LINC-NIRVANA

The LBT INterferometric Camera and Near-InfraRed / Visible Adaptive iNterferometer for Astronomy

A collaborative project of the MPIA Heidelberg, INAF-Arcetri, Universität zu Köln, and MPIfR Bonn http://www.mpia.de/LINC



LINC-NIRVANA

LN Data Reduction Software

Doc. No. LN-MPIFR-MAN-ASW-001 Issue 0.1 Date 28 October, 2008

 Prepared
 M. Heininger Name
 26. August, 2010

 Approved
 N. Surname Name
 Date
 Signature

 Released
 N. Surname Name
 Date
 Signature

Issue	Date	Section/Parag. affected	Reason/Remarks
0.1	28. April, 2009	All	Initial version

Change Record

Contents

1	Intro	oduction	L
	1.1	Data Reduction Schema	L
	1.2	Downloading the Software	2
	1.3	Installing the Software	2
2	Tute	orial	3
	2.1	Starting the Graphical User Interface	3
	2.2	Loading an Example Parameter Set	ł
	2.3	Inspecting the Example Data	5
	2.4	Calibrating the Raw Data	;
	2.5	Preprocessing the Calibrated Data	3
	2.6	Extracting a Target from the Preprocessed Data)
	2.7	Extracting a Psf from the Preprocessed Data 10)
	2.8	Reconstructing an Object 10)
ი	TT		
3	HOW	to start the software	5
	ა.1 ე.ე	Introduction))
	პ.2 ე.ე	Starting the Graphical User Interface))
	ა.ა ე_4	Using the Pipeline in Batch Mode It) 1
	3.4	Using the Configuration File Editor	Ł
4	How	to use the image window 16	3
	4.1	Introduction	3
	4.2	Tools	3
	4.3	View Tool	7
	4.4	Info Tool	7
	4.5	Probe Tool	3
	4.6	Pixel Tool)
5	How	to calibrate the raw data 22	2
	5.1	Introduction	2
6	How	to determine the preprocessing parameters 25	2
Ū	61	Introduction 25	ł
	6.2	Reference Star Tool	1
	6.3	Estimating the Transformation Parameters 26	3
	6.4	Recentering a Reference Star	, 7
	0.1		
7	How	to extract a target 28	3
	7.1	Introduction	3
	7.2	Target Tool	3
	7.3	Target Extraction 30)
0	TT		,
ð	\mathbf{P} \mathbf{P} \mathbf{P} \mathbf{P} \mathbf{P}	To define a psi 32 Introduction 32))
	0.1	DSE Star Teol	ند د
	0.2 0.2) 1
	రి.చ	паю 1001	Ł

	8.4	Psf Extraction	35
9	Hov 9.1	v to reconstruct an object	37 37
10	Soft 10.1 10.2 10.3	ware Architecture Introduction	39 39 39 39
11	Too 11.1 11.2 11.3	ls Introduction	41 41 41 42
12	Rec 12.1 12.2 12.3	entering objects	45 45 46 47 47 48
13	Sky 13.1	background Introduction	49 49
14	Defe 14.1 14.2 14.3 14.4	ect pixel interpolation Introduction	50 50 50 51 51
15	Rec 15.1 15.2	onstruction algorithms Introduction	55 55 55

List of Figures

1	The flowchart represents the data reduction and image reconstruction pipeline	1
2	The main window of the LN DRS (left panel) represents the pipeline (flowchart in the	
	left window) and provides access to the available algorithms and selected files. The	
	image window of the LN DRS (right panel) shows loaded images and provides access	
	to specific tools	3
3	Window, menu and dialog box to load a parameter set.	4
4	Load an existing image.	5
5	A loaded image is shown in the LN DRS Image GUI window.	6
6	Performing a calibration	7
7	Calibrated data.	7
8	Preprocessing	8

9	The image window shows the first preprocessed interferogram (for the first position	
	angle)	9
10	The algorithm-specific target creation setup dialog	9
11	The target image created from the target preprocessing step	10
12	The algorithm-specific psf creation setup dialog	11
13	The psf image created from the psf preprocessing step	11
14	The algorithm-specific reconstruction setup dialog.	12
15	The reconstructed target convolved with the reference psf (23m psf)	12
16	Configuration editor.	15
17	A loaded image is shown in the LN DRS Image GUI window	16
18	Toolbar in the "LN DRS Image GUI" window.	17
19	Graphical representation and dialog window of an image probe	19
20	Graphical representation and dialog window of an image pixel	20
21	Standard calibration setup dialog.	22
22	Dialog window containing the preprocessing parameters	23
23	Calibrated data.	24
24	Graphical representation and dialog window of a reference star	25
25	Preprocessed interferogram with a target object, two psf stars and an image probe	28
26	Graphical representation and dialog window of a target	29
27	The algorithm-specific target creation setup dialog	30
28	The target image created from the target preprocessing step	31
29	Preprocessed interferogram with a target object, two psf stars and an image probe	32
30	Graphical representation and dialog window of a psf star.	33
31	Graphical representation and dialog window of a target halo.	35
32	The algorithm-specific psf creation setup dialog	35
33	The psf image created from the psf preprocessing step	36
34	Left panel: a typical target used by the Richardson-Lucy deconvolution algorithm.	
	Right panel: the associated psf	37
35	Setup dialog window for the Richardson-Lucy algorithm.	37
36	Left panel: a reconstructed target using the Richardson-Lucy deconvolution algorithm.	
	Right panel: the same image convolved with an ideal 23m psf	38
37	Class hierarchy of the framework core classes.	39
38	Properties and dependencies of the framework core classes	40

List of Tables

1	Optional arguments for the GUI command.	13
2	Optional arguments for the pipeline command.	14
3	Predefined tools	17
4	View tool mouse button assignment	18
5	Info tool mouse button assignment	18
6	Probe tool mouse button assignment	19
7	Pixel tool mouse button assignment	20
8	Reference star tool mouse button assignment	25
9	Target tool mouse button assignment	29
10	Psf star tool mouse button assignment	33
11	Halo tool mouse button assignment	34
12	Tool template placeholders	43

1 Introduction

1.1 Data Reduction Schema

The LINC-NIRVANA Data Reduction Software (LN DRS) implements a data reduction and image reconstruction pipeline (see figure 1). The software provides all steps necessary to reconstruct high-resolution images from several LINC-NIRVANA raw images. The flowchart in figure 1 represents the data flow (from top to bottom).



Two different elements are shown in the flowchart. The blue rectangles represent data reduction and reconstruction steps; the other rectangles represent files with specific content. The processing steps are executed in the following order:

- 1. Calibration: compensates detector and instrument effects like flatfield and bad pixels.
- 2. Preprocessing: compensates observation effects like image rotation.
- 3. Target Preprocessing: extracts the target from the preprocessed data.
- 4. PSF Preprocessing: extracts the psf from the preprocessed data and removes the background.
- 5. Reconstruction: reconstructs an image from target and psf images.

The LN DRS uses the files with the following content:

- RAW Data: the raw interferograms of the science target from the LINC-NIRVANA instrument.
- SKY: sky images.
- Flat: the flatfield map for compensating the pixel gain.
- Bad Pix Map: a map in which all defective pixels are marked as bad pixels.
- Calibrated Data: the calibrated interferograms (without instrument effects).
- Preprocessed data: the preprocessed interferograms (without observational effects).
- Target: the extracted target.
- Ref PSF: the reference psf (23m).
- PSF: the extracted psf.
- RefImg: the convolved reference image for calculating the image error (used for the tests).
- REC: the reconstruction (not convolved).
- REC Convolved: the convolved (23m psf) reconstructed object.
- LOG: the file with logging information.

Figure 1: The flowchart represents the data reduction and image reconstruction pipeline.

An astronomer collects raw data images and calibration data (sky, flatfield and bad pixel map) from an observation and can use the graphical user interface to inspect these data and parameterize the pipeline. By executing the pipeline, controlled by these parameters, a high resolution image will be reconstructed.

1.2 Downloading the Software

The software of the LINC-NIRVANA Data Reduction System (LN DRS) uses the IDL programming language and consists of several components:

- 1. The idlastro library (mandatory) idlastro.tar.gz (4.7MB)
- 2. The LN DRS package itself (mandatory) lndrs.tar.gz (65KB)
- 3. The LN DRS documentation (HTML, optional) lndrs_manual.tar.gz (2.1MB)
- 4. An example (NGC4151, optional) NGC4151.tar.gz (121MB)
- 5. A configuration file editor written in Java (optional) lndrs.jar (54KB)

Please download at least the first two packages from the web page at http://www.mpifr-bonn.mpg.de/lndrs/index.html and store them in a proper directory (e.g. " \sim /lndrs").

1.3 Installing the Software

After downloading all packages into a directory (e.g. " \sim /lndrs"), please go into this directory and follow these steps:

- Unpack the idlastro library package with the command tar xzf idlastro.tar.gz
 A subdirectory with the name "idlastro" is created.
- 2. If you use the bash as your shell, then include export IDL_PATH="<IDL_DEFAULT>:~/lndrs/idlastro/pro" in your ".bash_profile" file. If you are using the tcsh, then include setenv IDL_PATH "<IDL_DEFAULT>:~/lndrs/idlastro/pro" in your ".cshrc" file. If you want to use the software from within any directory, you must add the source code

directory (e.g. "~/lndrs/idl") to the environment variable.

