 3x3 pixel window.

Accessed from the Operators – Morphological option

The source image may have any structure and BD data type

Dialog requests the number of iterations and the transformation option.

The source image is transformed to the displayed destination image. The number of cycle executed is described in the System Messages.

hit_or_miss_roi – Performs morphological operations on a ROI using a hit or miss transformation over a 3x3 pixel window.

Accessed from the Operators – Morphological option

The source ROI may have any structure.

Dialog requests the number of iterations and the transformation option.

The source ROI is transformed to the displayed ROI. The number of cycles executed is described in the System Messages.

hough_transform – Computes the Duda and Hart version of the Hough transform of a Boolean image.

Accessed from the Operators – Analysis option

The source image must have MON structure and BD data type.

Dialog requests the number of array columns and rows. A static array storage dialog request follows.

The source image is transformed to an SD static array that is displayed and optionally saved as designated in the storage dialog.

label_objects – Labels each pixel of each object in a pixel plane of a Boolean image with its object index.

Accessed from the Operators – Feature Extraction option

The source image may have any structure and BD data type, only one plane can be processed for each execution

Dialog requests the indices of the third, fourth and fifth dimensions and the maximum index, not to exceed 255

The designated source image is processed to the displayed destination image of same structure but of ND data type. The resulting label count is reported in the System Messages.

laws_texture_features – Computes Laws texture features of an image.

Accessed from the Operators – Feature Extraction option

The source image may have any structure and ND, SD or RD data type.

Dialog requests low pass filter from file or repository and texture mask from file or repository.

The designated source image is processed to displayed destination image of same structure but of RD data type.

level_slice – Performs level slicing of an image in which destination pixels are set to a specified value if the corresponding source pixels lie within specified limits.

Accessed from the Operators – Point menu option

The source image having any structure and ND, SD or RD data type.

Dialog requests upper and lower amplitude slice values, slice output value and output background option.

The designated source image is processed to the displayed destination image of same structure and type.

line_profile – Computes amplitudes of pixels along a line passing through an image.

Accessed from the Operators – Analysis menu option.

The source image may be existing source or existing destination image.

Dialog requests the choice of line dimensions by cursor applied to the desired image or by specified start/end dimensions and the number of samples along the line.

The designated line is displayed on the selected source image, the extracted pixels are listed in the System Messages and displayed as a histogram.

lookup – Performs lookup table manipulation of an image.

Accessed from the Operators – Point menu option

Source image having any structure and BD or SD data type

Dialog requests 1D or 2D option. With 1D, a single banded lookup table can be applied to each depth, time and band plane of an image. With 2D, a multidimensional lookup table can be applied to a single banded image to produce a three-banded colour image.

The designated source image is processed to the displayed destination image of option-directed structure and same data type as the lookup table output data type.

lookup_interpolated – Performs interpolated lookup table manipulation of an image.

Accessed from the Operators – Point menu option

The source image may have any structure and ND or SD data type.

Dialog requests the number of source offset bits followed by dialog that requests the .lut lookup table of ND, SD or RD output data type.

The designated source image is processed to the displayed destination image of same data type as the lookup table output data type.

luminance_generation – Generates a luminance image from a tri-chromatic colour image.

Accessed from the Operators – Colour option

The source image must have COLR, VC, TC or VTC structure and ND, SD or RD data type.

Dialog requests the colour option.

The designated source image is processed to the displayed destination image of MON, VOL, TEM, or VT relative structure and the same data type as the source image.

moments – Computes the mean and standard deviation moments of an image.

Accessed from the Operators – Analysis option

The source image may have any structure and ND, SD or RD data type

No dialog

The mean and standard deviation values are displayed in the System Messages.

monadic_arithmetic – Performs monadic arithmetic combination of an integer or real image and an integer or real constant.

Accessed from the Operators – Point option

The source image may have any structure and ND, SD or RD data type.

Dialog requests the value of the constant and the combination option.

The designated source image is processed to the displayed destination image of the same structure and data type.

monadic_complex – Performs monadic arithmetic combination of a complex image and a complex constant.

Accessed from the Operators – Point option

The source image may have any structure and CD data type

Dialog requests the real and imaginary values of the constant and the combination option.

The designated source image is processed to the displayed destination image of the same structure and data type.

monadic_logical – Performs monadic logical combination of a Boolean or integer image and a constant.

Accessed from the Operators – Point option

The source image may have any structure and BD, ND or SD data type.

Dialog requests the value of the constant and the combination option.

The designated source image is processed to the displayed destination image of the same structure and data type.

monadic_logical_roi – Performs monadic logical intersection or union combination of ROI and a Boolean constant.

Accessed from the Operators – Point option

The source ROI having any structure

Dialog requests the value of the constant (0 = FALSE, any other = TRUE) and the combination option (intersection or union).

The designated source ROI is processed to the displayed destination ROI of generic structure.

morphic_processor – Performs morphological processing on a Boolean image using a 3x3 hit or miss transformation with a specified transformation table.

Accessed from the Operators – Morphological option

The source image may have any structure and BD data type.

Dialog requests the number of iterations (default of –1 specifies iteration until the results remain unchanged) followed by a request for transformation lookup table (.lut).

The designated source image is processed to the displayed destination image of same structure and data type.

morphic_processor_roi – Performs morphological processing on a ROI using a 3x3 hit or miss transformation with a specified transformation table.

Accessed from the Operators – Morphological option

The source ROI may have any structure.

Dialog requests the number of iterations (default of –1 specifies iteration until the results remain unchanged) followed by a request for transformation lookup table (.lut).

The designated source ROI is processed to the displayed destination ROI of generic structure.

morphology – Performs connectivity-constrained morphological operations, such as shrinking and thinning, on a Boolean image using hit or miss transformations within a 3x3 window.

Accessed from the Operators – Morphological option

The source image may have any structure and BD data type.

Dialog requests number of iteration (-1 default specifies iteration until the results remain unchanged) and the morphological option.

The designated source image is processed to the displayed destination image of same structure and data type.

morphology_roi – Performs connectivity-constrained morphological operations, such as shrinking and thinning, on a ROI using hit or miss transformations within a 3x3 window.

Accessed from the Operators – Morphological option

The source ROI may have any structure.

Dialog requests number of iterations (-1 default specifies iteration until the results remain unchanged) and the morphological option.

The designated source ROI is processed to the displayed destination ROI of generic structure.

neighbour_count – Counts the number of TRUE state neighbours of a TRUE state pixel within a structuring element in a Boolean image.

Accessed from the Operators – Morphological option

The source image may have any structure and BD data type.

No dialog

The designated source image is processed to displayed destination image of same dimensions and structure but of SD data type.

noise_combination – Adds uniform or Gaussian distributed white noise to an image such that the noisy image has a prescribed signal-to-noise ratio.

Accessed from the Operators – Point option

The source image may have any structure and ND, SD or RD data type.

Dialog requests signal-to-noise ratio and probability density option.

The designated source image is processed to the displayed destination image of same dimensions and structure but of RD data type.

open_close – Performs opening or closing on a pixel plane of an image with a rectangular structuring element array.

Accessed from the Operators – Morphological option

The source image may have any structure and BD, ND, SD or RD data type.

Dialog requests selection of open or close and non-rotated or rotated structuring element for the erosion and for the dilation operations.

