Defectix user guide

A. BAILLARD for TERAPIX team (IAP)

May 16, 2005

Contents

Chapter 1

Introduction

1.1 Subject

Defectix must identify large artifacts directly from pixel data because it must create masks as pixel data as well. The main difficulty lies in the large scale range of defects. Compared to "real-life" images, lightning and perspective effects are not relevant for the analysis of astronomical data. For this reason machine learning is a simple and efficient approach. A supervised system based on neural networks should be able to treat arbitrary defects without the need to code any new detection algorithms. Our software deals with artifacts such as halos, diffraction spikes, satellite trails and scattered light. Once the network has been trained, Defectix acts as a non linear translation invariant filter.

This text corresponds to the documentation of defectix version 1.0.4.

Chapter 2

Technical overview

2.1 Neural networks

Neural networks used in Defectix are three layers Multi-Layer Perceptrons (MLP) as shown on figure 1.

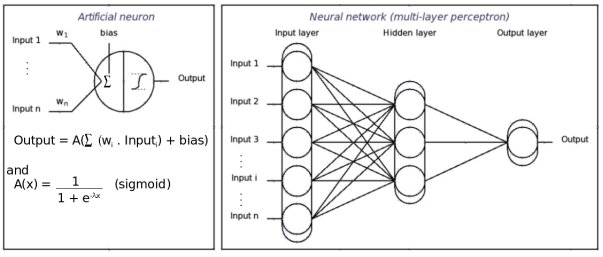

fig 1. Neuron and neural network

They are trained using rprop algorithm (M. Riedmiller and H. Braun), a fast batch back-propagation method.

2.2 Simulations

Original images are first rebinned to better match the seeing. A background model is then substracted. Defects are added with random positions, intensities and sizes. Corresponding masks are created simultaneously. Only one simulator is coded yet for halos. Images are dynamically compressed using the following function:

$$y(x) = \frac{1}{|x|} \ln\left(1 + \frac{x}{\sigma}\right)$$

where σ is the standard deviation of the global background.

2.3 Training

Principal Component Analysis (PCA) is carried out on blocks of n x n pixels. Only the k first components are used as inputs to the neural networks. Typically, $n = 8$ and $k = 16$ (among 64).

As shown on figure 2, the network is trained with:

- input: principal components of each block.
- output: value of the mask for the central pixel of the block.

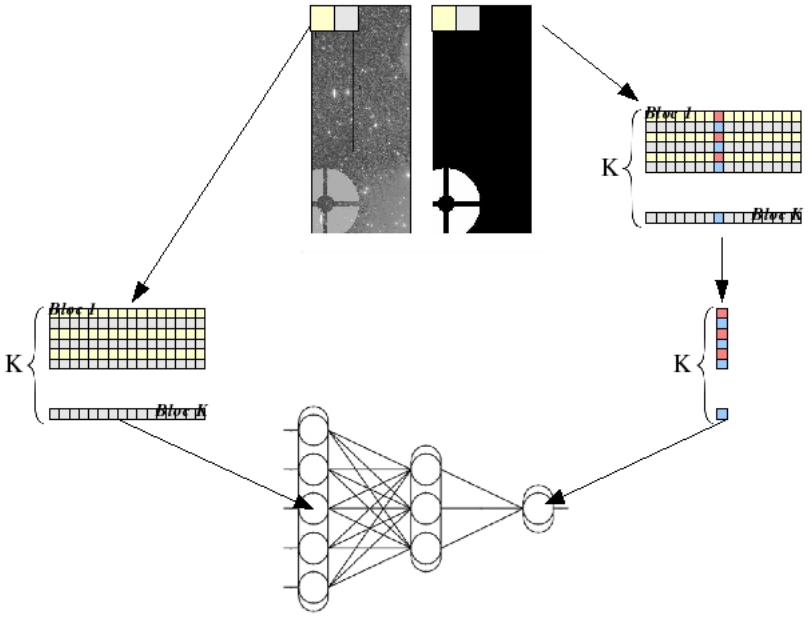

fig 2. Data used for training

2.4 Mask creation

Images are analysed as blocks as for training. Blocks slide along the image to compute a mask value for each pixel of the original image.

