MICADO Pipeline Reference Manual

0.2.0

Generated by Doxygen 1.9.1

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

Module Documentation

3.1 Recipe mcd_psfr_scao_final: Compute the reconstructed PSF

Functions

- static int mcd_psfr_scao_final (cpl_frameset *frameset, const cpl_parameterlist *parlist)

  Interpret the command line options and execute the data processing.
- static cpl_error_code mcd_psfr_scao_final_fill_parameterlist (cpl_parameterlist *self)

  Read the parameter list from the pipeline and fill the data structure.
- mcd_psfr_gridfunction * mcd_psfr_otf_inv (const mcd_psfr_parameters *params, const mcd_psfr_gridfunction *otf)
- mcd_psfr_otf * mcd_psfr_otf_multi_unify (const mcd_psfr_parameters *params, const mcd_psfr_otf *otf)
- mcd_psfr_gridfunction * mcd_psfr_otf_multi (const mcd_psfr_parameters *params, const mcd_psfr_otf *otf)

  Combine the single parts of the OTF.
- mcd_psfr_gridfunction * mcd_psfr_otf_nonao_vibration (const mcd_psfr_parameters *params, const mcd_psfr_gridfunction *otf)

  NonAO Vibration definition. Temporary this function returns the same input data, without any changes.
- mcd_psfr_gridfunction * mcd_psfr_otf_rotate (const mcd_psfr_parameters *params, const mcd_psfr_gridfunction *otf)

  rotate the input OTF by a given angle. Temporary this function returns the same input data, without any changes.

3.1.1 Detailed Description

Author

A. La Camera, M. Ravenna

Compute the reconstructed PSF from all the previous OTFs.

3.1.2 Function Documentation

3.1.2.1 mcd_psfr_otf_inv()

```
mcd_psfr_gridfunction* mcd_psfr_otf_inv (
            const mcd_psfr_parameters * params,
            const mcd_psfr_gridfunction * otf )
```

Inputs:

- [INTERM] OTF_TOTAL_IMG
- [EXT] REF_FILTER_CURVE

Output:

- [FINAL] PSF_ANISOPLANATIC_IMG

Definition at line 45 of file mcd_psfr_otf_inv.c.

References mcd_psfr_gridfunction_duplicate(), mcd_psfr_gridfunction_fft(), and mcd_psfr_gridfunction_pad_↩create().

3.1.2.2 mcd_psfr_otf_multi()

```
mcd_psfr_gridfunction* mcd_psfr_otf_multi (
            const mcd_psfr_parameters * params,
            const mcd_psfr_otf * otf )
```

Combine the single parts of the OTF.

Parameters

params	Parameter struct
otf	Structure with parts of the OTF

Returns

Single big OTF

Definition at line 139 of file mcd_psfr_otf_multi.c.

References mcd_psfr_cpl_image_fill(), mcd_psfr_gridfunction_new(), and mcd_psfr_otf_multi_unify().

Referenced by mcd_psfr_otf_compute().

3.1.2.3 mcd_psfr_otf_multi_unify()

```
mcd_psfr_otf* mcd_psfr_otf_multi_unify (
            const mcd_psfr_parameters * params,
            const mcd_psfr_otf * otf )
```

3.1.3 Variant 1:

Inputs:

- [FINAL] OTF_ELT_QUASI_STATIC_IMG
- [FINAL] OTF_ELT_ANISOPLANATIC_IMG
- [FINAL] OTF_PARA_LONG_IMG
- [FINAL] OTF_PERP_LONG_IMG

Output:

- [INTERM] OTF_COMB_IMG

3.1.4 Variant 2:

Inputs:

- [INTERM] OTF_ROT_IMG
- [FINAL] OTF_MICADO_QUASI_STATIC_IMG
- [FINAL] OTF_BLURRING_KERNEL_IMG

Output:

- [INTERM] OTF_TOTAL_IMG

Definition at line 63 of file mcd_psfr_otf_multi.c.

References mcd_psfr_domain_centered(), mcd_psfr_gridfunction_check_minmax_at_origin(), mcd_psfr_↩ gridfunction_evaluate_domain(), mcd_psfr_otf_delete(), and mcd_psfr_otf_new().

Referenced by mcd_psfr_otf_multi().

3.1.4.1 mcd_psfr_otf_nonao_vibration()

```
mcd_psfr_gridfunction* mcd_psfr_otf_nonao_vibration (
            const mcd_psfr_parameters * params,
            const mcd_psfr_gridfunction * otf )
```

NonAO Vibration definition. Temporary this function returns the same input data, without any changes.

Parameters

params	- input parameters data structure
otf	- input otf data structure

Returns

a copy of the input otf data structure

Definition at line 53 of file mcd_psfr_otf_nonao_vibration.c.

References mcd_psfr_gridfunction_duplicate().

3.1.4.2 mcd_psfr_otf_rotate()

```
mcd_psfr_gridfunction* mcd_psfr_otf_rotate (
            const mcd_psfr_parameters * params,
            const mcd_psfr_gridfunction * otf )
```

rotate the input OTF by a given angle. Temporary this function returns the same input data, without any changes.

Parameters

params	- input parameters data structure
otf	- input otf data structure

Returns

a copy of the input otf data structure

[TEIGA] To be implemented or not? The OTF is converted into an oversampled PSF. The result is then rotated by the specified rotation angle and transformed back to OTF space. It is TBC that this can also be done in the OTF space (to avoid Fourier transforms). TODO - convert otf (gridfunction ??)-->psf TODO - get parameter "angle" for rotation TODO - rotate psf with angle TODO - convert psf in otf ?? TODO Error: Missing user parameters, missing or wrong input products.

Definition at line 50 of file mcd_psfr_otf_rotate.c.

References mcd_psfr_gridfunction_duplicate().

3.1.4.3 mcd_psfr_scao_final()

```
static int mcd_psfr_scao_final (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

CPL_ERROR_NONE(0) if everything is ok

Create the recipe instance and make it available to the application using the interface.

Definition at line 131 of file mcd_psfr_scao_final.c.

References mcd_check_and_set_groups().

3.1.4.4 mcd_psfr_scao_final_fill_parameterlist()

```
static cpl_error_code mcd_psfr_scao_final_fill_parameterlist (
            cpl_parameterlist * self )  [static]
```

Read the parameter list from the pipeline and fill the data structure.

Parameters

self	the parameter list

Returns

0 if everything is ok

Definition at line 414 of file mcd_psfr_scao_final.c.

3.2 Recipe mcd_psfr_scao_otf_elt: Compute the OTF Telescope

Functions

- static int mcd_psfr_scao_otf_elt (cpl_frameset ∗frameset, const cpl_parameterlist ∗parlist)
  Interpret the command line options and execute the data processing.
- static cpl_error_code mcd_psfr_scao_otf_elt_fill_parameterlist (cpl_parameterlist ∗self)
  Read the parameter list from the pipeline and fill the data structure.
- CPL_ATTR_ALLOC mcd_psfr_gridfunction ∗ mcd_psfr_otf_telescope (const mcd_psfr_gridfunction ∗pupil)
  Compute the telescope part of the OTF by OTF_tel(r) = sum_x P(x)P(x+r), x, x+r in supp(P)

3.2.1 Detailed Description

Author

M. Ravenna, A. La Camera

Compute the OTF telescope from pupil image.

3.2.2 Function Documentation

3.2.2.1 mcd_psfr_otf_telescope()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_otf_telescope (
            const mcd_psfr_gridfunction * pupil )
```

Compute the telescope part of the OTF by OTF_tel(r) = sum_x P(x)P(x+r), x, x+r in supp(P)

Parameters

pupil	Telescope pupil P

Returns

OTF-telescope

Definition at line 45 of file mcd_psfr_otf_telescope.c.

References mcd_psfr_gridfunction_autocorrelation(), mcd_psfr_gridfunction_check_minmax_at_origin(), and mcd_psfr_gridfunction_delete().

Referenced by mcd_psfr_otf_compute(), and mcd_psfr_scao_otf_elt().

3.2.2.2 mcd_psfr_scao_otf_elt()

```
static int mcd_psfr_scao_otf_elt (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

CPL_ERROR_NONE(0) if everything is ok

Create the recipe instance and make it available to the application using the interface.

Definition at line 103 of file mcd_psfr_scao_otf_elt.c.

References mcd_check_and_set_groups(), mcd_psfr_fits_get_double(), mcd_psfr_otf_telescope(), mcd_psfr_↩ utility_gridfunction_from_img_px(), and mcd_psfr_utility_read_data().

3.2.2.3 mcd_psfr_scao_otf_elt_fill_parameterlist()

```
static cpl_error_code mcd_psfr_scao_otf_elt_fill_parameterlist (
            cpl_parameterlist * self )  [static]
```

Read the parameter list from the pipeline and fill the data structure.

Parameters

self	the parameter list

Returns

0 if everything is ok

Definition at line 193 of file mcd_psfr_scao_otf_elt.c.

3.3 Recipe mcd_psfr_scao_otf_micado: Compute the OTF Micado

Functions

- static int mcd_psfr_scao_otf_micado (cpl_frameset ∗frameset, const cpl_parameterlist ∗parlist)
  Interpret the command line options and execute the data processing.
- static cpl_error_code mcd_psfr_scao_otf_micado_fill_parameterlist (cpl_parameterlist ∗self)
  Read the parameter list from the pipeline and fill the data structure.
- cpl_error_code mcd_psfr_cal_ADC_mistrack (cpl_frameset ∗frames, const cpl_parameterlist ∗parameters)
  empty function
- cpl_error_code mcd_psfr_cal_NCPA_residual (cpl_frameset ∗frames, const cpl_parameterlist ∗parameters)
  empty function
- cpl_error_code mcd_psfr_cal_pupil_drift (cpl_frameset ∗frames, const cpl_parameterlist ∗parameters)
  empty function
- cpl_error_code mcd_psfr_elt_quasi_static_otf (cpl_frameset ∗frames, const cpl_parameterlist ∗parameters)
  empty function

3.3.1 Detailed Description

Author

A. La Camera, M. Ravenna

Compute the OTF MICADO QUASI STATIC Image.

3.3.2 Function Documentation

3.3.2.1 mcd_psfr_cal_ADC_mistrack()

```
cpl_error_code mcd_psfr_cal_ADC_mistrack (
            cpl_frameset * frames,
            const cpl_parameterlist * parameters )
```

empty function

Parameters

frames	the frameset
parameters	the parameters list

Returns

0 if everything is OK

Definition at line 40 of file mcd_psfr_cal_ADC_mistrack.c.

Referenced by mcd_psfr_scao_otf_micado().

3.3.2.2 mcd_psfr_cal_NCPA_residual()

```
cpl_error_code mcd_psfr_cal_NCPA_residual (
            cpl_frameset * frames,
            const cpl_parameterlist * parameters )
```

empty function

Parameters

frames	the frameset
parameters	the parameters list

Returns

0 if everything is OK

Definition at line 41 of file mcd_psfr_cal_NCPA_residual.c.

Referenced by mcd_psfr_scao_otf_micado().

3.3.2.3 mcd_psfr_cal_pupil_drift()

```
cpl_error_code mcd_psfr_cal_pupil_drift (
            cpl_frameset * frames,
            const cpl_parameterlist * parameters )
```

empty function

Parameters

frames	the frameset
parameters	the parameters list

Returns

0 if everything is OK

Definition at line 41 of file mcd_psfr_cal_pupil_drift.c.

Referenced by mcd_psfr_scao_otf_micado().

3.3.2.4 mcd_psfr_elt_quasi_static_otf()

```
cpl_error_code mcd_psfr_elt_quasi_static_otf (
            cpl_frameset * frames,
            const cpl_parameterlist * parameters )
```

empty function

Parameters

frames	the frameset
parameters	the parameters list

Returns

0 if everything is OK

Definition at line 43 of file mcd_psfr_elt_quasi_static_otf.c.

3.3.2.5 mcd_psfr_scao_otf_micado()

```
static int mcd_psfr_scao_otf_micado (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

CPL_ERROR_NONE(0) if everything is ok

Definition at line 103 of file mcd_psfr_scao_otf_micado.c.

References mcd_check_and_set_groups(), mcd_psfr_cal_ADC_mistrack(), mcd_psfr_cal_NCPA_residual(), mcd_psfr_cal_pupil_drift(), and mcd_psfr_utility_read_data().

3.3.2.6 mcd_psfr_scao_otf_micado_fill_parameterlist()

```
static cpl_error_code mcd_psfr_scao_otf_micado_fill_parameterlist (
            cpl_parameterlist * self )  [static]
```

Read the parameter list from the pipeline and fill the data structure.

Parameters

self	the parameter list

Returns

0 if everything is ok

Definition at line 232 of file mcd_psfr_scao_otf_micado.c.

3.4 Recipe mcd_psfr_scao_otf_par: Compute the OTF parallel

Functions

- static int mcd_psfr_scao_otf_par (cpl_frameset *frameset, const cpl_parameterlist *parlist)
  Interpret the command line options and execute the data processing.
- static cpl_error_code mcd_psfr_scao_otf_par_fill_parameterlist (cpl_parameterlist *self)
  Read the parameter list from the pipeline and fill the data structure.
- cpl_error_code mcd_psfr_scao_anisoplanatic_phase (mcd_psfr_otf *otf, const mcd_psfr_parameters *params)
  Compute the alias part of the OTF.
- cpl_error_code mcd_psfr_scao_cal_WFS_aliasing (mcd_psfr_otf *otf, const mcd_psfr_parameters *params)
  Compute the alias part of the OTF.
- cpl_error_code mcd_psfr_scao_cal_WFS_noise (mcd_psfr_otf *otf, const mcd_psfr_parameters *params)
- cpl_error_code mcd_psfr_scao_cal_WFS_slopes (mcd_psfr_parameters *params)
  Compute the DM updates from the DM commands.
- cpl_error_code mcd_psfr_scao_fried_param (mcd_psfr_parameters *parameters)
- cpl_error_code mcd_psfr_scao_parallel (mcd_psfr_otf *otf, const mcd_psfr_parameters *params)
  Compute the reconstructed part of the OTF.

3.4.1 Detailed Description

Author

D. Jodlbauer, A. La Camera, M. Ravenna

Compute the OTF of the controlled (parallel) modes using Pyramid WFS telemetry data (slopes and matrices) and atmospheric monitoring telemetry.

3.4.2 Function Documentation

3.4.2.1 mcd_psfr_scao_anisoplanatic_phase()

```
cpl_error_code mcd_psfr_scao_anisoplanatic_phase (
            mcd_psfr_otf * otf,
            const mcd_psfr_parameters * params )
```

Compute the alias part of the OTF.

Inputs:

- [FINAL] PUB_VIEWS_IMG
- [RAW] ELT_TELEMETRY_TAB
- [INTERM] PYR_WFS_CAL_SLOPES_TAB
- [INTERM] OUTER_SCALE_TAB
- [INTERM] RESID_PHASE_MODAL_SCAO_IMG
- [INTERM] FRIED_PARAMETER_TAB
- [RAW] CN2_PROFILE_TAB

Output:

- TODO NONE?

Parameters

otf	The OTF object to compute
params	The PSFR object to use

Returns

otf->anisoplanatic

Definition at line 61 of file mcd_psfr_scao_anisoplanatic_phase.c.

References mcd_psfr_cpl_image_add_scale(), mcd_psfr_cpl_wall_time(), and mcd_psfr_gridfunctions_get_const_data().

3.4.2.2 mcd_psfr_scao_cal_WFS_aliasing()

```
cpl_error_code mcd_psfr_scao_cal_WFS_aliasing (
            mcd_psfr_otf * otf,
            const mcd_psfr_parameters * params )
```

Compute the alias part of the OTF.

Inputs:

- [RAW] CN2_PROFILE_TAB
- [RAW] AO_CONFIGURATION_TAB
- [INTERM] OUTER_SCALE_TAB
- [INTERM] FRIED_PARAMETER_TAB

Output:

- [INTERM] PYR_WFS_ALIASING_TAB

Parameters

otf	The OTF object to compute
params	The PSFR object to use

Returns

otf->alias

Definition at line 54 of file mcd_psfr_scao_cal_WFS_aliasing.c.

Referenced by mcd_psfr_otf_compute().

3.4.2.3 mcd_psfr_scao_cal_WFS_noise()

```
cpl_error_code mcd_psfr_scao_cal_WFS_noise (
            mcd_psfr_otf * otf,
            const mcd_psfr_parameters * params )
```

Compute the noise part of the OTF.

Algorithm:

1. Compute the covariance matrix C of the dm shapes, based on the covariance matrix sigma^2 I of the measurements, affected by the linear reconstructor R via C = sigma^2 R^T I R
2. For each shift (a,b), average the covariance over all pairs of points with this distance, i.e., cov((x,y) , (x+a,x+b)) and store the result in the OTF-noise, with shift (0,0) at the center of the image.

Parameters

otf	The OTF object to compute
params	The PSFR object to use

Returns

otf->noise

Definition at line 78 of file mcd_psfr_scao_cal_WFS_noise.c.

References mcd_psfr_covariance_average(), mcd_psfr_cpl_wall_time(), mcd_psfr_gridfunction_wrap(), mcd_psfr_influence_functions_delete(), mcd_psfr_influence_functions_domain(), mcd_psfr_influence_functions_resize_all(), mcd_psfr_scao_cal_WFS_noise_compute_average_gain(), mcd_psfr_scao_cal_WFS_noise_compute_covariance_matrix(), and mcd_psfr_scao_cal_WFS_noise_m4_to_fried().

Referenced by mcd_psfr_otf_compute().

3.4.2.4 mcd_psfr_scao_cal_WFS_slopes()

```
cpl_error_code mcd_psfr_scao_cal_WFS_slopes (
            mcd_psfr_parameters * params )
```

Compute the DM updates from the DM commands.

Inputs:

- [RAW] PYR_WFS_MODAL_OPTIMIZATION_TAB
- [RAW] PYR_WFS_NORMALIZED_SLOPES_TAB
- [RAW] PYR_WFS_MODULATION_TAB
- [RAW] PYR_WFS_NCPA_OFFSET_TAB
- [RAW] DM_COMMANDS_SCAO_TAB
- [RAW] INTERACTION_MATRIX_SCAO_TAB
- [RAW] AO_CONFIGURATION_TAB

Output:

- [INTERM] PYR_WFS_CAL_SLOPES_TAB
- [INTERM] RESID_PHASE_MODAL_SCAO_IMG

Parameters

params	
params->dm_commands	
params->dm_(x|y)pos	
params->influfct	

Returns

params->dm_update

Definition at line 62 of file mcd_psfr_scao_cal_WFS_slopes.c.

References mcd_psfr_cpl_wall_time(), mcd_psfr_gridfunction_wrap(), mcd_psfr_gridfunctions_new(), mcd_psfr_influence_functions_domain(), mcd_psfr_influence_functions_size(), mcd_psfr_influence_functions_size_

x(), mcd_psfr_influence_functions_size_y(), mcd_psfr_influence_functions_sum(), mcd_psfr_utility_get_tilt(), and mcd_psfr_utility_get_tip().

Referenced by mcd_psfr_otf_compute().

3.4.2.5 mcd_psfr_scao_fried_param()

```
cpl_error_code mcd_psfr_scao_fried_param (
            mcd_psfr_parameters * parameters )
```

Inputs:

- [INTERM] PYR_WFS_CAL_SLOPES_TAB
- [INTERM] RESID_PHASE_MODAL_SCAO_IMG

Output:

- [INTERM] OUTER_SCALE_TAB
- [INTERM] FRIED_PARAMETER_TAB

Definition at line 45 of file mcd_psfr_scao_fried_param.c.

3.4.2.6 mcd_psfr_scao_otf_par()

```
static int mcd_psfr_scao_otf_par (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

CPL_ERROR_NONE(0) if everything is ok

Create the recipe instance and make it available to the application using the interface.

Definition at line 107 of file mcd_psfr_scao_otf_par.c.

References mcd_check_and_set_groups(), and mcd_psfr_parameters_new().

3.4.2.7 mcd_psfr_scao_otf_par_fill_parameterlist()

```
static cpl_error_code mcd_psfr_scao_otf_par_fill_parameterlist (
            cpl_parameterlist * self )  [static]
```

Read the parameter list from the pipeline and fill the data structure.

Parameters

self	the parameter list

Returns

0 if everything is ok

Definition at line 341 of file mcd_psfr_scao_otf_par.c.

References mcd_psfr_parameters_new().

3.4.2.8 mcd_psfr_scao_parallel()

```
cpl_error_code mcd_psfr_scao_parallel (
            mcd_psfr_otf * otf,
            const mcd_psfr_parameters * params )
```

Compute the reconstructed part of the OTF.

Inputs:

- [TODO] mcd_psfr_scao_anisoplanatic_phase
- [RAW] PYR_WFS_MISREGISTRATION_TAB
- [INTERM] PYR_WFS_ALIASING_TAB
- [INTERM] PYR_WFS_NOISE_TAB

Output:

- [FINAL] OTF_PARA_LONG_IMG

Parameters

otf	The OTF object to compute
params	The PSFR object to use

Returns

otf->rec

Definition at line 58 of file mcd_psfr_scao_parallel.c.

References mcd_psfr_cpl_imagelist_sample(), mcd_psfr_cpl_wall_time(), and mcd_psfr_gridfunctions_data().

Referenced by mcd_psfr_otf_compute().

3.5 Recipe mcd_psfr_scao_otf_perp: Compute the OTF of the orthogonal modes

Functions

- cpl_error_code mcd_psfr_scao_otf_perp_parameters_from_frameset (mcd_psfr_parameters *prm, const cpl_frameset *frameset, const cpl_parameterlist *params)

  Read the parameter list from the pipeline and fill the paramters data structure.
- static int mcd_psfr_scao_otf_perp (cpl_frameset *frameset, const cpl_parameterlist *parlist)

  Interpret the command line options and execute the data processing.
- static cpl_error_code mcd_psfr_scao_otf_perp_fill_parameterlist (cpl_parameterlist *self)

  Read the parameter list from the pipeline and fill the data structure.
- static CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_scao_orthogonal_atmosphere_psd (mcd_psfr_domain domain, double L0, double r0)
- cpl_error_code mcd_psfr_scao_orthogonal (mcd_psfr_otf *otf, const mcd_psfr_parameters *params)

  Compute the orthogonal part of the OTF.

3.5.1 Detailed Description

Author

A. La Camera, M. Ravenna

Compute the OTF of the non-controlled (orthogonal) modes using information on the Adaptive Optic system configuration and Fried parameters.

3.5.2 Function Documentation

3.5.2.1 mcd_psfr_scao_orthogonal()

```
cpl_error_code mcd_psfr_scao_orthogonal (
            mcd_psfr_otf * otf,
            const mcd_psfr_parameters * params )
```

Compute the orthogonal part of the OTF.

Parameters

otf	The OTF object to compute
params	The PSFR object to use

Returns

otf->perp

Definition at line 59 of file mcd_psfr_scao_orthogonal.c.

References mcd_psfr_cpl_wall_time(), mcd_psfr_domain_centered(), mcd_psfr_gridfunction_fill_distance(), and mcd_psfr_gridfunction_new().

Referenced by mcd_psfr_otf_compute().

3.5.2.2 mcd_psfr_scao_orthogonal_atmosphere_psd()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_scao_orthogonal_atmosphere_psd (
            mcd_psfr_domain domain,
            double L0,
            double r0 )  [static]
```

Inputs:

- [RAW] AO_CONFIGURATION_TAB
- [INTERM] OUTER_SCALE_TAB
- [INTERM] FRIED_PARAMETER_TAB

Output:

- [FINAL] OTF_PERP_LONG_IMG

Beltramo 2020 ("Pushing point spread function ...") Van-Karman PSD: W(k) = 0.0229 r0^{-5/3} (k^2 + 1/L0^2)^{-11/6} [-5/3 ∗ 11/3] = [6/3] = [m^2] [k] = 1/m [L0] = m Outer scale [r0] = m Cut frequencies |k| > k_AO, with k_AO ∼ nact_per_row/(2∗teldiam) ∼ 1/(2∗actspacing) (actspacing ∼ teldiam / nact_per_row) (0.0229 ∼ 0.1513^2) TODO: 0.0229 ∼ constant when using frequency f = 1/s instead of k = rad/s = 2pi/s

From https://github.com/ArcetriAdaptiveOptics/arte/blob/master/arte/atmo/von_karman_psd.py PSD(f,h) = c ∗ r_0(h)^{-5/3} ∗ (f^2+ \frac{1}{L_0^2})^{-11/6} c = (\frac{24}{5} \Gamma(6/5))^{5/6} \frac{\Gamma(11/6)^2}{2 \pi^{11/3} } = 0.0228955

Definition at line 210 of file mcd_psfr_scao_orthogonal.c.