The designated source image is processed to the displayed destination image with structure and data type same as the source image.

open_close_roi – Performs opening or closing on a ROI with a rectangular structuring element array.

Accessed from the Operators – Morphological option

The source ROI may have any structure.

Dialog requests selection of open or close and non-rotated or rotated structuring element for the erosion and for the dilation operations.

The designated source ROI is processed to the displayed destination ROI with generic structure.

outlier_removal – Performs removal of outlier pixels in pixel planes that differ significantly from their neighbours. Removed pixels are replaced by the average of their neighbours.

Accessed from the Operators – Enhancement option

The source image may have any structure and ND, SD or RD data type.

Dialog requests selection of an impulse response array (.imp) from file or from repository and the threshold value.

The designated source image is processed to the displayed destination image of same structure and data type.

paint_pixels – Paints pixels specified by an array of control points from a source image into a destination image.

Accessed from the Operators – Modification option

The source image may have any structure and BD, ND, SD or RD data type. The source image will be used for cursor tracing of the paintbrush path.

Dialog requests third, fourth and fifth indices, number of brush control points, painting option and brush option.

The designated source image plane and trace are displayed as the destination image of same dimensions and structure but of RD data type.

perimeter_code_generator – Generates the crack code of the outer perimeter of each object in a pixel plane of a Boolean image.

Accessed from the Operators – Shape menu option.

The source image may have any structure and BD data type but only one designated 2D plane will be extracted and processed.

Dialog requests third, fourth and fifth indices and the Object count.

Resulting crack codes extracted from designated source image are presented in the System Messages.

polar_to_cartesian – Performs conversion of a pixel plane in polar coordinates to a pixel plane in Cartesian coordinates.

Accessed from the Operators – Geometric menu option

The source image may any structure and BD, ND, SD or RD data type.

Dialog requests *rho_min*, *rho_max*, *theta_min* and *theta_max* parameters..

The designated source image is processed to the displayed destination image of same dimensions, structure but of RD data type.

power_law_scaling – Performs power law amplitude scaling of an image in which each source pixel is raised to a power law exponent.

Accessed from the Operators – Point menu option

The source image may have any structure and ND, SD or RD data type. WARNING, amplitude results for ND or SD images may exceed maximum amplitude from powers greater than unity.

Dialog requests the exponent to be applied as a floating point value.

The designated source image is processed to the displayed destination image of same dimensions, structure and data type.

pseudocolour – Performs pseudocolouring of a grey scale image in which each grey scale value is mapped into a red, green, blue tristimulus value through a lookup table transformation.

Accessed from Operators – Enhancement menu option

The source image must have MON, VOL, TEM or VT structure and ND or SD data type.

Dialog requests the single – banded monochrome to three-banded colour lookup table (.lut) and offset to the first table value.

The designated source image is processed to the displayed destination image of same dimensions, COLR, VC, TC or VTC structure and same data type as source.

rescale – Performs size rescaling of a five-dimensional window of an image.

Accessed from the Operators – Geometric menu option

The source image may have any structure and any data type.

Dialog requests the five dimensional scale factors and the lower and upper coordinate values or optional selection of two-dimensional lower and upper coordinate values by cursor control.

The designated source image is processed to the displayed destination image of same structure and data type.

rescale_roi - Performs size rescaling of a five-dimensional window of a ROI.

Accessed from the Operators – Geometric menu option

Source ROI may have any structure

Dialog requests the five dimensional scale factors and the lower and upper coordinate values or optional selection of two-dimensional lower and upper coordinate values by cursor control.

The designated source ROI is processed to the displayed destination ROI with generic structure.

resize – Performs resizing of an image by reducing or enlarging its coordinate sizes.

Accessed from the Operators – Geometric menu option

The source image may have any structure and data type.

Dialog requests the five size dimensions of the destination image.

The designated source image is processed to the displayed destination image of same structure and data type.

resize_roi – Performs resizing of a ROI by reducing or enlarging its coordinate sizes.

Accessed from the Operators – Geometric menu option

The source ROI may have any structure.

Dialog requests the five size dimensions of the destination ROI.

The designated source ROI is processed to the displayed destination ROI of generic structure.

rotate – Performs two-dimensional or three-dimensional rotation of an image about a point.

Accessed from the Operators – Geometric menu option

The source image may have any structure and data type; however, band rotation of colour images is not permitted.

Dialog requests the j, k and l centre coordinates of rotation, the j-k, j-l and k-l angles and the rotation 2D or 3D option.

The designated source image is processed to the displayed destination image of same structure and data type.

rotate_roi – Performs two-dimensional or three-dimensional rotation of a ROI about a point.

Accessed from the Operators – Geometric menu option

The source ROI may have any structure.

Dialog requests the j, k and l centre coordinates of rotation, the j-k, j-l and k-l angles and the rotation 2D or 3D option.

The designated source ROI is processed to the displayed destination ROI with generic structure.

rubber_band_scaling – Performs rubber band amplitude scaling of an image in which the image undergoes a piecewise linear transformation over two linear segments.

Accessed from the Operators – Point menu option

The source image may have any structure and ND, SD or RD data type.

Dialog requests the low, high, inflection input and output values.

The designated source image is processed to the displayed destination image of same structure and data type.

sequence_average – Computes the pixel plane average of an image sequence.

Accessed from the Operators – 3D Specific option

The source image may have any structure and ND, SD or RD data type.

Dialog requests the dimension option to be averaged.

The designated source image is processed to the displayed RD destination image with appropriate structure.

sequence_karhunen_loeve – Performs a Kahunen-Loeve transform of an image sequence based upon an application-supplied covariance matrix.

Accessed from the Operators – 3D Specific option

The source image may have any structure, but third dimension greater than 2, and ND, SD or RD data type.

Dialog requests the indices of the fourth and fifth dimensions and whether the transformation is to be forward or reverse. The covariance matrix (.mtx) is request follows.

The designated source image is transformed into the displayed RD destination image.

sequence_running_measures – Computes running difference, ratio and log ratio measures of depth, time and spectral sequences of pixel planes.

Accessed from the Operators – 3D Specific option

The source image may have any structure, with adequate depth, temporal or spectral dimension sufficient to support the dimension option selected, and ND, SD or RD data type.

Dialog requests selection of structure (volume, temporal or spectral) and measure (difference, ratio or log ratio) options.

The designated source image is processed according to options selected resulting in an RD destination image of one lower size in the selected structure dimension (depth, time or bands).

shape_metrics – Generates an array of shape metrics of objects in a pixel plane of a Boolean image.

Accessed from the Operators – Shape option

The source image may have any structure and BD data type.

Dialog requests third, fourth and fifth indices and the maximum object count.

No destination image is created. Resulting objects extracted from the designated source image are described in the System Messages.

spatial_moments_invariant – Generates an array of Hu's invariant spatial moments of a pixel plane of an image.

Accessed from the Operators – Shape option

The source image may have any structure and BD, ND, SD or RD data type.

Dialog requests third, fourth and fifth indices.

No destination image is created. Resulting objects extracted from the designated source image are described in the System Messages.

spatial_moments_scaled – Generates a list of scaled spatial moments of a pixel plane of an image.

Accessed from the Operators – Shape option

The source image may have any structure and BD, ND, SD or RD data type.

