

5.6. *CBFlib*: AN ANSI C LIBRARY FOR MANIPULATING IMAGE DATA

and *gain_esd*. The value of *_array_intensities.overload* can be retrieved or set as *overload*. The value of *_diffrn_scan_frame.integration_time* can be retrieved or set as *integration_time*.

Timestamp calls operate on the DATE entry in the DIFFRN_SCAN_FRAME category:

```
int cbf_get_timestamp (cbf_handle handle,
    unsigned int reserved, double *time,
    int *timezone);
int cbf_set_timestamp (cbf_handle handle,
    unsigned int reserved, double time,
    int timezone, double precision);
int cbf_get_datestamp (cbf_handle handle,
    unsigned int reserved, int *year, int *month,
    int *day, int *hour, int *minute, double *second,
    int *timezone);
int cbf_set_datestamp (cbf_handle handle,
    unsigned int reserved, int year, int month,
    int day, int hour, int minute, double second,
    int timezone, double precision);
int cbf_set_current_timestamp (cbf_handle handle,
    unsigned int reserved, int timezone)
```

cbf_get_timestamp and *cbf_set_timestamp* measure time in seconds since 1 January 1970. *cbf_get_datestamp* and *cbf_set_datestamp* work in terms of individual year, month, day, hour, minute and second. The optional collection time zone, *timezone*, is the difference from universal time in minutes; *precision* is the fraction, in seconds, to which the time will be recorded. *cbf_set_current_timestamp* sets the collection timestamp from the current time, to the nearest second.

Also in the DIFFRN_SCAN_FRAME category is the integration time of the image:

```
int cbf_get_integration_time (cbf_handle handle,
    unsigned int reserved, double *time);
int cbf_set_integration_time (cbf_handle handle,
    unsigned int reserved, double time);
```

Finally, these functions include routines for working with binary images:

```
int cbf_get_image_size (cbf_handle handle,
    unsigned int reserved,
    unsigned int element_number,
    size_t *ndim1, size_t *ndim2);
int cbf_get_image (cbf_handle handle,
    unsigned int reserved,
    unsigned int element_number,
    void *array, size_t elsize, int elsign,
    size_t ndim1, size_t ndim2);
int cbf_set_image (cbf_handle handle,
    unsigned int reserved,
    unsigned int element_number,
    unsigned int compression, void *array,
    size_t elsize, int elsign, size_t ndim1,
    size_t ndim2);
```

cbf_get_image_size retrieves the dimensions of detector element *element_number* from the ARRAY_STRUCTURE_LIST category, setting *ndim1* and *ndim2* to the slow and fast array dimensions, respectively. These dimensions can be used to allocate memory before calling *cbf_get_image*. *cbf_get_image* reads the image data from detector element *element_number* into a signed or unsigned integer array of size *ndim1 * ndim2* and *cbf_set_image* associates image data with a detector element. As in the description of the integer array functions, the compression argument can currently take one of three values: CBF_CANONICAL, for canonical-code com-

pression (see Section 5.6.3.1); CBF_PACKED, for CCP4-style packing (see Section 5.6.3.2); or CBF_NONE, for no compression.

5.6.2.4. Goniometer geometry functions

A CBF/imgCIF file includes a geometric description of the goniometer used to orient the sample during the experiment. Practical use of this information, however, is not trivial as it involves combining data from several categories and analysing in three dimensions the nested axes in which the description is framed (see Section 3.7.3 for a discussion of the axis system). *CBFlib* provides six functions to facilitate this task:

```
int cbf_construct_goniometer (cbf_handle handle,
    cbf_goniometer *goniometer);
int cbf_free_goniometer (cbf_goniometer goniometer);
int cbf_get_rotation_axis (cbf_goniometer goniometer,
    unsigned int reserved, double *vector1,
    double *vector2, double *vector3);
int cbf_get_rotation_range (cbf_goniometer goniometer,
    unsigned int reserved, double *start,
    double *increment);
int cbf_rotate_vector (cbf_goniometer goniometer,
    unsigned int reserved, double ratio,
    double initial1, double initial2, double initial3,
    double *final1, double *final2, double *final3);
int cbf_get_reciprocal (cbf_goniometer goniometer,
    unsigned int reserved, double ratio,
    double wavelength, double real1, double real2,
    double real3, double *reciprocal1,
    double *reciprocal2, double *reciprocal3);
```

cbf_construct_goniometer uses the data in the categories DIFFRN_MEASUREMENT, DIFFRN_MEASUREMENT_AXIS, AXIS, DIFFRN_SCAN_FRAME_AXIS and DIFFRN_SCAN_AXIS to construct a geometric representation of the goniometer and initializes the *cbf_goniometer handle, goniometer*. *cbf_free_goniometer* frees the goniometer structure. *cbf_get_rotation_axis* and *cbf_get_rotation_range* get the normalized rotation vector, and the starting value and increment of the first rotating axis of the goniometer, respectively. The *cbf_rotate_vector* call applies the goniometer axis rotation to the given initial vector, with the *ratio* value specifying the goniometer setting from 0.0 at the beginning of the exposure to 1.0 at the end, irrespective of the actual rotation range. Finally, *cbf_get_reciprocal* transforms the given real-space vector (*real1, real2, real3*) to the corresponding reciprocal-space vector (*reciprocal1, reciprocal2, reciprocal3*). As before, the transform corresponds to the goniometer initial position with a *ratio* of 0.0 and the goniometer final position with a *ratio* of 1.0.

5.6.2.5. Detector geometry functions

In a similar manner, a CBF/imgCIF file includes a description of the surface of each detector and the arrangement of the pixels in space. *CBFlib* provides eight functions for analysing this description:

```
int cbf_construct_detector (cbf_handle handle,
    cbf_detector *detector,
    unsigned int element_number);
int cbf_free_detector (cbf_detector detector);
int cbf_get_beam_center (cbf_detector detector,
    double *index1, double *index2,
    double *center1, double *center2);
int cbf_get_detector_distance (cbf_detector detector,
    double *distance);
```