- 3. Unpack the LN DRS package with the command tar xzf lndrs.tar.gz A subdirectory with the name "idl" is created.
- 4. If you downloaded the example file, unpack it with the command tar xzf NGC4151.tar.gz
 A subdimension in the subdimension of the subdi

A subdirectory "NGC4151" is created with all input files of a data reduction example.

3

2 Tutorial

2.1 Starting the Graphical User Interface

Before you start, a new login shell is needed for the environment variable "IDL_PATH" to include the idlastro package. Change your current directory with the command cd ~/lndrs/idl. Then, type idl, which starts the IDL programming environment. If you want to use the software from within another directory, you have to add the "~/lndrs/idl" directory to the "IDL_PATH" environment variable.

The LINC-NIRVANA Data Reduction Software (LN DRS) has two entry points:

- The graphical user interface, which is started with ldrs_gui.
- The pipeline, which is started with ldrs_pipeline,

This tutorial covers only the graphical user interface, which provides functions for manipulating a parameter file and executing algorithms. A parameter file is a simple text file which contains all file names, algorithm selections and parameters necessary to execute the pipeline, which uses the raw image data and produces a reconstructed image. After starting the graphical user interface with ldrs_gui, you will see two new windows (see figure 2).

				LN DRS Image GUI (on s	ervice64)		
		Recipe	Algorithm File	e Image Help			
		File:			Inage: < D	> Color: Gray	<u>×</u>
LN DI	RS Recipe GUI	Tools:	• • •	2 ? 🖪 🖬 🎯			
Recipe Algorithm File Image Help							
Recipe	Algorithm						
RAW Data SKY Flat Bad Pix Map Calibration							
Calibrated Data							
reprocessing	Edit Execute Pause Resume						
Preprocessed Data	Files						
Target Preprocessing PSF Preprocessing Target Ref PSF PSF							
Refing Reconstruction REC REC Convolved LOG	Prefix: I						
Open Save As	Add Edit Load Info Remove Prefix						
Info:							
		Info:					

Figure 2: The main window of the LN DRS (left panel) represents the pipeline (flowchart in the left window) and provides access to the available algorithms and selected files. The image window of the LN DRS (right panel) shows loaded images and provides access to specific tools.



• Section 3 "How to start the software"

2.2 Loading an Example Parameter Set

In the example NGC4151.tar.gz, a file with a parameter set is given in the file "recipe.conf". In order to load these parameters into the framework, please execute the following steps (see figure 3):

- 1. Select the menu item **Recipe/Open...** from the menu bar or **Open...** in the main window below the flowchart.
- 2. In the file selection dialog box switch to your directory with the NGC4151 example (e.g. " \sim /lndrs/NGC4151") and select the file "recipe.conf".
- 3. Click the **Ok** button in the file dialog box.

LN DRS Recip	De GUI (on service64)
Recipe Algorithm File Image Help	
Open Recipe	Algorithm
Save As SKY Flat Bad Pix Map Save As Calibration Calibrated Data Preprocessing Preprocessing Target Preprocessing Target PSF Reconstruction Ref PSF	Edi Selection Filter Selection Filter Carcel
Den Save Save As	Add Edit Load Info Remove

Figure 3: Window, menu and dialog box to load a parameter set.

After loading the parameter file, the elements in the flow diagram can be used to inspect file names, algorithm selections and additional parameters.

Important:

The method used by the LN DRS framework for storing filenames in parameter files has a strong influence on the organization of the files. The directory paths of the filenames stored in the parameter file are converted into relative paths based of the location of the parameter file. If, for example, the parameter file has the name

"/home/mhein/soft/lndrs/data/ngc4151/param.conf" and a raw data file called

"/home/mhein/soft/lndrs/data/ngc4151/raw_200101227T204710.fits", the raw data file is stored simply as "raw_200101227T204710.fits", because the complete directory path of the raw file is identical to the directory path of the parameter file.

The relative name of a data file is converted into an absolute name while loading; while storing, the absolute name is converted into a relative name. This method allows the complete directory content, including the data files and the parameter file, to be moved to a different location without jeopardizing the use of the parameter file.

2.3 Inspecting the Example Data

After loading the parameters, the available images can be inspected by selecting and loading a file. The example contains an artificially created sky, a flatfield and a bad pixel map. The raw file was created from simulated calibrated data by applying the inverse flatfield and adding the sky image.

An existing file can be viewed by carrying out the following steps (see also figure 4):

- 1. Select the image type in the flow diagram; for example, **RAW Data**.
- 2. Select the file name in the list box in the lower left part of the main window.
- 3. Click on the **Load** button below the list box.

LN DRS Recip	pe GUI (on service64)
Recipe Algorithm File Image Help	
1 Recipe	Algorithm
RAW Data SKY Flat Bad Pix Map	
Calibration	
Calibrated Data	
Preprocessing	
Programmed Date	Edit Execute Pause Resume
Preprocessed Data	Files 2
Target Preprocessing PSF Preprocessing	/iri/irisrv/howe/whein/soft/lndrs/data/NGC4151/raw.fits
Target PSF	
Reconstruction Ref PSF	
REC REC Convolved	3
Open Save Save As	Add Edit Load Info Remove
Info:	

Figure 4: Load an existing image.

After loading a file, the contents are shown in the LN DRS Image GUI window (see figure 5). If the file contains several images (several position angles), the first image is displayed.

This window provides the following functional interface:

- 1. The file name of the image currently shown in the window.
- 2. A mouse click on the < button switches to the preceeding image (lower image number).
- 3. A mouse click on the > button switches to the following image (higher image number).
- 4. The current color map can be switched by using this option menu.
- 5. These tools are necessary to parameterize the preprocessing, target preprocessing and PSF preprocessing steps.

By using the view tool (see section 4.3 "View Tool"), the user can explore an image in more detail or get an overview.

Additional information:

• Section 4 "How to use the image window"



Figure 5: A loaded image is shown in the LN DRS Image GUI window.

2.4 Calibrating the Raw Data

An astronomer receives the raw data, flatfield images, a valid bad pixel map and sky images from an observation. The raw images must be calibrated before further preprocessing and reconstruction can be applied. The calibration step only compensates for instrumental and detector effects. The mapping of the pixel to the sky is not changed. The steps necessary to execute a calibration are (see figure 6):

- 1. Select the calibration step in the flow diagram.
- 2. Select the specific algorithm.
- 3. Click the **Edit** button below the list box to display the algorithm-specific **Calibration Setup** dialog box, or double click on the algorithm name in the list box.
- 4. In the dialog box, set the specific parameters.
- 5. Click Apply after changing the parameters and on Close to close the dialog box.
- 6. The calibration step is executed by clicking on the **Execute** button below the list box.

Important:

Whether or not you are allowed to subtract the sky background depends on the algorithm used in the reconstruction. For example, in the Richardson-Lucy deconvolution algorithm, the target image must contain the sky background. In this case, the sky background must be kept and subtracted later during the psf preprocessing step.



Figure 6: Performing a calibration.

After the calibration is executed, the contents of the first calibrated file are displayed in the LN DRS Image GUI window (see figure 7).

LN DRS Image GUI (on service64)	_0×
Recipe Algorithm File Image Help	
File: Viri/irisrv/home/mhein/soft/lndrs/data/NGC4151/cal Image: < 0 > Color: Blackb	ody <u>v</u>
mis: N A II R 2 II H M	
Ref 1	
"Ref 2	
Info;	

Figure 7: Calibrated data.

Additional information:

• Section 5 "How to calibrate the raw data"

2.5 Preprocessing the Calibrated Data

The preprocessing step changes the mapping of the calibrated images. This means that the orientation (mapping on the sky) of all images is identical after this step. The steps necessary to preprocess the calibrated data are (see figure 8):

- 1. Select the preprocessing step in the flow diagram.
- 2. Select the specific algorithm.
- 3. Click on the **Edit** button below the list box to display the algorithm-specific **Preprocessing Setup** dialog box.
- 4. In the dialog box, set the specific parameters (see section 6 "How to determine the preprocessing parameters").
- 5. Click on the **Apply** button after changing the parameters and on the **Close** button to close the dialog box.
- 6. The preprocessing step is executed by clicking on the **Execute** button below the list box.

Important:

In order to correctly rotate and translate the images (map them at the same position and orientation in the sky), it is necessary to calculate or extract the transformation parameters. In this example, the images are from computer simulations; therefore, the rotation angles are well known.



Figure 8: Preprocessing.

After the data has been preprocessed, the contents of the first preprocessed file are displayed in the **LN DRS Image GUI** window (see figure 9). In this figure, it is obvious that at least one image is tilted compared to the original orientation in the calibrated image (see figure 7).

In the center of the image, the target is marked with a green square and the label **Target 1**. Both available psf stars are marked with green squares, too (labels **Psf 1** and **Psf 2**).



• Section 6 "How to determine the preprocessing parameters"

🗖 LN DF	S Image GUI (on se	ervice32)	_ O X
Recipe Algorithm File Image	Help		
File: Home/mhein/soft/lndrs/data	/NGC4151/int.fits	Image: < 🕅 > Color:	Blackbody <u>v</u>
Tools: 📐 🕀 🕩 🙋 🤅	' 🖾 🗷 🞯		
Probe 1		Psf I	
	Target 1		
Psf 2			
Info: L: zoom in, M: pan, R: zoom	out		

Figure 9: The image window shows the first preprocessed interferogram (for the first position angle).