Dialog requests third, fourth and fifth indices.

No destination image is created. Resulting objects extracted from the designated source image are described in the System Messages.

split_image – Generates split image views of a pair of monochrome or colour images.

Accessed from the Operators – Dyadic option

The source image must have MON or COLR structure and BD, ND, SD or RD data type. A second source image with structure and data type same as the source image will be requested.

Dialog requests source image combination option and split location

The designated source images are processed to a destination image of same structure and data type.

subsample – Performs spatial subsampling of an image. Pixels are extracted from a source window offset with respect to the source origin and placed in a destination image window offset with respect to the destination origin.

Accessed from the Operators – Geometric option

The source image may have any structure and any data type.

Dialog requests the j, k factors (required 1 or greater), j, k source offsets, j, k destination offsets and window width and height. The latter 6 may be assigned by cursor selection, the source offsets and window sizes from a right button box dragged over the source image and the subsequent destination offsets by a right button cursor click over the destination image. In addition, the initial destination image must be specified as none, image from file or exiting destination image. An initial destination image must be the same data type and structure as the source image.

The designated scaled image portion will be placed within the optional destination image or be itself the destination image according to the initially designated destination image.

subsample_roi – Performs spatial subsampling of a ROI. Pixels are extracted from a source window offset with respect to the source origin and placed in a destination image window offset with respect to the destination origin.

Accessed from the Operators – Geometric option

Source ROI may have any structure.

Dialog requests the j, k factors (required 1 or greater), j, k source offsets, j, k destination offsets and window width and height. The latter 6 may be assigned by cursor selection, the source offsets and window sizes from a right button box dragged over the source ROI and the subsequent destination offsets by a right button cursor click over the destination ROI. In addition, the initial destination ROI must be specified as none, ROI from file or exiting destination ROI. An initial destination ROI must may any structure.

The designated scaled ROI portion will be placed within the optional destination ROI or be itself the destination ROI according to the initially designated ROI.

template_match – Computes the cross-correlation function array of a pixel plane of an image and a template extracted from another image.

Accessed from the Operators – Correlation option

First source image may have any structure and ND, SD or RD data type and second source of same structure and data type.

Dialog requests third, fourth and fifth index offsets, first source image first and second index offsets, second source image first and second index offsets, number of window (from first source image) and template (from second source image) columns and rows. The first and second source image window offsets and dimensions may optionally be obtained by cursor.

No destination image. The destination static array will be displayed as a monochrome image and will be described in the system messages. The destination static array may be saved as a .pwr file.

threshold – Performs pixel amplitude thresholding of an image such that a destination pixel is set at a specified value if the source pixel is equal to or above a threshold, and the destination pixel is set at another specified value if the source pixel is below the threshold.

Accessed from the Operators – Point option

The source image may have any structure and ND, SD or RD data type.

Dialog requests above and below clip values, including imaginary values required for CD destination image type, threshold value and data type of the destination image

The designated source image is processed to the destination image of same structure as source image but with data type specified in the dialog.

transform_cosine – Performs a two-dimensional cosine transform in rectangular blocks on pixel planes of an image.

Accessed from the Operators – Unitary Transform

The source image may have any structure and ND, SD or RD data type.

Dialog requests block horizontal and vertical sizes and transform direction option.

The designated source image is transformed to the destination image with same structure but RD data type.

transform_fourier – Performs one- to five-dimensional Fourier transform of an image.

Accessed from the Operators – Unitary Transform

The source image may have any structure and ND, SD, RD or CD data type for forward transform but it must have CD data type for inverse transform.

Dialog requests the order option, transform direction option and frequency shift option.

The designated source image is transformed to the destination image with same structure. If the transform is forward, the destination image will be CD data type. If the transform is inverse, the destination image will be RD data type.

transform_hadamard – Performs a two-dimensional Hadamard transform on pixel planes of an image in rectangular blocks.

Accessed from the Operators – Unitary Transform

The source image may have any structure and ND, SD or RD data type.

Dialog requests the block size.

The designated source image is transformed to the destination image with same structure but and RD data type.

transform_hartley – Performs one- to 5-dimensional Hartley transform of an image.

Accessed from the Operators – Unitary Transform

The source image may have any structure and ND, SD or RD data type.

Dialog requests the order option.

The designated source image is transformed to the destination image with same structure but RD data type.

translate – Performs five-dimensional translation of an image with respect to its origin.

Accessed from the Operators – Geometric

The source image may have any structure and data type

Dialog requests the offset values.

The designated source image is processed to the destination image with same structure and data type.

translate_roi – Performs five-dimensional translation of a ROI with respect to its origin.

Accessed from the Operators – Geometric

The source ROI may have any structure.

Dialog requests the offset values

The designated source ROI is processed to the destination ROI with generic structure.

unary_integer – Performs unary point conversion of pixels of an integer data type image.

Accessed from the Operators – Point

The source image may have any structure and data type ND or SD.

Dialog requests the conversion option.

The designated source ROI is processed to the destination image of the same structure and data type.

unary_real – Performs unary point conversion of pixels of a real data type image.

Accessed from the Operators – Point

The source image may have any structure and data type RD.

Dialog requests the conversion option.

The designated source ROI is processed to the destination image of the same structure and data type.

unsharp_mask – Performs image detail sharpening by two-dimensional high pass filtering of pixel planes by forming the weighted difference between an image and a blurred version of the image.

Accessed from the Operators – Enhancement option

The source image may have any structure and ND, SD or RD data type.

Dialog requests source of impulse array (.imp file or repository), weight factors and the bias factor,

The designated source image is processed to the destination image of the same structure and RD data type.

value_bounds – Compiles a collection of all pixels in an image that lie within prescribed amplitude bounds.

Accessed from the Operators – Analysis

The source image may have any structure and ND, SD or RD data type.

Dialog requests upper and lower bounds and option to file resulting (.vbnd) array.

The designated source image is processed to a bounds array and a destination image of pixels captured.

wallis_statistical_differencing – Performs amplification of pixels by a gain factor that is dependent upon the moving window standard deviation of pixels within a pixel plane.

Accessed from the Operators – Enhancement

The source image may have any structure and ND, SD or RD data type.

Dialog requests the gain, average, deviation and proportionality parameters and the impulse response (.imp) array option

The designated source image is processed into the displayed destination image of the same structure and RD data type.

warp_control_point – Performs two-dimensional geometric warping of an image using polynomial inverse address calculation for which the polynomial coefficients are computed from control points.

Accessed from the Operators – Geometric

The source image may have any structure and data type.

A request to allow input of a stored control point (.ctl) file before the Warp Control Point dialog appears. Dialog reports the existing number of control point pairs, loads and provides options for adding control point pairs by cursor line drawing or manual entry. An option to save the resulting control point table (.ctl) follows.

The designated source image is processed into the displayed destination image of the same structure and data type.

warp_control_point_roi - Performs two-dimensional geometric warping of a ROI using polynomial inverse address calculation for which the polynomial coefficients are computed from control points.

Accessed from the Operators – Geometric

The source ROI may have any structure

A request will appear to allow input of a stored control point (.ctl file) before the Warp Control Point dialog appears. Dialog reports the existing number of control point pairs, loads and provides options for adding control point pairs by cursor line drawing or manual entry. An option to save the resulting control point table (.ctl) follows.