References mcd_psfr_domain_index_to_coordinate(), mcd_psfr_gridfunction_new(), mcd_psfr_gridfunction_set_vertex(), and MCD_PSFR_SQR.

3.5.2.3 mcd_psfr_scao_otf_perp()

```
static int mcd_psfr_scao_otf_perp (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

CPL_ERROR_NONE(0) if everything is ok

Create the recipe instance and make it available to the application using the interface.

Definition at line 111 of file mcd_psfr_scao_otf_perp.c.

References mcd_check_and_set_groups(), and mcd_psfr_parameters_new().

3.5.2.4 mcd_psfr_scao_otf_perp_fill_parameterlist()

```
static cpl_error_code mcd_psfr_scao_otf_perp_fill_parameterlist (
            cpl_parameterlist * self )  [static]
```

Read the parameter list from the pipeline and fill the data structure.

Parameters

self	the parameter list

Returns

0 if everything is ok

Definition at line 219 of file mcd_psfr_scao_otf_perp.c.

References mcd_psfr_parameters_new().

3.5.2.5 mcd_psfr_scao_otf_perp_parameters_from_frameset()

```
cpl_error_code mcd_psfr_scao_otf_perp_parameters_from_frameset (
            mcd_psfr_parameters * prm,
            const cpl_frameset * frameset,
            const cpl_parameterlist * params )
```

Read the parameter list from the pipeline and fill the paramters data structure.

Parameters

prm	the mcd_psfr_parameters data
frameset	the frames list
parlist	the parameters list

Returns

0 if everything is ok

Definition at line 264 of file mcd_psfr_scao_otf_perp.c.

3.6 Recipe mcd_spec_flat: Compute spectroscopic flat field

Functions

- static int mcd_spec_flat_create (cpl_plugin ∗plugin)

  Setup the recipe options.
- static int mcd_spec_flat_exec (cpl_plugin ∗plugin)

  Execute the plugin instance given by the interface.
- static int mcd_spec_flat_destroy (cpl_plugin ∗plugin)

  Destroy what has been created by the 'create' function.
- static cpl_error_code mcd_spec_flat (cpl_frameset ∗frameset, const cpl_parameterlist ∗parlist)

  Interpret the command line options and execute the data processing.

3.6.1 Detailed Description

Author

O. Czoske, W. Kausch

Measure wavelength-dependent variations in pixel sensitivity at small (pixel-to-pixel) and large (blaze function) scale. Determination of the inter-order background model (cf. [R-MCD-131]) and the slit deviation induced by the filter wheel. Order detection.

3.6.2 Function Documentation

3.6.2.1 mcd_spec_flat()

```
static cpl_error_code mcd_spec_flat (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

0 if everything is ok

Create SFLAT_SLIT_IMG

Create MASTER_SFLAT

Definition at line 273 of file mcd_spec_flat.c.

Referenced by mcd_spec_flat_exec().

3.6.2.2 mcd_spec_flat_create()

```
static int mcd_spec_flat_create (
            cpl_plugin * plugin )  [static]
```

Setup the recipe options.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Defining the command-line/configuration parameters for the recipe.

Definition at line 170 of file mcd_spec_flat.c.

3.6.2.3 mcd_spec_flat_destroy()

```
static int mcd_spec_flat_destroy (
            cpl_plugin * plugin )  [static]
```

Destroy what has been created by the 'create' function.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 248 of file mcd_spec_flat.c.

3.6.2.4 mcd_spec_flat_exec()

```
static int mcd_spec_flat_exec (
            cpl_plugin * plugin )  [static]
```

Execute the plugin instance given by the interface.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 227 of file mcd_spec_flat.c.

References mcd_spec_flat().

3.7 Recipe mcd_spec_flux: Compute flux calibration

Functions

- static int mcd_spec_flux_create (cpl_plugin *plugin)
  Setup the recipe options.
- static int mcd_spec_flux_exec (cpl_plugin *plugin)
  Execute the plugin instance given by the interface.
- static int mcd_spec_flux_destroy (cpl_plugin *plugin)
  Destroy what has been created by the 'create' function.
- static cpl_error_code mcd_spec_flux (cpl_frameset *frameset, const cpl_parameterlist *parlist)
  Interpret the command line options and execute the data processing.

3.7.1 Detailed Description

Author

O. Czoske, W. Kausch

Determine the instrumental response function

3.7.2 Function Documentation

3.7.2.1 mcd_spec_flux()

```
static cpl_error_code mcd_spec_flux (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

0 if everything is ok

Create the fstd images (object and sky)

Create RESPONSE_SPEC

Definition at line 256 of file mcd_spec_flux.c.

Referenced by mcd_spec_flux_exec().

3.7.2.2 mcd_spec_flux_create()

```
static int mcd_spec_flux_create (
            cpl_plugin * plugin )  [static]
```

Setup the recipe options.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Defining the command-line/configuration parameters for the recipe.

Definition at line 175 of file mcd_spec_flux.c.

3.7.2.3 mcd_spec_flux_destroy()

```
static int mcd_spec_flux_destroy (
            cpl_plugin * plugin )  [static]
```

Destroy what has been created by the 'create' function.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 232 of file mcd_spec_flux.c.

3.7.2.4 mcd_spec_flux_exec()

```
static int mcd_spec_flux_exec (
            cpl_plugin * plugin )  [static]
```

Execute the plugin instance given by the interface.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 211 of file mcd_spec_flux.c.

References mcd_spec_flux().

3.8 Recipe mcd_spec_mf_calctrans: calculate transmission

Functions

- static int mcd_spec_mf_calctrans_create (cpl_plugin ∗plugin)
  Setup the recipe options.
- static int mcd_spec_mf_calctrans_exec (cpl_plugin ∗plugin)
  Execute the plugin instance given by the interface.
- static int mcd_spec_mf_calctrans_destroy (cpl_plugin ∗plugin)
  Destroy what has been created by the 'create' function.
- static cpl_error_code mcd_spec_mf_calctrans (cpl_frameset ∗frameset, const cpl_parameterlist ∗parlist)
  Interpret the command line options and execute the data processing.

3.8.1 Detailed Description

Author

O. Czoske, W. Kausch, J. Brink

Telluric correction modelling recipe

3.8.2 Function Documentation

3.8.2.1 mcd_spec_mf_calctrans()

```
static cpl_error_code mcd_spec_mf_calctrans (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

0 if everything is ok

Sort the input frameset

PRODUCTS

Definition at line 255 of file mcd_spec_mf_calctrans.c.

Referenced by mcd_spec_mf_calctrans_exec().

3.8.2.2 mcd_spec_mf_calctrans_create()

```
static int mcd_spec_mf_calctrans_create (
            cpl_plugin * plugin )  [static]
```

Setup the recipe options.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Defining the command-line/configuration parameters for the recipe.

Definition at line 170 of file mcd_spec_mf_calctrans.c.

3.8.2.3 mcd_spec_mf_calctrans_destroy()

```
static int mcd_spec_mf_calctrans_destroy (
            cpl_plugin * plugin )  [static]
```

Destroy what has been created by the 'create' function.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 229 of file mcd_spec_mf_calctrans.c.

3.8.2.4 mcd_spec_mf_calctrans_exec()

```
static int mcd_spec_mf_calctrans_exec (
            cpl_plugin * plugin )  [static]
```

Execute the plugin instance given by the interface.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 208 of file mcd_spec_mf_calctrans.c.

References mcd_spec_mf_calctrans().

3.9 Recipe mcd_spec_mf_correct: calculate transmission

Functions

- static int mcd_spec_mf_correct_create (cpl_plugin *plugin)
  Setup the recipe options.
- static int mcd_spec_mf_correct_exec (cpl_plugin *plugin)
  Execute the plugin instance given by the interface.
- static int mcd_spec_mf_correct_destroy (cpl_plugin *plugin)
  Destroy what has been created by the 'create' function.
- static cpl_error_code mcd_spec_mf_correct (cpl_frameset *frameset, const cpl_parameterlist *parlist)
  Interpret the command line options and execute the data processing.

3.9.1 Detailed Description

Author

O. Czoske, W. Kausch, J. Brink

Telluric correction recipe

3.9.2 Function Documentation

3.9.2.1 mcd_spec_mf_correct()

```
static cpl_error_code mcd_spec_mf_correct (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

0 if everything is ok

Sort the input frameset

PRODUCTS

Definition at line 247 of file mcd_spec_mf_correct.c.

Referenced by mcd_spec_mf_correct_exec().

3.9.2.2 mcd_spec_mf_correct_create()

```
static int mcd_spec_mf_correct_create (
            cpl_plugin * plugin )  [static]
```

Setup the recipe options.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Defining the command-line/configuration parameters for the recipe.

Definition at line 162 of file mcd_spec_mf_correct.c.

3.9.2.3 mcd_spec_mf_correct_destroy()

```
static int mcd_spec_mf_correct_destroy (
            cpl_plugin * plugin )  [static]
```

Destroy what has been created by the 'create' function.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 221 of file mcd_spec_mf_correct.c.

3.9.2.4 mcd_spec_mf_correct_exec()

```
static int mcd_spec_mf_correct_exec (
            cpl_plugin * plugin )  [static]
```

Execute the plugin instance given by the interface.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 200 of file mcd_spec_mf_correct.c.

References mcd_spec_mf_correct().

3.10 Recipe mcd_spec_mf_model: Model telluric correction

Functions

- static int mcd_spec_mf_model_create (cpl_plugin *plugin)

  Setup the recipe options.
- static int mcd_spec_mf_model_exec (cpl_plugin *plugin)

  Execute the plugin instance given by the interface.
- static int mcd_spec_mf_model_destroy (cpl_plugin *plugin)

  Destroy what has been created by the 'create' function.
- static cpl_error_code mcd_spec_mf_model (cpl_frameset *frameset, const cpl_parameterlist *parlist)

  Interpret the command line options and execute the data processing.

3.10.1 Detailed Description

Author

O. Czoske, W. Kausch, J. Brink

Telluric correction modelling recipe

3.10.2 Function Documentation

3.10.2.1 mcd_spec_mf_model()

```
static cpl_error_code mcd_spec_mf_model (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

0 if everything is ok

Sort the input frameset

PRODUCTS

Definition at line 264 of file mcd_spec_mf_model.c.

Referenced by mcd_spec_mf_model_exec().

3.10.2.2 mcd_spec_mf_model_create()

```
static int mcd_spec_mf_model_create (
            cpl_plugin * plugin )  [static]
```

Setup the recipe options.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Defining the command-line/configuration parameters for the recipe.

Definition at line 181 of file mcd_spec_mf_model.c.

3.10.2.3 mcd_spec_mf_model_destroy()

```
static int mcd_spec_mf_model_destroy (
            cpl_plugin * plugin )  [static]
```

Destroy what has been created by the 'create' function.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 238 of file mcd_spec_mf_model.c.

3.10.2.4 mcd_spec_mf_model_exec()

```
static int mcd_spec_mf_model_exec (
            cpl_plugin * plugin )  [static]
```

Execute the plugin instance given by the interface.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 217 of file mcd_spec_mf_model.c.

References mcd_spec_mf_model().

3.11 Recipe mcd_spec_sci: Spectroscopic science reduction

Functions

- static int mcd_spec_sci_create (cpl_plugin *plugin)
  Setup the recipe options.
- static int mcd_spec_sci_exec (cpl_plugin *plugin)
  Execute the plugin instance given by the interface.
- static int mcd_spec_sci_destroy (cpl_plugin *plugin)
  Destroy what has been created by the 'create' function.
- static cpl_error_code mcd_spec_sci (cpl_frameset *frameset, const cpl_parameterlist *parlist)
  Interpret the command line options and execute the data processing.

3.11.1 Detailed Description

Author

O. Czoske, W. Kausch

Main spectroscopic science reduction recipe

3.11.2 Function Documentation

3.11.2.1 mcd_spec_sci()

```
static cpl_error_code mcd_spec_sci (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

0 if everything is ok

Sort the input frameset

Background subtraction

offset mode

direct mode

SCI_FLUX_2D_ORDERS: separate rectified orders

SCI_FLUX_2D: merged orders (single extension)

SCI_FLUX_1D: extracted spectra

Definition at line 261 of file mcd_spec_sci.c.

Referenced by mcd_spec_sci_exec().

3.11.2.2 mcd_spec_sci_create()

```
static int mcd_spec_sci_create (
            cpl_plugin * plugin )  [static]
```

Setup the recipe options.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Defining the command-line/configuration parameters for the recipe.

Definition at line 180 of file mcd_spec_sci.c.

3.11.2.3 mcd_spec_sci_destroy()

```
static int mcd_spec_sci_destroy (
            cpl_plugin * plugin )  [static]
```

Destroy what has been created by the 'create' function.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 237 of file mcd_spec_sci.c.

3.11.2.4 mcd_spec_sci_exec()

```
static int mcd_spec_sci_exec (
            cpl_plugin * plugin )  [static]
```

Execute the plugin instance given by the interface.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 216 of file mcd_spec_sci.c.

References mcd_spec_sci().

3.12 Recipe mcd_spec_tellcorr: Telluric correction

Functions

- static int mcd_spec_tellcorr_create (cpl_plugin ∗plugin)
  Setup the recipe options.
- static int mcd_spec_tellcorr_exec (cpl_plugin ∗plugin)
  Execute the plugin instance given by the interface.
- static int mcd_spec_tellcorr_destroy (cpl_plugin ∗plugin)
  Destroy what has been created by the 'create' function.
- static cpl_error_code mcd_spec_tellcorr (cpl_frameset ∗frameset, const cpl_parameterlist ∗parlist)
  Interpret the command line options and execute the data processing.

3.12.1 Detailed Description

Author

O. Czoske, W. Kausch

Telluric correction recipe

3.12.2 Function Documentation

3.12.2.1 mcd_spec_tellcorr()

```
static cpl_error_code mcd_spec_tellcorr (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

0 if everything is ok

Sort the input frameset

SCI_FLUX_1D_TELL: corrected spectra

Definition at line 249 of file mcd_spec_tellcorr.c.

Referenced by mcd_spec_tellcorr_exec().

3.12.2.2 mcd_spec_tellcorr_create()

```
static int mcd_spec_tellcorr_create (
            cpl_plugin * plugin )  [static]
```

Setup the recipe options.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Defining the command-line/configuration parameters for the recipe.

Definition at line 166 of file mcd_spec_tellcorr.c.

3.12.2.3 mcd_spec_tellcorr_destroy()

```
static int mcd_spec_tellcorr_destroy (
            cpl_plugin * plugin )  [static]
```

Destroy what has been created by the 'create' function.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 223 of file mcd_spec_tellcorr.c.

3.12.2.4 mcd_spec_tellcorr_exec()

```
static int mcd_spec_tellcorr_exec (
            cpl_plugin * plugin )  [static]
```

Execute the plugin instance given by the interface.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 202 of file mcd_spec_tellcorr.c.

References mcd_spec_tellcorr().

3.13 Recipe mcd_spec_wave: Compute wavelength solution

Functions

- static int mcd_spec_wave_create (cpl_plugin ∗plugin)

  Setup the recipe options.
- static int mcd_spec_wave_exec (cpl_plugin ∗plugin)

  Execute the plugin instance given by the interface.
- static int mcd_spec_wave_destroy (cpl_plugin ∗plugin)

  Destroy what has been created by the 'create' function.
- static cpl_error_code mcd_spec_wave (cpl_frameset ∗frameset, const cpl_parameterlist ∗parlist)

  Interpret the command line options and execute the data processing.

3.13.1 Detailed Description

Author

O. Czoske, W. Kausch

Determine orders on Focal Plane Array (FPA), rectification solution, and wavelength calibration on the rectified orders.

3.13.2 Function Documentation

3.13.2.1 mcd_spec_wave()

```
static cpl_error_code mcd_spec_wave (
            cpl_frameset * frameset,
            const cpl_parameterlist * parlist )  [static]
```

Interpret the command line options and execute the data processing.

Parameters

frameset	the frames list
parlist	the parameters list

Returns

0 if everything is ok

Stack the wave_raw images

Create order trace tables

Create rect_solution

Create 2D_WAVE_SOL

Definition at line 259 of file mcd_spec_wave.c.

Referenced by mcd_spec_wave_exec().

3.13.2.2 mcd_spec_wave_create()

```
static int mcd_spec_wave_create (
            cpl_plugin * plugin )  [static]
```

Setup the recipe options.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Defining the command-line/configuration parameters for the recipe.

Definition at line 178 of file mcd_spec_wave.c.

3.13.2.3 mcd_spec_wave_destroy()

```
static int mcd_spec_wave_destroy (
            cpl_plugin * plugin )  [static]
```

Destroy what has been created by the 'create' function.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 235 of file mcd_spec_wave.c.

3.13.2.4 mcd_spec_wave_exec()

```
static int mcd_spec_wave_exec (
            cpl_plugin * plugin )  [static]
```

Execute the plugin instance given by the interface.

Parameters

plugin	the plugin

Returns

0 if everything is ok

Definition at line 214 of file mcd_spec_wave.c.

References mcd_spec_wave().

3.14 Error handling

Macros

- #define mcd_check_error_code(MSG)
  Error handling macro.
- #define mcd_check(BOOL, CODE, MSG) cpl_error_ensure(BOOL, CODE, goto cleanup, MSG)
  Macro ensuring that a condition is met.

3.14.1 Detailed Description

3.14.2 Macro Definition Documentation

3.14.2.1 mcd_check

```
#define mcd_check(
             BOOL,
             CODE,
             MSG )  cpl_error_ensure(BOOL, CODE, goto cleanup, MSG)
```

Macro ensuring that a condition is met.

Parameters

BOOL	The condition to check
CODE	The CPL error code to set if BOOL is non-zero (i.e. false)
MSG	A printf-style message for cpl_error_set_message().

This macro checks that a condition evaluates to true (zero). If that is not the case, the macro sets the specified CPL error code CODE, then goes to a code section tagged `cleanup`, after which all relevant cleanup operations (deletion of framesets etc.) are performed and the function is exited.

Definition at line 62 of file mcd_defs.h.

3.14.2.2 mcd_check_error_code

```
#define mcd_check_error_code(
              MSG )
```

Value:

```
    cpl_error_ensure(cpl_error_get_code()==CPL_ERROR_NONE, \
                     cpl_error_get_code(), goto cleanup, MSG)
```

Error handling macro.

Parameters

MSG	A printf-style message for cpl_error_set_message().

This macro typically follows a function call that sets a cpl_error_code on failure. If that code is not CPL_ERROR↩ _NONE, the macro goes to a code section tagged `cleanup`, after which all relevant cleanup operations (deletion of framesets etc.) are performed and the function is exited.

Definition at line 44 of file mcd_defs.h.

3.15 DFS related functions

Functions

- int mcd_dfs_set_groups (cpl_frameset ∗set)

  Set the group as RAW or CALIB in a frameset.

3.15.1 Detailed Description

3.15.2 Function Documentation

3.15.2.1 mcd_dfs_set_groups()

```
int mcd_dfs_set_groups (
             cpl_frameset * set )
```

Set the group as RAW or CALIB in a frameset.

Parameters

set	the input frameset

Returns

0 if ok, -1 in error case

Definition at line 49 of file mcd_dfs.c.

Referenced by mcd_check_and_set_groups().

3.16 Functions to detrend detector effects

Functions

- cpl_error_code mcd_img_det_bkg (hdrl_imagelist *mcd_imglist)
  Subtract the MasterDark.
- cpl_bivector * mcd_img_det_get_offsets_from_hds (const cpl_frameset *frameset, const cpl_size det_id)
  Extract offset from headers of a frameset.

3.16.1 Detailed Description

3.16.2 Function Documentation

3.16.2.1 mcd_img_det_bkg()

```
cpl_error_code mcd_img_det_bkg (
            hdrl_imagelist * mcd_imglist )
```

Subtract the MasterDark.

Parameters

mcd_imglist	input micado imglist (hdrl_imagelist)
master_dark	MasterDark to subtracted

Returns

CPL_ERROR_NONE if OK

Describe here

- the function algorithm

- How the parameters are used - their validity range
- What needs to be deallocated after the function call

Definition at line 58 of file mcd_img_det.c.

3.16.2.2 mcd_img_det_get_offsets_from_hds()

```
cpl_bivector* mcd_img_det_get_offsets_from_hds (
            const cpl_frameset * frameset,
            const cpl_size det_id )
```

Extract offset from headers of a frameset.

Parameters

frameset	input micado imglist (hdrl_imagelist)
det_id	Detector ID in the DFS, valid from 1 to 9.

Returns

cpl_bivector offs in pixels; need to deallocate

Use 1st frame as the reference; Get CRVAL from the headers' wcs; Convert to pixel using the wcs;

Definition at line 103 of file mcd_img_det.c.

References mcd_io_extkw_load_from_frame().

3.17 Functions to derive and apply distortion corrections

Functions

- static void * mcd_find_nearest_pos (const double *posx, const double *posy, cpl_size npos, int *indx, int *nearest_indx, double *nearest_dist)

  Find the nearest position for each of npos positions.
- cpl_table * mcd_aperture_to_table (const cpl_apertures *aperts)

  Store aperture position into tables.
- static cpl_error_code mcd_pos_table_find_nearest (cpl_table *pos_tab)

  Find nearest neighbor's ID and distances from catalogue and store into the table columns
- static cpl_table * mcd_pos_get_binary (cpl_table *pos_tab, double max_bi_sep, double *med_sep)

  Extract a table only including binaries from catalogue table and erase these from the original
- static cpl_error_code mcd_pos_tab_get_dist (cpl_table *pos_tab, double ref_x, double ref_y)

  Add distance to reference center (DIST) into the position table.
- static cpl_error_code mcd_pos_tab_sort_dist_xy (cpl_table *pos_tb, double ref_x, double ref_y)

Sort the position table according to the ascending distance, x, and y.