Figure 3 shows some results obtain with a real image.

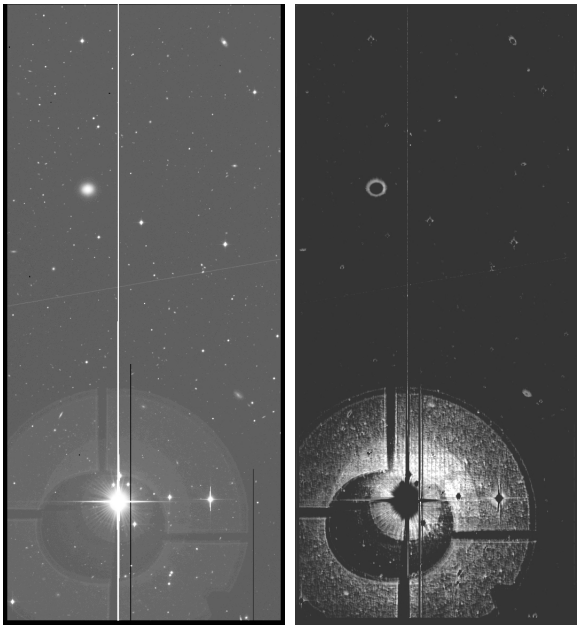

fig 3. Left: original image. Right: computed mask

2.5 Global process

Figure 4 shows the global process followed by defectix.