The designated source ROI is processed into the displayed destination ROI with generic structure.

warp_lut – Performs two-dimensional geometric warping of an image using lookup table inverse address calculation.

Accessed from the Operators – Geometric

The source image may have any structure and data type.

Row and column virtual lookup table images (.header) are requested from file.

The designated source image is processed into the displayed destination image of the same structure and data type but with J and K dimensions corresponding to the lookup table images J and K dimensions.

warp_lut_roi – Performs two-dimensional geometric warping of a ROI using lookup table inverse address calculation.

Accessed from the Operators – Geometric

The source ROI may have any structure.

Row and column virtual lookup table images (.header) are requested from file.

The designated source ROI is processed into the displayed generic ROI with J and K dimensions corresponding to the lookup table images J and K dimensions.

warp_polynomial – Performs two-dimensional geometric warping of an image using polynomial inverse address calculation.

Accessed from the Operators – Geometric

The source image may have any structure and data type.

A request to allow input of a stored coefficient array (.array) file appears before the Warp Polynomial dialog. Dialog reports the existing coefficients and supports editing of the coefficients. An option to save the resulting coefficients follows.

The designated source image is processed into the displayed destination image of the same structure and data type.

warp_polynomial_roi – Performs two-dimensional geometric warping of a ROI using polynomial inverse address calculation.

Accessed from the Operators – Geometric

The source ROI may have any structure.

A request to allow input of a stored coefficient (.array) file appears before the Warp Polynomial dialog. Dialog reports the existing coefficients and supports editing of the coefficients. An option to save the resulting coefficients follows.

The designated source ROI is processed into the displayed destination ROI of generic structure.

window_level – Performs window-level scaling of an image in which high amplitude source pixels above the window value are clipped to a specified value, low amplitude source pixels below the window lvalue are clipped to a specified value and intermediate source pixels are linearly rescaled in amplitude.

Accessed from the Operators – Point

The source image may have any structure and ND, SD or RD data type.

Dialog requests above and below level clip values, window width and window level.

The designated source image is processed into the displayed destination image of the same structure and data type.

window_statistics – Computes two-dimensional moving window statistics of an image.

Accessed from the Operators – Feature Extraction

The source image may have any structure and ND, SD or RD data type.

Dialog requests third, fourth and fifth index values, window odd-valued size and Mean, Standard Deviation, Median or Mode option.

The designated source image is processed into the displayed destination image of same structure but RD data type.

z_merge – Copies source image pixels into a destination image based upon the relative values of two Z-merge control images.

Accessed from the Operators – Ensemble

The source images (first, second and third images) may have any but same structure and same ND, SD or RD data type.

No operator-specific dialog.

The designated source images are processed into the displayed destination image of the same structure and data type.

zoom – Performs zooming of an image by magnifying its pixel planes by integer scale factors using pixel replication interpolation.

Accessed from the Operators – Geometric

The source image may have any structure and data type.

Dialog requests horizontal, vertical, depth, temporal and spectral integral scale factors. Zooming of the band dimension of a colour image is not permitted.

The designated source image is magnified into the displayed destination image of the same structure and data type.

zoom_roi – Performs zooming of a ROI by magnifying its pixel planes by integer scale factors using pixel replication interpolation.

Accessed from the Operators – Geometric

The source ROI may have any structure.

Dialog requests horizontal, vertical, depth, temporal and spectral integral scale factors.

The designated source ROI is magnified into the displayed destination ROI of generic structure.

SUPPORT - Support tools include PIKS mechanisms, tools and utilities. In addition, some support elements that are not in the PIKS standard have been added for the convenience of the PIKSTool user.

MECHANISMS – All mechanisms are accessed from the Support – Mechanisms menu option.

bind_roi – Binds a ROI object to an image.

Dialog requests the 5-dimensional offset values. A source ROI may be bound if it displayed as source or destination ROI or it must be chosen from (.roi) file. Note that ROI control must be enabled by *set_globals* for bound ROIs to be effective upon the image to which it is bound.

The specified ROI is bound to the designated image.

define_sub_image – Creates a reference to a sub-image.

Dialog requests the source image sizes and offsets.

The designated source image is converted into the generic displayed destination image according to the dimensions specified.

set_globals – Sets the PIKS global modes.

Dialog requests selection of error reporting, image resampling, ROI control and ROI resampling modes. The default modes are error reporting = enabled, image resampling = 2 bilinear, ROI control = disabled and ROI resampling = 2 bilinear.

TOOLS – All tools are accessed from the Support – Tools menu option.

array_to_lut – Inserts an external array of values into a lookup table at a specified index offset.

The source array may be obtained from the *create_array*, External Numeric Collection option.

The request for an input array is followed by a dialog requesting input and output data types and the conversion parameters: Table Entries per Band, Table Bands, Entry Offset and Band Offset.

The array is converted to LUT and a request for optional saving of the .lut file is presented.

colour_conversion_matrix – Generates 3x3 colour conversion matrices for conversion between XYZ and RGB colour spaces.

Dialog requests the chromaticity coordinates. This is followed by dialogs to optionally save the XYZ to RGB matrix and, subsequently the RGB to XYZ matrix.

There is no information displayed.

create_array – Supports creation of array objects, a PIKS non-standard extension tool.

Dialog requests the choice of external numeric, neighbourhood or pixel array type.

- For external numeric arrays, a dialog requests the output data type, numbers of rows and columns, output values and an option for interpolated fill between output values assuming a specified input value range. Anticipated row and column indices are returned to the dialog as values are entered and an option to save the resulting array as a .array file will follow completion entering values to satisfy the array dimensions.
- For neighborhood arrays, a dialog requests the data type, label, scale factor (applicable for SD data type), 5-dimensional size and keys of the array followed by a repeated data dialog that also displays the count of entered values. Values may be entered until Done is clicked or sufficient to satisfy the array dimensions. When the array is completed, dialog will appear to permit optional saving of the file dependent upon the label selected (.nbhood for GENERIC, .dit for DITHER, .imp for IMPULSE, .msk for MASK).
- For pixel arrays, a repeated dialog requests the relative coordinates and value of the 5D array contents Entered as CP values though they may be utilized as any of the PIKS internal data types. A count of pixels entered is displayed in the dialog box. When Done is clicked the tool proceeds to allow optional saving of the .pxl array file.

create_lookup_table – Supports creation of a lookup table, a PIKS non-standard extension tool.

Dialog requests the choice of filled or discrete table values, the number of entries if discrete values are to be entered, the input and output data types, the output bands and the band and entries offsets followed by a data entry dialog which displays the current band and entry indices and requests the input and output values

The resulting lookup table is displayed and described in the System Messages

create_matrix – Supports creation of a matrix, a PIKS non-standard extension tool.

Dialog requests the 2D column/row dimensions followed by dialog display of coordinate indices and request for input values.

The resulting matrix is displayed in the System Messages and a dialog allows saving of the matrix as a .mtx file

filter_butterworth – Generates a Butterworth filter two-dimensional transfer function static array data object.

Dialog requests the column and row sizes, filter order, cutoff frequency and pass type.

The resulting CD 2D static array destination object is displayed and described in the System Messages. A dialog follows to allow optional saving of the .xfrmf filter object.

filter_gaussian – Generates a Gaussian-shaped filter two-dimensional transfer function static array data object.