- static cpl_error_code mcd_pos_tab_sort (cpl_table *pos_tb, const char *key)

  Sort the position table according to the ascending distance, x, and y.
- static cpl_error_code mcd_pos_tab_lexsort (cpl_table *pos_tb, const char *primary, const char *secondary)

  Sort the position table according to the ascending distance, x, and y.
- static cpl_error_code mcd_pos_tab_to_grid (cpl_table *pos_tab, double grid_size, double ref_x, double ref_y)

  Assign row, col and grid index to the positions

- static int mcd_find_cluster (cpl_table *pos_tab, double radius)

  Get pattern codes and centers of binary clusters.
- static cpl_table * mcd_get_binary_pattern (cpl_table *binary_tab, double grid_size)

  Get pattern codes and centers of binary clusters.
- cpl_table * mcd_get_binary_pattern_from_pos (cpl_table *pinhole_tb, double max_bi_sep)

  Get pattern codes and centers of binary clusters.
- static cpl_error_code mcd_match_ref_pos (cpl_table *measured_pos, cpl_table *ref_pos, const char *match_key)

  Match binary patterns to the reference catalogue using the 8-bit codes.
- static cpl_matrix * mcd_legendre2d_tensor (cpl_matrix *grid_x, cpl_matrix *grid_y, int order)

  Get 2-D Legendre tensor for (grid_x, grid_y)
- static cpl_matrix * mcd_fit_legendre2d (cpl_matrix *grid_x, cpl_matrix *grid_y, int order, cpl_matrix *to_fit)

  Get 2-D Legendre coefficients to fit (grid_x, grid_y, to_fit)
- static cpl_matrix * mcd_eval_legendre2d (cpl_matrix *grid_x, cpl_matrix *grid_y, cpl_matrix *coeffs)

  Get 2-D Legendre coefficients to fit (grid_x, grid_y, to_fit)
- static cpl_matrix * mcd_fit_distortion (cpl_table *match_tab, int order, double rms[3])

  Fit distortion using pinholes

- static cpl_error_code mcd_apply_distortion (cpl_table *pos_tab, cpl_matrix *coeffs)

  Fit distortion using pinholes

- static cpl_error_code mcd_fit_distortion_poly (cpl_table *match_tab, int order, cpl_polynomial *coeffs_x, cpl_polynomial *coeffs_y)

  Fit distortion to 2d polynomials using pinhole positions.
- static cpl_error_code mcd_apply_distortion_poly (cpl_table *pos_tab, cpl_polynomial *coeffs_x, cpl_polynomial *coeffs_y)

  Fit distortion using pinholes with 2d polynomials fitting.
- cpl_error_code mcd_match_ref_pos_closest (cpl_table *measured_pos, const cpl_table *ref_pos, double max_dist)

  Match pin hole positions to the reference catalogue using closest distance.
- cpl_table * mcd_get_low_order_corrected_pinhole (const cpl_image *raw_wam, cpl_table *ref_binary_pattern, double sigma, cpl_wcs *wcs, double max_bi_sep)

  Get the pin-hole positions corrected with low-oder polynomials derived from binaries.

- cpl_table * mcd_get_distortion_from_pinhole (cpl_table *apert_pos_tab, const cpl_table *ref_pinhole_tb, int order, double rms[3])

  Derive distortion coefficients table for pin-hole positions.

- static cpl_error_code mcd_get_centroid (const cpl_image *img, double *xcen, double *ycen)

  Find centers of a star on an image.
- static cpl_error_code mcd_pos_tab_centroid (const cpl_image *img, cpl_table *pos_tab)

  Find centers of a star on an image.
- static cpl_matrix * mcd_pos_tab_to_matrix (cpl_table *pos_tb)

Convert position table into matrices for catalogue matching

- cpl_error_code mcd_pos_tab_trans (cpl_table ∗pos_tab, cpl_wcs ∗wcs, cpl_wcs_trans_mode transform)
  Convert positions position table according to the wcs.

3.17.1 Detailed Description

3.17.2 Function Documentation

3.17.2.1 mcd_aperture_to_table()

```
cpl_table* mcd_aperture_to_table (
            const cpl_apertures * aperts )
```

Store aperture position into tables.

Parameters

aperts	cpl_apertures, e.g. extracted from an image

Returns

aperts_tab cpl_table wiht colums POS_X, POS_Y, FLUX

Definition at line 134 of file mcd_img_distort.c.

Referenced by mcd_get_low_order_corrected_pinhole(), and mcd_img_phot_match_refcat_from_aperts().

3.17.2.2 mcd_apply_distortion()

```
cpl_error_code mcd_apply_distortion (
            cpl_table * pos_tab,
            cpl_matrix * coeffs )  [static]
```

Fit distortion using pinholes

Parameters

pos_tab	Catalogue position table includes POS_X, POS_Y
coeffs	Legendre polinomials matrix (order+1)^2 rows x 2 cols

Returns

Definition at line 839 of file mcd_img_distort.c.

References mcd_legendre2d_tensor().

3.17.2.3 mcd_apply_distortion_poly()

```
static cpl_error_code mcd_apply_distortion_poly (
            cpl_table * pos_tab,
            cpl_polynomial * coeffs_x,
            cpl_polynomial * coeffs_y )  [static]
```

Fit distortion using pinholes with 2d polynomials fitting.

Parameters

pos_tab	Catalogue position table includes POS_X, POS_Y
coeffs	Legendre polinomials matrix (order+1)^2 rows x 2 cols

Returns

Definition at line 982 of file mcd_img_distort.c.

Referenced by mcd_get_low_order_corrected_pinhole().

3.17.2.4 mcd_eval_legendre2d()

```
static cpl_matrix* mcd_eval_legendre2d (
            cpl_matrix * grid_x,
            cpl_matrix * grid_y,
            cpl_matrix * coeffs )  [static]
```

Get 2-D Legendre coefficients to fit (grid_x, grid_y, to_fit)

Parameters

grid_x	Matrix for X postions 1xn or nx1
grid_y	Matrix for Y postions 1xn or nx1
order	Order of the Legendre polinomials

Returns

Coefficients of the Legendre series (order+1)^2

Definition at line 726 of file mcd_img_distort.c.

References mcd_legendre2d_tensor().

3.17.2.5 mcd_find_cluster()

```
static int mcd_find_cluster (
            cpl_table * pos_tab,
            double radius )  [static]
```

Get pattern codes and centers of binary clusters.

Parameters

pos_tab	Position table includes POS_X, POS_Y
radius	Radius for neighborhood searching

Returns

cluster_id Count of the clusters found in the table

Definition at line 420 of file mcd_img_distort.c.

References mcd_pos_tab_get_dist().

Referenced by mcd_get_binary_pattern().

3.17.2.6 mcd_find_nearest_pos()

```
static void* mcd_find_nearest_pos (
            const double * posx,
            const double * posy,
            cpl_size npos,
            int * indx,
            int * nearest_indx,
            double * nearest_dist )  [static]
```

Find the nearest position for each of npos positions.

Parameters

posx	X positions
posy	Y positions
npos	Number of the points
nearest_indx	The nearest neighbor index
nearest_dist	The nearest neighbor distance

Returns

Definition at line 79 of file mcd_img_distort.c.

Referenced by mcd_pos_table_find_nearest().

3.17.2.7 mcd_fit_distortion()

```
static cpl_matrix * mcd_fit_distortion (
            cpl_table * match_tab,
            int order,
            double rms[3] )  [static]
```

Fit distortion using pinholes

Parameters

match_tab	Catalogue position table includes matched POS_X, POS_Y, POS_X_REF, POS_Y_REF in arcsec
order	Order of the Legendre polinomials to be fitted

Returns

Coefficients of the Legendre series (order+1)^2 x 2

Definition at line 745 of file mcd_img_distort.c.

References mcd_legendre2d_tensor().

Referenced by mcd_get_distortion_from_pinhole().

3.17.2.8 mcd_fit_distortion_poly()

```
static cpl_error_code mcd_fit_distortion_poly (
            cpl_table * match_tab,
            int order,
            cpl_polynomial * coeffs_x,
            cpl_polynomial * coeffs_y )  [static]
```

Fit distortion to 2d polynomials using pinhole positions.

Parameters

match_tab	Catalogue position table includes matched POS_X, POS_Y, POS_X_REF, POS_Y_REF in arcsec
order	Order of the Legendre polinomials to be fitted

Returns

Coefficients of the Legendre series (order+1)^2 x 2

Definition at line 900 of file mcd_img_distort.c.

References mcd_pos_tab_to_matrix().

Referenced by mcd_get_low_order_corrected_pinhole().

3.17.2.9 mcd_fit_legendre2d()

```
static cpl_matrix* mcd_fit_legendre2d (
            cpl_matrix * grid_x,
            cpl_matrix * grid_y,
            int order,
            cpl_matrix * to_fit )  [static]
```

Get 2-D Legendre coefficients to fit (grid_x, grid_y, to_fit)

Parameters

grid_x	Matrix for X postions 1xn or nx1
grid_y	Matrix for Y postions 1xn or nx1
order	Order of the Legendre polinomials

Returns

Coefficients of the Legendre series (order+1)^2

Definition at line 707 of file mcd_img_distort.c.

References mcd_legendre2d_tensor().

3.17.2.10 mcd_get_binary_pattern()

```
static cpl_table * mcd_get_binary_pattern (
            cpl_table * binary_tab,
            double grid_size )  [static]
```

Get pattern codes and centers of binary clusters.

Parameters

binary_tab	Binary cluster pattern, includes POS_X, POS_Y
grid_size	grid size of the pattern

Returns

binary_pattern_tab with CODE to be mathced with reference catalogue

Definition at line 478 of file mcd_img_distort.c.

References mcd_find_cluster(), mcd_pos_tab_lexsort(), and mcd_pos_tab_to_grid().

Referenced by mcd_get_binary_pattern_from_pos().

3.17.2.11 mcd_get_binary_pattern_from_pos()

```
cpl_table* mcd_get_binary_pattern_from_pos (
            cpl_table * pinhole_tb,
            double max_bi_sep )
```

Get pattern codes and centers of binary clusters.

Parameters

pinhole_tb	Pinhole table containing binaries, with POS_X, POS_Y columns
grid_size	grid size of the pattern

Returns

Definition at line 587 of file mcd_img_distort.c.

References mcd_get_binary_pattern(), and mcd_pos_get_binary().

Referenced by mcd_get_low_order_corrected_pinhole().

3.17.2.12 mcd_get_centroid()

```
static cpl_error_code mcd_get_centroid (
            const cpl_image * img,
            double * xcen,
            double * ycen )  [static]
```

Find centers of a star on an image.

Parameters

img	a full image which may include a collection of stars
xcen	initial x position of the star
ycen	initial y position of the star

Returns

Definition at line 1304 of file mcd_img_distort.c.

Referenced by mcd_pos_tab_centroid().

3.17.2.13 mcd_get_distortion_from_pinhole()

```
cpl_table* mcd_get_distortion_from_pinhole (
            cpl_table * apert_pos_tab,
            const cpl_table * ref_pinhole_tb,
            int order,
            double rms[3] )
```

Derive distortion coefficients table for pin-hole positions.

Parameters

apert_pos_tab	Pinhole positions with low-order correction applied.
ref_pinhole↩ _tb	Reference pin-hole position table, including POS_X, POS_Y
order	Order of the distortion fitting

Returns

Definition at line 1168 of file mcd_img_distort.c.

References mcd_fit_distortion(), and mcd_match_ref_pos_closest().

3.17.2.14 mcd_get_low_order_corrected_pinhole()

```
cpl_table* mcd_get_low_order_corrected_pinhole (
            const cpl_image * raw_wam,
            cpl_table * ref_binary_pattern,
            double sigma,
            cpl_wcs * wcs,
            double max_bi_sep )
```

Get the pin-hole positions corrected with low-oder polynomials derived from binaries.

Parameters

raw_wam	RAW WAM image with pinholes
ref_binary_pattern	Reference binary pattern table, including POS_X, POS_Y
sigma	Sigma for extracting pinhole apertures
wcs	WCS of the RAW WAM image
max_bi_sep	Maximal seperation for identifying binaries, i.e. sources are binaries if seperation $<$ max_bi_sep

Returns

apert_pos_tab with corrected positions C_POS_X and C_POS_Y

Definition at line 1114 of file mcd_img_distort.c.

References mcd_aperture_to_table(), mcd_apply_distortion_poly(), mcd_fit_distortion_poly(), mcd_get_binary_↩ pattern_from_pos(), mcd_match_ref_pos(), mcd_pos_tab_centroid(), and mcd_pos_tab_trans().

3.17.2.15 mcd_legendre2d_tensor()

```
static cpl_matrix* mcd_legendre2d_tensor (
            cpl_matrix * grid_x,
            cpl_matrix * grid_y,
            int order )  [static]
```

Get 2-D Legendre tensor for (grid_x, grid_y)

Parameters

grid↩ _x	Matrix for X postions 1xn or nx1
grid↩ _y	Matrix for Y postions 1xn or nx1
order	Order of the Legendre polinomials

Returns

Tensor matrix n_grid x(order+1)^2

Definition at line 670 of file mcd_img_distort.c.

Referenced by mcd_apply_distortion(), mcd_eval_legendre2d(), mcd_fit_distortion(), and mcd_fit_legendre2d().

3.17.2.16 mcd_match_ref_pos()

```
static cpl_error_code mcd_match_ref_pos (
            cpl_table * measured_pos,
            cpl_table * ref_pos,
            const char * match_key )  [static]
```

Match binary patterns to the reference catalogue using the 8-bit codes.

Parameters

measured_pos	Measured inary pattern table, including POS_X, POS_Y, key
ref_pos	Reference binary pattern table, including POS_X, POS_Y, key

Returns

Definition at line 610 of file mcd_img_distort.c.

Referenced by mcd_get_low_order_corrected_pinhole().

3.17.2.17 mcd_match_ref_pos_closest()

```
cpl_error_code mcd_match_ref_pos_closest (
            cpl_table * measured_pos,
            const cpl_table * ref_pos,
            double max_dist )
```

Match pin hole positions to the reference catalogue using closest distance.

Parameters

measured_pos	Measured position table, including C_POS_X, C_POS_Y
ref_pos	Reference pinhole position table, including POS_X, POS_Y
max_dist	Tolerance

Returns

Definition at line 1045 of file mcd_img_distort.c.

References mcd_pos_tab_get_dist().

Referenced by mcd_get_distortion_from_pinhole().

3.17.2.18 mcd_pos_get_binary()

```
static cpl_table * mcd_pos_get_binary (
             cpl_table * pos_tab,
             double max_bi_sep,
             double * med_sep )  [static]
```

Extract a table only including binaries from catalogue table and erase these from the original

Parameters

pos_tab	catalogue position table includes POS_X, POS_Y
max_bi_sep	maximum distance for binaries

Returns

binary_pos_tab cpl_table with colums POS_X, POS_Y, COL_ID, ROW_ID, GRID_ID

Definition at line 206 of file mcd_img_distort.c.

References mcd_pos_tab_lexsort(), mcd_pos_tab_to_grid(), and mcd_pos_table_find_nearest().

Referenced by mcd_get_binary_pattern_from_pos().

3.17.2.19 mcd_pos_tab_centroid()

```
cpl_error_code mcd_pos_tab_centroid (
             const cpl_image * img,
             cpl_table * pos_tab )  [static]
```

Find centers of a star on an image.

Parameters

img	a full image which may include a collection of stars
pos_tab	catalogue position table includes POS_X, POS_Y in units of pixels

Returns

Definition at line 1343 of file mcd_img_distort.c.

References mcd_get_centroid().

Referenced by mcd_get_low_order_corrected_pinhole().

3.17.2.20 mcd_pos_tab_get_dist()

```
static cpl_error_code mcd_pos_tab_get_dist (
             cpl_table * pos_tab,
             double ref_x,
             double ref_y )  [static]
```

Add distance to reference center (DIST) into the position table.

Parameters

pos_tab	catalogue position table includes POS_X, POS_Y
ref_x	reference (origin) positon of x
ref_y	reference (origin) position of y

Returns

Definition at line 261 of file mcd_img_distort.c.

Referenced by mcd_find_cluster(), mcd_match_ref_pos_closest(), and mcd_pos_tab_sort_dist_xy().

3.17.2.21 mcd_pos_tab_lexsort()

```
static cpl_error_code mcd_pos_tab_lexsort (
             cpl_table * pos_tb,
             const char * primary,
             const char * secondary )  [static]
```

Sort the position table according to the ascending distance, x, and y.

Parameters

pos_tab	catalogue position table includes primary and sencondary keys
primary	primary key for sorting
secondary	secondary key for sorting

Returns

Definition at line 339 of file mcd_img_distort.c.

Referenced by mcd_get_binary_pattern(), and mcd_pos_get_binary().

3.17.2.22 mcd_pos_tab_sort()

```
static cpl_error_code mcd_pos_tab_sort (
            cpl_table * pos_tb,
            const char * key )  [static]
```

Sort the position table according to the ascending distance, x, and y.

Parameters

pos_tab	catalogue position table includes primary and sencondary keys
key	key for sorting

Returns

Definition at line 318 of file mcd_img_distort.c.

3.17.2.23 mcd_pos_tab_sort_dist_xy()

```
static cpl_error_code mcd_pos_tab_sort_dist_xy (
            cpl_table * pos_tb,
            double ref_x,
            double ref_y )  [static]
```

Sort the position table according to the ascending distance, x, and y.

Parameters

pos_tab	catalogue position table includes POS_X, POS_Y
ref_x	reference (origin) positon of x
ref_y	reference (origin) position of y

Returns

Definition at line 293 of file mcd_img_distort.c.

References mcd_pos_tab_get_dist().

3.17.2.24 mcd_pos_tab_to_grid()

```
static cpl_error_code mcd_pos_tab_to_grid (
             cpl_table * pos_tab,
             double grid_size,
             double ref_x,
             double ref_y )  [static]
```

Assign row, col and grid index to the positions

Parameters

pos_tab	catalogue position table includes POS_X, POS_Y
grid_size	grid size
ref_x	origin on pos_x
ref_y	origin on pos_y

Returns

Definition at line 370 of file mcd_img_distort.c.

Referenced by mcd_get_binary_pattern(), and mcd_pos_get_binary().

3.17.2.25 mcd_pos_tab_to_matrix()

```
cpl_matrix * mcd_pos_tab_to_matrix (
             cpl_table * pos_tb )  [static]
```

Convert position table into matrices for catalogue matching

Parameters

pos_tab	catalogue position table includes POS_X, POS_Y

Returns

Matrix with x position in 1st row and y postion in 2nd row.

Definition at line 1378 of file mcd_img_distort.c.

Referenced by mcd_fit_distortion_poly(), and mcd_pos_tab_trans().

3.17.2.26 mcd_pos_tab_trans()

```
cpl_error_code mcd_pos_tab_trans (
            cpl_table * pos_tab,
            cpl_wcs * wcs,
            cpl_wcs_trans_mode transform )
```

Convert positions position table according to the wcs.

Parameters

pos_tab	catalogue position table includes POS_X, POS_Y
wcs	The input cpl_wcs structure
transform	transformation mode, CPL_WCS_PHYS2WORLD, CPL_WCS_WORLD2PHYS, CPL_WCS_WORLD2STD, CPL_WCS_PHYS2STD

Returns

Definition at line 1401 of file mcd_img_distort.c.

References mcd_pos_tab_to_matrix().

Referenced by mcd_get_low_order_corrected_pinhole(), and mcd_img_phot_match_refcat_from_aperts().

3.17.2.27 mcd_pos_table_find_nearest()

```
static cpl_error_code mcd_pos_table_find_nearest (
            cpl_table * pos_tab )  [static]
```

Find nearest neighbor's ID and distances from catalogue and store into the table columns

Parameters

pos_tab	catalogue position table includes POS_X, POS_Y

Returns

out_tab cpl_table wiht colums POS_X, POS_Y, FLUX, NEAREST_ID, NEAREST_DIST

Definition at line 166 of file mcd_img_distort.c.

References mcd_find_nearest_pos().

Referenced by mcd_pos_get_binary().

3.18 Kdtree libray used for finding nearest neighbours

3.19 Functions to derive and apply photometry and illum corrections

Functions

- static double * mcd_img_phot_match_refcat_from_aperts (const cpl_apertures *aperts, const cpl_wcs *wcs, const cpl_table *ref_cat, double max_dist)

  Match the apertures to a refenrece catalogue.
- static double * mcd_img_phot_match_refcat_from_hcat (const hdrl_catalogue_result *hcat, const cpl_table *ref_cat, double max_dist)

  Match the apertures to a refenrece catalogue.
- static cpl_polynomial * mcd_img_phot_fit2d_illum_from_hcat (const hdrl_catalogue_result *hcat, const double *ref_flux, double *mse)

  Get the illumination pattern from the matched flux using hdrl_catalogue_result.
- static cpl_image * mcd_img_phot_illum_image (const cpl_polynomial *fit2d, cpl_size nx, cpl_size ny)

  Get fitted values of the 2D polynomial
- static cpl_polynomial * mcd_img_phot_fit2d_illum_from_apert (const cpl_apertures *aperts, const double *illum)

  Get the illumination pattern from the matched flux.
- static double * mcd_img_phot_get_illum_from_aperts (const cpl_apertures *aperts, const double *ref_flux)

  Get the illumination pattern from the matched flux.
- static double mcd_img_phot_angsep (double posx1, double posy1, double posx2, double posy2)

  Get angular seperation of two positions.
- cpl_image * mcd_img_phot_offset_saa (hdrl_imagelist *illum_list, cpl_bivector *offs)

  Shift and combine images.
- hdrl_catalogue_result * mcd_img_phot_get_hdrl_catalogue (const cpl_image *img, const cpl_wcs *wcs, double snr_thresh, double obj_fwhm)

  Extract stars catalogue from an image.
- static double * mcd_img_phot_match_refcat (const double *posx, const double *posy, cpl_size n_pos, const double *ref_x, const double *ref_y, const double *ref_f, cpl_size n_ref, double max_dist)

  Match the apertures to a refenrece catalogue.
- static double * mcd_img_phot_get_illum (const double *flux, const double *ref_flux, cpl_size n_flux)

  Get the illumination pattern from the matched flux.
- static cpl_polynomial * mcd_img_phot_fit2d_illum (const double *posx, const double *posy, const double *illum, const double *illum_err, cpl_size n_pos, double *mse)

  Get the illumination pattern from the matched flux.
- hdrl_image * mcd_img_phot_fill_illum (cpl_polynomial *fit2d, double mse, cpl_size nx, cpl_size ny)

  Fill the image with fitted 2D polynomial
- static double mcd_img_phot_fit1d_zeropoint (const double *flux, const double *ref_flux, cpl_size n_pos, double *mse, double *coeff0)

  Get the illumination pattern from the matched flux.
- double mcd_img_phot_fit_zeropoint_from_hcat (const hdrl_catalogue_result *hcat, const cpl_table *ref_cat, double min_sep, double *mse, double *coeff0)

  Get zeropoint using extracted catalogue and reference catalogue.

3.19.1 Detailed Description

3.19.2 Function Documentation

3.19.2.1 mcd_img_phot_angsep()

```
static double mcd_img_phot_angsep (
             double posx1,
             double posy1,
             double posx2,
             double posy2 )  [static]
```

Get angular seperation of two positions.

Parameters

posx_1	1st positon x (RA)
posy_1	1st positon y (DEC)
posx_2	2nd positon x (RA)
posy_2	2nd positon x (RA)

Returns

Array of seperation in angle

Reference:
https://en.wikipedia.org/wiki/Great-circle_distance https://docs.astropy.org/en/stable/_modules/astropy/coordinates/angles/utils.html#angular_separation

Definition at line 92 of file mcd_img_phot.c.

Referenced by mcd_img_phot_match_refcat().

3.19.2.2 mcd_img_phot_fill_illum()

```
hdrl_image* mcd_img_phot_fill_illum (
             cpl_polynomial * fit2d,
             double mse,
             cpl_size nx,
             cpl_size ny )
```

Fill the image with fitted 2D polynomial

Parameters

fit2d	a pointer to cpl_apertures extracted from an image.
mse	error value
nx	number of the columns of the image, i.e. from hdrl_image_get_size_x
ny	number of the rows of the image, i.e. from hdrl_image_get_size_y

Returns

hdrl_image with each pixel fitted from the fit2d and empty mask

Definition at line 741 of file mcd_img_phot.c.

References mcd_img_phot_illum_image().

3.19.2.3 mcd_img_phot_fit1d_zeropoint()

```
static double mcd_img_phot_fit1d_zeropoint (
             const double * flux,
             const double * ref_flux,
             cpl_size n_pos,
             double * mse,
             double * coeff0 )  [static]
```

Get the illumination pattern from the matched flux.

Parameters

flux	a pointer to measured flux
ref_flux	reference flux
n_pos	Number of the positions
mse	fit residual statistic
coeff0	nomalized coeff0

Returns

photoflam (1/throughput)

Description:

- Interface the aperts and illum array into sample position matrix and value vector
- Use cpl_polynomial_fit_2d_create to fit the illumination

Definition at line 778 of file mcd_img_phot.c.

Referenced by mcd_img_phot_fit_zeropoint_from_hcat().

3.19.2.4 mcd_img_phot_fit2d_illum()

```
static cpl_polynomial* mcd_img_phot_fit2d_illum (
            const double * posx,
            const double * posy,
            const double * illum,
            const double * illum_err,
            cpl_size n_pos,
            double * mse )  [static]
```

Get the illumination pattern from the matched flux.

