

dtfind

SBC01-????-00

April 26, 1995

Prepared by: J.W. Pflugrath
Molecular Structure Corporation
3200 Research Forest Drive
The Woodlands, Texas 77381
(713) 363-1033
(713) 364-3628 FAX
jwp@msc.com

Prepared for: Contract No. 943072401
Mary Westbrook
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439
(708) 252-8914
(708) 252-4021 FAX
westbrook@anl.el.anl.gov

Reviewed by:

Approved by:

Copyright © 1995, 1994 Molecular Structure Corporation
RESTRICTED RIGHT NOTICE SHORT FORM (JUNE 1987)

Use, reproduction, or disclosure is subject to restrictions set forth in Contract No. W-31
109-ENG-38 and Contract No. 943072401 with the University of Chicago, Operator of
Argonne National Laboratory

TABLE OF CONTENTS

1 Scope and Purpose.....	3
1.1 Definitions and Abbreviations.....	3
2 Background.....	5
3 Input Images.....	6
3.1 Scan image header keywords.....	6
3.2 Image header keywords.....	7
3.3 Non-uniformity image header keywords.....	7
4 Running dtfind.....	8
4.1 Command line options.....	8
4.2 Examples.....	10
4.3 Results.....	10
4.4 Overcoming difficulties.....	12
5 Accessory programs.....	12
6 File formats.....	15
6.1 Reflection file format.....	15

1 Scope and Purpose

This document describes how to use the dtfind module of d*TREK. dtfind finds peaks in detector images, determines their centroids and writes the results to an output file.

1.1 Definitions and Abbreviations

Image An **image** is the output of a position-sensitive detector. It has a header that describes the image and how it was made. dtfind reads image headers to get information about images, so that users do not need to describe image to the program.

Dark image A dark image is an image exposed with the shutter closed with no photons events recorded. It records the dark current counts in a detector. It may be required to correct images for non-uniformity of response if they have dark current. The dark current in an image is both time and temperature dependent.

DC offset image A DC offset image is an image which shows the direct current offset of the analog-to-digital convert. It does not contain any dark current. It should be neither time nor temperature dependent.

Non-uniformity image A non-uniformity image contains values that are used to scale other images on a pixel by pixel basis. The non-uniformity image also designates bad pixels. The scale factor can be contained directly in the image or it can be calculated from the pixel value and another number. The other number is designated in the header of the non-uniformity image with the keyword NONUNF_DENOMINATOR if it is the denominator of the scale factor or NONUNF_NUMERATOR if it is the numerator of the scale factor.

Rotation A **rotation** describes the goniometer movement while an image is exposed. It has the following properties:

Start angle	The angle of the rotating axis at the start of exposure
End angle	The angle of the rotating axis at the end of exposure
Increment	The difference between the Start angle and End angle
Time	The exposure time in seconds of the image
Oscillations	The number of time the axis oscillated from the Start angle to the End angle back to the Start angle. A simple rotation will have Oscillations equal to 0.