Dialog requests the column and row sizes and index spread factors.

The resulting CD 2D static array destination object is displayed and described in the System Messages. A dialog follows to allow optional saving of the .xfrmf filter object.

filter_inverse – Generates an inverse filter two-dimensional transfer function static array data object.

Dialog requests the column and row sizes and cutoff frequencies followed by a dialog request for the image blur CD 2D static array transfer function

The resulting CD 2D static array destination object is displayed and described in the System Messages. A dialog follows to allow optional saving of the .xfrmf filter object.

filter_matched – Generates a matched filter two-dimensional transfer function static array data object.

Dialog requests the column and row sizes and offsets followed by a dialog request for the image blur CD 2D static array transfer function and followed by a dialog request for the CD 2D noise power spectrum static array.

The resulting CD 2D static array destination object is displayed and described in the System Messages. A dialog follows to allow optional saving of the .xfrmf filter object.

filter_wiener – Generates a Wiener filter two-dimensional transfer function static array data object.

Dialog requests the column and row sizes followed by a dialog request for the image blur CD 2D static array transfer function, followed by a dialog request for the CD 2D image power spectrum static array and followed by a dialog request for the CD 2D noise power spectrum static array.

The resulting CD 2D static array destination object is displayed and described in the System Messages. A dialog follows to allow optional saving of the .xfrmf filter object.

filter_zonal – Generates a zonal filter two-dimensional transfer function static array data object.

Dialog requests the column and row sizes and cutoff frequencies and the filter type option.

The resulting CD 2D static array destination object is displayed and described in the System Messages. A dialog follows to allow optional saving of the .xfrmf filter object.

image_bar_chart – Creates a bar chart monochrome image with eight vertical bars ranging in amplitude from zero to one or creates a colour image with eight bars of different colours.

Dialog requests the image horizontal and vertical dimensions, maximum pixel amplitude value (≥ 255), colour or non-colour structure and ND, SD or RD data type. An image (.header) storage dialog request follows.

The resulting image is displayed as destination image.

image_constant – Creates an image whose pixels have the same constant value

Dialog requests the 5D image dimensions, colour or non-colour structure option, data type option and pixel values. An image (.header) storage dialog request follows.

The resulting image, not optimized in amplitude, is displayed as destination image.

image_gaussian – Creates a monochrome image with a Gaussian-shaped amplitude.

Dialog requests the image horizontal and vertical dimensions, horizontal and vertical centres and horizontal and vertical standard deviations. An image (.header) storage dialog request follows.

The resulting RD image is displayed as destination image.

image_grey_scale – Creates a monochrome grey scale image whose amplitude increases linearly from left-to-right, right-to-left, top-to-bottom or bottom-to-top.

Dialog requests the image horizontal and vertical dimensions, maximum pixel value (≥ 255), ND, SD or RD data type and orientation option. An image (.header) storage dialog request follows.

The resulting image is displayed as destination image.

image_random_number – Creates an image with pixels whose amplitudes are random numbers.

Dialog requests the image five-dimensional size, colour or non-colour structure, probability density mean and standard deviation, pixel maximum and minimum amplitude and uniform or Gaussian probability density option. An image (.header) storage dialog request follows.

The resulting image is displayed as destination image.

impulse_boxcar – Generates a boxcar two-dimensional impulse response function neighbourhood array object.

Dialog requests the column and row sizes and key positions, SD or RD data type and horizontal or vertical derivative option. An impulse function (.imp) storage dialog request follows.

The resulting impulse response neighbourhood array is described in the System Messages.

impulse_derivative_of_gaussian – Generates a derivative of Gaussian two-dimensional impulse response function neighbourhood array.

Dialog requests the column and row sizes and key positions, column and row standard deviations and horizontal or vertical derivative option. An impulse function (.imp) storage dialog request follows.

The resulting impulse response neighbourhood array is described in the System Messages.

impulse_difference_of_gaussians – Generates a difference of Gaussians two-dimensional impulse response function neighbourhood array object.

Dialog requests the column and row sizes and key positions and narrow and wide, column and row standard deviations. An impulse function (.imp) storage dialog request follows.

The resulting impulse response neighbourhood array is described in the System Messages.

impulse_elliptical – Generates a uniform amplitude elliptical-shaped two-dimensional impulse response function neighbourhood array object.

Dialog requests the column and row sizes and key positions and requests the SD or RD data type. An impulse function (.imp) storage dialog request follows.

The resulting impulse response neighbourhood array is described in the System Messages.

impulse_gaussian – Generates a Gaussian-shaped two-dimensional impulse response function neighbourhood array object.

Dialog requests the column and row sizes and key positions and requests the horizontal and vertical standard deviation values. An impulse function (.imp) storage dialog request follows.

The resulting impulse response neighbourhood array is described in the System Messages.

impulse_laplacian_of_gaussian – Generates a Marr-Hildreth, Laplacian of Gaussian two-dimensional impulse response function neighbourhood array object.

Dialog requests the column and row sizes and key positions and requests standard deviation value. An impulse function storage (.imp) dialog request follows.

The resulting impulse response neighbourhood array is described in the System Messages.

impulse_pyramid – Generates a pyramid-shaped two-dimensional impulse response function neighbourhood array object.

Dialog requests the column and row sizes and key positions and requests the SD or RD data type and pyramid shape options. An impulse function (.imp) storage dialog request follows.

The resulting impulse response neighbourhood array is described in the System Messages.

impulse_rectangular – Generates a uniform amplitude rectangular-shaped two-dimensional impulse response function neighbourhood array object.

Dialog requests the column and row sizes and key positions and requests the SD or RD data type option. An impulse function storage (.imp) dialog request follows.

The resulting impulse response neighbourhood array is described in the System Messages.

impulse_sinc – Generates a sinc function two-dimensional impulse response function neighbourhood array object.

Dialog requests the column and row sizes, key positions and the scale factors. An impulse function (.imp) storage dialog request follows.

The resulting impulse response neighbourhood array is described in the System Messages.

markov_spectrum – Generates a Markov process two-dimensional power spectrum static array object.

Dialog requests the column and row sizes and correlation factors and requests the symmetry option. A power static array (.pwr) storage dialog request follows

The resulting static array is displayed as real and imaginary CD data type destination images.

roi_coordinate – Generates a five-dimensional coordinate collection ROI object for which the ROI is logically TRUE or FALSE at specified coordinates in an external coordinate array, and is logically FALSE or TRUE at all other coordinates.

Dialog requests the five overall size dimensions of the ROI, the polarity of the coordinates to appear in the resulting ROI of opposite polarity and whether the 5-dimensional coordinates (j, k, l, m, n) are acquired from file or manual entry. Manual entry will allow the optional selection of an existing coordinate array file (.crd) to permit expansion. Successive coordinates can be entered by dialog. The resulting array can be saved and it will be passed on to the *roi_coordinate* tool.

The resulting ROI is displayed as one plane of the destination ROI as directed by the Display Control.

roi_elliptical – Generates a two-dimensional, three-dimensional, four-dimensional or five-dimensional elliptically- shaped ROI object.

Dialog requests the size dimensions of the ROI, the ellipse centres, the ellipse lengths, the internal polarity of the ellipse object, the dimensions of the ROI (2D through 5D) and the index identifiers.