2.6 Extracting a Target from the Preprocessed Data

Before extracting a target from the preprocessed images, it is necessary to create a target object in the first image of each file. In the provided example, the target is already marked and ready to use. The steps necessary to define a target including sky background are described in a separate section (see section 7 "How to extract a target").

The target preprocessing is executed like the calibration or preprocessing step. The algorithm-specific dialog window is presented in figure 10 and the result is shown in figure 11.

🗖 Target Creatio 🗕 🗖 🗙		
Add: One 🗹		
Background: 🗖 Remove		
Fourier Mask: 🗖 Use		
Centering: 🗖 Recenter		
Apply Close		

Figure 10: The algorithm-specific target creation setup dialog.



Figure 11: The target image created from the target preprocessing step.

Additional information:

• Section 7 "How to extract a target"

2.7 Extracting a Psf from the Preprocessed Data

Before extracting a psf from the preprocessed images, a graphical representation of at least one psf star must be created in the first image of each file. In the provided example, two psf stars are already marked and ready to use.

The psf preprocessing is executed like the target preprocessing step. The algorithm-specific dialog window is presented in figure 12 and the result is shown in figure 13.

Additional information:

• Section 8 "How to define a psf"

2.8 Reconstructing an Object

After creating the target and psf image as an input to the reconstruction step, an algorithm is selected and specific parameters are set (figure 14).

During reconstruction, intermediate results are displayed in the image GUI. It is possible to interrupt and resume or terminate a reconstruction (buttons **Pause** and **Resume** below the algorithm list



Figure 12: The algorithm-specific psf creation setup dialog.

LN DRS Image Gl	JI (on service32)	
Recipe Algorithm File Image Help		
File: //home/mhein/soft/lndrs/data/NGC4151/psf.f	its Image: < D > Color: Black	kbody <u>v</u>
Tools: 🕀 🕂 🖉 ? 🖾 🖽		
		-
		- 1.
		12
Info:		

Figure 13: The psf image created from the psf preprocessing step.

box in the LN DRS Recipe GUI window (see the left panel in see figure 2). The result of the reconstruction step is automatically displayed afterward (figure 15).

Additional information:

• Section 9 "How to reconstruct an object"

🗖 Deconvolution 🗕 🗆 🗙		
Iterations:	I 5000	
Show every:	ž 50	
Apply Close		

Figure 14: The algorithm-specific reconstruction setup dialog.



Figure 15: The reconstructed target convolved with the reference psf (23m psf).

3 How to start the software

3.1 Introduction

The LN DRS consists of a data reduction framework and a set of modules which provide algorithms for all processing steps (calibration, preprocessing, target preprocessing, psf preprocessing and reconstruction). A complete data reduction and image reconstruction setup is specified by:

- A set of input files which are the result of an observation run (raw data, sky images, flatfield and bad pixel map).
- A set of intermediate files (calibrated data, preprocessed data, target and psf).
- The reconstructed object (reconstruction and the convolved reconstruction).
- A selected algorithm for each processing step and the specific parameters.

This information is stored in a parameter file which can be created and modified by hand or with the graphical user interface (started with the command ldrs_gui). If a parameter file is already created, all processing steps can be carried out without a GUI by executing the pipeline with the command ldrs_pipeline.

The LN DRS framework uses a mandatory configuration file ("ldrs.cfg") to load all available plug-ins into the framework. In the LN DRS distribution one configuration file is provided which describes all plug-ins contained in the distribution. If a user implements additional plug-ins, a new extended version of this configuration file is needed.

3.2 Starting the Graphical User Interface

The graphical user interface is started with the command ldrs_gui. The user can provide the optional parameters described in table 1.

CFG= <filename></filename>	The name of the configuration file relative to the current
	directory. The file "ldrs.cfg" is the default.
/FORCE	If this flag is used, the configuration file will be loaded even
	if a configuration is already available.
RECIPE= <filename></filename>	The name of a parameter file relative to the current direc-
	tory.

Table 1: Optional arguments for the GUI command.

3.3 Using the Pipeline in Batch Mode

The pipeline is started with the command ldrs_pipeline. The user can provide the optional parameters described in table 2.

It is possible to use the pipeline in a shell script by invoking it with

CFG = < filename >	The name of the configuration file relative to the curent
	directory. The file "ldrs.cfg" is the default.
/FORCE	If this flag is used, the configuration will be loaded even if a
	configuration is already available.
RECIPE= <filename></filename>	The name of a parameter file relative to the current direc-
	tory.
PROC= <name></name>	The name or tag of one or more processing steps. They are
PROC=[<name1>,</name1>	executed in the given order.
<name2>,]</name2>	

Table 2: Optional arguments for the pipeline command.

```
echo "ldrs_pipeline,_recipe='recipe.conf',_proc='Calibration'" | idl
```

Source 1: Invoking the pipeline from a shell script.

which executes the calibration step of the first example in the tutorial. Alternatively, a short batch file can be created with the contents:

ldrs_pipeline, recipe='data/ngc4151/recipe.conf', proc='Calibration'

Source 2: Invoking the pipeline from an IDL startup file.

which is executed by starting the idl runtime environment after specifying that file as a startup file. An idl startup file is defined by setting the environment variable "IDL_STARTUP" to the filename (see the **Introducing IDL/Startup Files** chapter in the **Using IDL** manual). The pipeline can be launched from the IDL command prompt with:

ldrs_pipeline, \$
recipe='/home/mhein/soft/lndrs/data/NGC4151/recipe.conf', \$
proc=['CAL', 'PREP', 'TARCR', 'PSFCR', 'REC']

Source 3: Invoking the pipeline from the IDL command prompt.

which executes the complete pipeline and produces a reconstruction.

3.4 Using the Configuration File Editor

If a user wants to extend the LN DRS (for example, implementing a new reconstruction algorithm), he has to modify the configuration file "ldrs.cfg". A small application simplifies modification of this file. A screenshot of the configuration editor is presented in figure 16.

The configuration editor is launched from the command line with the command java -jar lndrs.jar. The jar file needed is provided in the distribution (see section 1.2 "Downloading the Software") and must be given as an argument to the java command.



Figure 16: Configuration editor.

4 How to use the image window

4.1 Introduction

After loading a file, the contents are shown in the LN DRS Image GUI window (see figure 17). If the file contains several images (several position angles), the first image is displayed.



Figure 17: A loaded image is shown in the LN DRS Image GUI window.

This window provides the following functional interface:

- 1. The file name of the image currently shown in the window.
- 2. A mouse click on the <-button switches to the preceeding image (lower image number).
- 3. A mouse click on the >-button switches to the following image (higher image number).
- 4. The current color map can be switched by using this option menu.
- 5. These tools are necessary to parameterize the preprocessing, target preprocessing and PSF preprocessing steps.

4.2 Tools

The toolbar in the image window (see figure 18 and table 3) provides tools for navigating a loaded image, getting information about a part of an image or creating and modifying objects necessary for specific data processing steps (for example, creating reference stars for the preprocessing step).

Tools: 📐 🕀 💀 🖻 🥐 🖾 🗮 🎯

Figure 18: Toolbar in the "LN DRS Image GUI" window.

Except for the view tool and the info tool, each tool has an accompanying object which is created, manipulated and deleted through that tool.

Icon	Name	Description
k	View tool	Provides functions for zooming and panning images.
		888888
Ð	Reference star tool	Provides functions for creating and modifying reference stars (preprocessing step).
•••	PSF star tool	Provides functions for creating and modifying psf stars (psf preprocessing step).
0	Target tool	Provides functions for creating and modifying targets (target preprocessing step).
?	Info tool	Provides access to created objects and their specification.
k	Probe tool	Provides statistical information of an image part.
	Pixel tool	Provides information about an individual pixel and its direct neighbours.
0	Halo tool	Creates and modifies a target halo (psf preprocessing).

Table 3: Predefined tools

4.3 View Tool

This tool provides functions for zooming and panning (moving) an image in the LN DRS Image GUI window. Other tools also include this function, but the Shift key must be pressed simultaneously.

The zoom function is equivalent to binning or enlarging pixels. The zoom steps are integral values. This means that the values go from -20 to 20, where positive numbers mean enlarging (zooming in) and negative values shrinking (binning). A value of 0 means a 1:1 image display.

4.4 Info Tool

The info tool provides access to the information of graphical objects. By selecting this tool and an existing object with the left mouse button, the dialog window containing the object-specific data

Button	Function
Left click	Zooms into the image and the click position is used as the
	new image center.
Middle click	Uses the click position as the new image center.
Right click	Zooms out of the image and the click position is used as the
	new image center.
Middle drag	Moves the image.

Table 4: View tool mouse button assignment

is shown. In addition, the functionality of the view tool is still available by pressing the shift key simultaneously with a mouse button.

Mouse button	Function
Left click	Shows the dialog window containing information about the
	selected object.
Shift left click	Zooms into the image and the click position is used as the
	new image center (from view tool).
Shift middle click	Uses the click position as the new image center (from view
	tool).
Shift right click	Zooms out of the image and the click position is used as the
	new image center (from view tool).
Shift middle drag	Moves the image (from view tool).