Parameters

posx	a pointer to catalogue positons x (RA)
posy	a pointer to to catalogue positon y (DEC)
illum	Array of illumination factor for each aperture
n_pos	Number of the positions

Returns

cpl_polynomial, 2D polinomial coefficients

Description:

- Interface the aperts and illum array into sample position matrix and value vector
- Use cpl_polynomial_fit_2d_create to fit the illumination

Definition at line 509 of file mcd_img_phot.c.

Referenced by mcd_img_phot_fit2d_illum_from_apert(), and mcd_img_phot_fit2d_illum_from_hcat().

3.19.2.5 mcd_img_phot_fit2d_illum_from_apert()

```
static cpl_polynomial * mcd_img_phot_fit2d_illum_from_apert (
            const cpl_apertures * aperts,
            const double * illum )  [static]
```

Get the illumination pattern from the matched flux.

Parameters

aperts	a pointer to cpl_apertures extracted from an image.
illum	Array of illumination factor for each aperture

Returns

cpl_polynomial, 2D polinomial coefficients

Description:

- Interface the aperts and illum array into sample position matrix and value vector
- Use cpl_polynomial_fit_2d_create to fit the illumination

Definition at line 619 of file mcd_img_phot.c.

References mcd_img_phot_fit2d_illum().

3.19.2.6 mcd_img_phot_fit2d_illum_from_hcat()

```
static cpl_polynomial * mcd_img_phot_fit2d_illum_from_hcat (
            const hdrl_catalogue_result * hcat,
            const double * ref_flux,
            double * mse )  [static]
```

Get the illumination pattern from the matched flux using hdrl_catalogue_result.

Parameters

hcat	hdrl_catalogue_result
ref_flux	matched reference catalogue cpl_table colume FLUX
mse	the pointer to return the nominal fitting residual

Returns

cpl_polynomial, 2D polinomial coefficients

Description:

- Interface the hcat and illum array into sample position matrix and value vector
- Use cpl_polynomial_fit_2d_create to fit the illumination

Definition at line 657 of file mcd_img_phot.c.

References mcd_img_phot_fit2d_illum(), and mcd_img_phot_get_illum().

3.19.2.7 mcd_img_phot_fit_zeropoint_from_hcat()

```
double mcd_img_phot_fit_zeropoint_from_hcat (
            const hdrl_catalogue_result * hcat,
            const cpl_table * ref_cat,
            double min_sep,
            double * mse,
            double * coeff0 )
```

Get zeropoint using extracted catalogue and reference catalogue.

Parameters

hcat	extracted hdrl_catalogue_result
ref_cat	reference catalogue table
min_sep	Minimal seperation in degree to match the catalogue
mse	fit residual statistic
coeff0	nomalized coeff0

Returns

photoflam (1/throughput)

Description:

- Interface the aperts and illum array into sample position matrix and value vector
- Use cpl_polynomial_fit_2d_create to fit the illumination

Definition at line 892 of file mcd_img_phot.c.

References mcd_img_phot_fit1d_zeropoint(), and mcd_img_phot_match_refcat_from_hcat().

3.19.2.8 mcd_img_phot_get_hdrl_catalogue()

```
hdrl_catalogue_result* mcd_img_phot_get_hdrl_catalogue (
            const cpl_image * img,
            const cpl_wcs * wcs,
            double snr_thresh,
            double obj_fwhm )
```

Extract stars catalogue from an image.

Parameters

snr_thresh	obj_threshold
obj_fwhm	obj_core_radius

Returns

hdrl_catalogue_result

Definition at line 177 of file mcd_img_phot.c.

3.19.2.9 mcd_img_phot_get_illum()

```
static double* mcd_img_phot_get_illum (
            const double * flux,
            const double * ref_flux,
            cpl_size n_flux )  [static]
```

Get the illumination pattern from the matched flux.

Parameters

flux	a pointer to measuref flux
ref_flux	array of flux matched by ref_cat for each aperture,
n_flux	number of the flux measurements
max_dist	Tolerance of the maximum distance to match,

Returns

Array of illumination factor for each aperture

Description:

- Convert reference flux to arbitrary ADU, use a fake zeropoint
- Derive the illumination from meausred_flux/reference_flux
- Nomralize the illumination pattern to maximum = 1

Definition at line 410 of file mcd_img_phot.c.

Referenced by mcd_img_phot_fit2d_illum_from_hcat(), and mcd_img_phot_get_illum_from_aperts().

3.19.2.10 mcd_img_phot_get_illum_from_aperts()

```
static double * mcd_img_phot_get_illum_from_aperts (
            const cpl_apertures * aperts,
            const double * ref_flux )  [static]
```

Get the illumination pattern from the matched flux.

Parameters

aperts	aperts a pointer to cpl_apertures extracted from an image.
ref_flux	matched reference catalogue cpl_table colume FLUX

Returns

Array of illumination factor for each aperture

Description:

- Convert reference flux to arbitrary ADU, use a fake zeropoint
- Derive the illumination from meausred_flux/reference_flux
- Nomralize the illumination pattern to maximum = 1

Definition at line 474 of file mcd_img_phot.c.

References mcd_img_phot_get_illum().

3.19.2.11 mcd_img_phot_illum_image()

```
static cpl_image * mcd_img_phot_illum_image (
            const cpl_polynomial * fit2d,
            cpl_size nx,
            cpl_size ny )  [static]
```

Get fitted values of the 2D polynomial

Parameters

fit2d	a pointer to cpl_apertures extracted from an image.
nx	number of the columns of the image, i.e. from hdrl_image_get_size_x
ny	number of the rows of the image, i.e. from hdrl_image_get_size_y

Returns

cpl_image with each pixel fitted from the fit2d

Definition at line 693 of file mcd_img_phot.c.

Referenced by mcd_img_phot_fill_illum().

3.19.2.12 mcd_img_phot_match_refcat()

```
static double* mcd_img_phot_match_refcat (
            const double * posx,
            const double * posy,
            cpl_size n_pos,
            const double * ref_x,
            const double * ref_y,
            const double * ref_f,
            cpl_size n_ref,
            double max_dist )  [static]
```

Match the apertures to a refenrece catalogue.

Parameters

posx	a pointer to catalogue positons x (RA)
posy	a pointer to to catalogue positon y (DEC)
n_pos	numper of the catalogue positions.
ref_x	reference catalogue cpl_table POS_X (RA), POS_Y (DEC), FLUX
ref_y	reference catalogue cpl_table POS_Y (DEC), FLUX
ref_f	reference catalogue cpl_table FLUX
n_ref	number of the reference positions
max_dist	Tolerance of the maximum distance to match, in the same unit of ref_cat POS_X(Y)

Returns

Array of Flux from ref_cat for each aperture.

Description:

- Use wcs convert apertures pixel coord into sky coord.
- Match the reference table using the minimal distance
- What needs to be deallocated after the function call

Definition at line 236 of file mcd_img_phot.c.

References mcd_img_phot_angsep().

Referenced by mcd_img_phot_match_refcat_from_aperts(), and mcd_img_phot_match_refcat_from_hcat().

3.19.2.13 mcd_img_phot_match_refcat_from_aperts()

```
static double * mcd_img_phot_match_refcat_from_aperts (
            const cpl_apertures * aperts,
            const cpl_wcs * wcs,
            const cpl_table * ref_cat,
            double max_dist )  [static]
```

Match the apertures to a refenrece catalogue.

Parameters

aperts	a pointer to cpl_apertures extracted from an image.
wcs	a pointer to the cpl_wcs of the image.
ref_cat	reference catalogue cpl_table, including POS_X (RA), POS_Y (DEC), FLUX
max_dist	Tolerance of the maximum distance to match, in the same unit of ref_cat POS_X(Y)

Returns

Array of Flux from ref_cat for each aperture.

Description:

- Use wcs convert apertures pixel coord into sky coord.
- Match the reference table using the minimal distance
- What needs to be deallocated after the function call

Definition at line 319 of file mcd_img_phot.c.

References mcd_aperture_to_table(), mcd_img_phot_match_refcat(), and mcd_pos_tab_trans().

3.19.2.14 mcd_img_phot_match_refcat_from_hcat()

```
static double * mcd_img_phot_match_refcat_from_hcat (
            const hdrl_catalogue_result * hcat,
            const cpl_table * ref_cat,
            double max_dist )  [static]
```

Match the apertures to a refenrece catalogue.

Parameters

hcat	hdrl_catalogue_result from hdrl_catalogue_compute
ref_cat	reference catalogue cpl_table, including POS_X (RA), POS_Y (DEC), FLUX
max_dist	Tolerance of the maximum distance to match, in the same unit of ref_cat POS_X(Y)

Returns

Array of Flux from ref_cat for each aperture.

Description:

- Use wcs convert apertures pixel coord into sky coord.
- Match the reference table using the minimal distance
- What needs to be deallocated after the function call

Definition at line 370 of file mcd_img_phot.c.

References mcd_img_phot_match_refcat().

Referenced by mcd_img_phot_fit_zeropoint_from_hcat().

3.19.2.15 mcd_img_phot_offset_saa()

```
cpl_image* mcd_img_phot_offset_saa (
             hdrl_imagelist * illum_list,
             cpl_bivector * offs )
```

Shift and combine images.

Parameters

illum_list	hdrl_imagelist
offs	offsets

Returns

Pointer to the combined image (cpl_image)

Definition at line 123 of file mcd_img_phot.c.

References mcd_io_himglist_to_cimglist().

3.20 I/O related functions

Functions

- hdrl_image * mcd_io_image_load_from_file (const char *file, const cpl_type det_id, int mode)

  Load the image for one specific detector from a fits file.
- hdrl_image * mcd_io_image_load_from_frame (const cpl_frame *frame, const cpl_type det_id, int mode)

  Load the image for one specific detector from a CPL frame.
- cpl_propertylist * mcd_io_extkw_load_from_file (const char *file, const cpl_type ext_id)

  Load the extension header for one specific detector from a fits file.
- cpl_propertylist * mcd_io_extkw_load_from_frame (const cpl_frame *frame, const cpl_type det_id)

  Load the extension header for one specific detector from a CPL frame.
- hdrl_imagelist * mcd_io_imagelist_load_from_frameset (const cpl_frameset *frameset, const cpl_size det_id, int mode)

Load multiple images from the frameset to an hdrl_imagelist for one detector.

- cpl_imagelist * mcd_io_himglist_to_cimglist (const hdrl_imagelist *hlist)

  Convert hdrl_imagelist into cpl_imagelist to use cpl functions.
- static char * mcd_io_create_extname (int det_id, int mode)

  Create Extname.
- static int mcd_io_get_ext_idx (const char *filename, int det_id, int mode)

  Get extension number of a detector given the det_id and mode.
- int mcd_io_save_himg (const char *filename, cpl_frameset *allframes, const cpl_frameset *usedframes, const cpl_frame *inherit, const cpl_parameterlist *parlist, hdrl_image **himg, const cpl_propertylist *qc_list, cpl_propertylist **ext_plist, cpl_type type, const char *recipe, const char *procatg, const char *proclass)

  Save a multi extension images.
- int mcd_io_save_var_himg (const char *filename, cpl_frameset *allframes, const cpl_frameset *usedframes, const cpl_frame *inherit, const cpl_parameterlist *parlist, int nimages, hdrl_image **himg, const cpl_↩ propertylist *qc_list, cpl_propertylist **ext_plist, cpl_type type, const char *recipe, const char *procatg, const char *proclass)

  Save a list of images as a DFS-compliant multi-extension FITS file.
- int mcd_io_save_table (const char *filename, cpl_frameset *allframes, const cpl_frameset *usedframes, const cpl_frame *inherit, const cpl_parameterlist *parlist, cpl_table **tables, const cpl_propertylist *qc_list, cpl_propertylist **ext_plist, const char *recipe, const char *procatg, const char *proclass)

  Save a multi extension table.
- cpl_table * mcd_io_table_load_from_file (const char *file, const cpl_type det_id)

  Load the image for one specific detector from a fits file.
- cpl_error_code mcd_io_save_var_table (const char *filename, cpl_frameset *allframes, const cpl_frameset *usedframes, const cpl_frame *inherit, const cpl_parameterlist *parlist, int ntables, cpl_table **tables, const cpl_propertylist *qc_list, cpl_propertylist **ext_plist, const char *recipe, const char *procatg, const char *proclass)

  Save a list of tables as a DFS-compliant multi-extension FITS file.

3.20.1 Detailed Description

3.20.2 Function Documentation

3.20.2.1 mcd_io_create_extname()

```
static char * mcd_io_create_extname (
            int det_id,
            int mode )  [static]
```

Create Extname.

Parameters

det↩ _id	The wished det_id (1 to MICADO_NB_DETECTORS)
mode	1 for the data image, 2 for the error, 3 for bpm, 0 for raw simulated data

Returns

the newly allocated string with the EXTNAME

Definition at line 289 of file mcd_io.c.

Referenced by mcd_io_get_ext_idx(), mcd_io_save_himg(), and mcd_io_save_table().

3.20.2.2 mcd_io_extkw_load_from_file()

```
cpl_propertylist* mcd_io_extkw_load_from_file (
            const char * file,
            const cpl_type ext_id )
```

Load the extension header for one specific detector from a fits file.

Parameters

file	Fits file to be loaded
det↩ _id	Detector ID in the DFS, valid from 1 to 9.

Returns

1 extension header (cpl_propertylist) or NULL in case of error

Definition at line 171 of file mcd_io.c.

Referenced by mcd_io_extkw_load_from_frame(), and mcd_io_image_load_from_file().

3.20.2.3 mcd_io_extkw_load_from_frame()

```
cpl_propertylist* mcd_io_extkw_load_from_frame (
            const cpl_frame * frame,
            const cpl_type det_id )
```

Load the extension header for one specific detector from a CPL frame.

Parameters

frame	Fits file to be loaded
det↩ _id	Detector ID in the DFS, valid from 1 to 9.

Returns

1 extension header (cpl_propertylist) or NULL in case of error

See also

mcd_io_extkw_load_from_file

Definition at line 193 of file mcd_io.c.

References mcd_io_extkw_load_from_file(), and mcd_io_get_ext_idx().

Referenced by mcd_img_det_get_offsets_from_hds().

3.20.2.4 mcd_io_get_ext_idx()

```
static int mcd_io_get_ext_idx (
            const char * filename,
            int det_id,
            int mode )  [static]
```

Get extension number of a detector given the det_id and mode.

Parameters

filename	The FITS file name
det_id	The wished det_id (1 to MICADO_NB_DETECTORS)
mode	1 for the data image, 2 for the error, 3 for bpm, 0 for raw data

Returns

The extention index of -1 in error case, to be used to point to the data from the filename.

Get the wished name from mcd_io_create_extname. Loop over each extension header to serach for the wished one.

Definition at line 324 of file mcd_io.c.

References mcd_io_create_extname().

Referenced by mcd_io_extkw_load_from_frame(), mcd_io_image_load_from_file(), and mcd_io_table_load_from↩_file().

3.20.2.5 mcd_io_himglist_to_cimglist()

```
cpl_imagelist* mcd_io_himglist_to_cimglist (
            const hdrl_imagelist * hlist )
```

Convert hdrl_imagelist into cpl_imagelist to use cpl functions.

Parameters

hlist	hdrl_imagelist

Returns

1 cpl_imagelist object or NULL in case of error

Needs cpl_imagelist_unwrap() to free

Definition at line 262 of file mcd_io.c.

Referenced by mcd_img_phot_offset_saa().

3.20.2.6 mcd_io_image_load_from_file()

```
hdrl_image* mcd_io_image_load_from_file (
             const char * file,
             const cpl_type det_id,
             int mode )
```

Load the image for one specific detector from a fits file.

Parameters

file	Fits file to be loaded
det↩ _id	Detector ID in the DFS, valid from 1 to 9.
mode	0 for raw data

Returns

1 hdrl_image or NULL in case of error

Load the IMG and extkw from the fits file

IF mode == 0: Create ERR and DQ ELSE : Load the ERR and DQ layers from the fits file

Set error to IMG if it is not raw data (mode == 0).

Definition at line 67 of file mcd_io.c.

References mcd_io_extkw_load_from_file(), and mcd_io_get_ext_idx().

Referenced by mcd_io_image_load_from_frame(), and mcd_io_imagelist_load_from_frameset().

3.20.2.7 mcd_io_image_load_from_frame()

```
hdrl_image* mcd_io_image_load_from_frame (
              const cpl_frame * frame,
              const cpl_type det_id,
              int mode )
```

Load the image for one specific detector from a CPL frame.

Parameters

frame	CPL frame
det_id	Detector ID in the DFS, valid from 1 to 9.
mode	0 values will be used for create shot noise for raw data

Returns

1 hdrl_image or NULL in case of error

See also

mcd_io_image_load_from_file

Load the IMG and extkw from a CPL frame

IF mode == 0: Create ERR and DQ ELSE : Load the ERR and DQ layers from from a CPL frame

Set error to IMG if it is not raw data (mode == 0).

Definition at line 148 of file mcd_io.c.

References mcd_io_image_load_from_file().

3.20.2.8 mcd_io_imagelist_load_from_frameset()

```
hdrl_imagelist* mcd_io_imagelist_load_from_frameset (
              const cpl_frameset * frameset,
              const cpl_size det_id,
              int mode )
```

Load multiple images from the frameset to an hdrl_imagelist for one detector.

Parameters

frameset	
det_id	Detector ID in the DFS, valid from 1 to 9.
mode	1 for the data image, 2 for the error, 3 for bpm, 0 for raw simulated data

Returns

1 hdrl_imagelist object or NULL in case of error

Loop over each frame in the frameset Get the file from the frameset Use mcd_io_image_load_from_file(file, det_id, gain) to load data Set to into the hilist

Definition at line 222 of file mcd_io.c.

References mcd_io_image_load_from_file().

3.20.2.9 mcd_io_save_himg()

```
int mcd_io_save_himg (
             const char * filename,
             cpl_frameset * allframes,
             const cpl_frameset * usedframes,
             const cpl_frame * inherit,
             const cpl_parameterlist * parlist,
             hdrl_image ** himg,
             const cpl_propertylist * qc_list,
             cpl_propertylist ** ext_plist,
             cpl_type type,
             const char * recipe,
             const char * procatg,
             const char * proclass )
```

Save a multi extension images.

Parameters

filename	The FITS file name
allframes	The recipe input frames
usedframes	The recipe used input frames
inherit	NULL or product frames inherit their header from this frame
parlist	The recipe input parameters
himg	The hdrl_image array to save (data and error per det_id)
qc_list	The QC parameters
ext_plist	The extensions property lists
type	CPL_TYPE_DOUBLE, CPL_TYPE_INT,...
recipe	The recipe name
procatg	PRO.CATG
proclass	PRO.CLASS

Returns

0 if ok, -1 in error case

Definition at line 398 of file mcd_io.c.

References mcd_io_create_extname().

Referenced by mcd_detector_imagelist_save().

3.20.2.10 mcd_io_save_table()

```
int mcd_io_save_table (
             const char * filename,
             cpl_frameset * allframes,
             const cpl_frameset * usedframes,
             const cpl_frame * inherit,
             const cpl_parameterlist * parlist,
             cpl_table ** tables,
             const cpl_propertylist * qc_list,
             cpl_propertylist ** ext_plist,
             const char * recipe,
             const char * procatg,
             const char * proclass )
```

Save a multi extension table.

Parameters

filename	The FITS file name
allframes	The recipe input frames
usedframes	The recipe used input frames
inherit	NULL or product frames inherit their header from this frame
parlist	The recipe input parameters
tables	The cpl_table array to save
qc_list	The QC parameters
ext_plist	The extensions property lists
recipe	The recipe name
procatg	PRO.CATG
proclass	PRO.CLASS

Returns

0 if ok, -1 in error case

Definition at line 719 of file mcd_io.c.

References mcd_io_create_extname().

3.20.2.11 mcd_io_save_var_himg()

```
int mcd_io_save_var_himg (
             const char * filename,
```

```
cpl_frameset * allframes,
const cpl_frameset * usedframes,
const cpl_frame * inherit,
const cpl_parameterlist * parlist,
int nimages,
hdrl_image ** himg,
const cpl_propertylist * qc_list,
cpl_propertylist ** ext_plist,
cpl_type type,
const char * recipe,
const char * procatg,
const char * proclass )
```

Save a list of images as a DFS-compliant multi-extension FITS file.

Parameters

filename	The FITS file name
allframes	The recipe input frames
usedframes	The list of raw/calibration frames used for this product
inherit	NULL or product frames inherit their header from this frame
parlist	The list of recipe parameters
nimages	Number of hdrl_image to save
himg	Array of hdrl_image to save (data and error per det_id)
qc_list	Property list containing QC parameters (primary header)
ext_plist	Extensions property lists
type	CPL_TYPE_DOUBLE, CPL_TYPE_INT,...
recipe	The recipe name
procatg	PRO.CATG
proclass	PRO.CLASS

Returns

CPL_ERROR_NONE or the relevant CPL error code on error

See also

mcd_io_save_himg()

This function saves an arbitrary number of images to a multi-extension FITS file (unlike mcd_io_save_himg(), which assumes that each image corresponds to a MICADO detector). Extension names are not set automatically; naming of extensions via the EXTNAME property in the extension property lists is the responsability of the user. Save the extensions

Definition at line 557 of file mcd_io.c.

Referenced by mcd_image_list_save().

3.20.2.12 mcd_io_save_var_table()

```
cpl_error_code mcd_io_save_var_table (
             const char * filename,
             cpl_frameset * allframes,
             const cpl_frameset * usedframes,
             const cpl_frame * inherit,
             const cpl_parameterlist * parlist,
             int ntables,
             cpl_table ** tables,
             const cpl_propertylist * qc_list,
             cpl_propertylist ** ext_plist,
             const char * recipe,
             const char * procatg,
             const char * proclass )
```

Save a list of tables as a DFS-compliant multi-extension FITS file.

Parameters

filename	The FITS file name
allframes	The recipe input frames
usedframes	The list of raw/calibration frames used for this product
inherit	NULL or product frames inherit their header from this frame
parlist	The list of recipe parameters
ntables	Number of cpl_table to save
tables	Array of cpl_table to save
qc_list	Property list containing QC parameters
ext_plist	Extension property lists
recipe	The recipe name
procatg	PRO.CATG
proclass	PRO.CLASS

Returns

CPL_ERROR_NONE or the relevant CPL error code on error

See also

mcd_io_save_table()

This function saves an arbitrary number of tables to a multi-extension FITS file (unlike mcd_io_save_table(), which assumes that each table corresponds to a MICADO detector). Extension names are not set automatically; naming of extensions via the EXTNAME property in the extension property lists is the responsability of the user.

Definition at line 844 of file mcd_io.c.

References mcd_check_error_code.

Referenced by mcd_table_list_save().

3.20.2.13 mcd_io_table_load_from_file()

```
cpl_table* mcd_io_table_load_from_file (
            const char * file,
            const cpl_type det_id )
```

Load the image for one specific detector from a fits file.

Parameters

file	Fits file to be loaded
det_id	Detector ID in the DFS, valid from 1 to 9.

Returns

1 cpl_table or NULL in case of error

Definition at line 806 of file mcd_io.c.

References mcd_io_get_ext_idx().

3.21 FITS header protected access

Functions

- const char * mcd_pfits_get_arcfile (const cpl_propertylist *plist)
  find out the arcfile
- double mcd_pfits_get_dit (const cpl_propertylist *plist)
  find out the DIT value

3.21.1 Detailed Description

3.21.2 Function Documentation

3.21.2.1 mcd_pfits_get_arcfile()

```
const char* mcd_pfits_get_arcfile (
            const cpl_propertylist * plist )
```

find out the arcfile

Parameters

plist	property list to read from

Returns

pointer to statically allocated character string

Definition at line 52 of file mcd_pfits.c.