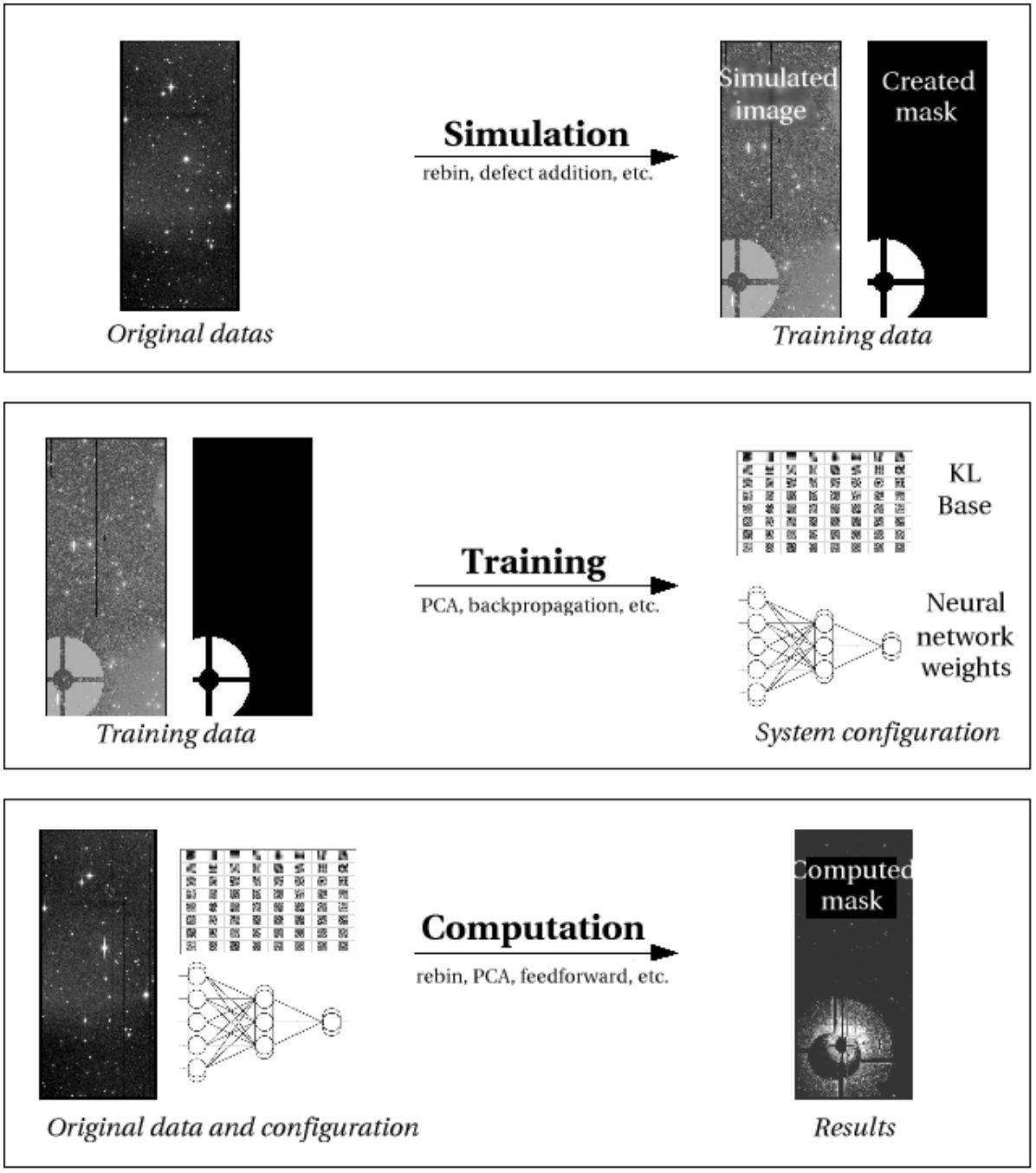

fig 4. Global process

2.6 Future

Defectix is still being developed and improved.

- Provide realistic simulations of a wilder range of defects
- Optimize and parallel code

- Bug reports

Chapter 3

Installation

3.1 Retrieve

Tarballs of defectix are available on Terapix team website: http://terapix.iap.fr

3.2 Install

Defectix was created using autotools so it is easy to install.

Get into the directory where you downloaded the tarball *defectix-X.Y.Z.tar.gz*, where *X.Y.Z* is the version. Type in:

$> tar xvzf defectix-X.Y.Z.tar.gz

The project is uncompressed in a directory called *defectix-X.Y.Z*. Type in:

$> cd defectix.X.Y.Z ; ./configure

Autotools will try to install defectix so you can compile the source. You need to have a version of cfitsio library installed. If every thing works, just type in:

$> make

and defectix will be compiled in *src* subdirectory. If you want to make defectix available for every users, log as root and type in:

$> make install

defectix will be install in */usr/local/bin* and configuration files will be copied in */usr/local/defectix*.

3.3 Documentation

If you have doxygen, you can create Defectix documentation by typing in:

$> cd dox ; doxygen doxconf

Chapter 4

How to use

4.1 Generalities

Defectix is invoked on command line. There are two ways to invoke defectix. The first one is to type each option on the command line. The second, and more efficient, one is to set each option in a configuration file and to choose this file on the command line.

The first options to know is *-h* or *–help*. Indeed, this option displays the list of available options :

$> ./defectix –help

```
Usage: defectix -M <mode> {-C <name> | [options]}

  -M, --mode <mode>             specify the defectix mode to use :
                                  halo
  trail
                                  training
                                  computing

  -C, --conf-file <name>        specify the configuration file to load
                                when loaded, command line options are ignored
  -I, --input-file <name>       specify the MEF file to treat
  -K, --mask-file <name>        specify the MEF file for masks
  -O, --output-file <name>      specify the MEF file for output (if needed)
  -L, --evec-file <name>        specify the gsl_matrix_float for the KL-base
  -N, --nn-file <name>          specify the neural network

Options:
  -d, --defaults                display default options and exit
  -h, --help                    display this help and exit
  -V, --version                 display the version and exit
  -v, --verbose                 display error messages

  -S, --substract-bg <bool>     ask for a background substraction on image
  -D, --reduce-dynamic <bool>   ask for a reduction of the image dynamic range
```

```
  -r, --rebin-factor <value>    specify a rebin factor for image loading
  -b, --bloc-size <value>       specify a size for PCA blocs

For 'trainer' and 'computer' mode only
  -p, --nb-pc <value>           specify the number of components to use
  -t, --nb-vector <value>       specicy the number of vectors used for training

  -e, --energy-treshold <value> define a treshold for neural network convergence
  -m, --max-iteration <value>   define a maximum number of iterations for
                                neural network convergence
  -x, --max-prior <value>       define a maximum intensity value for priors
  -n, --min-prior <value>       define a minimum intensity value for priors
  -l, --non-prior <value>       define a part of blocs to use without priors

  -s, --mask-treshold <value>   define a treshold for mask creation

  -E, --print-evec              print eigen vectors of the PCA into files
  -R, --print-rebuilt           print the rebuilt image by the PCA into a file

For simulating modes only
  -a, --adder-type <type>       specify the defect adder type :
                                  extension (nb-defect per extension)
                                  mosaic (nb-defect on the whole mosaic)
  -i, --intensity <value>       specify the defect intensity as a divisor
  -f, --nb-defect <value>       specify the number of defects to add
```