Table 5: Info tool mouse button assignment

4.5 Probe Tool

The probe tool allows the creation and modification of image probes. These probes define an image part and show statistical information about that part. The primary purpose of an image probe is to give information about the background.

Figure 19 shows the the graphical representation of an image probe and the dialog window with the statistical information. It is possible to automatically recenter the image probe by using the intensity peak, intensity mean or fourier phase (see section 12 "Recentering objects").

The dialog window of an image probe shows the following information:

- 1. The label of an image probe is automatically created and updated.
- 2. The position and size of an image probe uses integral values. It is possible to recenter the image probe or set the position and size explicitly. The **Update** button recalculates the statistical information after changing the position or size of an image probe.
- 3. The sky background specification is based on the image probe area. These values give only a rough estimate of the sky background.
- 4. The integral intensity (using all pixels inside the probe) and number of pixels are shown here.
- 5. The smallest and largest pixel value in the probe area and their positions (relative to the complete image) are shown.

Mouse button	Function
Left click	Creates a new image probe at the mouse position.
Middle click	Shows the dialog window containing all information about
	the selected image probe.
Right click	Deletes the selected image probe.
Control left click	Enlarges the size of the selected image probe by a factor of
	two.
Control right click	Shrinks the size of the selected image probe by a factor of
	two.
Middle drag	Moves the selected image probe to a different position.
Shift left click	Zooms into the image and the click position is used as the
	new image center (from view tool).
Shift middle click	Uses the click position as the new image center (from view
	tool).
Shift right click	Zooms out of the image and the click position is used as the
	new image center (from view tool).
Shift middle drag	Moves the image (from view tool).

Table 6: Probe tool mouse button assignment



Figure 19: Graphical representation and dialog window of an image probe.

- 6. The centroid is equal to the intensity mean.
- 7. The statistical data gives useful information about the sky background:
 - If the mean and variance have the same value, the background is dominated by poisson noise and no addditional point sources exist inside the probe area. The mean value then gives a good estimate for a constant sky background.
 - A skew value not equal to zero (including a small uncertainty range) is a strong hint that

the background is contaminated with weak point sources.

4.6 Pixel Tool

The pixel tool and the accompanying pixel object shows information about one pixel and compares the value to the nearest eight pixels.

Mouse button	Function
Left click	Creates a new pixel probe at the mouse position.
Middle click	Shows the dialog window containing all information about
	the selected pixel.
Right click	Deletes the selected pixel probe.
Middle drag	Moves the selected pixel probe to a different position.
Shift left click	Zooms into the image and the click position is used as the
	new image center (from view tool).
Shift middle click	Uses the click position as the new image center (from view
	tool).
Shift right click	Zooms out of the image and the click position is used as the
	new image center (from view tool).
Shift middle drag	Moves the image (from view tool).

Table 7: Pixel tool mouse button assignment

Figure 20 shows the graphical representation of a pixel probe and the dialog window with the statistical information. It is possible to automatically recenter the pixel probe by using the intensity peak, intensity mean or fourier phase (see section 12 "Recentering objects").

		Pixel (on sei	rvice32)	_ O ×	
	Label:	Pixel 1			
	Position:	Intensity Peak 🗵	Recenter		
	x=	ĭ 1691.00	y=	ĭ 1491.00	2
	Values:	-1	Û	+1	
and the second se	+1	ĭ 5602₊00	Ĭ 16285₊0	ĭ 11685.0	
4	0	ĭ 5862₊00	16585.0	ĭ12071₊0	3
	-1	ĭ 5462₊00	Ĭ <u>15525</u> ₊0	ľ 11138₊0	
	Statistics: mean=	10453.8	median=	11685.0	4
	var=	1.91231e+07	sigma=	ž 4373₊00	
	Deviation:	ĭ 1.40207	5		
	Apply Close U	pdate			

Figure 20: Graphical representation and dialog window of an image pixel.

The dialog window of an image pixel shows the following information:

1. The label of an image pixel is automatically created and updated.

20

- 2. The position of an image pixel uses integral values. It is possible to recenter the image pixel or set the position explicitly. The **Update** button recalculates the statistical information after changing the position of an image pixel.
- 3. The values of the selected image pixel and the eight neighbors are shown. The orientation in the dialog window and in the displayed image is identical.
- 4. The statistical information uses only the neighbour pixels; the central pixel is ignored.
- 5. The deviation describes the pixel value as $mean + \sigma \times deviation$. If the deviation is too big, the central pixel may be a defect pixel.

5 How to calibrate the raw data

5.1 Introduction

The standard calibration algorithm executes the following steps for each raw data image (if requested, see the dialog window in figure 21):

- 1. Subtracts the sky image from the raw image.
- 2. Divides the result by the flatfield.
- 3. The value of each bad pixel is replaced by a value calculated from some neighbor pixels.

Important:

Whether or not you are allowed to subtract the sky background depends on the algorithm used in the reconstruction. For example, in the Richardson-Lucy deconvolution algorithm, the target image must contain the sky background. In this case, the sky background must be kept and subtracted later during the psf preprocessing step.



Figure 21: Standard calibration setup dialog.

The defect pixel interpolation is provided by an algorithm which uses up to two pairs of good pixels. The pixels of a pair are on the opposite side of a defect pixel, and an interpolation schema calculates the replacement value.

Additional information:

• 14 "Defect pixel interpolation"

6 How to determine the preprocessing parameters

6.1 Introduction

Because all reconstruction algorithms require identical mapping of the target images onto the sky, the preprocessing is an essential step. The standard preprocessing algorithm uses five parameters to rotate, scale and translate an image. Therefore, the parameters specified in the dialog window (see figure 22) must be precise.

Preproce //iri/irisrv/home/m	ssing Setup (on s hein/soft/lndrs/data/l	ervice64) 4GC4151/cal.fi	
2 Image: <	þ	≥ 🗆 🖾 Ret	erence 3
Transformation:	Cubic 🗹	4	
Parameters:	Translation from St	ars 🔟 Estima	ate 5
Scale: sx=	1,00000	sy= Ĭ	1,00000
Rotate:	ĭ 310₊000	[•
Translate: x=	Ŭ 0.00000	9= I	0.00000
Apply Close			

Figure 22: Dialog window containing the preprocessing parameters.

The following parameters are presented in the preprocessing setup:

- 1. The list of files containing calibrated images is used to select the file for which the preprocessing parameters should be inspected and/or modified.
- 2. If a file contains more than one image (different position angles), the <- and >-buttons are used to select a specific image number.
- 3. It is possible to specify an image as a reference image. The orientation of that image is then used as a reference if the transformation parameters of the other images must be calculated.
- 4. Several algorithms are available for transforming an image:
 - Nearest: This is a fast transformation algorithm.
 - Bilinear: A simple bilinear interpolation schema is used for the transformation.
 - Cubic: A more sophisticated cubic interpolation schema is used.
 - Fourier: A fourier space method is used for rotation and translation.
- 5. There are several ways to determine the transformation parameters:
 - Use FITS Header: The instrument-specific keyword values in the FITS header of the calibrated data are used to calculate the transformation parameters. This method is not implemented!

• Rotation from Stars: The reference stars are used to calculate the rotation angle.

Issue 0.1

- Translation from Stars: The reference stars are used to calculate the translation vector.
- 6. All available transformation parameters use up to five parameters for scaling, rotation, and translation. Image scaling is currently not implemented.

If the FITS header keywords are available and sufficient, they are preferred over a calculation of the transformation parameters. If an explicit calculation of the parameters is required, a set of reference points (reference stars) is needed. These reference stars are created by using the reference star tool. During an inspection of the calibrated data, several reference points or stars are identified. At least two stars are needed in order to calculate the rotation angle. All reference stars must be defined in each calibrated image, and each star must have the same reference number in each image. Figure 23 shows a calibrated image with two defined reference stars. Because the calculation of the rotation angle or translation vector works only between two images, one image must be marked as the reference image (see **Reference** checkbox in the preprocessing parameter dialog window).



Figure 23: Calibrated data.

6.2 Reference Star Tool

The reference star tool (see section 4.2 "Tools") provides the usual image navigation interface (zooming and panning of an image) and allows the creation, modification, and deletion of reference stars. The dialog window of a reference star shows the following information:

1. The label of a reference star is automatically created and updated.

Mouse button	Function
Left click	Creates a new reference star at the mouse click position.
Middle click	Shows the dialog window containing all information about
	the selected reference star.
Right click	Deletes the selected reference star.
Middle drag	Moves the selected reference star to a different position.
Shift left click	Zooms into the image, and the click position is used as the
	new image center (from view tool).
Shift middle click	Uses the click position as the new image center (from view
	tool).
Shift right click	Zooms out of the image, and the click position is used as the
	new image center (from view tool).
Shift middle drag	Moves the image (from view tool).

Table 8: Reference star tool mouse button assignment



Figure 24: Graphical representation and dialog window of a reference star.