3.21.2.2 mcd_pfits_get_dit()

```
double mcd_pfits_get_dit (
            const cpl_propertylist * plist )
```

find out the DIT value

Parameters

plist	property list to read from

Returns

the requested value

Definition at line 64 of file mcd_pfits.c.

3.22 PSFR Domain Utilities

Functions

- mcd_psfr_domain mcd_psfr_domain_invalid (void)
- mcd_psfr_domain mcd_psfr_domain_centered (cpl_size nx, double pixel_size)
- mcd_psfr_domain mcd_psfr_domain_frequency (mcd_psfr_domain physical)
- cpl_error_code mcd_psfr_domain_fix (mcd_psfr_domain ∗domain)
- bool mcd_psfr_domain_is_valid (mcd_psfr_domain domain)
- cpl_error_code mcd_psfr_domain_center (mcd_psfr_domain ∗domain)
- cpl_size mcd_psfr_domain_n_pixels (mcd_psfr_domain domain)
- bool mcd_psfr_domain_equal (mcd_psfr_domain a, mcd_psfr_domain b)
- mcd_psfr_domain mcd_psfr_domain_change_n_points (mcd_psfr_domain domain, cpl_size n_points)
- double mcd_psfr_domain_first_vertex (mcd_psfr_domain domain)
- double mcd_psfr_domain_last_vertex (mcd_psfr_domain domain)
- double mcd_psfr_domain_index_to_coordinate (mcd_psfr_domain domain, cpl_size i)

3.22.1 Detailed Description

Author

D. Jodlbauer

Domain Utilities for PSFR functions

3.22.2 Function Documentation

3.22.2.1 mcd_psfr_domain_center()

```
cpl_error_code mcd_psfr_domain_center (
            mcd_psfr_domain * domain )
```

Adjust x0 such that the domain is cenetered.

Definition at line 139 of file mcd_psfr_domain.c.

References mcd_psfr_domain_centered(), and mcd_psfr_domain_fix().

3.22.2.2 mcd_psfr_domain_centered()

```
mcd_psfr_domain mcd_psfr_domain_centered (
            cpl_size nx,
            double pixel_size )
```

Return a Q1 description with given number of points and resolution. The number of "pixels" is nx - 1.

Definition at line 63 of file mcd_psfr_domain.c.

References mcd_psfr_domain_n_pixels().

Referenced by mcd_psfr_domain_center(), mcd_psfr_domain_frequency(), mcd_psfr_otf_multi_unify(), mcd_psfr↩_scao_orthogonal(), and mcd_psfr_utility_gridfunction_from_img_px().

3.22.2.3 mcd_psfr_domain_change_n_points()

```
mcd_psfr_domain mcd_psfr_domain_change_n_points (
            mcd_psfr_domain domain,
            cpl_size n_points )
```

Change the amount of points/pixels used to represent the domain. This may slightly adjust x0 to make sure the origin is exactly represented.

Definition at line 181 of file mcd_psfr_domain.c.

Referenced by mcd_psfr_gridfunction_wrap().

3.22.2.4 mcd_psfr_domain_equal()

```
bool mcd_psfr_domain_equal (
            mcd_psfr_domain a,
            mcd_psfr_domain b )
```

Compare two domains

Definition at line 160 of file mcd_psfr_domain.c.

Referenced by mcd_psfr_gridfunctions_set().

3.22.2.5 mcd_psfr_domain_first_vertex()

```
double mcd_psfr_domain_first_vertex (
            mcd_psfr_domain domain )
```

Return the physical position of the first vertex in the grid, corresponding to the first coefficient stored for the function. Assumes the domain is square.

Definition at line 214 of file mcd_psfr_domain.c.

3.22.2.6 mcd_psfr_domain_fix()

```
cpl_error_code mcd_psfr_domain_fix (
            mcd_psfr_domain * domain )
```

Return a domain with fixed /valid) attributes, i.e., computes missing parts of n_points, pixel_size, length.

Returns

New domain with valid attributes.

Definition at line 90 of file mcd_psfr_domain.c.

References mcd_psfr_domain_is_valid(), and mcd_psfr_domain_n_pixels().

Referenced by mcd_psfr_domain_center(), and mcd_psfr_gridfunction_new().

3.22.2.7 mcd_psfr_domain_frequency()

```
mcd_psfr_domain mcd_psfr_domain_frequency (
            mcd_psfr_domain physical )
```

Compute the domain after fourier transform.

Definition at line 77 of file mcd_psfr_domain.c.

References mcd_psfr_domain_centered().

3.22.2.8 mcd_psfr_domain_index_to_coordinate()

```
double mcd_psfr_domain_index_to_coordinate (
            mcd_psfr_domain domain,
            cpl_size i )
```

Convert the given vertex index to its coordinate.

Parameters

d	Domain
i	Vertex index for one coordinate direction starting at 0

Returns

Coordinate x for one direction such that f(x) = value[i] Note: assumes square domains

Definition at line 242 of file mcd_psfr_domain.c.

Referenced by mcd_psfr_gridfunctions_save(), and mcd_psfr_scao_orthogonal_atmosphere_psd().

3.22.2.9 mcd_psfr_domain_invalid()

```
mcd_psfr_domain mcd_psfr_domain_invalid (
            void  )
```

Return an invalid domain.

Definition at line 53 of file mcd_psfr_domain.c.

Referenced by mcd_psfr_gridfunction_domain(), mcd_psfr_gridfunctions_domain(), mcd_psfr_gridfunctions_new(), mcd_psfr_influence_functions_domain(), and mcd_psfr_influence_functions_new().

3.22.2.10 mcd_psfr_domain_is_valid()

```
bool mcd_psfr_domain_is_valid (
            mcd_psfr_domain domain )
```

Check if the given domain is valid.

Definition at line 125 of file mcd_psfr_domain.c.

References mcd_psfr_domain_n_pixels().

Referenced by mcd_psfr_domain_fix().

3.22.2.11 mcd_psfr_domain_last_vertex()

```
double mcd_psfr_domain_last_vertex (
            mcd_psfr_domain domain )
```

Return the physical position of the last vertex in the grid, corresponding to the last coefficient stored for the function. Assumes the domain is square.

Definition at line 227 of file mcd_psfr_domain.c.

3.22.2.12 mcd_psfr_domain_n_pixels()

```
cpl_size mcd_psfr_domain_n_pixels (
            mcd_psfr_domain domain )
```

Return the size in pixels per coordinate direction. If this domain represents a sampled function, its contains n_↩ pixels+1 data points, otherwise the number of data points is n_pixels.

Definition at line 151 of file mcd_psfr_domain.c.

Referenced by mcd_psfr_domain_centered(), mcd_psfr_domain_fix(), and mcd_psfr_domain_is_valid().

3.23 PSFR Grid Function Utilities

Functions

- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_gridfunction_new (mcd_psfr_domain domain)
- void mcd_psfr_gridfunction_delete (mcd_psfr_gridfunction *a)
- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_gridfunction_duplicate (const mcd_psfr_gridfunction *a)
- bool mcd_psfr_gridfunction_compatible (const mcd_psfr_gridfunction *a, const mcd_psfr_gridfunction *b)
- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_gridfunction_wrap (cpl_image *data, mcd_psfr_domain domain)
- cpl_image * mcd_psfr_gridfunction_coeffs (mcd_psfr_gridfunction *a)
- mcd_psfr_domain mcd_psfr_gridfunction_domain (const mcd_psfr_gridfunction *a)
- double mcd_psfr_gridfunction_pixelsize (const mcd_psfr_gridfunction *a)
- double mcd_psfr_gridfunction_length (const mcd_psfr_gridfunction *a)
- double mcd_psfr_gridfunction_evaluate (const mcd_psfr_gridfunction *a, double x, double y)
- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_gridfunction_evaluate_domain (const mcd_psfr_gridfunction *a, mcd_psfr_domain domain)
- double mcd_psfr_gridfunction_at_vertex (const mcd_psfr_gridfunction *f, cpl_size i, cpl_size j)
- cpl_error_code mcd_psfr_gridfunction_set_vertex (mcd_psfr_gridfunction *f, cpl_size i, cpl_size j, double value)
- bool mcd_psfr_gridfunction_check_minmax_at_origin (const mcd_psfr_gridfunction *a)
- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_gridfunction_resize (const mcd_psfr_gridfunction *a, cpl_size nx, cpl_size ny)
- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_gridfunction_resample (const mcd_psfr_gridfunction *a, cpl_size nx, cpl_size ny)
- double mcd_psfr_gridfunction_integrate (const mcd_psfr_gridfunction *a)
- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_gridfunction_pad_create (const mcd_psfr_gridfunction *a, cpl_size fftsize)
- cpl_error_code mcd_psfr_gridfunction_fft (mcd_psfr_gridfunction *real, mcd_psfr_gridfunction *imag, unsigned int flags)
- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_gridfunction_autocorrelation (const mcd_psfr_gridfunction *a)
- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_gridfunction_structure_function (const mcd_psfr_gridfunction *a, const mcd_psfr_gridfunction *pupil)
- cpl_size mcd_psfr_gridfunction_get_size_x (const mcd_psfr_gridfunction *a)
- cpl_size mcd_psfr_gridfunction_get_size_y (const mcd_psfr_gridfunction *a)
- cpl_error_code mcd_psfr_gridfunction_fill_distance (mcd_psfr_gridfunction *f)
- cpl_error_code mcd_psfr_gridfunction_save (const mcd_psfr_gridfunction *f, const char *filename)

3.23.1 Detailed Description

Author

D. Jodlbauer, A. La Camera, M. Ravenna

Note

This is different from PSFR Grid Functions Utilities

Grid Function Utilities for PSFR functions

3.23.2 Function Documentation

3.23.2.1 mcd_psfr_gridfunction_at_vertex()

```
double mcd_psfr_gridfunction_at_vertex (
            const mcd_psfr_gridfunction * f,
            cpl_size i,
            cpl_size j )
```

Evaluate the gridfunction exactly on the vertex (i,j) (0-based)

Parameters

f	Gridfunction to evaluate
i	x-index of vertex
j	y-index of vertex

Returns

Function value f(x_i, y_j)

Definition at line 464 of file mcd_psfr_grid_function.c.

3.23.2.2 mcd_psfr_gridfunction_autocorrelation()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_gridfunction_autocorrelation (
            const mcd_psfr_gridfunction * a )
```

Compute the autocorrelation.

Definition at line 801 of file mcd_psfr_grid_function.c.

Referenced by mcd_psfr_otf_telescope().

3.23.2.3 mcd_psfr_gridfunction_check_minmax_at_origin()

```
bool mcd_psfr_gridfunction_check_minmax_at_origin (
            const mcd_psfr_gridfunction * a )
```

Check if the minimum or maximum is exactly at the origin.

Definition at line 495 of file mcd_psfr_grid_function.c.

Referenced by mcd_psfr_otf_multi_unify(), and mcd_psfr_otf_telescope().

3.23.2.4 mcd_psfr_gridfunction_coeffs()

```
cpl_image* mcd_psfr_gridfunction_coeffs (
            mcd_psfr_gridfunction * a )
```

Return the coefficients of the gridfunction.

Definition at line 234 of file mcd_psfr_grid_function.c.

3.23.2.5 mcd_psfr_gridfunction_compatible()

```
bool mcd_psfr_gridfunction_compatible (
            const mcd_psfr_gridfunction * a,
            const mcd_psfr_gridfunction * b )
```

Check if two functions are valid and live on the same domain.

Parameters

a	First gridfunction
b	Second gridfunction

Definition at line 160 of file mcd_psfr_grid_function.c.

3.23.2.6 mcd_psfr_gridfunction_delete()

```
void mcd_psfr_gridfunction_delete (
            mcd_psfr_gridfunction * a )
```

Delete the given gridfunction.

Parameters

a	gridfunction to delete

Definition at line 132 of file mcd_psfr_grid_function.c.

Referenced by mcd_psfr_otf_compute(), mcd_psfr_otf_delete(), mcd_psfr_otf_telescope(), and mcd_psfr_↩ parameters_delete().

3.23.2.7 mcd_psfr_gridfunction_domain()

```
mcd_psfr_domain mcd_psfr_gridfunction_domain (
            const mcd_psfr_gridfunction * a )
```

Return the domain description.

Definition at line 243 of file mcd_psfr_grid_function.c.

References mcd_psfr_domain_invalid().

3.23.2.8 mcd_psfr_gridfunction_duplicate()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_gridfunction_duplicate (
            const mcd_psfr_gridfunction * a )
```

Duplicate a gridfunction and return the copy.

Parameters

a	gridfunction to duplicate

Definition at line 147 of file mcd_psfr_grid_function.c.

Referenced by mcd_psfr_otf_inv(), mcd_psfr_otf_nonao_vibration(), and mcd_psfr_otf_rotate().

3.23.2.9 mcd_psfr_gridfunction_evaluate()

```
double mcd_psfr_gridfunction_evaluate (
            const mcd_psfr_gridfunction * a,
            double x,
            double y )
```

Evaluate the grid function at point (x,y).

Definition at line 305 of file mcd_psfr_grid_function.c.

3.23.2.10 mcd_psfr_gridfunction_evaluate_domain()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_gridfunction_evaluate_domain (
            const mcd_psfr_gridfunction * a,
            mcd_psfr_domain domain )
```

Evaluate the function on a new domain.

Parameters

a	Function to evaluate
domain	New domain

Returns

New function on new domain

Definition at line 408 of file mcd_psfr_grid_function.c.

Referenced by mcd_psfr_otf_multi_unify().

3.23.2.11 mcd_psfr_gridfunction_fft()

```
cpl_error_code mcd_psfr_gridfunction_fft (
            mcd_psfr_gridfunction * real,
            mcd_psfr_gridfunction * imag,
            unsigned int flags )
```

Compute the Fourier-transform. This function takes care of the fftshift. Note: result->pixel_size := 1 / a->length.

Definition at line 724 of file mcd_psfr_grid_function.c.

Referenced by mcd_psfr_otf_inv().

3.23.2.12 mcd_psfr_gridfunction_fill_distance()

```
cpl_error_code mcd_psfr_gridfunction_fill_distance (
            mcd_psfr_gridfunction * f )
```

Fill the given gridfunction with value f(x,y) = ||x-y||.

Parameters

f	Gridfunction with initialized domain to fill.

Definition at line 909 of file mcd_psfr_grid_function.c.

Referenced by mcd_psfr_scao_orthogonal().

3.23.2.13 mcd_psfr_gridfunction_get_size_x()

```
cpl_size mcd_psfr_gridfunction_get_size_x (
            const mcd_psfr_gridfunction * a )
```

Return the size of the discrete representation of the function.

Definition at line 886 of file mcd_psfr_grid_function.c.

3.23.2.14 mcd_psfr_gridfunction_get_size_y()

```
cpl_size mcd_psfr_gridfunction_get_size_y (
            const mcd_psfr_gridfunction * a )
```

Return the size of the discrete representation of the function.

Definition at line 897 of file mcd_psfr_grid_function.c.

3.23.2.15 mcd_psfr_gridfunction_integrate()

```
double mcd_psfr_gridfunction_integrate (
            const mcd_psfr_gridfunction * a )
```

Compute the integral over the whole domain.

Definition at line 659 of file mcd_psfr_grid_function.c.

3.23.2.16 mcd_psfr_gridfunction_length()

```
double mcd_psfr_gridfunction_length (
            const mcd_psfr_gridfunction * a )
```

Return the length of the domain.

Returns

length := pixelsize ∗ size(data)

Definition at line 263 of file mcd_psfr_grid_function.c.

3.23.2.17 mcd_psfr_gridfunction_new()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_gridfunction_new (
            mcd_psfr_domain domain )
```

Create a new gridfunction living on the given domain.

Parameters

domain	Domain of the function

Definition at line 106 of file mcd_psfr_grid_function.c.

References mcd_psfr_domain_fix().

Referenced by mcd_psfr_otf_multi(), mcd_psfr_scao_orthogonal(), and mcd_psfr_scao_orthogonal_atmosphere_psd().

3.23.2.18 mcd_psfr_gridfunction_pad_create()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_gridfunction_pad_create (
            const mcd_psfr_gridfunction * a,
            cpl_size fftsize )
```

Zero-pad the domain.

Parameters

a	Original function
fftsize	New size

Returns

Center-padded gridfunction

Definition at line 689 of file mcd_psfr_grid_function.c.

Referenced by mcd_psfr_otf_inv().

3.23.2.19 mcd_psfr_gridfunction_pixelsize()

```
double mcd_psfr_gridfunction_pixelsize (
            const mcd_psfr_gridfunction * a )
```

Return the resolution of the domain.

Returns

resolution := length/size

Definition at line 253 of file mcd_psfr_grid_function.c.

3.23.2.20 mcd_psfr_gridfunction_resample()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_gridfunction_resample (
            const mcd_psfr_gridfunction * a,
            cpl_size nx,
            cpl_size ny )
```

Evaluate the function on the same domain using

Parameters

nx	x
ny	points.
a	Gridfunction to resample
nx	New size in x direction
ny	New size in y direction

Returns

Newly created gridfunction Note: keeps the min/max at zero!

Definition at line 624 of file mcd_psfr_grid_function.c.

3.23.2.21 mcd_psfr_gridfunction_resize()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_gridfunction_resize (
            const mcd_psfr_gridfunction * a,
            cpl_size nx,
            cpl_size ny )
```

Change the data size of the gridfunction. Note: Effectively changes the pixel size. Note: does not keep the min/max at zero!

Definition at line 582 of file mcd_psfr_grid_function.c.

3.23.2.22 mcd_psfr_gridfunction_save()

```
cpl_error_code mcd_psfr_gridfunction_save (
            const mcd_psfr_gridfunction * f,
            const char * filename )
```

Save gridfunction with additional information about domain.

Definition at line 941 of file mcd_psfr_grid_function.c.

3.23.2.23 mcd_psfr_gridfunction_set_vertex()

```
cpl_error_code mcd_psfr_gridfunction_set_vertex (
            mcd_psfr_gridfunction * f,
            cpl_size i,
            cpl_size j,
            double value )
```

Evaluate the gridfunction exactly on the vertex (i,j) (0-based)

Parameters

f	Gridfunction to modify
i	x-index of vertex
j	y-index of vertex
value	Function value f(x_i, y_j)

Definition at line 482 of file mcd_psfr_grid_function.c.

Referenced by mcd_psfr_scao_orthogonal_atmosphere_psd().

3.23.2.24 mcd_psfr_gridfunction_structure_function()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_gridfunction_structure_function (
            const mcd_psfr_gridfunction * a,
            const mcd_psfr_gridfunction * pupil )
```

Compute the structure function of a gridfunction.

Parameters

a	Input gridfunction
pupil	Input pupil

Definition at line 835 of file mcd_psfr_grid_function.c.

3.23.2.25 mcd_psfr_gridfunction_wrap()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_gridfunction_wrap (
            cpl_image * data,
            mcd_psfr_domain domain )
```

Create new gridfunction from image and domain.

Parameters

data	Coefficients as image for the gridfunction
domain	Domain for the new gridfunction

Returns

New gridfunction, needs to be freed with mcd_psfr_gridfunction_unwrap.

Definition at line 181 of file mcd_psfr_grid_function.c.

References mcd_psfr_domain_change_n_points().

Referenced by mcd_psfr_scao_cal_WFS_noise(), mcd_psfr_scao_cal_WFS_slopes(), mcd_psfr_utility_↩ gridfunction_from_img_length(), and mcd_psfr_utility_gridfunction_from_img_px().

3.24 PSFR Grid Functions Utilities

Functions

- CPL_ATTR_ALLOC mcd_psfr_gridfunctions ∗ mcd_psfr_gridfunctions_new (void)
- void mcd_psfr_gridfunctions_delete (mcd_psfr_gridfunctions ∗list)
- cpl_imagelist ∗ mcd_psfr_gridfunctions_unwrap (mcd_psfr_gridfunctions ∗list)
- cpl_error_code mcd_psfr_gridfunctions_set (mcd_psfr_gridfunctions ∗list, mcd_psfr_gridfunction ∗element, cpl_size position)
- mcd_psfr_domain mcd_psfr_gridfunctions_domain (const mcd_psfr_gridfunctions ∗list)
- double mcd_psfr_gridfunctions_pixelsize (const mcd_psfr_gridfunctions ∗list)
- double mcd_psfr_gridfunctions_length (const mcd_psfr_gridfunctions ∗list)
- cpl_size mcd_psfr_gridfunctions_get_size (const mcd_psfr_gridfunctions ∗list)
- cpl_size mcd_psfr_gridfunctions_get_size_x (const mcd_psfr_gridfunctions ∗list)
- cpl_size mcd_psfr_gridfunctions_get_size_y (const mcd_psfr_gridfunctions ∗list)
- cpl_imagelist ∗ mcd_psfr_gridfunctions_data (mcd_psfr_gridfunctions ∗list)
- const cpl_image ∗ mcd_psfr_gridfunctions_get_const_data (const mcd_psfr_gridfunctions ∗list, cpl_size position)
- CPL_ATTR_ALLOC mcd_psfr_gridfunctions ∗ mcd_psfr_gridfunctions_resize (const mcd_psfr_gridfunctions ∗list, cpl_size nx, cpl_size ny)
- cpl_error_code mcd_psfr_gridfunctions_save (mcd_psfr_gridfunctions ∗list, const char ∗filename, cpl_↩ propertylist ∗plist)

3.24.1 Detailed Description

Author

D. Jodlbauer, A. La Camera, M. Ravenna, R. Wagner

Note

This is different from PSFR Grid Function Utilities

Grid Frunctions Utilities for PSFR functions

3.24.2 Function Documentation

3.24.2.1 mcd_psfr_gridfunctions_data()

```
cpl_imagelist* mcd_psfr_gridfunctions_data (
            mcd_psfr_gridfunctions * list )
```

Return the raw imagelist data

Parameters

list	Gridfunction list

Returns

Pointer to raw data

Definition at line 252 of file mcd_psfr_gridfunctions.c.

Referenced by mcd_psfr_scao_parallel().

3.24.2.2 mcd_psfr_gridfunctions_delete()

```
void mcd_psfr_gridfunctions_delete (
            mcd_psfr_gridfunctions * list )
```

Delete a gridfunction list and its contained objects.

Parameters

list	List to delete

Definition at line 75 of file mcd_psfr_gridfunctions.c.

Referenced by mcd_psfr_parameters_delete().

3.24.2.3 mcd_psfr_gridfunctions_domain()

```
mcd_psfr_domain mcd_psfr_gridfunctions_domain (
            const mcd_psfr_gridfunctions * list )
```

Return the domain description.

Definition at line 186 of file mcd_psfr_gridfunctions.c.

References mcd_psfr_domain_invalid().

3.24.2.4 mcd_psfr_gridfunctions_get_const_data()

```
const cpl_image* mcd_psfr_gridfunctions_get_const_data (
            const mcd_psfr_gridfunctions * list,
            cpl_size position )
```

Return the raw imagelist data

Parameters

list	Gridfunction list

Returns

Pointer to raw data

Definition at line 264 of file mcd_psfr_gridfunctions.c.

Referenced by mcd_psfr_scao_anisoplanatic_phase().

3.24.2.5 mcd_psfr_gridfunctions_get_size()

```
cpl_size mcd_psfr_gridfunctions_get_size (
            const mcd_psfr_gridfunctions * list )
```

Obtain the number of stored functions

Parameters

list	Gridfunction list

Returns

Number of stored functions

Definition at line 219 of file mcd_psfr_gridfunctions.c.

3.24.2.6 mcd_psfr_gridfunctions_get_size_x()

```
cpl_size mcd_psfr_gridfunctions_get_size_x (
            const mcd_psfr_gridfunctions * list )
```

Obtain the size of the stored functions in y-direction

Parameters

list	Gridfunction list

Returns

Number of pixels

Definition at line 230 of file mcd_psfr_gridfunctions.c.

3.24.2.7 mcd_psfr_gridfunctions_get_size_y()

```
cpl_size mcd_psfr_gridfunctions_get_size_y (
            const mcd_psfr_gridfunctions * list )
```

Obtain the size of the stored functions in x-direction

Parameters

list	Gridfunction list

Returns

Number of pixels

Definition at line 241 of file mcd_psfr_gridfunctions.c.

3.24.2.8 mcd_psfr_gridfunctions_length()

```
double mcd_psfr_gridfunctions_length (
            const mcd_psfr_gridfunctions * list )
```

Obtain the (physical) size of each function

Parameters

list	Gridfunction list

Returns

Domain length

Definition at line 208 of file mcd_psfr_gridfunctions.c.

3.24.2.9 mcd_psfr_gridfunctions_new()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunctions* mcd_psfr_gridfunctions_new (
            void  )
```

Create a new gridfunction list. Pixel size will be set once the first gridfunction is added.