Usage explains there are the two ways to use defectix. The first one is to set options on the command line, every option being described in the section 4.2. This mode in invoked with *-M* or *–mode*.

The second mode uses a configuration file chosen with *-C* or *–conf-file* option. Configuration files are described in the section 4.3.

4.2 Command line

Command line mode

-M, –mode <mode>

Choose the execution mode of defectix.*mode* can be *halo, trail, training* or *computing*. Once a mode is chosen, complementary options can be chosen. There is no default value.
halo and trail sets defectix to create simulated defects on an image and to generate corresponding mask.
training sets defectix to create a KL-Base and train a neural network using an image and the corresponding mask.
computing sets defectix to create a mask for an image according to a KL-Base and a given neural network.

File management

-I, –input-file <name>	Choose the input file. For each mode, it corresponds to the image to treat and the file is never modified. There is no default value.
-K, –mask-file <name>	Choose the mask file. For simulating and computing modes, the created mask is saved in this file. For the training mode, the mask is loaded from this file. There is no default value.
-O, –output-file <name>	Choose the ouput file. In fact, it is currently useless. There is no default value.
-L, –evec-file <name>	Choose the eigen vectors file, that is the KL-Base. For training and computing modes. If the files exists and its format is correct, the KL-Base is loaded, else a new KL-Base is computed and saved in this file. There is no default value.

Global options

-d, –defaults	Display default options as in a configuration file and exit.
-h, –help	Display an usage message and exit.
-V, –version	Display defectix version and exit
-v, –verbose	Set verbose mode to display C++ debug informations.
-S, –substract-bg <bool>	Ask for background substraction on the input image. Default value is *true*.
-D, –reduce-dynamic <bool>	Ask for dynamic reduction on the input image. If both *-S* and *-D* options are set to *true*, background substraction is computed before dynamic reduction. Default value is *true*.
-r, –rebin-factor <value>	Choose a rebin factor for the input image. If the factor is 1, no rebin is computed. Else, rebin is done before background substraction. Default value is *4*.
-b, –bloc-size <value>	Define the size of the blocks for the PCA. Resulting blocks contain *value*x*value* pixels. Default value is *8*.

Training and computing options

-p, –nb-pc <value>	Specify the number of principal components from the KL-Base to use. This value depends on the *-b* option because the KL-Base dimension corresponds to the size of the blocs. Default value is *32* because the default value of *-b* being 8, the KL-Base contains 64 components.