- 2. The position of a reference star uses fractional values. It is possible to recenter the reference star or set the position explicitly. The **Update** button recalculates the statistical information after changing the position of a reference star.
- 3. The values of the pixel in the center of the reference star and the eight neighbors are shown. The orientation in the dialog window and in the displayed image is identical.
- 4. The statistical information uses only the neighbor pixels; the central pixel is ignored.
- 5. The deviation describes the pixel value as $mean + \sigma \times deviation$. If the deviation is too big, it is possible that the central pixel is a defect pixel.

6.3 Estimating the Transformation Parameters

The code snippet in source 4 contains the method of calculating the rotation angle of an image according to the fixed rotation angle of a reference image. For each reference star, the basic algorithm calculates the angle between the x-axis and the line from the center of the reference star set (center of gravity) to that reference star. The distance of that star from the center of gravity determines the significance (weight) of the calculated angle.

```
; refAngle
               : rotation angle of the reference image
               : x coordinate of a star in the reference image
; refPos [0,*]
; refPos[1,*] : y coordinate of a star in the reference image
; refCX, refCY : center of gravity of all stars in the reference image
 imgPos[0,*] : x coordinate of a star in the current image
; imgPos [1,*]
              : y coordinate of a star in the current image
; imgCX, imgCY : center of gravity of all stars in the current image
angle = 0.0
weight = 0.0
FOR nr=0, nstars -1 DO BEGIN
    dx = refPos[0, nr] - refCX
    dy = refPos[1, nr] - refCY
    d = SQRT(dx*dx + dy*dy); the distance from the center as weight
    refAngle = ATAN(refPos[1, nr] - refCY, refPos[0, nr] - refCX)
    imgAngle = ATAN(imgPos[1, nr] - imgCY, imgPos[0, nr] - imgCX)
    diffAngle = imgAngle - refAngle
    IF diffAngle GT !PI THEN diffAngle -= 2.0*!PI
    IF diffAngle LT -!PI THEN diffAngle += 2.0*!PI
    angle += d*diffAngle
    weight += d
ENDFOR
angle = weight
```

Source 4: Code snippet for calculating the rotation angle.

The code snippet in source 5 contains the method of calculating the translation vector of an image according to the fixed position of a reference image. This method requires a precise rotation angle because this angle is used to rotate the reference stars in an image and calculate the vectors between the same stars in the reference image.

; refAngle : rotation angle of the reference image	
; $refPos[0,*]$: x coordinate of a star in the reference ima	ge
; $refPos[1,*]$: y coordinate of a star in the reference ima	ge
; $imgPos[0,*]$: x coordinate of a star in the current image	
; $imgPos[1,*]$: y coordinate of a star in the current image	
phi = !DTOR*(r - refAngle)	
FOR $nr=0$, $nstars -1$ DO BEGIN	
xs = imgPos[0, nr] * COS(phi) - imgPos[1, nr] * SIN(phi)	
ys = imgPos[0, nr] * SIN(phi) + imgPos[1, nr] * COS(phi)	
$\operatorname{transX}[\operatorname{nr}] = \operatorname{refPos}[0, \operatorname{nr}] - \operatorname{xs}$	
$\operatorname{transY}[\operatorname{nr}] = \operatorname{refPos}[1, \operatorname{nr}] - \operatorname{ys}$	
ENDFOR	
tx = MEAN(transX)	
ty = MEAN(transY)	

Source 5: Code snippet for calcuating the translation vector.

6.4 Recentering a Reference Star

If a closer look at a reference star (high magnification) shows that the position is not correct, some methods to automatically recenter the graphical object exist:

- Intensity Peak: Searches the highest pixel value in a part of the image (20 pixels), which is then used as the new position. The new position has integral coordinates. This method is sensitive to noise and can give wrong coordinates for weak reference stars.
- **Intensity Mean**: Uses the weighted center of all pixels in a small window that have values above a certain limit. The new position has fractional coordinates. This method is not sensitive to a tilted background (for example, the object halo).
- Fourier Phase: Uses the fourier phase of the smallest frequency in a small image part as a displacement vector. This method is sensitive to a tilted background.

Additional information:

• Section 12 "Recentering objects"

7 How to extract a target

7.1 Introduction

The reconstruction of an object needs a set of target and psf images. At least one psf image is needed for each target image. The exact configuration (e.g. one image for each position angle or a summed up image) depends on the algorithm used for the reconstruction. The target preprocessing needs the part of the preprocessed data which contains the target and some additional information (e.g. sky background) as an input. Normally, the following steps are necessary:

- 1. Find and define the area of the target in the preprocessed data.
- 2. Specify the form and level of the sky background if it was not removed during the calibration step.

In figure 25, an image window shows the target in the center, two psf stars near the corners, and an image probe in the upper left corner. The image probe is used to estimate the sky background in the image. Alternatively, an image probe in a sky image can be used.

LN DRS Image GUI (on service32)	_ 0 ×
Recipe Algorithm File Image Help	
File: Vhome/mhein/soft/lndrs/data/NGC4151/int.fits Image: < > Color: BlackE	ody <u>z</u>
Tools: 🕂 🕂 🖉 ? 🖾 🗷 🞯	
Prote 1	
Target 1	
Pst 2	



7.2 Target Tool

The purpose of the target tool (see section 4.2 "Tools") is to specify the target area in at least the first image of each preprocessed data file. This target area is used to cut out the part of an image later

used for reconstruction. The target definition is the major input for the **Target Preprocessing** step.

Mouse button	Function
Left click	Creates a new target object centered at the mouse position.
Middle click	Shows a dialog window with the details of the selected target
	object (figure 26).
Right click	Removes the target object at the selected position.
Middle drag	Moves the selected target object to a different position.
Control left click	Enlarges the selected target object by a factor of 2.
Control right click	Shrinks the selected target object by a factor of 2.
Shift left click	Zooms into the image, and the click position is used as the
	new image center (from view tool).
Shift middle click	Uses the click position as the new image center (from view
	tool).
Shift right click	Zooms out of the image, and the click position is used as the
	new image center (from view tool).
Shift middle drag	Moves the image (from view tool).

Table 9: Target tool mouse button assignment

In the preprocessed data, exactly one target area must be defined. The graphical representation is shown in figure 26. When selecting the graphic object with the middle mouse button (target tool) or the left mouse button (info tool), a dialog box containing all parameters for that target (figure 26 right panel) appears.

	🗖 Ta	arget (on servic	e64)		
	Label:	∬arget 1	1		
	Position:	Intensity Peak 🗵	Recenter	J	
	x=	ĭ 896.000	y= Ĭ	896.000	1
	Size: width=	ľ 256₊000	height=	256,000	
Target 1	Sky:	Constant 🛛 🗹	Estimate		
	p1=	ž 223,700	р2= Ĭ	906,955	B
	p3=	ž 906₊955	р4= Ĭ	906,955	
	Apply Close				

Figure 26: Graphical representation and dialog window of a target.

The target dialog box contains the following information:

- 1. The label (automatically generated).
- 2. The position and size of the target area. The position gives the coordinates of the lower left corner. It is possible to recenter the target area using one of the available methods.

3. The sky background subtracted from each extracted target is specified by one or four parameters.

The sky background must be defined even if it should be kept during the target extraction, because some algorithms require the background level to begin. The overall shape and the respective parameters must be specified in the dialog window.

7.3 Target Extraction

30

The target preprocessing is executed like the calibration or preprocessing step. The algorithm-specific dialog window is presented in figure 27; in figure 28, an extracted target is shown.

🗖 Target Creatio 🗕 🗖 🗙	
Add: One 🗹	1
Background: 🗖 Remove	2
Fourier Mask: 🗖 Use	3
Centering: 🗖 Recenter	4
Apply Close	

Figure 27: The algorithm-specific target creation setup dialog.

The setup dialog box contains the following elements:

- 1. Depending on the algorithm used for the reconstruction, different kinds of target images are generated:
 - **One**: All target images of all files and all position angles are summed up into exactly one target image.
 - File: For each preprocessed file, a target image is generated.
 - Image: For each image in each preprocessed file, a target image is generated.
- 2. Check if the sky background should be removed.
- 3. Check if a fourier mask should be used. This mask sets all values outside the transfer function to zero.
- 4. Check if the target should be recentered using the fourier phase.

Important:

Whether or not you are allowed to subtract the sky background depends on the algorithm used in the reconstruction. For example, in the Richardson-Lucy deconvolution algorithm, the target image must contain the sky background. In this case, the sky background must be kept and subtracted later during the psf preprocessing step.

Additional information:

- Section 12 "Recentering objects"
- Section 13 "Sky background"



Figure 28: The target image created from the target preprocessing step.

8 How to define a psf

8.1 Introduction

Before extracting a psf from the preprocessed images, at least one psf star must be marked in the first image fo each file. In the provided example, two psf stars are already marked and ready to use. The psf preprocessing is executed like the target preprocessing step. The psf preprocessing needs the part of the preprocessed data containing at least one psf star and some additional information (e.g. sky background) as an input. Normally, the following steps are necessary:

- 1. Find and define at least one psf star in the preprocessed data.
- 2. Specify the form and level of the sky background if it was not removed during the calibration step.
- 3. If a psf star is influenced by an extended target, a target halo must be specified.

In figure 29, an image window shows the target in the center, two psf stars near the corners and an image probe in the upper left corner. The image probe is used to estimate the sky background in the image. Alternatively, an image probe in a sky image can be used.