Returns

New object

Definition at line 62 of file mcd_psfr_gridfunctions.c.

References mcd_psfr_domain_invalid().

Referenced by mcd_psfr_scao_cal_WFS_slopes().

3.24.2.10 mcd_psfr_gridfunctions_pixelsize()

```
double mcd_psfr_gridfunctions_pixelsize (
            const mcd_psfr_gridfunctions * list )
```

Obtain the (physical) pixel size of each function

Parameters

list	Gridfunction list

Returns

Pixel size

Definition at line 197 of file mcd_psfr_gridfunctions.c.

3.24.2.11 mcd_psfr_gridfunctions_resize()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunctions* mcd_psfr_gridfunctions_resize (
            const mcd_psfr_gridfunctions * list,
            cpl_size nx,
            cpl_size ny )
```

Resize the gridfunctions contained in this list to a new size.

Parameters

list	List of gridfunctions
nx	New size in x-direction
ny	New size in x-direction

Returns

New list with modified sizes

Definition at line 278 of file mcd_psfr_gridfunctions.c.

3.24.2.12 mcd_psfr_gridfunctions_save()

```
cpl_error_code mcd_psfr_gridfunctions_save (
            mcd_psfr_gridfunctions * list,
            const char * filename,
            cpl_propertylist * plist )
```

Save the gridfunctions list to a cpl_imagelist FITS file.

Parameters

list	List of gridfunctions to be saved
filename	Filename of the output file
plist	Header of the output, other information is added here

Returns

cpl_error_code

Definition at line 295 of file mcd_psfr_gridfunctions.c.

References mcd_psfr_domain_index_to_coordinate().

3.24.2.13 mcd_psfr_gridfunctions_set()

```
cpl_error_code mcd_psfr_gridfunctions_set (
            mcd_psfr_gridfunctions * list,
            mcd_psfr_gridfunction * element,
            cpl_size position )
```

Add a new function to this list. On first entry, sets the pixel size, otherwise checks that all functions have the same physical domain.

Parameters

list	Gridfunction list
element	Function object to add
position	Function index

Definition at line 124 of file mcd_psfr_gridfunctions.c.

References mcd_psfr_domain_equal().

3.24.2.14 mcd_psfr_gridfunctions_unwrap()

```
cpl_imagelist* mcd_psfr_gridfunctions_unwrap (
            mcd_psfr_gridfunctions * list )
```

Destroy the list but keep the underlying imagelist data.

Parameters

list	Gridfunction list

Returns

Raw imagelist data

Definition at line 107 of file mcd_psfr_gridfunctions.c.

3.25 PSFR OTF Utilities

Functions

- mcd_psfr_otf * mcd_psfr_otf_new (void)
  Create a new structure of OTF parts.
- void mcd_psfr_otf_delete (mcd_psfr_otf *otf)
  Free memory of the OTFs.
- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_otf_compute (mcd_psfr_parameters *params)
- cpl_error_code mcd_psfr_otf_save (const mcd_psfr_otf *otfs, const char *prefix, const char *postfix)

3.25.1 Detailed Description

Author

D. Jodlbauer, A. La Camera, M. Ravenna, R. Wagner

OTF Utilities for PSFR functions

3.25.2 Function Documentation

3.25.2.1 mcd_psfr_otf_compute()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_otf_compute (
            mcd_psfr_parameters * params )
```

Compute the full OTF from the PSF parameters.

Parameters

params	Parameters (TODO const)

Returns

OTF image of size fft_size x fft_size

Definition at line 85 of file mcd_psfr_otf.c.

References mcd_psfr_gridfunction_delete(), mcd_psfr_otf_delete(), mcd_psfr_otf_multi(), mcd_psfr_otf_new(), mcd_psfr_otf_save(), mcd_psfr_otf_telescope(), mcd_psfr_scao_cal_WFS_aliasing(), mcd_psfr_scao_cal_WFS↩ _noise(), mcd_psfr_scao_cal_WFS_slopes(), mcd_psfr_scao_orthogonal(), and mcd_psfr_scao_parallel().

3.25.2.2 mcd_psfr_otf_delete()

```
void mcd_psfr_otf_delete (
            mcd_psfr_otf * otf )
```

Free memory of the OTFs.

Parameters

otf	The OTF to delete

Definition at line 63 of file mcd_psfr_otf.c.

References mcd_psfr_gridfunction_delete().

Referenced by mcd_psfr_otf_compute(), and mcd_psfr_otf_multi_unify().

3.25.2.3 mcd_psfr_otf_new()

```
mcd_psfr_otf* mcd_psfr_otf_new (
            void  )
```

Create a new structure of OTF parts.

Returns

Newly created OTF object

Definition at line 53 of file mcd_psfr_otf.c.

Referenced by mcd_psfr_otf_compute(), and mcd_psfr_otf_multi_unify().

3.25.2.4 mcd_psfr_otf_save()

```
cpl_error_code mcd_psfr_otf_save (
            const mcd_psfr_otf * otfs,
            const char * prefix,
            const char * postfix )
```

Save the contained OTFs. Filename is generated as <prefix><name><postfix>.fits, i.e.,

Parameters

prefix	can be used to specify a folder, and
postfix	can be used to specify a version or similar.

Definition at line 179 of file mcd_psfr_otf.c.

Referenced by mcd_psfr_otf_compute().

3.26 PSFR Parameters Utilities

Functions

- mcd_psfr_parameters * mcd_psfr_parameters_new (void)
  Create a new parameter structure for PSFR.
- cpl_error_code mcd_psfr_parameters_anisoplanatic (mcd_psfr_anisoplanatic_parameters *params)
- mcd_psfr_parameters * mcd_psfr_parameters_new_compass (const char *folder)
- cpl_error_code mcd_psfr_parameters_internal (mcd_psfr_parameters *params)
- cpl_error_code mcd_psfr_parameters_from_frameset (mcd_psfr_parameters *prm, const cpl_frameset *frameset, const cpl_parameterlist *params)
- void mcd_psfr_parameters_delete (mcd_psfr_parameters *params)
  Free memory of the PSF parameters.

3.26.1 Detailed Description

Author

D. Jodlbauer, A. La Camera, M. Ravenna

Parameters Utilities for PSFR functions

3.26.2 Function Documentation

3.26.2.1 mcd_psfr_parameters_anisoplanatic()

```
cpl_error_code mcd_psfr_parameters_anisoplanatic (
            mcd_psfr_anisoplanatic_parameters * params )
```

Default parameters for the anisoplanatic phase computation.

Definition at line 98 of file mcd_psfr_parameters.c.

Referenced by mcd_psfr_parameters_new().

3.26.2.2 mcd_psfr_parameters_delete()

```
void mcd_psfr_parameters_delete (
            mcd_psfr_parameters * params )
```

Free memory of the PSF parameters.

Parameters

params	The PSF to delete

Definition at line 409 of file mcd_psfr_parameters.c.

References mcd_psfr_gridfunction_delete(), mcd_psfr_gridfunctions_delete(), and mcd_psfr_influence_functions↩_delete().

3.26.2.3 mcd_psfr_parameters_from_frameset()

```
cpl_error_code mcd_psfr_parameters_from_frameset (
            mcd_psfr_parameters * prm,
            const cpl_frameset * frameset,
            const cpl_parameterlist * params )
```

Fill PSF parameters from a given frameset.

Definition at line 270 of file mcd_psfr_parameters.c.

Referenced by mcd_psfr_parameters_new_compass().

3.26.2.4 mcd_psfr_parameters_internal()

```
cpl_error_code mcd_psfr_parameters_internal (
            mcd_psfr_parameters * params )
```

Compute some internal variables

Definition at line 154 of file mcd_psfr_parameters.c.

References mcd_psfr_utility_next_power_of_two().

3.26.2.5 mcd_psfr_parameters_new()

```
mcd_psfr_parameters* mcd_psfr_parameters_new (
            void  )
```

Create a new parameter structure for PSFR.

Returns

Newly created PSFR object

Definition at line 50 of file mcd_psfr_parameters.c.

References mcd_psfr_parameters_anisoplanatic().

Referenced by mcd_psfr_parameters_new_compass(), mcd_psfr_scao_otf_par(), mcd_psfr_scao_otf_par_fill_↩parameterlist(), mcd_psfr_scao_otf_perp(), and mcd_psfr_scao_otf_perp_fill_parameterlist().

3.26.2.6 mcd_psfr_parameters_new_compass()

```
mcd_psfr_parameters* mcd_psfr_parameters_new_compass (
            const char * folder )
```

Create parameter object from COMPASS data

Definition at line 142 of file mcd_psfr_parameters.c.

References mcd_psfr_parameters_from_frameset(), mcd_psfr_parameters_new(), and mcd_psfr_utility_frameset_from_compass_data().

3.27 PSFR Projection Utilities

Author

D. Jodlbauer

3.28 PSFR Miscellaneous Utilities

Macros

- #define MCD_PSFR_ZERO_TOLERANCE 1.0e-12

  Tolerance at which a value is considered zero. Used for compatibility with Matlab to convert doubles to logicals.
- #define MCD_PSFR_ARCSEC_TO_RAD (CPL_MATH_RAD_DEG / 3600.0)

  Conversion factor from arc-seconds to radians.
- #define MCD_PSFR_M_TO_UM 1.0e6

  Conversion factor from meter to µm (and vice-versa)
- #define MCD_PSFR_SQR(x) ((x) ∗ (x))

  Square the argument.
- #define MCD_PSFR_TIME_SECTION(name, code)

Functions

- mcd_psfr_box mcd_psfr_box_unify (mcd_psfr_box a, mcd_psfr_box b)
- mcd_psfr_box mcd_psfr_box_shift (mcd_psfr_box a, int shift_x, int shift_y)
- bool mcd_psfr_box_valid (mcd_psfr_box a)
- cpl_size mcd_psfr_box_get_size_x (mcd_psfr_box a)
- cpl_size mcd_psfr_box_get_size_y (mcd_psfr_box a)
- mcd_psfr_box mcd_psfr_box_image_bounding_box (const cpl_image ∗image)
- CPL_ATTR_ALLOC cpl_image ∗ mcd_psfr_box_image_crop (const cpl_image ∗img)
- mcd_psfr_box mcd_psfr_box_common_area (int a, int b, int n)
- CPL_ATTR_ALLOC cpl_image ∗ mcd_psfr_covariance_average (const cpl_matrix ∗covariance_matrix, cpl_size n)
- cpl_error_code mcd_psfr_cpl_image_transpose (cpl_image ∗in)
- cpl_error_code mcd_psfr_cpl_image_fill (cpl_image ∗image, double value)

  Fill the whole image with a value.
- double mcd_psfr_cpl_image_get_or_default (const cpl_image ∗a, cpl_size i, cpl_size j, double def)
- cpl_error_code mcd_psfr_cpl_image_add_scale (cpl_image ∗a, const cpl_image ∗b, double factor_b)

Compute a = a + factor_b + b.

- cpl_imagelist * mcd_psfr_cpl_imagelist_load_ext (const char *filename, cpl_type im_type, int xtnum, int ind-explane)

  Load an imagelist from a FITS file.
- CPL_ATTR_ALLOC cpl_image * mcd_psfr_cpl_imagelist_sum (const cpl_imagelist *images, const cpl_↩ vector *weights, cpl_size N)
- CPL_ATTR_ALLOC cpl_imagelist * mcd_psfr_cpl_imagelist_sample (cpl_imagelist *images, cpl_size start, cpl_size step)
- double mcd_psfr_cpl_wall_time (void)
- double mcd_psfr_fits_get_double (const char *filename, const char *variable_name)
- static CPL_ATTR_ALLOC mcd_psfr_influence_functions * mcd_psfr_influence_functions_new (void)
- static mcd_psfr_box mcd_psfr_image_bounding_box_mode (const cpl_image *influ, enum mcd_psfr_↩ embedding mode)
- static mcd_psfr_box mcd_psfr_influence_functions_bounding_box (const mcd_psfr_influence_functions *ifs, enum mcd_psfr_embedding mode)
- void mcd_psfr_influence_functions_delete (mcd_psfr_influence_functions *ifs)
- CPL_ATTR_ALLOC mcd_psfr_influence_functions * mcd_psfr_influence_functions_load (const char *fits↩ _file)
- mcd_psfr_domain mcd_psfr_influence_functions_domain (const mcd_psfr_influence_functions *ifs)
- cpl_size mcd_psfr_influence_functions_size (const mcd_psfr_influence_functions *ifs)
- cpl_size mcd_psfr_influence_functions_size_x (const mcd_psfr_influence_functions *ifs)
- cpl_size mcd_psfr_influence_functions_size_y (const mcd_psfr_influence_functions *ifs)
- const cpl_imagelist * mcd_psfr_influence_functions_get (const mcd_psfr_influence_functions *ifs)
- const cpl_array * mcd_psfr_influence_functions_get_x (const mcd_psfr_influence_functions *ifs)
- const cpl_array * mcd_psfr_influence_functions_get_y (const mcd_psfr_influence_functions *ifs)
- CPL_ATTR_ALLOC mcd_psfr_influence_functions * mcd_psfr_influence_functions_resize_all (const mcd↩ _psfr_influence_functions *ifs, cpl_size nx, cpl_size ny)
- CPL_ATTR_ALLOC mcd_psfr_influence_functions * mcd_psfr_influence_functions_resize (const mcd_↩ psfr_influence_functions *ifs, cpl_size nx, cpl_size ny)
- CPL_ATTR_ALLOC mcd_psfr_influence_functions * mcd_psfr_influence_functions_embed (const mcd_↩ psfr_influence_functions *ifs, enum mcd_psfr_embedding mode)
- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_influence_functions_sum (const mcd_psfr_↩ influence_functions *ifs, const cpl_vector *weights, cpl_size N)
- CPL_ATTR_ALLOC mcd_psfr_gridfunction * mcd_psfr_influence_functions_support (const mcd_psfr_↩ influence_functions *ifs)
- bool mcd_psfr_influence_functions_is_embedded (const mcd_psfr_influence_functions *ifs)
- static cpl_error_code mcd_psfr_find_intervals (const cpl_vector *interpolation_points, const cpl_vector *interval_points, cpl_size *idx_a, cpl_size *idx_b, double *wght_a)

  For each point x find points a,b such that a <= x <= b.
- cpl_imagelist * mcd_psfr_interpolate_2d_batch (const cpl_vector *x_given, const cpl_vector *y_given, const cpl_imagelist *f_given, const cpl_vector *x_interp, const cpl_vector *y_interp, double extension)

  Interpolate a list of functions from one grid to another grid contained in the first one. Grids are repesented by the tensor product of x and y coordinates.
- cpl_error_code mcd_psfr_interpolate_2d (const cpl_vector *x_given, const cpl_vector *y_given, const cpl↩ _image *f_given, const cpl_bivector *interpolation_points, cpl_vector *interpolated_values)

  Interpolate a function from one grid to another grid contained in the first one. The given grids is repesented by the tensor product of x and y coordinates, the interpolation points are given by vectors with x and y coordinates.
- cpl_error_code mcd_psfr_interpolate_2d_tp (const cpl_vector *x_given, const cpl_vector *y_given, const cpl_image *f_given, const cpl_vector *x_interp, const cpl_vector *y_interp, cpl_image *f_interp, double extension)

  Interpolate a function from one grid to another grid contained in the first one. Grids are repesented by the tensor product of x and y coordinates.
- cpl_matrix * mcd_psfr_scao_cal_WFS_noise_m4_to_fried (const mcd_psfr_influence_functions *scaled_ifs) CPL_ATTR_ALLOC
- static cpl_matrix * mcd_psfr_scao_cal_WFS_noise_compute_covariance_matrix (const cpl_matrix *transfer, const cpl_matrix *basis, const cpl_vector *modal_gains_avg, const cpl_matrix *cmat) CPL_ATTR_ALLOC

- static cpl_vector * mcd_psfr_scao_cal_WFS_noise_compute_average_gain (const cpl_image *modal_↩ gains) CPL_ATTR_ALLOC
- cpl_size mcd_psfr_utility_image_rws (cpl_size n, cpl_size first, cpl_size second)
- cpl_error_code mcd_psfr_utility_matrix_add (cpl_matrix *a, cpl_size r, cpl_size c, double value)
- cpl_size mcd_psfr_utility_matrix_nonzero (const cpl_matrix *a)
- cpl_size mcd_psfr_utility_matrix_size (const cpl_matrix *a)
- double mcd_psfr_utility_image_relative_error (const cpl_image *a, const cpl_image *b)
- cpl_vector * mcd_psfr_utility_vector_linspace_create (cpl_size size, double start, double end)

  Create a vector with n values from start to end.
- cpl_size mcd_psfr_utility_next_power_of_two (cpl_size x)
- bool mcd_psfr_utility_is_power_of_two (cpl_size x)
- CPL_ATTR_ALLOC cpl_image * mcd_psfr_utility_image_pad_create (const cpl_image *image, cpl_size fft↩ _size)

  Pad the image with zeros.
- CPL_ATTR_ALLOC cpl_image * mcd_psfr_utility_image_unpad_create (const cpl_image *padded, cpl_size size, cpl_size zero_freq_location)

  Unpad an image from a padded FFT. CPL-FFT with CPL_FFT_SWAP_HALVES places the zero-frequency at pixel fftsize/2+1. If no target zero-frequency location is given, use the center pixel (n+1)/2 (odd image dimensions), or n/2 (even image dimensions).
- cpl_error_code mcd_psfr_utility_fftshift (cpl_image *image)
- CPL_ATTR_ALLOC cpl_image * mcd_psfr_utility_image_autocorrelation (const cpl_image *image)
- cpl_error_code mcd_psfr_utility_image_add_scale_window (cpl_image *dst, const cpl_image *add, double weight, cpl_size start_x, cpl_size start_y)
- CPL_ATTR_ALLOC cpl_imagelist * mcd_psfr_utility_imagelist_resize (const cpl_imagelist *source, cpl_size new_x, cpl_size new_y)
- CPL_ATTR_ALLOC cpl_image * mcd_psfr_utility_image_resize (const cpl_image *source, cpl_size new_x, cpl_size new_y)
- cpl_size mcd_psfr_utility_last_nonzero_row (const cpl_matrix *m)
- cpl_size mcd_psfr_utility_last_nonzero_col (const cpl_matrix *m)
- CPL_ATTR_ALLOC cpl_frameset * mcd_psfr_utility_frameset_from_compass_data (const char *folder)
- cpl_image * mcd_psfr_utility_transpose_image_if_required (cpl_image *in, cpl_size expected_nx, cpl_size expected_ny)
- CPL_ATTR_ALLOC cpl_image * mcd_psfr_utility_get_tip (cpl_size nx, cpl_size ny)
- CPL_ATTR_ALLOC cpl_image * mcd_psfr_utility_get_tilt (cpl_size nx, cpl_size ny)
- mcd_psfr_gridfunction * mcd_psfr_utility_gridfunction_from_img_px (cpl_image *img, double pixel_size)

  Writes the input image into a gridfunction, the domain is defined by the pixel_size.
- mcd_psfr_gridfunction * mcd_psfr_utility_gridfunction_from_img_length (cpl_image *img, double length)

  Writes the input image into a gridfunction, the domain is defined by the length of the domain.
- cpl_image * mcd_psfr_utility_read_data (cpl_frameset *frameset, char *tag)

  Reads the input data, finds the tag and returns the image read from the corresponding fits file.

3.28.1 Detailed Description

Author

D. Jodlbauer, A. La Camera, M. Ravenna, R. Wagner

Miscellaneous Utilities for PSFR functions

3.28.2 Macro Definition Documentation

3.28.2.1 MCD_PSFR_TIME_SECTION

```
#define MCD_PSFR_TIME_SECTION(
               name,
               code )
```

Value:

```
    const double CPL_CONCAT2X(clock, __LINE__) = mcd_psfr_cpl_wall_time(); \
    code cpl_msg_info("mcd",                                               \
                      "Time for %s/%s:  %f s",                              \
                      cpl_func,                                            \
                      name,                                                \
                      mcd_psfr_cpl_wall_time() - CPL_CONCAT2X(clock, __LINE__));
```

Section timing

Definition at line 62 of file mcd_psfr_utility.h.

3.28.3 Function Documentation

3.28.3.1 mcd_psfr_box_common_area()

```
mcd_psfr_box mcd_psfr_box_common_area (
             int a,
             int b,
             int n )
```

Compute the maximal common area between an image and itself shifted by (a,b)

Parameters

a	Shift in the first component
b	Shift in the second component
n	Size of the image

Returns

Bounding box of the ommon area

Definition at line 206 of file mcd_psfr_box.c.

3.28.3.2 mcd_psfr_box_get_size_x()

```
cpl_size mcd_psfr_box_get_size_x (
             mcd_psfr_box a )
```

Return the extent in x-direction

Definition at line 75 of file mcd_psfr_box.c.

3.28.3.3 mcd_psfr_box_get_size_y()

```
cpl_size mcd_psfr_box_get_size_y (
            mcd_psfr_box a )
```

Return the extent in y-direction

Definition at line 84 of file mcd_psfr_box.c.

3.28.3.4 mcd_psfr_box_image_bounding_box()

```
mcd_psfr_box mcd_psfr_box_image_bounding_box (
            const cpl_image * image )
```

Compute the bounding box of the non-zero region of an image.

Parameters

image	Image

Returns

Bounding box containing all non-zero parts

Definition at line 95 of file mcd_psfr_box.c.

Referenced by mcd_psfr_box_image_crop(), and mcd_psfr_image_bounding_box_mode().

3.28.3.5 mcd_psfr_box_image_crop()

```
CPL_ATTR_ALLOC cpl_image* mcd_psfr_box_image_crop (
            const cpl_image * img )
```

Extract the non-zero region of an image

Parameters

img	Image to crop

Returns

New cropped image

Definition at line 193 of file mcd_psfr_box.c.

References mcd_psfr_box_image_bounding_box().

3.28.3.6 mcd_psfr_box_shift()

```
mcd_psfr_box mcd_psfr_box_shift (
             mcd_psfr_box a,
             int shift_x,
             int shift_y )
```

Return a shifted a bounding box by a certain amount

Definition at line 53 of file mcd_psfr_box.c.

3.28.3.7 mcd_psfr_box_unify()

```
mcd_psfr_box mcd_psfr_box_unify (
             mcd_psfr_box a,
             mcd_psfr_box b )
```

Return the bounding box of two bounding boxes

Definition at line 41 of file mcd_psfr_box.c.

3.28.3.8 mcd_psfr_box_valid()

```
bool mcd_psfr_box_valid (
             mcd_psfr_box a )
```

Check if a bounding box is valid

Definition at line 67 of file mcd_psfr_box.c.

3.28.3.9 mcd_psfr_covariance_average()

```
CPL_ATTR_ALLOC cpl_image* mcd_psfr_covariance_average (
             const cpl_matrix * covariance_matrix,
             cpl_size n )
```

Compute the average covariance between pixels with equal distance (a,b), i.e., between pixels (x,y) and (x+a,y+b) for all x,y that make sense. Result is stored in a 2n-1 x 2n-1 matrix, with shift (a,b) = (0,0) in the center.

Parameters

covariance_matrix	Covariance matrix
n	Square-root of the number of rows of the covariance matrix

Returns

Structure function

Definition at line 50 of file mcd_psfr_covariance.c.

Referenced by mcd_psfr_scao_cal_WFS_noise().