-t, –nb-vector <value>	Specify the number of vectors to use for training. Blocks are randomly chosen in the input image to create a training set for the network training. The bigger the set is, the better the network should learn and the longer the training should be. Default value is *500.000* but must be reduced if there is not enough RAM on the computer.
-e, –energy-treshold <value>	Define a treshold for neural network convergence. The energy (or error) is computed at each training cycle. If the error is lower than the chosen treshold, the network is considered to be trained. Default value is *0.025*.
-m, –max-iteration <value>	Define a maximum number of iterations. This option is a security to avoid an infinite convergence. Indeed, if the chosen treshold (*-e* option) cannot be reached, the training will stop if the energy value does not decrease during *value* successive iterations. Default value is *200*.
-E, –print-evec	Save eigen vectors in the file given by *-L* option. If *-E* is not set, the KL-Base is not saved, even if *-L* is chosen. By default, the KL-Base is not saved.
-R, –print-rebuilt	Save a rebuilt version of the original image in a file. Currently not working.
-x, –max-prior <value>	Define a maximum intensity value for priors. If a bloc is brighter than *max * value* where *max* is the maximum intensity of the image, it is NOT consider as a prior. Only prior blocs are used for training.
-n, –min-prior <value>	Define a minimum intensity value for priors. If a bloc is darker than *max * value* where *max* is the maximum intensity of the image, it is NOT consider as a prior. Only prior blocs are used for training.
-l, –non-prior <value>	Define a part of blocs to use without priors. If one wants to use non prior blocs for training, it can specify a ratio. *value * 100*% of the blocs will not be prior blocs.
-s, –mask-treshold <value>	Define a treshold for mask creation. If *value* is 2, no treshold is apply and the result is a linear mask. If *value* is between 0 and 1, it is used as a treshold. Pixels brighter than *value* are set to 1, else, they are set to 0;

Simulating options

-a, –adder-type <type>	Specify the way to add defects. *type* can be *extension* or *mosaic*. The default value is *extension*. *extension* type add the number of defects given by *-d* option on each extension of the mosaic. *mosaic* type add the number of defects given by *-d* option on the whole mosaic.
-i, –intensity <value>	Choose the defect intensity as a divisor. The intensity of defects are computed according to the maximum value of the image. Indeed, the average intensity is the result of the division of the maximum value by *value*. Default value is *200*.
-d, –nb-defect <value>	Choose the number of defects to add according to the type. Default value is 5. The default values adds 5 defects on each extension.

4.3 Configuration files

Configuration file mode

-C, –conf-file <name>	Choose the configuration file.

Once a configuration file has been chosen on the command line using option *-C*, any other option on the command line is not considered. Options must be set in the configuration file using options long names. For example, the energy treshold is set typing:

```
ENERGY-TRESHOLD 0.001
```

Option *-d* or *–defaults* displays on standard output all that can be found in a configuration file with usefull comments. Blank lines are authorised and any line starting with a # is treated as a comment.

Considering cfitsio library, defectix respects some considerations regarding to file management. Indeed, to overwrite a file, it is necessary to add a *!* before the file name. For example, */home/ab/mask/721548p.fits* will not allow defectix to write or replace the file whereas *!/home/ab/mask/721548p.fits* will.

4.4 Standard use

The standard use of defectix requires 3 configuration files.

The first is used for defects simulation. Indeed, *simconf* rebins the image, substracts the background, adds defects and then reduces the dynamic of the image, including defects.

The second file is *trainconf* and is used to compute a KL-Base and to train the network.

The third file is *computeconf* and is used to create masks according to files obtained during training. This last file is the most usefull. Indeed, defect simulation and training can be done only once whereas computation is done for each image one wants to treat.

```
## configuration file for defectix
## use 'defectix -C' or 'defectix --conf-file'
## /!\ configuration file options cancel command line options
## add # at the begin of a line to comment it

####################### following options are always used ######################
## execution mode ('halo', 'trail, 'training' or 'computing')
MODE halo

## input file (no default, must be chosen)
## do not forget '!' to create or overwrite a file
INPUT-FILE /home/nis/ab/src/718798p.fits
MASK-FILE !/home/nis/ab/simmask/718798p.simmask.fits
OUTPUT-FILE !/home/nis/ab/siminput/718798p.siminput.fits

## rebin factor (default is 4)
#REBIN-FACTOR 8

## ask for substraction of the background (default true)
#SUBSTRACT-BG false

## reduce dynamic range of the image (default true)
#REDUCE-DYN false

############## following options are used for 'simulating' mode ###############
## defect adder type
## extension = NB-DEFECTS per extension
## mosaic = NB-DEFECTS on the whole mosaic
ADDER-TYPE extension

# intensity of the defect compared to maximum (1 / INTENSITY) (default 200)
INTENSITY 800

## number of defects to add corresponding to the defect adder type
NB-DEFECT 1

## end of file
```