Figure 29: Preprocessed interferogram with a target object, two psf stars and an image probe.

8.2 PSF Star Tool

The purpose of the psf star tool (see section 4.2 "Tools") is to specify at least one psf star in the first image of each preprocessed data file. This psf star is used to cut out the part of an image later used for reconstruction. The psf star definition is the major input for the **Psf Preprocessing** step.

Mouse button	Function
Left click	Creates a new psf star centered at the mouse position.
Middle click	Shows a dialog window with the details of the selected psf
	star (figure 30).
Right click	Removes the psf star at the selected position.
Middle drag	Moves the selected psf star to a different position.
Control left click	Enlarges the selected psf star by a factor of 2.
Control right click	Shrinks the selected psf star by a factor of 2.
Shift left click	Zooms into the image and the click position becomes the
	new image center (from view tool).
Shift middle click	Uses the click position as the new image center (from view
	tool).
Shift right click	Zooms out of the image and the click position becomes the
	new image center (from view tool).
Shift middle drag	Moves the image (from view tool).

Table 10: Psf star tool mouse button assignment

In the preprocessed data, at least one psf star must be defined. If more stars are used as psf stars, either the quality of the extracted psf will increase or a space variant psf can be used for deconvolution. When selecting the graphic object with the middle mouse button (psf star tool) or the left mouse button (info tool), a dialog box appears which contains all parameters for that psf star (figure 30, right panel)



Figure 30: Graphical representation and dialog window of a psf star.

The psf star dialog box contains the following elements:

- 1. The label (automatically generated).
- 2. The position and size of the psf star. The position gives the center of the psf. It is possible to recenter the psf star using one of the available methods.
- 3. The sky background subtracted from each extracted psf (specified by one or four parameters).
- 4. If the psf star is influenced by the target halo, the type of compensation is specified here.

The sky background must be defined even if it will be kept during the psf extraction, because some algorithms need the background level. The overall shape and the respective parameters must be specified in the dialog window.

8.3 Halo Tool

Sometimes a bright or extended target shows an extended halo, which can influence one or more psf stars. The halo must be removed from these psfs. The basis for this compensation is the target halo. It defines an area of the preprocessed image where the target halo is smooth and symmetric and can therefore be removed. The target halo tool (see section 4.2 "Tools") allows the creation and modification of such a target halo.

Mouse button	Function
Left click	Creates a new target halo centered at the mouse position.
Middle click	Shows a dialog window with the details of the selected target
	halo (figure 31).
Right click	Removes the target halo at the selected position.
Middle drag	Moves the selected target halo to a different position.
Control left click	Enlarges the selected target halo by a factor of 2.
Control right click	Shrinks the selected target halo by a factor of 2.
Shift left click	Zooms into the image and the click position becomes the
	new image center (from view tool).
Shift middle click	Uses the click position as the new image center (from view
	tool).
Shift right click	Zooms out of the image and the click position becomes the
	new image center (from view tool).
Shift middle drag	Moves the image (from view tool).

Table 11: Halo tool mouse button assignment

In the preprocessed data, only one target halo can be defined and used. The graphical representation is shown in figure 31. When the graphic object is selected with the middle mouse button (halo tool) or the left mouse button (info tool), a dialog box containing all parameters for that target appears (figure 31, right panel).

The halo dialog box contains the following values:

- 1. The label is automatically generated.
- 2. The position of the target halo. The position gives the center of the halo and not the center of the target. It is possible to recenter the halo.

	Halo (on service	e32)		
Label:)Halo 1			
Position:	Intensity Mean 🗵	Recenter		2
x=	1018.00]	y=	1028.00 I	
Sky:	Constant 🗵	Estimate		
param1=	ž 224₊473	param2=	ž 224,473	3
param3=	ž 224₊473	param4=	ž 224 . 473	
Ring: inner=	Ĭ 200₊000	outer=	I 250,000	4
Limits: inner=	ĭ 150₊000	outer=	1 800.000	5
Angle:	ľ 135₊000	6		
Apply Close				

Figure 31: Graphical representation and dialog window of a target halo.

- 3. The sky background subtracted from the halo is specified by one or four parameters.
- 4. The inner and outer radii of the ring define the part of the halo used for recentering.
- 5. The inner and outer limits define the part of the image where the halo is sufficiently symmetric and smooth.
- 6. The angle defines a sector of the halo where no background stars influence the halo smoothness.

Because the sky background at the halo position can differ from the background at the psf star position, it is only necessary to compensate for the halo component. The sky background at the halo position must therefore be defined. Recentering the halo uses an approach in which only the pixels between the two dashed circles are used to calculate the intensity mean.

8.4 Psf Extraction

The psf preprocessing is carried out like the target preprocessing step. The algorithm-specific dialog window is presented in figure 32; an extracted psf is shown in figure 33.

PSF Creation :	
Add: One 🗹	1
Background: 🗖 Remove	2
Fourier Mask: 🗖 Use	3
Centering: 🗖 Recenter	4
Apply Close	

Figure 32: The algorithm-specific psf creation setup dialog.

The setup dialog box contains the following elements:

- 1. Depending in the algorithm used for the reconstruction, different kinds of psf images are generated:
 - **One**: All psf images of all files and all position angles are summed up into exactly one psf image.
 - File: For each preprocessed file, a psf image is generated.
 - Image: For each image in each preprocessed file, a psf image is generated.
 - **Object**: For each psf star in each image in each file, a psf image is generated. This allows a reconstruction with a space-variant psf.
- 2. Check if the sky background should be removed.
- 3. Check if a fourier mask should be used. This mask sets all values outside the transfer function to zero.
- 4. Check if the psf should be recentered using the fourier phase.



Figure 33: The psf image created from the psf preprocessing step.

Additional information:

- Section 12 "Recentering objects"
- Section 13 "Sky background"

9 How to reconstruct an object

9.1 Introduction

Reconstructing high-resolution images from several target images at different position angles and associated psfs is the primary task of the LINC-NIRVANA Data Reduction Software (LN DRS). All previous steps provide the input for the last deconvolution step. The kind of target and psf images depends on the algorithm selected for deconvolution. Typical images used by the Richardson-Lucy (RL) algorithm are shown in figure 34.



Figure 34: Left panel: a typical target used by the Richardson-Lucy deconvolution algorithm. Right panel: the associated psf.

The necessary parameters of each deconvolution algorithm can be set in the deconvolution algorithm dialog box and on the target and psf images. Further information is given in the reference section (see section 15 "Reconstruction algorithms"). A typical dialog window is shown in figure 35 (Richardson-Lucy).

🗖 Deconv	volution 🗕 🗆 🗙	Ŋ
[] Iterations:	I 5000	1
Show every:	Ĭ 50	2
Apply Clo	se	

Figure 35: Setup dialog window for the Richardson-Lucy algorithm.

In the dialog window, the following elements are shown:

1. The total number of iterations the algorithm should be use. The RL algorithm does not stop until this number of iterations is executed.

2. If the algorithm is written in IDL, this number means that the result from each specified iteration should be shown in the image window.

The execution of an algorithm written in IDL may be interrupted by using the menu item Algorithm/Pause or the Pause button below the algorithm list in the recipe GUI. With the menu item Algorithm/Resume or the Resume button below the algorithm list in the recipe GUI, the execution can be resumed. Figure 36 shows a reconstruction convolved with the ideal 23m psf in addition to the normal reconstruction.



Figure 36: Left panel: a reconstructed target using the Richardson-Lucy deconvolution algorithm. Right panel: the same image convolved with an ideal 23m psf.

Additional information:

• Section 15 "Reconstruction algorithms"

10 Software Architecture

10.1 Introduction

The LINC-NIRVANA Data Reduction Software (LN DRS) consists of a set of core IDL classes and LN DRS specific extension classes.

10.2 Class Inheritance Diagram



Figure 37: Class hierarchy of the framework core classes.

10.3 Class Dependency Diagram



Figure 38: Properties and dependencies of the framework core classes.

11 Tools

11.1 Introduction

11.2 Creating a new Tool

The IDL template shown in source 6 is a minimal skeleton for a new tool. For more sophisticated applications, additional functionality must be implemented. In this case, a user should take a closer look at the available source code. The template covers only the implementation of a simple tool. For example, the PSF star tool is based completely on this template.

```
FUNCTION </HYTOOL>::Init , view
self.dlgTitle = '<MYTITLE>'
self.cfgPlace = '<MYCFG>'
self.objClass = '<MYOBJCLASS>'
self.labelPrefix = '<MYPREFIX>'
; Additional initialization code
RETURN, 1
END
PRO <MYTOOL>::InitObject, obj, x, y
self-><SUPERCLASS>::InitObject, obj, x, y
; object specific extensions
END
PRO </HYTOOL>::ReadDynamicContent
self-><SUPERCLASS>::ReadDynamicContent
; update additional dialog elements (depends on the current object)
END
PRO </MYTOOL>::WriteDynamicContent
self-><SUPERCLASS>::WriteDynamicContent
; store additional dialog content in the current object
. . .
END
PRO </Pre>MYTOOL>::CreateContent
self-><SUPERCLASS>::CreateContent
row = self \rightarrow GetFreeRow()
; create additional dialog window elements
. . .
END
PRO <MYTOOL>...define
struct = { <\!\!MYTOOL\!\!>, $
           INHERITS <SUPERCLASS>, $
           ; tool specific extensions
           . . .
         }
END
```

Source 6: IDL template for a tool.