The resulting ROI is displayed as one plane of the destination ROI as directed by the Display Control.

roi_polygon – Generates a two-dimensional polygon-shaped ROI object from an external array of polygon coordinates.

Dialog requests the five overall size dimensions of the ROI, the polarity of the polygon to appear inside the resulting ROI of opposite polarity, whether the 5-dimensional coordinates (j, k, l, m, n) acquired from file (.crd) or from manual entry and the index identifiers.

The resulting ROI is displayed as one plane of the destination ROI as directed by the Display Control.

roi_rectangular – Generates a one- to five-dimensional rectangular-shaped ROI object.

Dialog requests the five overall size dimensions of the ROI, the start and the end coordinates of the of the 1- to 5-dimensional rectangle, polarity of the rectangular object to appear inside the resulting ROI of opposite polarity, the dimensions of the rectangle (1D through 5D) and the index identifiers.

The resulting ROI is displayed as one plane of the destination ROI as directed by the Display Control.

windowing_function – Generates a windowing function two-dimensional array.

Dialog requests 2D Column and Row dimension values, 2D Column and Row Guard band values, windowing option (Bartlett, Blackman, Hamming, Hanning or rectangular), and an option to allow saving the resulting function as a (.static) file.

The function array is displayed as a destination image.

UTILITIES – all utilities are accessed from the Support – Utilities menu option.

constant_predicate – Determines if all pixels in an image are equal to a constant within a specified tolerance.

The source image may have any structure and ND, SD or RD data type.

Dialog requests the pixel value and tolerance

The results of the comparison are reported in the System Messages.

convert_array_to_image – Converts a matrix, neighbourhood or static array object to a monochrome image.

Dialog requests choice of matrix, neighbourhood array label or static array label. A dialog requesting the appropriate array type from file follows.

The designated source static array is converted to the displayed destination image.

convert_image_data_type – Converts a PIKS image from one arithmetic data type to another.

The source image may have any structure and any data type.

Dialog requests the data type choice for the conversion.

The designated source image is converted to a destination image with the specified data type.

convert_image_to_array – Converts a monochrome image into a matrix, neighbourhood or static array object.

The source image must have MON structure and specified data type as required by the conversion selected.

Dialog requests matrix, neighbourhood array or static array choice

- Conversion to matrix accepts only source images with ND, SD or RD data type.
- Conversion to neighbourhood array accepts source images with any data type and presents a dialog requesting key offsets, label and scale factor for SD data type arrays
- Conversion to static array accepts source images with any data type and presents a dialog requesting label

The designated source image is converted to the designated array and a dialog appears to allow saving the array as the appropriate file type.

convert_image_to_ROI – Converts an image to a ROI object.

The source image may have any structure and BD data type.

No dialog

The designated source image is converted to a displayed array ROI.

convert_roi_to_image – Converts a ROI object to an image.

The source ROI may have any structure.

No dialog

The designated ROI is converted to a displayed Boolean image.

copy_window – Converts a multi-dimensional rectangular-shaped window from a source image into a destination image.

The source image may have any structure and data type.

Dialog requests optional selection of window width and height by cursor and destination offset set equal to source offsets plus the window dimensions, if not specified by cursor, and the source and destination offsets.

The specified window from the source image is included in a new destination image if one did not preexist or added to the existing destination image.

copy_window_roi – Converts a multi-dimensional rectangular-shaped window from a source ROI into a destination ROI.

The source ROI may have any structure.

Dialog requests optional selection of window width and height by cursor and destination offset set equal to source offsets plus the window dimensions, if not specified by cursor, and the source and destination offsets.

The specified window from the source ROI is included in a new destination ROI if one did not preexist or added to the existing destination ROI.

equal_predicate – Determines if all pixels at corresponding indices of a pair of images are equal within a specified tolerance.

Dialog requests the second source image to be compared to the existing source image, both images may be any but same structure and have the same ND, SD or RD data types and the dialog requests the pixel tolerance

The specified images are compared and the results are presented in the System Messages.

extract_pixel_plane – Extracts a monochrome image from a multi-plane image.

Dialog requests the third, fourth and fifth indices

The specified image plane is extracted and displayed as the destination image.

extract_pixel_plane_roi – Extracts a ROI plane from a multi-plane ROI.

Dialog requests the third, fourth and fifth indices

The specified ROI plane is extracted and displayed as the destination ROI.

get_pixels – Services *get_colour_pixel*, *get_pixel_array*, *get_pixel* and *get_pixel_record* pixel data access.

Dialog provides checkbox if the user desires j,k rectangular coordinates selected by cursor and requests indices and size for the 5 dimensions (j,k entries ignored if cursor option checked). Indices must be within the image and sizes and will be automatically restricted to be within the image. A save dialog follows to permit pixel array (.pxl) file saving.

The coordinates and pixel values are displayed in the System Messages.

get_pixels_roi – Services *get_pixel_array_roi*, *get_pixel_roi* and *get_pixel_record_roi* pixel data access.

Dialog provides checkbox if the user desires j,k rectangular coordinates selected by cursor and requests indices and size for the 5 dimensions (j,k entries ignored if cursor option checked). Indices must be within the ROI and sizes and will be automatically restricted to be within the ROI. A save dialog follows to permit pixel array (.pxl) file saving.

The coordinates and pixel values are displayed in the System Messages.

inquire_elements – Returns an array of information regarding an implementations's support of elements.

Dialog requests a PIKS element to be entered. More element numbers can be subsequently entered until DONE is clicked.

A private label, the repository class option, the availability of the requested elements and the validity of the inquiry are displayed in the System Messages.

inquire_image – Returns image attribute information.

No dialog

The image attributes and the validity of the inquiry are displayed in the System Messages.

inquire_index_assignment – Returns the index assignment state.

No dialog

The x, y, z, t, b index assignments to j, k, l, m, n dimensions and the validity of the inquiry are presented in the System Messages

inquire_non_image_object – Returns non-image data object attribute information.

Dialog requests the object type followed by the appropriate type open file dialog to request the object.

The object type (Array, Coordinate Collection, Control Point Array, Dither Array, 1D Histogram, 2D Histogram, Impulse Response Array, Lookup Table, Mask Structuring Element, Matrix, Paint Brush Array, Pixel Collection, Power Array, ROI, Static Array, Transfer Function Array, Transform Array or Value Bounds Collection), its attribute values and the validity of the inquiry are displayed in the System Messages.

inquire_piks_implementation – Returns information about a PIKS implementation.

No Dialog

The PIKS implementation and the validity of the inquiry are presented in the System Messages.

inquire_piks_status – Returns information about PIKS status and global control mode status.

No Dialog

The status of PIKS functionality and the validity of the inquiry are presented in the System Messages.

inquire_repository – Returns an array of information regarding an implementation's support of data object repository enties.

Dialog requests the repository class option (entry locked upon first enter) and the ENTER of registration numbers until DONE is clicked.

A private label, the repository class option, the availability (TRUE) of the requested repository element and the validity of the inquiry are presented in the System Messages.

inquire_resampling – Returns an array of information regarding an implementation's support of resampling options.

Dialog requests registration numbers of resampling options be ENTERed until clicking DONE.

A private label, the availability of the requested resampling options and the validity of the inquiry are presented in the System Messages.

insert_pixel_plane – Inserts a monochrome image into a multi-plane image.