3.28.3.10 mcd_psfr_cpl_image_add_scale()

```
cpl_error_code mcd_psfr_cpl_image_add_scale (
            cpl_image * a,
            const cpl_image * b,
            double factor_b )
```

Compute a = a + factor_b + b.

Parameters

a	Result and left image
b	right image
factor_b	right factor

Definition at line 86 of file mcd_psfr_cpl_ext.c.

Referenced by mcd_psfr_cpl_imagelist_sum(), mcd_psfr_scao_anisoplanatic_phase(), and mcd_psfr_utility_image_add_scale_window().

3.28.3.11 mcd_psfr_cpl_image_fill()

```
cpl_error_code mcd_psfr_cpl_image_fill (
            cpl_image * image,
            double value )
```

Fill the whole image with a value.

Parameters

image	The image to modify
value	The value to fill

Definition at line 55 of file mcd_psfr_cpl_ext.c.

Referenced by mcd_psfr_otf_multi().

3.28.3.12 mcd_psfr_cpl_image_get_or_default()

```
double mcd_psfr_cpl_image_get_or_default (
            const cpl_image * a,
            cpl_size i,
            cpl_size j,
            double def )
```

Return the (i,j)-th element, or some default value if it is outside the index range.

Definition at line 69 of file mcd_psfr_cpl_ext.c.

Referenced by mcd_psfr_interpolate_2d_batch().

3.28.3.13 mcd_psfr_cpl_image_transpose()

```
cpl_error_code mcd_psfr_cpl_image_transpose (
            cpl_image * in )
```

Transpose an image

Parameters

in	Image to transpose

Returns

Error code

Definition at line 45 of file mcd_psfr_cpl_ext.c.

Referenced by mcd_psfr_utility_transpose_image_if_required().

3.28.3.14 mcd_psfr_cpl_imagelist_load_ext()

```
cpl_imagelist* mcd_psfr_cpl_imagelist_load_ext (
            const char * filename,
            cpl_type im_type,
            int xtnum,
            int indexplane )
```

Load an imagelist from a FITS file.

Parameters

filename	Name of the file to load from
im_type	Type of the created images
xtnum	Extension number in the file (0 for primary HDU)
indexplane	Plane number for image index

Returns

1 newly allocated imagelist or NULL on error

Examples: indexplane = 0: result[i][x][y] = raw[i][x][y] indexplane = 1: result[i][x][y] = raw[x][i][y] indexplane = 2: result[i][x][y] = raw[x][y][i]

Definition at line 115 of file mcd_psfr_cpl_ext.c.

Referenced by mcd_psfr_influence_functions_load().

3.28.3.15 mcd_psfr_cpl_imagelist_sample()

```
CPL_ATTR_ALLOC cpl_imagelist* mcd_psfr_cpl_imagelist_sample (
            cpl_imagelist * images,
            cpl_size start,
            cpl_size step )
```

Select every n-th image from an imagelist. Result needs to be deleted using cpl_imagelist_unwrap.

Parameters

images	List of images
start	Starting index
step	Stepsize

Returns

Subset of images

Definition at line 234 of file mcd_psfr_cpl_ext.c.

Referenced by mcd_psfr_scao_parallel().

3.28.3.16 mcd_psfr_cpl_imagelist_sum()

```
CPL_ATTR_ALLOC cpl_image* mcd_psfr_cpl_imagelist_sum (
            const cpl_imagelist * images,
            const cpl_vector * weights,
            cpl_size N )
```

Weighted sum over an imagelist Computes Sum(weight[i]∗image[i] for i = 0 .. N-1)

Parameters

images	List of images
weights	Weight vector, or NULL (weight == 1)
N	Use the first N images (all if N == 0)

Returns

Resulting image

Definition at line 194 of file mcd_psfr_cpl_ext.c.

References mcd_psfr_cpl_image_add_scale().

3.28.3.17 mcd_psfr_cpl_wall_time()

```
double mcd_psfr_cpl_wall_time (
            void  )
```

Return the elapsed wall time in seconds

Definition at line 247 of file mcd_psfr_cpl_ext.c.

Referenced by mcd_psfr_scao_anisoplanatic_phase(), mcd_psfr_scao_cal_WFS_noise(), mcd_psfr_scao_cal_↩ WFS_noise_compute_covariance_matrix(), mcd_psfr_scao_cal_WFS_noise_m4_to_fried(), mcd_psfr_scao_cal↩ _WFS_slopes(), mcd_psfr_scao_orthogonal(), and mcd_psfr_scao_parallel().

3.28.3.18 mcd_psfr_find_intervals()

```
static cpl_error_code mcd_psfr_find_intervals (
            const cpl_vector * interpolation_points,
            const cpl_vector * interval_points,
            cpl_size * idx_a,
            cpl_size * idx_b,
            double * wght_a )  [static]
```

For each point x find points a,b such that a <= x <= b.

Parameters

interpolation_points	The points x to fit into an interval
interval_points	Points defining the intervals
idx_a	Array[interpolation_points] of indices of the left points a for each interpolation point
idx_b	Array[interpolation_points] of indices of the right points b for each interpolation point
wght_a	Array[interpolation_points] of weights for the left point, (1-wgh_a[i]) for the right point

Returns

Definition at line 53 of file mcd_psfr_interpolate.c.

Referenced by mcd_psfr_interpolate_2d(), and mcd_psfr_interpolate_2d_batch().

3.28.3.19 mcd_psfr_fits_get_double()

```
double mcd_psfr_fits_get_double (
            const char * filename,
            const char * variable_name )
```

Search the fits file for the given variable and return its value

Parameters

filename	FITS filename
variable_name	Variable name

Returns

Value as double, or NAN in case of failure

Definition at line 46 of file mcd_psfr_fits.c.

Referenced by mcd_psfr_influence_functions_load(), and mcd_psfr_scao_otf_elt().

3.28.3.20 mcd_psfr_image_bounding_box_mode()

```
static mcd_psfr_box mcd_psfr_image_bounding_box_mode (
            const cpl_image * influ,
            enum mcd_psfr_embedding mode )  [static]
```

Return the bounding box of an influence function

Parameters

influ	Influence function
mode	If mode is MCD_PSFR_TIGHTEST, remove zero-regions

Returns

Bounding box

Definition at line 819 of file mcd_psfr_influence_functions.c.

References mcd_psfr_box_image_bounding_box().

3.28.3.21 mcd_psfr_influence_functions_bounding_box()

```
static mcd_psfr_box mcd_psfr_influence_functions_bounding_box (
            const mcd_psfr_influence_functions * ifs,
            enum mcd_psfr_embedding mode )  [static]
```

Return the bounding box of all influence functions (with their correct position)

Parameters

ifs	Influence functions
mode	Determine truncation mode of zero regions

Returns

Bounding box

Definition at line 735 of file mcd_psfr_influence_functions.c.

3.28.3.22 mcd_psfr_influence_functions_delete()

```
void mcd_psfr_influence_functions_delete (
            mcd_psfr_influence_functions * ifs )
```

Destroy an influence function struct

Parameters

ifs	Object to destroy

Definition at line 89 of file mcd_psfr_influence_functions.c.

Referenced by mcd_psfr_influence_functions_load(), mcd_psfr_parameters_delete(), and mcd_psfr_scao_cal_↩ WFS_noise().

3.28.3.23 mcd_psfr_influence_functions_domain()

```
mcd_psfr_domain mcd_psfr_influence_functions_domain (
            const mcd_psfr_influence_functions * ifs )
```

Return the domain description

Definition at line 206 of file mcd_psfr_influence_functions.c.

References mcd_psfr_domain_invalid().

Referenced by mcd_psfr_scao_cal_WFS_noise(), and mcd_psfr_scao_cal_WFS_slopes().

3.28.3.24 mcd_psfr_influence_functions_embed()

```
CPL_ATTR_ALLOC mcd_psfr_influence_functions* mcd_psfr_influence_functions_embed (
            const mcd_psfr_influence_functions * ifs,
            enum mcd_psfr_embedding mode )
```

Place each influence function at the correct position in a larger image.

Parameters

ifs	Struct with influence functions and positions
mode	Determine the placement mode, e.g.: MCD_PSFR_NATURAL: Place each influence function at the given position, resulting image is [1,max(pos+if_size)] MCD_PSFR_TIGHT: Place each influence function at the given position, but crops influence functions to non-zero regions, resulting image is [min(pos),max(pos+if_size)] MCD_PSFR_TIGHTEST: MCD_PSFR_NATURAL, with all nonzero regions stripped away, resulting image is [min(pos+support_start),max(pos+support_end)]
square	Force a square domain

Returns

List of extended influence functions

Definition at line 467 of file mcd_psfr_influence_functions.c.

3.28.3.25 mcd_psfr_influence_functions_get()

```
const cpl_imagelist* mcd_psfr_influence_functions_get (
            const mcd_psfr_influence_functions * ifs )
```

Return the stored influence functions

Parameters

ifs	Influence function object

Returns

List of influence functions

Definition at line 260 of file mcd_psfr_influence_functions.c.

Referenced by mcd_psfr_scao_cal_WFS_noise_m4_to_fried().

3.28.3.26 mcd_psfr_influence_functions_get_x()

```
const cpl_array* mcd_psfr_influence_functions_get_x (
            const mcd_psfr_influence_functions * ifs )
```

Return the x-coordinates

Parameters

ifs	Influence function object

Returns

List of x-coordinates

Definition at line 272 of file mcd_psfr_influence_functions.c.

3.28.3.27 mcd_psfr_influence_functions_get_y()

```
const cpl_array* mcd_psfr_influence_functions_get_y (
            const mcd_psfr_influence_functions * ifs )
```

Return the y-coordinates

Parameters

ifs	Influence function object

Returns

List of y-coordinates

Definition at line 284 of file mcd_psfr_influence_functions.c.

3.28.3.28 mcd_psfr_influence_functions_is_embedded()

```
bool mcd_psfr_influence_functions_is_embedded (
            const mcd_psfr_influence_functions * ifs )
```

Check if the influence functions are embedded in a larger grid already, i.e., all influence functions will be placed at (0, 0).

Definition at line 716 of file mcd_psfr_influence_functions.c.

Referenced by mcd_psfr_scao_cal_WFS_noise_m4_to_fried().

3.28.3.29 mcd_psfr_influence_functions_load()

```
CPL_ATTR_ALLOC mcd_psfr_influence_functions* mcd_psfr_influence_functions_load (
            const char * fits_file )
```

Load influence functions and positions from a COMPASS fits file

Parameters

fits_file	Filename of fits file

Returns

Newly created influence function structure

Definition at line 109 of file mcd_psfr_influence_functions.c.

References mcd_psfr_cpl_imagelist_load_ext(), mcd_psfr_fits_get_double(), mcd_psfr_influence_functions_↩ delete(), and mcd_psfr_influence_functions_new().

3.28.3.30 mcd_psfr_influence_functions_new()

```
static CPL_ATTR_ALLOC mcd_psfr_influence_functions * mcd_psfr_influence_functions_new (
            void  )  [static]
```

Create a new influence function struct

Definition at line 74 of file mcd_psfr_influence_functions.c.

References mcd_psfr_domain_invalid().

Referenced by mcd_psfr_influence_functions_load().

3.28.3.31 mcd_psfr_influence_functions_resize()

```
CPL_ATTR_ALLOC mcd_psfr_influence_functions* mcd_psfr_influence_functions_resize (
            const mcd_psfr_influence_functions * ifs,
            cpl_size nx,
            cpl_size ny )
```

Scale influence functions support to a different support size, and update its positions accordingly.

Parameters

ifs	Struct with influence functions and positions
nx	Resulting size of each influence functions support in x-direction
ny	Resulting size of each influence functions support in y-direction

Returns

Scaled influence functions and positions

Definition at line 363 of file mcd_psfr_influence_functions.c.

3.28.3.32 mcd_psfr_influence_functions_resize_all()

```
CPL_ATTR_ALLOC mcd_psfr_influence_functions* mcd_psfr_influence_functions_resize_all (
            const mcd_psfr_influence_functions * ifs,
            cpl_size nx,
            cpl_size ny )
```

Resize already embedded influence functions to the given size.

Parameters

ifs	Influence function struct
nx	Size in x direction
ny	Size in y direction

Definition at line 297 of file mcd_psfr_influence_functions.c.

Referenced by mcd_psfr_scao_cal_WFS_noise().

3.28.3.33 mcd_psfr_influence_functions_size()

```
cpl_size mcd_psfr_influence_functions_size (
            const mcd_psfr_influence_functions * ifs )
```

Return the number of influence functions

Parameters

ifs	Influence function object

Returns

Number of influence functions

Definition at line 218 of file mcd_psfr_influence_functions.c.

Referenced by mcd_psfr_influence_functions_size_x(), mcd_psfr_influence_functions_size_y(), mcd_psfr_scao_↩ cal_WFS_noise_m4_to_fried(), and mcd_psfr_scao_cal_WFS_slopes().

3.28.3.34 mcd_psfr_influence_functions_size_x()

```
cpl_size mcd_psfr_influence_functions_size_x (
            const mcd_psfr_influence_functions * ifs )
```

Return the size of each influence function in x-direction

Parameters

ifs	Influence function object

Returns

Size

Definition at line 230 of file mcd_psfr_influence_functions.c.

References mcd_psfr_influence_functions_size().

Referenced by mcd_psfr_scao_cal_WFS_noise_m4_to_fried(), and mcd_psfr_scao_cal_WFS_slopes().

3.28.3.35 mcd_psfr_influence_functions_size_y()

```
cpl_size mcd_psfr_influence_functions_size_y (
            const mcd_psfr_influence_functions * ifs )
```

Return the size of each influence function in y-direction

Parameters

ifs	Influence function object

Returns

Size

Definition at line 245 of file mcd_psfr_influence_functions.c.

References mcd_psfr_influence_functions_size().

Referenced by mcd_psfr_scao_cal_WFS_noise_m4_to_fried(), and mcd_psfr_scao_cal_WFS_slopes().

3.28.3.36 mcd_psfr_influence_functions_sum()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_influence_functions_sum (
            const mcd_psfr_influence_functions * ifs,
            const cpl_vector * weights,
            cpl_size N )
```

A more memory efficient implementation of the weighted sum over all influence functions

Parameters

ifs	Influence functions
weights	Weight per influence function
N	Take first N image, all if N == 0

Returns

Summed image

Definition at line 577 of file mcd_psfr_influence_functions.c.

Referenced by mcd_psfr_scao_cal_WFS_slopes().

3.28.3.37 mcd_psfr_influence_functions_support()

```
CPL_ATTR_ALLOC mcd_psfr_gridfunction* mcd_psfr_influence_functions_support (
            const mcd_psfr_influence_functions * ifs )
```

Compute the support of the influence functions

Parameters

ifs	Influence functions

Returns

Indicator function (1 = some influence function is non-zero at a pixel, 0 = otherwise)

Definition at line 684 of file mcd_psfr_influence_functions.c.

3.28.3.38 mcd_psfr_interpolate_2d()

```
cpl_error_code mcd_psfr_interpolate_2d (
            const cpl_vector * x_given,
            const cpl_vector * y_given,
            const cpl_image * f_given,
            const cpl_bivector * interpolation_points,
            cpl_vector * interpolated_values )
```

Interpolate a function from one grid to another grid contained in the first one. The given grids is repesented by the tensor product of x and y coordinates, the interpolation points are given by vectors with x and y coordinates.

Parameters

x_given	x-coordinates of the original grid
y_given	y-coordinates of the original grid
f_given	function values on the original grid stored as f[i+1][j+1] = f(x_i,y_j)
interpolation_points	x,y-coordinates of the new grid
interpolated_values	computed function values on the new grid as f[i] = f(x[i],y[i])

Returns

Definition at line 314 of file mcd_psfr_interpolate.c.

References mcd_psfr_find_intervals().

3.28.3.39 mcd_psfr_interpolate_2d_batch()

```
cpl_imagelist* mcd_psfr_interpolate_2d_batch (
            const cpl_vector * x_given,
            const cpl_vector * y_given,
            const cpl_imagelist * f_given,
            const cpl_vector * x_interp,
            const cpl_vector * y_interp,
            double extension )
```

Interpolate a list of functions from one grid to another grid contained in the first one. Grids are repesented by the tensor product of x and y coordinates.

Parameters

x_given	x-coordinates of the original grid
y_given	y-coordinates of the original grid
f_given	list of functions, with function values on the original grid stored as f[i+1][j+1] = f(x_i,y_j)
x_interp	x-coordinates of the new grid
y_interp	y-coordinates of the new grid
f_interp	computed function values on the new grid as f[i+1][j+1] = f(x_i,y_j)
extension	value to use for points outside the data region Note: +1 because CPL images start at 1

Returns

Interpolated list of images

Definition at line 186 of file mcd_psfr_interpolate.c.

References mcd_psfr_cpl_image_get_or_default(), and mcd_psfr_find_intervals().

Referenced by mcd_psfr_interpolate_2d_tp(), and mcd_psfr_utility_imagelist_resize().

3.28.3.40 mcd_psfr_interpolate_2d_tp()

```
cpl_error_code mcd_psfr_interpolate_2d_tp (
            const cpl_vector * x_given,
            const cpl_vector * y_given,
            const cpl_image * f_given,
```

```
            const cpl_vector * x_interp,
            const cpl_vector * y_interp,
            cpl_image * f_interp,
            double extension )
```

Interpolate a function from one grid to another grid contained in the first one. Grids are repesented by the tensor product of x and y coordinates.

Parameters

x_given	x-coordinates of the original grid
y_given	y-coordinates of the original grid
f_given	function values on the original grid stored as f[i+1][j+1] = f(x_i,y_j)
x_interp	x-coordinates of the new grid
y_interp	y-coordinates of the new grid
f_interp	computed function values on the new grid as f[i+1][j+1] = f(x_i,y_j) Note: +1 because CPL images start at 1
extension	Value outside of the domain

Returns

Definition at line 425 of file mcd_psfr_interpolate.c.

References mcd_psfr_interpolate_2d_batch().

3.28.3.41 mcd_psfr_scao_cal_WFS_noise_compute_average_gain()

```
static CPL_ATTR_ALLOC cpl_vector * mcd_psfr_scao_cal_WFS_noise_compute_average_gain (
            const cpl_image * modal_gains )  [static]
```

Inputs:

- [RAW] AO_CONFIGURATION_TAB
- [RAW] CN2_PROFILE_TAB
- [RAW] PYR_WFS_BUFFER_TAB

Output:

- [INTERM] PYR_WFS_NOISE_TAB

Definition at line 422 of file mcd_psfr_scao_cal_WFS_noise.c.

Referenced by mcd_psfr_scao_cal_WFS_noise().

3.28.3.42 mcd_psfr_scao_cal_WFS_noise_compute_covariance_matrix()

```
static CPL_ATTR_ALLOC cpl_matrix * mcd_psfr_scao_cal_WFS_noise_compute_covariance_matrix (
            const cpl_matrix * transfer,
            const cpl_matrix * basis,
            const cpl_vector * modal_gains_avg,
            const cpl_matrix * cmat )  [static]
```

Compute the reconstruction covariance matrix by multiplying the single matrices in an optimal way.

Returns

New covariance matrix

Definition at line 317 of file mcd_psfr_scao_cal_WFS_noise.c.

References mcd_psfr_cpl_wall_time().

Referenced by mcd_psfr_scao_cal_WFS_noise().

3.28.3.43 mcd_psfr_scao_cal_WFS_noise_m4_to_fried()

```
CPL_ATTR_ALLOC cpl_matrix * mcd_psfr_scao_cal_WFS_noise_m4_to_fried (
            const mcd_psfr_influence_functions * scaled_ifs )
```

Compute the basis transformation from influence functions to fried geometry, i.e., a matrix of size B = nFried^2 x nInfl such that f = B c with a coefficient vector c in R^nInfl, and f in R^nFried^2 (unrolled image) converted from to [m]. Tip/tilt influence functions are added at the end.

Parameters

scaled_ifs	Influence function information (already embedded in a larger grid)

Returns

Matrix of size nq^2 x len(M4) matrix, containing the projected influence functions per row TODO each row contains the raw image data, might need a transpose?

Definition at line 222 of file mcd_psfr_scao_cal_WFS_noise.c.

References mcd_psfr_cpl_wall_time(), mcd_psfr_influence_functions_get(), mcd_psfr_influence_functions_is↩_embedded(), mcd_psfr_influence_functions_size(), mcd_psfr_influence_functions_size_x(), and mcd_psfr_↩influence_functions_size_y().

Referenced by mcd_psfr_scao_cal_WFS_noise().

3.28.3.44 mcd_psfr_utility_fftshift()

```
cpl_error_code mcd_psfr_utility_fftshift (
            cpl_image * image )
```

Implementation of Matlabs fftshift.

Parameters

image	Image to shift inplace

Definition at line 381 of file mcd_psfr_utility.c.

3.28.3.45 mcd_psfr_utility_frameset_from_compass_data()

```
CPL_ATTR_ALLOC cpl_frameset* mcd_psfr_utility_frameset_from_compass_data (
            const char * folder )
```

Create a test frameset from Compass output.

Parameters

folder	Folder containing Compass output

Returns

New frameset

Definition at line 621 of file mcd_psfr_utility.c.

Referenced by mcd_psfr_parameters_new_compass().

3.28.3.46 mcd_psfr_utility_get_tilt()

```
CPL_ATTR_ALLOC cpl_image* mcd_psfr_utility_get_tilt (
            cpl_size nx,
            cpl_size ny )
```

Return the tilt matrix [-1, 1]

Definition at line 862 of file mcd_psfr_utility.c.

References mcd_psfr_utility_image_resize().

Referenced by mcd_psfr_scao_cal_WFS_slopes().

3.28.3.47 mcd_psfr_utility_get_tip()

```
CPL_ATTR_ALLOC cpl_image* mcd_psfr_utility_get_tip (
            cpl_size nx,
            cpl_size ny )
```

Return the tip matrix [-1] [1]

Definition at line 849 of file mcd_psfr_utility.c.

References mcd_psfr_utility_image_resize().

Referenced by mcd_psfr_scao_cal_WFS_slopes().

3.28.3.48 mcd_psfr_utility_gridfunction_from_img_length()

```
mcd_psfr_gridfunction* mcd_psfr_utility_gridfunction_from_img_length (
            cpl_image * img,
            double length )
```

Writes the input image into a gridfunction, the domain is defined by the length of the domain.

Parameters

img	the cpl_image data
length	the value of the length

Returns

new_gridfunction

Definition at line 920 of file mcd_psfr_utility.c.

References mcd_psfr_gridfunction_wrap().

3.28.3.49 mcd_psfr_utility_gridfunction_from_img_px()

```
mcd_psfr_gridfunction* mcd_psfr_utility_gridfunction_from_img_px (
            cpl_image * img,
            double pixel_size )
```

Writes the input image into a gridfunction, the domain is defined by the pixel_size.

Parameters

img	the cpl_image data
pixel_size	the value of the pixel size

Returns

new_gridfunction

Definition at line 882 of file mcd_psfr_utility.c.

References mcd_psfr_domain_centered(), and mcd_psfr_gridfunction_wrap().

Referenced by mcd_psfr_scao_otf_elt().

3.28.3.50 mcd_psfr_utility_image_add_scale_window()

```
cpl_error_code mcd_psfr_utility_image_add_scale_window (
            cpl_image * dst,
            const cpl_image * add,
            double weight,
            cpl_size start_x,
            cpl_size start_y )
```

Add an image to another image at a given position

Parameters

dst	Destination image
add	Image to add
weight	Scaling factor
start_x	x-coordinate in destination image
start_y	y-coordinate in destination image

Definition at line 458 of file mcd_psfr_utility.c.

References mcd_psfr_cpl_image_add_scale().

3.28.3.51 mcd_psfr_utility_image_autocorrelation()