11.3 Description in the Configuration File

Each tool is described in a section of the configuration file "ldrs.cfg" (see source 7). These sections must follow the subsequent rules:

Placeholder	Description
< MYTOOL >	The IDL class name of the tool (CLASS element in the
	configuration file, for example "ldrsPSFTOOL").
$<\!\!SUPERCLASS\!>$	(for example "ldrsAREATOOL")
< <i>MYTITLE</i> >	The dialog title (for example "Psf").
< MYCFG >	The place in the parameter file (for example "PSFSTAR").
$<\!MYOBJCLASS\!>$	The tool specific graphic object (for example "ldrsPSFS-
	TAR").
< <i>MYPREFIX</i> >	The prefix used in the graphical representation (for example
	"Psf ")

LN-MPIFR-MAN-ASW-001

Table 12: Tool template placeholders

- The section is a child of a "TOOL" section. Each tool section must have a unique name (for example "TOOL4").
- The section must have a "TAG" element which contains an internally used tag.
- The "LABEL" element contains the GUI label.
- The IDL class name is specified with a "CLASS" element.
- The tool icon is specified with 24 strings, where a "*" character represents black pixels in the icon. Each string must contain exactly 24 characters.

Source 7: Excerpt from the configuration file for a tool.

12 Recentering objects

12.1 Introduction

It is occasionally necessary to recenter an image object (reference star, psf star, or target) because the current position is not centered (misplaced). The available algorithms are specified in the configuration file "ldrs.cfg". In this file, a special procedure group is defined, where each entry gives the name of an existing IDL procedure. Source 8 shows a sample of the configuration file:

```
PGROUP8.TAG = "CALCCENTER"

PGROUP8.LABEL = "Calculate Image Center"

PGROUP8.PROC1.TAG = "PEAK"

PGROUP8.PROC1.LABEL = "Intensity Peak"

PGROUP8.PROC1.PRO = "LDRS_CALC_CENTER_PEAK"

PGROUP8.PROC2.TAG = "MEAN"

PGROUP8.PROC2.LABEL = "Intensity Mean"

PGROUP8.PROC2.PRO = "LDRS_CALC_CENTER_MEAN"

PGROUP8.PROC3.TAG = "PHASE"

PGROUP8.PROC3.LABEL = "Fourier Phase"

PGROUP8.PROC3.PRO = "LDRS_CALC_CENTER_PHASE"
```

Source 8: Snippet from the configuration file concerning recentering.

The tag "CALCCENTER" is used in the framework and must not be changed. The label "Calculate Image Center" is used in the graphical user interface and can be changed to a different string. The tags for the available procedures ("PEAK", "MEAN", "PHASE", ...) should not be modified or the existing parameter files will be jeopardized. The standard distribution of the LN DRS contains three methods:

- LDRS_CALC_CENTER_PEAK: Uses the pixel with the largest value inside a small area as the new center.
- LDRS_CALC_CENTER_MEAN: Uses the weighted center of all pixels in a small window that have values above a certain limit.
- LDRS_CALC_CENTER_PHASE: Uses the fourier phase of the smallest frequency in a small image part as a displacement vector.

12.2 Creating a new Method

The template in source 9 can be used as a starting point to implement a new recentering method.

; IN image : $PTR_NEW(FLTARR(nx, ny, nz))$	image cube
; IN x : FLOAT	old center
; IN y : FLOAT	old center
; IN z : FLOAT	image plane
; $OUT \ cx$: $FLOAT$	new image center
; OUT cy : FLOAT	new image center
PRO <my_pro_name>, image, x, y, z, cx, cy,</my_pro_name>	WINSIZE
IF N_ELEMENTS(ws) EQ 0 THEN ws = 32	
s = SIZE(image, /DIMENSIONS)	
left = FIX(x - ws)	
right = FIX(x + ws)	
bottom = $FIX(y - ws)$	
top = FIX(y + ws)	
IF left LT 0 THEN left $= 0$	
IF right GE $s[0]$ THEN right = $s[0]-1$	
IF bottom LT 0 THEN bottom $= 0$	
IF top GE $s[1]$ THEN top $= s[1]-1$	
w=right-left+1	
h=top-bottom+1	
n = w * h	
FND	

Source 9: Recentering method template.

In the configuration file, the following excerpt from source 10 must be appended immediately after the last method concerning recentering (see source 8).

PGROUP8.PROC<N>.TAG = "<MY_TAG>" PGROUP8.PROC<N>.LABEL = "<MY_LABEL>" PGROUP8.PROC<N>.PRO = "<MY_PRO_NAME>"

Source 10: Part of a configuration file concerning a new recentering method.

The placeholders $\langle MY_PRO_NAME \rangle$, $\langle MY_TAG \rangle$, $\langle MY_LABEL \rangle$, and $\langle N \rangle$ must be replaced by the correct values.

12.3 Recentering Methods

In the LN DRS distribution three methods to automatically recenter a graphical object (reference star, psf star, target) exist:

- Intensity Peak: Searches the highest pixel value inside a small image part which is used as the new position. The new position has integral coordinates. This method is sensitive to noise and can give wrong coordinates for weak reference stars.
- Intensity Mean: Uses the weighted center of all pixels in a small window which have values above a certain limit. The new position has fractional coordinates. This method is not sensitive to a tilted background (for example, the object halo).

• Fourier Phase: Uses the fourier phase of the smallest frequency in a small image part as a displacement vector. This method is sensitive to a tilted background.

12.3.1 Recentering using the Intensity Peak

```
; left, right, bottom, top : borders of a small image part
; w,h : width and height of that image part
; z : image number
ipart = image[left:right,bottom:top,z]
m = MAX(ipart, idx)
cx = left + idx MOD w
cy = bottom + idx / w
```

Source 11: Excerpt of code for recentering using the intensity peak.

12.3.2 Recentering using the Intensity Mean

```
; left, right, bottom, top : borders of a small image part
; w,h : width and height of that image part
; z : image number
ipart = image[left:right,bottom:top,z]
tot = TOTAL(ipart)
\max = MAX(ipart, imaxIdx)
mean = tot/n
sumx = 0.0
sumv = 0.0
sum = 0.0
limit = mean + 0.1*(imax - mean)
FOR i=0, w-1 DO BEGIN
    FOR j=0,h-1 DO BEGIN
        IF ipart[i,j] GT limit THEN BEGIN
            sum += ipart[i, j]
            sumx += i*ipart[i,j]
            sumy += j*ipart[i,j]
        ENDIF
   ENDFOR
ENDFOR
cx = left + sumx/sum
cy = bottom + sumy/sum
```

Source 12: Excerpt of code for recentering using the intensity mean.

12.3.3 Recentering using the Fourier Phase

```
; left, right, bottom, top : borders of a small image part
; w,h : width and height of that image part
; z : image number
ipart = image[left:right, bottom:top,z]
ipfft = FFT(ipart)
hcx = ipfft[1,0]/ABS(ipfft[1,0])*COMPLEX(-1.0, 0.0)
hcy = ipfft[0,1]/ABS(ipfft[0,1])*COMPLEX(-1.0, 0.0)
xshift = 0.5*ATAN(IMAGINARY(hcx), REAL_PART(hcx))/!PI*w;
yshift = 0.5*ATAN(IMAGINARY(hcy), REAL_PART(hcy))/!PI*h;
cx = left + w/2 - xshift
cy = bottom + h/2 - yshift
```

Source 13: Excerpt of code for recentering using the fourier phase.

13 Sky background

13.1 Introduction

```
PGROUP1.TAG = "SKYC"
PGROUP1.LABEL = "Sky Compensation"
PGROUP1.PROC1.TAG = "NONE"
PGROUP1.PROC1.LABEL = "None"
PGROUP1.PROC1.PRO = "LDRS_SKY_NONE"
PGROUP1.PROC2.TAG = "CONST"
PGROUP1.PROC2.LABEL = "Constant"
PGROUP1.PROC2.PRO = "LDRS_SKY_CONSTANT"
PGROUP1.PROC3.TAG = "BILIN"
PGROUP1.PROC3.LABEL = "Bilinear"
PGROUP1.PROC3.PRO = "LDRS_SKY_BILINEAR"
PGROUP2.TAG = "SKYPE"
PGROUP2.LABEL = "Sky Parameter Estimation"
PGROUP2.PROC1.TAG = "NONE"
PGROUP2.PROC1.LABEL = "None"
PGROUP2.PROC1.PRO = "LDRS_SKY_NONE_GUESS"
PGROUP2.PROC2.TAG = "CONST"
PGROUP2.PROC2.LABEL = "Constant"
PGROUP2.PROC2.PRO = "LDRS_SKY_CONSTANT_GUESS"
PGROUP2.PROC3.TAG = "BILIN"
PGROUP2.PROC3.LABEL = "Bilinear"
PGROUP2.PROC3.PRO = "LDRS_SKY_BILINEAR_GUESS"
```

Source 14: Excerpt from the configuration file concerning the sky background.