Figure 4.1: simconf, create simulated masks and simulated images

```
## configuration file for defectix
## use 'defectix -C' or 'defectix --conf-file'
## /!\ configuration file options cancel command line options
## add # at the begin of a line to comment it


####################### following options are always used ######################
## execution mode ('simulating' 'training' or 'computing')
MODE training

## display error messages (default no display)
# VERBOSE

## input file (no default, must be chosen)
INPUT-FILE /home/nis/ab/siminput/718798p.siminput.fits
MASK-FILE /home/nis/ab/simmask/718798p.simmask.fits
EVEC-FILE evec.mat.gsl
NN-FILE network.net

## size of blocs for the PCA (default 8)
BLOC-SIZE 8

## rebin factor (default is 4)
REBIN-FACTOR 1

## ask for substraction of the background (default true)
SUBSTRACT-BG false

## reduce dynamic range of the image (default true)
REDUCE-DYN false


############### following options are use for 'training' mode #################
## output results of the PCA in files (default does not output)
PRINT-EVEC

## number of principal components to use after PCA (default 500000)
NB-VECTOR 400000

## number of principal components to use after PCA (default 32)
NB-PC 16

## energy treshold for training (default 0.025)
ENERGY-TRESHOLD 0.001
```

```
## maximum number of iterations with the same error value for training
## (default 200)
MAX-ITERATION 20

## prior management for training
MAX-PRIOR 1
MIN-PRIOR 0.3
NON-PRIOR 0.5

## end of file
```

Figure 4.2: trainconf, create a KL base and train a network

```
## configuration file for defectix
## use 'defectix -C' or 'defectix --conf-file'
## /!\ configuration file options cancel command line options
## add # at the begin of a line to comment it


####################### following options are always used ######################
## execution mode ('simulating' 'training' or 'computing')
MODE computing

## input file (no default, must be chosen)
## do not forget '!' to create or overwrite a file
INPUT-FILE /home/nis/ab/src/718798p.fits
MASK-FILE !/home/nis/ab/mask/718798p.mask.fits
EVEC-FILE evec.mat.gsl
NN-FILE network.net

## size of blocs for the PCA (default 8)
BLOC-SIZE 8

## rebin factor (default is 4)
REBIN-FACTOR 8

## ask for substraction of the background (default true)
#SUBSTRACT-BG false

## reduce dynamic range of the image (default true)
#REDUCE-DYN false


############### following options are use for 'training' mode #################
## output results of the PCA in files (default does not output)
#PRINT-EVEC

## number of principal components to use after PCA (default 32)
NB-PC 16

## mask treshold for training (default 2 = no treshold)
MASK-TRESHOLD 0.8

## end of file
```

Figure 4.3: computeconf; compute a mask for a given image

Chapter 5

Messages and errors

5.1 Output

There are three types of messages written by defectix :

- information messages, written with a green header. These messages indicates the steps of the normal process.
- alert messages, written with a yellow header. These messages are printed when a option was badly set so the default value is chosen.
- error messages, written with a red header. The messages result from an error during the process. The program usually exits after such a message.

5.2 Error codes

Error messages are usually explicite but more information can be obtained using *–verbose* option.

0	NO_ERROR	no error.
1	NO_EXTENSION	try to use an extension not loaded.
2	MALLOC	cannot allocate memory. Usually corresponds to a malloc() failure.
3	FITS	cfitsio error occured. Usually, cfitsio error message is also printed.
4	NOT_MEF	file is not a standard MEF.
5	NOT_IMGEXT	extension is not an image.
6	NO_EXT	no extension at given position.
7	NO_MASK	no mask at given position.
8	WRONG_BITPIX	bitpix not supported.
9	MASK_SIZE	mask dimensions are different from image ones.
10	NOT_ALLOC_MAT	matrix not allocated before use.
11	NOT_ALLOC_TABLE	table not allocated before use.
12	NO_KLBASE	cannot load KL-base.