```
CPL_ATTR_ALLOC cpl_image* mcd_psfr_utility_image_autocorrelation (
            const cpl_image * image )
```

Compute the auto-correlation of an image using the FFT.

Parameters

image	Input image nxn

Returns

Auto-correlation of size 2n-1

Definition at line 398 of file mcd_psfr_utility.c.

References mcd_psfr_utility_image_pad_create(), mcd_psfr_utility_image_unpad_create(), and mcd_psfr_utility↩
_next_power_of_two().

3.28.3.52 mcd_psfr_utility_image_pad_create()

```
CPL_ATTR_ALLOC cpl_image* mcd_psfr_utility_image_pad_create (
            const cpl_image * image,
            cpl_size fft_size )
```

Pad the image with zeros.

Parameters

image	Image to pad
fft_size	Size to pad up to, or 0 to use the next power of 2

Definition at line 255 of file mcd_psfr_utility.c.

References mcd_psfr_utility_next_power_of_two().

Referenced by mcd_psfr_utility_image_autocorrelation().

3.28.3.53 mcd_psfr_utility_image_relative_error()

```
double mcd_psfr_utility_image_relative_error (
            const cpl_image * a,
            const cpl_image * b )
```

Compute the relative error between two images

Parameters

a	First image
b	Second image

Returns

|a-b| / (|a|+|b|)

Definition at line 108 of file mcd_psfr_utility.c.

3.28.3.54 mcd_psfr_utility_image_resize()

```
CPL_ATTR_ALLOC cpl_image* mcd_psfr_utility_image_resize (
            const cpl_image * source,
            cpl_size new_x,
            cpl_size new_y )
```

Resize an image to a new size by interpolation

Parameters

source	Input image
new_x	New size in x-direction
new_y	New size in y-direction

Returns

Resized image

Definition at line 549 of file mcd_psfr_utility.c.

References mcd_psfr_utility_imagelist_resize().

Referenced by mcd_psfr_utility_get_tilt(), and mcd_psfr_utility_get_tip().

3.28.3.55 mcd_psfr_utility_image_rws()

```
cpl_size mcd_psfr_utility_image_rws (
            cpl_size n,
            cpl_size first,
            cpl_size second )
```

CPL index scheme for images. Arguments are 1-based analogously to cpl_image.

Definition at line 50 of file mcd_psfr_utility.c.

3.28.3.56 mcd_psfr_utility_image_unpad_create()

```
CPL_ATTR_ALLOC cpl_image* mcd_psfr_utility_image_unpad_create (
            const cpl_image * padded,
            cpl_size size,
            cpl_size zero_freq_location )
```

Unpad an image from a padded FFT. CPL-FFT with CPL_FFT_SWAP_HALVES places the zero-frequency at pixel fftsize/2+1. If no target zero-frequency location is given, use the center pixel (n+1)/2 (odd image dimensions), or n/2 (even image dimensions).

Parameters

padded	Image to unpad
size	Result size n (size before padding)
zero_freq_location	Location of the zero-frequency in the result image, or 0

Definition at line 317 of file mcd_psfr_utility.c.

References mcd_psfr_utility_is_power_of_two().

Referenced by mcd_psfr_utility_image_autocorrelation().

3.28.3.57 mcd_psfr_utility_imagelist_resize()

```
CPL_ATTR_ALLOC cpl_imagelist* mcd_psfr_utility_imagelist_resize (
            const cpl_imagelist * source,
            cpl_size new_x,
            cpl_size new_y )
```

Resize all images of an image list the the new size by interpolation

Parameters

source	List of images
new_x	New size in x-direction
new_y	New size in y-direction

Returns

List of resized images

Definition at line 496 of file mcd_psfr_utility.c.

References mcd_psfr_interpolate_2d_batch(), and mcd_psfr_utility_vector_linspace_create().

Referenced by mcd_psfr_utility_image_resize().

3.28.3.58 mcd_psfr_utility_is_power_of_two()

```
bool mcd_psfr_utility_is_power_of_two (
            cpl_size x )
```

Check whether the given integer is a power of 2.

Parameters

x	Integer to check

Returns

True/False

Definition at line 245 of file mcd_psfr_utility.c.

Referenced by mcd_psfr_utility_image_unpad_create().

3.28.3.59 mcd_psfr_utility_last_nonzero_col()

```
cpl_size mcd_psfr_utility_last_nonzero_col (
            const cpl_matrix * m )
```

Return the index of the last nonzero column

Parameters

m	Matrix to analyze

Returns

Index

Definition at line 600 of file mcd_psfr_utility.c.

3.28.3.60 mcd_psfr_utility_last_nonzero_row()

```
cpl_size mcd_psfr_utility_last_nonzero_row (
            const cpl_matrix * m )
```

Return the index of the last nonzero column

Parameters

m	Matrix to analyze

Returns

Index

Definition at line 580 of file mcd_psfr_utility.c.

3.28.3.61 mcd_psfr_utility_matrix_add()

```
cpl_error_code mcd_psfr_utility_matrix_add (
            cpl_matrix * a,
            cpl_size r,
            cpl_size c,
            double value )
```

Add a value to the given matrix index.

Parameters

a	Matrix to add to
r	Row index
c	Column index
value	Value to add

Definition at line 71 of file mcd_psfr_utility.c.

3.28.3.62 mcd_psfr_utility_matrix_nonzero()

```
cpl_size mcd_psfr_utility_matrix_nonzero (
            const cpl_matrix * a )
```

Count the number of nonzero entries in the given matrix.

Parameters

a	Matrix to analyse

Returns

Number of nonzero entries

Definition at line 81 of file mcd_psfr_utility.c.

3.28.3.63 mcd_psfr_utility_matrix_size()

```
cpl_size mcd_psfr_utility_matrix_size (
            const cpl_matrix * a )
```

Return the number of entries in the matrix

Definition at line 96 of file mcd_psfr_utility.c.

3.28.3.64 mcd_psfr_utility_next_power_of_two()

```
cpl_size mcd_psfr_utility_next_power_of_two (
            cpl_size x )
```

Return the smallest n such that x <= 2^k = n

Parameters

x	Value

Returns

Next n == 2^k

Definition at line 230 of file mcd_psfr_utility.c.

Referenced by mcd_psfr_parameters_internal(), mcd_psfr_utility_image_autocorrelation(), and mcd_psfr_utility_↩ image_pad_create().

3.28.3.65 mcd_psfr_utility_read_data()

```
cpl_image* mcd_psfr_utility_read_data (
            cpl_frameset * frameset,
            char * tag )
```

Reads the input data, finds the tag and returns the image read from the corresponding fits file.

Parameters

frameset	the input frameset
tag	the tag to be searched

Returns

new_gridfunction

Definition at line 961 of file mcd_psfr_utility.c.

Referenced by mcd_psfr_scao_otf_elt(), and mcd_psfr_scao_otf_micado().

3.28.3.66 mcd_psfr_utility_transpose_image_if_required()

```
cpl_image* mcd_psfr_utility_transpose_image_if_required (
            cpl_image * in,
            cpl_size expected_nx,
            cpl_size expected_ny )
```

Tries to fix (transpose) the given image if x/y are swapped, fails if x/y do not match entirely.

Parameters

in	Image to eventually transpose
expected_nx	Expected size nx or 0 (any size)
expected_ny	Expected size ny or 0 (any size)

Returns

Fixed image

Definition at line 783 of file mcd_psfr_utility.c.

References mcd_psfr_cpl_image_transpose().

3.28.3.67 mcd_psfr_utility_vector_linspace_create()

```
cpl_vector* mcd_psfr_utility_vector_linspace_create (
            cpl_size size,
            double start,
            double end )
```

Create a vector with n values from start to end.

Parameters

size	Size of the vector
start	Initial value
end	Last value

Returns

Newly created vector

Definition at line 142 of file mcd_psfr_utility.c.

Referenced by mcd_psfr_utility_imagelist_resize().

3.29 Data structures for pipeline products

Data Structures

- struct _mcd_spectrum1D_list_
  Structure to hold spectral response spectra.
- struct _mcd_image_list_
  Structure to hold a list of images.
- struct _mcd_detector_imagelist_
  Structure to hold a MICADO detector images.
- struct _mcd_table_list_
  Structure to a list of tables.

Typedefs

- typedef struct _mcd_spectrum1D_list_ mcd_spectrum1D_list

  Structure to hold spectral response spectra.
- typedef struct _mcd_image_list_ mcd_image_list

  Structure to hold a list of images.
- typedef struct _mcd_detector_imagelist_ mcd_detector_imagelist

  Structure to hold a MICADO detector images.
- typedef struct _mcd_table_list_ mcd_table_list

  Structure to a list of tables.

Functions

- mcd_spectrum1D_list * mcd_spectrum1D_list_new (char *filename, int norder)

  Create a list of hdrl_spectrum1D.
- cpl_error_code mcd_spectrum1D_list_delete (mcd_spectrum1D_list *speclist)

  Delete a list of hdrl_spectrum1D.
- mcd_image_list * mcd_image_list_new (char *filename, int nimages)

  Create a list of images.
- cpl_error_code mcd_image_list_delete (mcd_image_list *imagelist)

  Delete a list of images.
- cpl_error_code mcd_image_list_save (mcd_image_list *imlist, cpl_frameset *allframes, const cpl_parameterlist *parlist, const char *recipe, const char *procatg, const char *proclass)

  Save a list of images as a DFS-compliant pipeline product.
- mcd_detector_imagelist * mcd_detector_imagelist_new (char *filename)

  Create a list of detector images.
- cpl_error_code mcd_detector_imagelist_delete (mcd_detector_imagelist *imlist)

  Delete a list of images.
- cpl_error_code mcd_detector_imagelist_save (mcd_detector_imagelist *imlist, cpl_frameset *allframes, const cpl_parameterlist *parlist, const char *recipe, const char *procatg, const char *proclass)

  Save a list of detector images as a DFS-compliant pipeline product.
- mcd_table_list * mcd_table_list_new (char *filename, int ntables)

  Create a list of tables.
- cpl_error_code mcd_table_list_delete (mcd_table_list *tablelist)

  Delete a list of tables.
- cpl_error_code mcd_table_list_save (mcd_table_list *tablelist, cpl_frameset *allframes, const cpl_parameterlist *parlist, const char *recipe, const char *procatg, const char *proclass)

  Save a list of tables as a DFS-compliant pipeline product.

3.29.1 Detailed Description

This module defines data structures for pipeline products. Each structure contains the product data as well as the information necessary to save the product as a DFS-compliant multi-extension FITS file.

For each structure there are methods to create a new instance, to delete an instance (freeing all memory associated with it) and to save the product.

Synopsis:

```
#include "mcd_spec_datastructures.h"
```

3.29.2 Function Documentation

3.29.2.1 mcd_detector_imagelist_delete()

```
cpl_error_code mcd_detector_imagelist_delete (
            mcd_detector_imagelist * imlist )
```

Delete a list of images.

Parameters

imagelist	Pointer to the image list to be deleted

Returns

CPL_ERROR_NONE on success

Definition at line 223 of file mcd_spec_datastructures.c.

3.29.2.2 mcd_detector_imagelist_new()

```
mcd_detector_imagelist* mcd_detector_imagelist_new (
            char * filename )
```

Create a list of detector images.

list of detector imagelists (consisting of nine images of detector size)

Parameters

filename	Name of output file associated with the list

Returns

the newly created image list

Note

This list always holds nine images - the number of detectors in MICADO. mcd_image_list_new() creates a list with an arbitrary number of images.

Definition at line 200 of file mcd_spec_datastructures.c.

3.29.2.3 mcd_detector_imagelist_save()

```
cpl_error_code mcd_detector_imagelist_save (
            mcd_detector_imagelist * imlist,
            cpl_frameset * allframes,
            const cpl_parameterlist * parlist,
            const char * recipe,
            const char * procatg,
            const char * proclass )
```

Save a list of detector images as a DFS-compliant pipeline product.

Parameters

imlist	The list of images to be saved
allframes	The list of input frames for the recipe
parlist	The list of input parameters
recipe	The recipe name
procatg	The PRO.CATG of the pipeline product
proclass	The PRO.CLASS of the pipeline product

Returns

CPL_ERROR_NONE on success

Definition at line 249 of file mcd_spec_datastructures.c.

References mcd_io_save_himg().

3.29.2.4 mcd_image_list_delete()

```
cpl_error_code mcd_image_list_delete (
            mcd_image_list * imagelist )
```

Delete a list of images.

Parameters

imagelist	Pointer to the image list to be deleted

Returns

CPL_ERROR_NONE on success

Definition at line 138 of file mcd_spec_datastructures.c.

3.29.2.5 mcd_image_list_new()

```
mcd_image_list* mcd_image_list_new (
            char * filename,
            int nimages )
```

Create a list of images.

List of images

Parameters

filename	Name of output file associated with the list
nimages	Number of images in the list

Returns

the newly created image list

Definition at line 113 of file mcd_spec_datastructures.c.

3.29.2.6 mcd_image_list_save()

```
cpl_error_code mcd_image_list_save (
            mcd_image_list * imlist,
            cpl_frameset * allframes,
            const cpl_parameterlist * parlist,
            const char * recipe,
            const char * procatg,
            const char * proclass )
```

Save a list of images as a DFS-compliant pipeline product.

Parameters

imlist	The list of images to be saved
allframes	The list of input frames for the recipe
parlist	The list of input parameters
recipe	The recipe name
procatg	The PRO.CATG of the pipeline product
proclass	The PRO.CLASS of the pipeline product

Returns

CPL_ERROR_NONE on success

Definition at line 165 of file mcd_spec_datastructures.c.

References mcd_io_save_var_himg().

3.29.2.7 mcd_spectrum1D_list_delete()

```
cpl_error_code mcd_spectrum1D_list_delete (
            mcd_spectrum1D_list * speclist )
```

Delete a list of hdrl_spectrum1D.

Parameters

speclist	Pointer to mcd_spectrum1D_list to be deleted

Returns

CPL_ERROR_NONE on success

Definition at line 89 of file mcd_spec_datastructures.c.

3.29.2.8 mcd_spectrum1D_list_new()

```
mcd_spectrum1D_list* mcd_spectrum1D_list_new (
            char * filename,
            int norder )
```

Create a list of hdrl_spectrum1D.

list of hdrl_spectrum1D

Parameters

filename	the output filename to save the list
norder	number of spectra in the list

Returns

the newly created mcd_spectrum1D_list

Definition at line 65 of file mcd_spec_datastructures.c.

3.29.2.9 mcd_table_list_delete()

```
cpl_error_code mcd_table_list_delete (
            mcd_table_list * tablelist )
```

Delete a list of tables.

Parameters

tablelist	Pointer to the table list to be deleted

Returns

CPL_ERROR_NONE on success

Definition at line 306 of file mcd_spec_datastructures.c.

3.29.2.10 mcd_table_list_new()

```
mcd_table_list* mcd_table_list_new (
            char * filename,
            int ntables )
```

Create a list of tables.

List of tables

Parameters

filename	Name of output file associated with the list
ntables	Number of tables in the list

Returns

the newly created table list

Definition at line 282 of file mcd_spec_datastructures.c.

3.29.2.11 mcd_table_list_save()

```
cpl_error_code mcd_table_list_save (
            mcd_table_list * tablelist,
            cpl_frameset * allframes,
            const cpl_parameterlist * parlist,
            const char * recipe,
            const char * procatg,
            const char * proclass )
```

Save a list of tables as a DFS-compliant pipeline product.

Parameters

tablelist	The list of tables to be saved

Parameters

allframes	The list of input frames for the recipe
parlist	The list of input parameters
recipe	The recipe name
procatg	The PRO.CATG of the pipeline product
proclass	The PRO.CLASS of the pipeline product

Returns

CPL_ERROR_NONE on success

Definition at line 332 of file mcd_spec_datastructures.c.

References mcd_io_save_var_table().

3.30 Miscellaneous Utilities

Functions

- const char * mcd_get_license (void)

  Get the pipeline copyright and license.
- cpl_image * mcd_extract_image (const cpl_frame *in_frame, int center)

  Extract an image from a frame.
- double mcd_compute_quality (cpl_image *in_ima)

  Compute the image quality.
- cpl_frameset * mcd_extract_frameset (const cpl_frameset *frameset, const char *tag)

  Extract frame(s) with the given tag from a frameset.
- cpl_error_code mcd_check_and_set_groups (cpl_frameset *frameset)

  check the entries in the recipe and classify the frameset with the tags

3.30.1 Detailed Description

3.30.2 Function Documentation

3.30.2.1 mcd_check_and_set_groups()

```
cpl_error_code mcd_check_and_set_groups (
            cpl_frameset * frameset )
```

check the entries in the recipe and classify the frameset with the tags

Parameters

frameset	input set of frames

Returns

cpl_error_code

Definition at line 185 of file mcd_utils.c.

References mcd_dfs_set_groups().

Referenced by mcd_psfr_scao_final(), mcd_psfr_scao_otf_elt(), mcd_psfr_scao_otf_micado(), mcd_psfr_scao_↩otf_par(), and mcd_psfr_scao_otf_perp().

3.30.2.2 mcd_compute_quality()

```
double mcd_compute_quality (
            cpl_image * in_ima )
```

Compute the image quality.

Parameters

Input	image

Returns

The Quality parameter

The function computes the median of the input image Describe here

- the function algorithm
- How the parameters are used - their validity range
- What needs to be deallocated after the function call

Definition at line 129 of file mcd_utils.c.

3.30.2.3 mcd_extract_frameset()

```
cpl_frameset* mcd_extract_frameset (
            const cpl_frameset * frameset,
            const char * tag )
```

Extract frame(s) with the given tag from a frameset.

Parameters

frameset	A non-empty frameset
tag	The tag for extracted frameset

Returns

New frameset containing frames with the given tag or NULL on error.

Definition at line 146 of file mcd_utils.c.

3.30.2.4 mcd_extract_image()

```
cpl_image* mcd_extract_image (
            const cpl_frame * in_frame,
            int center )
```

Extract an image from a frame.

Parameters

Input	frame
center	- if 0 returns the whole image, otherwise only the center

Returns

The newly created extracted image

Describe here

- the function algorithm
- How the parameters are used - their validity range
- What needs to be deallocated after the function call

Definition at line 86 of file mcd_utils.c.

3.30.2.5 mcd_get_license()

```
const char* mcd_get_license (
            void  )
```

Get the pipeline copyright and license.

Returns

The copyright and license string

The function returns a pointer to the statically allocated license string. This string should not be modified using the returned pointer.

Definition at line 50 of file mcd_utils.c.

Chapter 4

Data Structure Documentation

4.1 _mcd_detector_imagelist_ Struct Reference

Structure to hold a MICADO detector images.

`#include <mcd_spec_datastructures.h>`

4.1.1 Detailed Description

Structure to hold a MICADO detector images.

Definition at line 75 of file mcd_spec_datastructures.h.

The documentation for this struct was generated from the following file:

- mcd_spec_datastructures.h

4.2 _mcd_image_list_ Struct Reference

Structure to hold a list of images.

`#include <mcd_spec_datastructures.h>`

4.2.1 Detailed Description

Structure to hold a list of images.

Definition at line 59 of file mcd_spec_datastructures.h.

The documentation for this struct was generated from the following file:

- mcd_spec_datastructures.h

4.3 _mcd_spectrum1D_list_ Struct Reference

Structure to hold spectral response spectra.

`#include <mcd_spec_datastructures.h>`

4.3.1 Detailed Description

Structure to hold spectral response spectra.

Definition at line 43 of file mcd_spec_datastructures.h.

The documentation for this struct was generated from the following file:

- mcd_spec_datastructures.h

4.4 _mcd_table_list_ Struct Reference

Structure to a list of tables.

`#include <mcd_spec_datastructures.h>`

4.4.1 Detailed Description

Structure to a list of tables.

Definition at line 90 of file mcd_spec_datastructures.h.

The documentation for this struct was generated from the following file:

- mcd_spec_datastructures.h

4.5 mcd_psfr_anisoplanatic_parameters Struct Reference

Structure mcd_psfr_anisoplanatic_parameters.

`#include <mcd_psfr_parameters.h>`

4.5.1 Detailed Description

Structure mcd_psfr_anisoplanatic_parameters.

Definition at line 61 of file mcd_psfr_parameters.h.

The documentation for this struct was generated from the following file:

- mcd_psfr_parameters.h

4.6 mcd_psfr_atmosphere Struct Reference

Structure mcd_psfr_atmosphere.

`#include <mcd_psfr_parameters.h>`

4.6.1 Detailed Description

Structure mcd_psfr_atmosphere.

Definition at line 51 of file mcd_psfr_parameters.h.

The documentation for this struct was generated from the following file:

- mcd_psfr_parameters.h

4.7 mcd_psfr_box Struct Reference

Bounding box representation.

`#include <mcd_psfr_box.h>`

4.7.1 Detailed Description

Bounding box representation.

Definition at line 31 of file mcd_psfr_box.h.

The documentation for this struct was generated from the following file:

- mcd_psfr_box.h

4.8 mcd_psfr_domain Struct Reference

Represent an interval [x0, x0+length], uniformly discretize with n_points points.

`#include <mcd_psfr_domain.h>`

4.8.1 Detailed Description

Represent an interval [x0, x0+length], uniformly discretize with n_points points.

Parameters

n_points	Number of degrees of freedoms (== number of vertices) per direction
pixel_size	Mesh size: spacing between two vertices
x0	Position of the first vertex
length	Size of domain: length = pixel_size $*$ (n_points - 1)

Definition at line 42 of file mcd_psfr_domain.h.

The documentation for this struct was generated from the following file:

- mcd_psfr_domain.h

4.9 mcd_psfr_gridfunction Struct Reference

Structure mcd_psfr_gridfunction.

`#include <mcd_psfr_grid_function.h>`

4.9.1 Detailed Description

Structure mcd_psfr_gridfunction.

Represent a Q1 (element-wise bilinear) function with N degrees of freedom. The mesh is given by vertices (x_{ij}, y_{ij}) = (x_0, y_0) + (i, j)$*$h, for i,j = 0, ..., N-1 The total length of the support is L = h $*$ (N-1), since we have one element less than vertices.

Basis-functions are indexed by (ij), given by piecewise bilinear functions defined as \phi_{ij}(x,y) such that \phi_↩{ij}(x_k,y_l) = \delta_{ik} \delta_{jl}. Hence, each basis-function \phi_{ij} has support in [x_{i-1}, x_{i+1}] x [y_{j-1}, y_{j+1}].

The represented function is given by f(x,y) = \sum_{ij} \phi_{ij}(x,y) coeff_{ij}

The value at gridpoints is given by the coefficients as such F(x_i, y_j) = f_{ij}.

Definition at line 53 of file mcd_psfr_grid_function.h.

The documentation for this struct was generated from the following file:

- mcd_psfr_grid_function.h

4.10 mcd_psfr_index_pair Struct Reference

Structure mcd_psfr_index_pair.

`#include <mcd_psfr_utility.h>`

4.10.1 Detailed Description

Structure mcd_psfr_index_pair.

CPL index scheme for images. Arguments are 1-based analogously to cpl_image.

Definition at line 134 of file mcd_psfr_utility.h.

The documentation for this struct was generated from the following file:

- mcd_psfr_utility.h

4.11 mcd_psfr_layer Struct Reference

Structure mcd_psfr_layer.

`#include <mcd_psfr_parameters.h>`

4.11.1 Detailed Description

Structure mcd_psfr_layer.

Definition at line 36 of file mcd_psfr_parameters.h.

The documentation for this struct was generated from the following file:

- mcd_psfr_parameters.h

4.12 mcd_psfr_otf Struct Reference

Manage different parts of the OTF.

`#include <mcd_psfr_otf.h>`

4.12.1 Detailed Description

Manage different parts of the OTF.

Definition at line 34 of file mcd_psfr_otf.h.

The documentation for this struct was generated from the following file:

- mcd_psfr_otf.h

4.13 mcd_psfr_parameters Struct Reference

Structure mcd_psfr_parameters.

`#include <mcd_psfr_parameters.h>`

4.13.1 Detailed Description

Structure mcd_psfr_parameters.

Definition at line 84 of file mcd_psfr_parameters.h.

The documentation for this struct was generated from the following file:

- mcd_psfr_parameters.h

Index