Dialog requests the third, fourth and fifth indices of the destination image into which the source image plane is to be inserted.

The source image replaces the designated image plane in the destination image and the destination image plane designated by the Display Control is displayed.

insert_pixel_plane_roi – Inserts a ROI plane into a multi-plane ROI.

Dialog requests the third, fourth and fifth indices of the destination ROI into which the source ROI plane is to be inserted.

The source ROI replaces the designated ROI plane in the destination ROI and the destination ROI plane designated by the Display Control is displayed.

put_pixels - Services *put_colour_pixel*, *put_pixel_array*, *put_pixel* and *put_pixel_record* requirements.

Dialog requests the method for image j and k index specification (cursor position or manual input), pixel source (file and/or manual input), remaining 3-dimensional manual index values and offset relative to input data (actual or minimum offset dimensions)

- If file source is specified, the file open dialog for .pxl files is presented for selection.
- If manual pixel source input is specified, a Pixel Data dialog appears requesting relative coordinate and pixel values that can be successively ENTERed for each pixel to be tabled followed by DONE when finished. A .pxl file save dialog follows.

The pixel data is subsequently added to the designated image and displayed.

put_pixels_roi - Services *put_pixel_array_roi*, *put_pixel_roi* and *put_pixel_record_roi* requirements.

Dialog requests the method for ROI j and k index specification (cursor position or manual input), pixel source (file and/or manual input) and the remaining 3-dimensional manual index values.

- If file source is specified, the file open dialog with .pxl files is presented for selection.
- If manual pixel source input is specified, a Pixel Data dialog appears requesting relative coordinate and pixel values that can be successively ENTERed for each pixel to be tabled followed by DONE when finished. A .pxl file save dialog follows.

The pixel data is subsequently added to the designated ROI and displayed.

SETTINGS – All settings are accessed from the Settings menu option

coordinate_display - Dialog presents checkbox that, if checked will allow System Messages display of image or ROI coordinates at cursor locations when the right mouse button is clicked.

display_control - Dialog allows selection of the third, fourth and fifth indices at which the plane or colour image will be displayed. The “Default Indices” button sets the third, fourth and fifth dimensions to 0. The “Repeat Display” button, when clicked, will result in the redisplay of the image with the currently selected third, fourth and fifth indices. The ‘Optimize Amplitudes’ checkbox, when checked, will allow reoptimization of the newly specified plane image amplitudes.

Appendix C. PIKS Object Files

PIKS non-image object files are stored as readable numeric text. Images are stored as two files, a .header of readable numeric text used to describe the image and a .data data stream file to hold the contents of the image. The numeric codes used in the descriptions of the files are defined below following the file definitions.

Array (*.array) Files

Array files describe the array content and hold numeric collections used as external data to a number of PIKS elements, e.g., *warp_polynomial*.

The first line of array files contains the internal data type code, the column count and the row count. Subsequent lines, numbering same as the row count, contain the data values, each line containing a number of values equal to the column count.

Coordinate (*.crd) Files

Coordinate files describe the collection and hold coordinate positions used as external data to a number of PIKS elements, e.g., *roi_coordinate*.

The first line of coordinate files contains the coordinate count and the dimensions represented. Subsequent lines, numbering same as the coordinate count, each contain the number of NP values equal to the dimensions.

Control Point (*.ctl) Files

Control Point files describe the collection and hold coordinate pairs used as external data to *warp_control_point*.

The first line of control point files contains the coordinate pairs count. Subsequent lines, numbering same as the coordinate pairs count, each contain pairs of 5-dimensional NP coordinates.

Dither (*.dithr) Files

Dither files describe the collection of the dither neighbourhood array object and its values used as external data in the *dither* operator.

The first line of dither array files contains the internal data type code, the INBHOOD_LABEL_DITHER label option and the scale factor. The second line contains the 5-dimensional key offsets. The third line contains the 5-dimensional key sizes. The fourth line contains a list of array values in number equal to the product of the 5-dimension sizes.

Histogram 1D (*.hst) Files

Histogram 1D files describe the histogram and hold the sequences of pixel sums produced by *histogram_1d* and are used as external data to PIKS elements, e.g., *histogram_shape_1d*.

The first line of histogram 1D files contains the bin count, lower amplitude limit value and upper amplitude limit value. The remaining rows, equal in number to the bin count, contain the histogram real bin data.

Histogram 2D (*.hst2d) Files

Histogram 2D files describe the histogram 2D static array object and hold the results from *histogram_2d* and are used as external data to PIKS elements, e.g., *histogram_shape_2d*.

The first line of histogram 2D files contain the internal data type code, the static array label ISTATIC_HIST, the horizontal dimension, the vertical dimension, the depth dimension, the temporal dimension and the spectral dimension. The remaining lines of the array contain the columns by row values of the static array plane data in the data type specified for the array.

Image (*.header and *.data) Files

Image header files describe the image to be obtained from its .data file as a data stream. The first line of the header contains the horizontal, vertical, depth, temporal and spectral dimensions of the image. The second line contains the precision as bit count per pixel, data type code, organization code, structure code and colour code. The third line is the name of the image .data file.

Images without structure and colour codes in their header files can be read by PIKSTool but these codes will be included in all headers written by PIKSTool.

Impulse Response (*.imp) Files

Impulse response files describe the neighbourhood array objects that are used as impulse response arrays for numerous PIKS elements, e.g., *unsharp_mask*.

The first line of impulse response array files contains the internal data type code, the INBHOOD_LABEL_IMPULSE label option and the scale factor. The second line contains the 5-dimensional key offsets. The third line contains the 5-dimensional sizes. The fourth line contains a list of array values in number equal to the product of the 5-dimension sizes.

Lookup Table (*.lut) Files

Lookup table files describe the lookup table objects to be used as external data for manipulation of images in operators, e.g., *lookup*.

The first line of lookup table files contain the source image internal data type code, the destination image internal data type code, the number of entries per band, the number of bands, the entries offset and the band offset. The subsequent lines, numbering same as the entries count minus the entries offset, present the lookup table output values. Each lookup value line lists lookup table value numbering same as the number of bands minus band offset.

Mask (*.msk) Files

Mask files describe the neighbourhood array objects that are used as external data to numerous PIKS elements for structuring, e.g., *erosion_dilation_boolean*.

The first line of mask files contains the BD internal data type code, the INBHOOD_LABEL_MASK label option and the scale factor. The second line contains the 5-dimensional key offsets. The third line contains the 5-dimensional sizes. The fourth line contains a list of array values in number equal to the product of the 5-dimension sizes.

Matrix (*.mtx) Files

Matrix files describe the matrix object that are used as external data for linear conversions by PIKS operators, e.g., *colour_lookup_linear*.

The first line of matrix files contains the internal data type code, the column count, the row count, the input colour code and the output colour code. The matrix values are presented in the subsequent lines numbering same as the row count, each line containing column count values.

Paint Brush (*.pbrsh) Files

Paint brush files describe the paint brush neighbourhood array object that is used for the *paint_brush* operator.

The first line of the mask file contains the internal data type code, the INBHOOD_LABEL_GENERIC label option and the scale factor. The second line contains the 5-dimensional key offsets. The third line contains the 5-dimensional sizes. The fourth line contains a list of array values in number equal to the product of the 5-dimension sizes.