14 Defect pixel interpolation

14.1 Introduction

The calibration step uses an algorithm to calculate replacement values for defective pixels. The available algorithms are specified in the configuration file "ldrs.cfg". In this file, a special procedure group is defined in which each entry gives the name of an existing IDL procedure. The code snippet in source 15 shows a sample part of the configuration file:

PGROUP3.TAG = "BPMC" PGROUP3.LABEL = "Bad Pixel Compensation" PGROUP3.PROC1.TAG = "NONE" PGROUP3.PROC1.LABEL = "None" PGROUP3.PROC1.PRO = "LDRS_BPM_NONE" PGROUP3.PROC2.TAG = "STD" PGROUP3.PROC2.LABEL = "Nearest Pairs Interpolation" PGROUP3.PROC2.PRO = "LDRS_BPM_PAIR_INTERPOLATION"

Source 15: Excerpt from the configuration file concerning the defective pixel interpolation.

The tag "BPMC" is used in the framework and must not be changed. The label "Bad Pixel Compensation" is used in the graphical user interface and can be changed to a different string. The tags for the available procedures ("NONE", "STD", ...) should not be modified or the existing parameter files will be jeopardized. The standard distribution of the LN DRS contains two methods:

- LDRS_BPM_NONE: The defective pixels are not modified.
- LDRS_BPM_PAIR_INTERPOLATION: The values of the defective pixels are replaced by a value calculated from up to two pixel pairs.

14.2 Creating a new Method

The template in source 16 can be used as a starting point to implement a new defective pixel interpolation method.

```
; INOUT data : FLTARR(nx, ny)
         bpm : PTR_NEW(FLTARR(nx, ny))
; IN
PRO <MY_PRO_NAME>, data, bpm
; check the dimensions of the data and the bpm
ds = SIZE(data, /DIMENSIONS)
bs = SIZE(*bpm, /DIMENSIONS)
IF (ds[0] \mathbb{N}E bs[0]) OR (ds[1] \mathbb{N}E bs[1]) THEN BEGIN
    ; the dimensions differ \implies not \ compatible
    PRINT, '</PRO_NAME>:_dimensions_of_data_and_BPM_differ!'
    REIURN
ENDIF
nx = ds [0]
ny = ds [1]
; bi are the 1-dimensional indices of the bad pixels
bi = WHERE(*bpm EQ 0.0, count)
END
```

Source 16: Defective pixel interpolation method template.

In the configuration file, the following excerpt must be appended immediately after the last method concerning the defective pixel interpolation (see source 15).

PGROUP3.PROC<N>.TAG = "<MY_TAG>" PGROUP3.PROC<N>.LABEL = "<MY_LABEL>" PGROUP3.PROC<N>.PRO = "<MY_PRO_NAME>"

Source 17: Part of a configuration file concerning a new defective pixel interpolation.

The placeholders $<\!MY_PRO_NAME>$, $<\!MY_TAG>$, $<\!MY_LABEL>$, and $<\!N>$ must be replaced with the correct values.

14.3 Interpolation Methods

In the LN DRS distribution, two methods to calculate replacement values for defective pixels exist:

- None: This method does not calculate a replacement value.
- Nearest Pairs Interpolation: Uses up to two pairs of pixels in the neighborhood of the defective pixel to calculate a replacement value.

14.4 Nearest Pairs Interpolation

This method uses up to two pairs of good pixels to calculate a replacement value for a defective pixel. The pixels of a pair are on the opposite side of a defective pixel. The algorithm is described in the following code snippets.

```
; replacement pixels are positioned horizontal, vertical, or
; diagonal relative to the bad pixel:
  0 | 1 | 2
;
  3
     \mid X \mid
           4
;
  5 | 6 | 7
;
; the 2-dimensional indices of a bad pixel are (bx, by)
rpf = INTARR(8); replacement pixel flag
rpw = FLTARR(8); replacement pixel weight/dist
rpv = FLTARR(8); replacement pixel value
rpdx = [-1, 0, 1, -1, 1, -1, 0, 1]
rpdy = [1, 1, 1, 0, 0, -1, -1, -1]
; search the replacement pixels
FOR j = 0,7 DO BEGIN
    rx = bx + rpdx[j]
    ry = by + rpdy[j]
    rpw[j] = 1.0
    WHILE rpf [j] EQ 0 DO BEGIN
        IF (rx LT 0) OR (rx GE nx) THEN BREAK
        IF (ry LT 0) OR (ry GE ny) THEN BREAK
        IF (*bpm) [rx, ry] EQ 0.0 THEN BEGIN
             ; the replacement pixel is also a bad pixel
             ; \Rightarrow move to the next replacement pixel
             rx += rpdx[j]
             ry += rpdy[j]
             rpw[j] += 1.0
            CONTINUE
        ENDIF
        ; we found a useful replacement pixel
        rpf[j] = 1
        rpv[j] = data[rx, ry]
    ENDWHILE
ENDFOR
```

Source 18: Determine the pixel pairs.

```
; recalculate the replacement values and weights for pairs
FOR j=0,3 DO BEGIN
    k = 7 - j
    IF rpf [j] EQ 1 THEN BEGIN
        IF rpf[k] EQ 1 THEN BEGIN
             ; we have a pair, recalculate the value and weight
             rpv[j] = (rpv[j]*rpw[k] + rpv[k]*rpw[j])/(rpw[j] + rpw[k])
             \operatorname{rpw}[j] = \operatorname{rpw}[j]^{2} + \operatorname{rpw}[k]^{2}
             ; remove the weight at k
             rpf[k] = 0
             rpw[k] = !VALUES.F_NAN
             rpv[k] = 0.0
        ENDIF ELSE BEGIN
             ; only a value at j exists, recalculate the weight
             \operatorname{rpw}[j] = 2 \operatorname{rpw}[j]^2 + 1; a little bit worse than a pair!
             ; remove the weight at k
             rpw[k] = !VALUES.F_NAN
        ENDELSE
    ENDIF ELSE BEGIN
        IF rpf [k] EQ 1 THEN BEGIN
             ; only a value at k exists, recalculate the weight
             rpw[k] = 2*rpw[k]^2 + 1; a little bit worse than a pair!
             ; remove the weight at j
             rpw[j] = !VALUES.F_NAN
        ENDIF ELSE BEGIN
             ; no value at j or k, remove both weights
             rpw[j] = !VALUES.F_NAN
             rpw[k] = !VALUES.F_NAN
        ENDELSE
    ENDELSE
ENDFOR
```

Source 19: Calculate the replacement values.

```
; find the values with the smallest weights and use up to 2 values
vl = SORT(rpw)
; we have to distinguish between three different cases:
; 1. no replacement value exists (extremely rare)
; 2. only one replacement value exists (very rare)
; 3. two or more replacement values exist (normal case)
IF rpf[v1[0]] NE !VALUES.F_NAN THEN BEGIN
    v1 = rpv[v1[0]]
   w1 = rpw[v1[0]]
   IF rpf[vl[1]] NE !VALUES.F_NAN THEN BEGIN
        v2 = rpv[vl[1]]
        w2 = rpw[vl[1]]
        data [bx, by] = (v1*w2 + v2*w1)/(w1 + w2)
   ENDIF ELSE BEGIN
        data[bx, by] = v1
   ENDELSE
ENDIF
```

Source 20: Replacing the value of a defective pixel.

15 Reconstruction algorithms

15.1 Introduction

All reconstruction algorithms that should be used by the LN DRS must be implemented as a subclass of "ldrsDECONVOLVE" or an already implemented algorithm. The code excerpt in source 21 shows the part of the configuration file covering all aspects of the reconstruction step.

```
MODULE4.TAG = "REC"

MODULE4.LABEL = "Reconstruction"

MODULE4.COLOR = "51 153 255"

MODULE4.BOX = "3.25 1.75 3.75 0.75"

MODULE4.IN1 = "PSF"

MODULE4.IN2 = "TAR"

MODULE4.IN3 = "RPSF"

MODULE4.OUT1 = "IMG"

MODULE4.OUT2 = "REC"

MODULE4.ALG01.TAG = "SBB"

MODULE4.ALG01.LABEL = "Simple Building Block"

MODULE4.ALG01.CLASS = "ldrsDECONV_SBB"

MODULE4.ALG02.TAG = "RL"

MODULE4.ALG02.LABEL = "Richardson-Lucy"

MODULE4.ALG02.CLASS = "ldrsDECONV_RL"
```

Source 21: Excerpt from the configuration file concerning the reconstruction step.

Property	Meaning
TAG	The tag is used internally and in the parameter files.
LABEL	The label is used in the flow diagram.
COLOR	The color of the box in the flow diagram (red, green, blue).
BOX	The position and size of the box in the flow diagram (x, y, width, height).
IN	The tags of the files used as input.
OUT	The tags of the files produced as output.
ALGO <i>.TAG</i>	The tag of an algorithm.
ALGO <i>.LABEL</i>	The label of an algorithm used in the algorithm list.
ALGO <i>.CLASS</i>	The name of an IDL class that implements an algorithm.

In the code excerpt, the properties have the semantics listed in table 13.

Table 13: Properties of the reconstruction step in the configuration file.

15.2 Creating a New Algorithm

To implement a new reconstruction algorithm, the IDL class "ldrsDECONVOLVE" and one of the subclasses "ldrsDECONV_RL", "ldrsBUILDINGBLOCK" or "ldrsDECONV_SBB" should be inspected first.