Pixel (*.pxl) Files

Pixel files describe the pixel collection data that follows and is used as external data for operators, e.g., *draw_pixels*.

The first line of the pixel file contains the pixel count to follow. Subsequent lines, pixel count in number, contain lists of the pixels as five coordinate indices, real and (optional) imaginary values of the pixel.

Power (*.pwr) Files

Power files describe the power static array object that are used by a number of elements, e.g., *filter_matched*.

The first line of the power file contains the internal data type code, the ISTATIC_POWER label option code, the five valued five-dimensional sizes. Subsequent lines contain the object data values arranged in planes.

Region of Interest (*.roi) Files

Region of Interest files, better known as ROI files, describe ROI objects that are optionally used as image controls for most operators.

The first line of the ROI file contains the ROI structure code, the background polarity code and the number of data entries related to the specified ROI type. Subsequent lines of the file data depend upon the ROI type.

Array (code 1) and Generic (code 4)

The second line presents the overall 5-dimensional size values. Subsequent lines, in x, y, z, t, b order, present the ROI values at each position of the ROI, one value per line.

Coordinate (code 2)

The third value of the first line represents the coordinate entries count. The second line presents the overall 5-dimensional size values. Subsequent lines, numbering same as the entries count, list 5-dimensional coordinate position at which a pixel of polarity the complement of the background polarity is to be located.

Elliptical (code 3)

The third value of the first line represents the dimension code of the ellipse. The second line presents the overall 5-dimensional size values. The third line presents the five ellipse centre coordinates. The fourth line presents the five ellipse length values. The fifth line presents the five index manipulation alpha codes.

Polygon (code 5)

The third value of the first line represents the number of coordinate entries listed. The second line presents the overall 5-dimensional size values. The subsequent lines, numbering same as the entries count, list the two-dimensional coordinates. The last line presents the five index manipulation alpha codes.

Rectangular (code 6)

The third value of the first line represents the dimension code of the rectangle. The second line presents the overall 5-dimensional size values. The third line presents the 5-dimensional start coordinate positions. The fourth line presents the 5-dimensional end coordinate positions. The fifth line presents the five index manipulation alpha codes.

Static Array (*.static) Files

Static array files describe a static array object and hold the results from tools, e.g., *windowing_function*.

The first line of static array files contains the internal data type code, the static array label code, the horizontal dimension, the vertical dimension, the depth dimension, the temporal dimension and the spectral dimension. The remaining lines of the array contain the columns by row values of the static array plane data in the data type specified for the array.

Transfer Function (*.xfrf) Files

Transfer function files describe the transfer function static array object that are used by a number of elements, e.g., *filtering_homomorphic*.

The first line of the power file contains the internal data type code, the ISTATIC_TRANSFER label option code, the five valued five-dimensional sizes. Subsequent lines contain the object data values arranged in planes.

Transform (*.xfrm) Files

Transform files describe the transform static array object that are produced by a number of operators, e.g., *hough_transform*.

The first line of the power file contains the internal data type code, the ISTATIC_TRANSFORM label option code, the five valued five-dimensional sizes. Subsequent lines contain the object data values arranged in planes.

Value Bounds (*.vbnd) Files

Value bounds files contain the results from the *value_bounds* operator.

The first line of value bounds files contains the entries count, the lower and upper amplitude bound values and the internal data type of the image data. Subsequent lines, numbering same as the entries count, list the 5-dimensional coordinate positions and the pixel amplitude value.

*represents the name assigned to a file.

Internal Data Type Codes

1	BD	Bit
2	ND	Unsigned Char
3	SD	Short
4	RD	Float
5	CD	Real & Imaginary Float

External Data Type

1	BI	Bit (images) or Unsigned Char
2	NI	Unsigned Char
3	SI	Short
4	RF	Float
5	CF	Real & Imaginary Float

Image Organization

0	Planar
1	Interleaved

Image Structure Codes

1	MON	monochrome
2	VOL	volume
3	TEM	temporal
4	COLR	colour
5	SPE	spectral
6	VT	volume-temporal
7	VC	volume-colour
8	VS	volume-spectral
9	TC	temporal-colour
10	TS	temporal-spectral
11	VTC	volume-temporal-colour
12	VTS	volume-temporal-spectral
13	GEN	generic

Image Colour Codes

1	unspecified
2	non-standard RGB
3	linear CCIR illuminant D65 RGB
4	linear CIE illuminant E RGB
5	linear EBU illuminant C RGB
6	linear EBU illuminant D65 RGB
7	linear NTSC illuminant C RGB
8	linear NTSC illuminant D65 RGB
9	linear SMPTE illuminant D65 RGB
10	gamma CCIR illuminant D65 RGB
11	gamma EBU illuminant C RGB
12	gamma EBU illuminant D65 RGB
13	gamma NTSC illuminant C RGB
14	gamma NTSC illuminant D65 RGB
15	gamma SMPTE illuminant D65 RGB
16	EBU illuminant C YUV
17	EBU illuminant D65 YUV
18	NTSC illuminant C YIQ
19	NTSC illuminant D65 YIQ
20	SMPTE illuminant D65 YCbCr
21	CIE XYZ
22	CIE UVW
23	CIE Yxy
24	CIE Yuv
25	CIE L*a*b*
26	CIE L*u*v*
27	IHS
28	CMY
29	CMYK

ROI Type Codes

1	ARRAY
2	COORDINATE
3	ELLIPTICAL
4	GENERIC
5	POLYGON
6	RECTANGULAR

ROI Polarity Codes

0	IPOLARITY_TRUE,
1	IPOLARITY_FALSE

Dimension Codes

0	IDIMENSION_1D,
1	IDIMENSION_2D,
2	IDIMENSION_3D,
3	IDIMENSION_4D,
4	IDIMENSION_5D

Static Array Codes

1	GENERIC
2	POWER
3	TRANSFER
4	WINDOWING
5	HIST
6	TRANSFORM

Codes Utilized in Object Files

Appendix D. PIKS Scientific Elements Utilized Internally in PIKSTool

allocate_display_image
allocate_histogram
allocate_image
allocate_lookup_table
allocate_matrix
allocate_neighbourhood_array
allocate_pixel_record
allocate_roi
allocate_static_array
allocate_tuple
allocate_value_bounds
bind_roi
boolean_display
close_piks
close_piks_emergency
close_window
colour_display
create_tuple
deallocate_data_object
error_handler
error_logger
error_test
export_[all]
event_delay
get_[all] except get_pixel_array
get_file
import_[all]
monochrome_display
open_piks
open_window
put_file
return_repository_identifier

Appendix E. PIKS Scientific Elements Not Included in PIKSTool Version 3.2.1

abort_synchronous_execution
allocate_chain
allocate_composite_identifier_array
allocate_composite_identifier_list
allocate_composite_identifier_record
allocate_vreg
bind_match_point
composite_identifier_[all]
input_object
inquire_chain_environment
inquire_chain_status
inquire_non_image_object options
 identifier_array
 identifier_list
 identifier_record
 pixel record
 tuple
 virtual register
output_object
pseudocolour_display
set_globals (roi_process_mode, index_assign_mode and error_reporting_mode)
set_image_attributes
set_index_assignment
synchronize