Dark	The number of dark images collected before the start of the rotation and used for non-uniformity correction.								
DarkUpdate	The dark image update interval. After this number of images, a new dark image should be made.								
DarkLimit	The dark image change limit. If the dark current changes by more than this fraction, then a new dark image should be collected.								
DCoffset	The number of DC offset images collected before the start of the rotation and used for non-uniformity correction.								
DC Update	The DC offset image update interval. After this number of images, a new DC offset image should be made. This should probably always be 0.								
Scan	<p>A scan is a set of contiguous (in rotation angle) images with similar properties. The images are generated by a single detector so they have the same dimensions, exposure time, angular increment and non-uniformity of response. A scan has the same global properties as described for Rotation, but the Scan End Angle is the end angle of the last image in the scan. In addition to the Scan Rotation properties, a scan has a file naming template and sequence info:</p> <table><tr><td>Template</td><td>A filename with number signs (#) which will be filled in by the sequence numbers of images contained in the scan to generate the image filenames.</td></tr><tr><td>Sequence</td><td>The start sequence number, the end sequence number and the sequence number increment for images in the scan. The sequence number for the ninth image in the scan is then Start + (8 * increment). This is used along with the Template to create the image filename.</td></tr><tr><td>Axis name</td><td>The name of the goniostat rotation axis that is rotated.</td></tr><tr><td>Axis reference</td><td>The number of axes of the crystal goniostat, the axis number of the rotation axis and the angles of the goniostat when the Rotation Start angle is 0 (whether any images were collected at that position or not).</td></tr></table>	Template	A filename with number signs (#) which will be filled in by the sequence numbers of images contained in the scan to generate the image filenames.	Sequence	The start sequence number, the end sequence number and the sequence number increment for images in the scan. The sequence number for the ninth image in the scan is then Start + (8 * increment). This is used along with the Template to create the image filename.	Axis name	The name of the goniostat rotation axis that is rotated.	Axis reference	The number of axes of the crystal goniostat, the axis number of the rotation axis and the angles of the goniostat when the Rotation Start angle is 0 (whether any images were collected at that position or not).
Template	A filename with number signs (#) which will be filled in by the sequence numbers of images contained in the scan to generate the image filenames.								
Sequence	The start sequence number, the end sequence number and the sequence number increment for images in the scan. The sequence number for the ninth image in the scan is then Start + (8 * increment). This is used along with the Template to create the image filename.								
Axis name	The name of the goniostat rotation axis that is rotated.								
Axis reference	The number of axes of the crystal goniostat, the axis number of the rotation axis and the angles of the goniostat when the Rotation Start angle is 0 (whether any images were collected at that position or not).								

Axis direction The 3D vectorial representation of the rotation axis in world coordinates. The axis goes from the crystal position (0,0,0) in the Axis direction direction such that a right-hand rotation is the positive direction. For example, if the omega axis points along X, then Axis direction is (1, 0, 0). General axes may be easily specified this way.

Non-uniformity correction

Many detectors exhibit some kind of non-uniformity of pixel response which must be corrected when the images are analyzed. This is also used to flag bad pixels on the detector. There are several possible kinds of correction for this:

None There are no bad pixels and no non-uniformity of response. There is no dark current or DC offset that needs to be considered when processing images.

Simple_scale

Each pixel is scaled by a constant image-wide factor. A non-uniformity image designates bad pixels.

Dark_only

A dark image is subtracted from each image, then a non-uniformity image that contains individual pixel scale factors is used to scale the result.

Dark_dcoffset

A DC offset image is subtracted from the dark image, the dark image is corrected for possible changes in the dark current, then the DC offset image is added back to the dark image. This is subtracted from each image, then a non-uniformity image that contains individual pixel scale factors is used to scale the result.

2 Background

One of the steps undertaken when analyzing single crystal diffraction images from position-sensitive detectors is the determination of the crystal properties so that the pattern of diffracted reflections can be accurately predicted. The observed centroids of a few dozen reflections with unknown Miller indices can be used to calculate the crystal cell parameters and the crystal orientation -- provided the detector properties, the source

properties and the image properties are accurately known. The `dtfind` module finds reflection centroids in one or more images and writes the pixel coordinates and the angular position of the reflections to an output file. All information that `dtfind` needs is found in the headers of the images except for the name of the scan or name of the images themselves.

The criteria for finding peaks in images is simple, but also sophisticated enough to avoid finding noise peaks. The basic criteria is that a peak is represented by a set of contiguous pixels that are significantly above the local background. This is explained in more detail below.

3 Input images

The `dtfind` requires one or more images as input. These images are the non-uniformity calibration images and the images to be searched for peaks. The non-uniformity images are specified either in the header of the first image searched or in the header of an image given on the command line after the `-nonunf` keyword (see Section 4.1.1).

Since it is assumed by `dtfind` that the spatial distortion of the detector does not vary drastically in a region as small as a reflection, the spatial distortion information is not required nor used by `dtfind`.

3.1 Scan image header keywords

The following keywords are required in the header of the input scan image specified on the command line with the `-scan` option:

`SCAN_ROTATION` followed by 10 values for the *SCAN* Rotation as described in 1.1.

`SCAN_TEMPLATE` followed by a single Template string.

`SCAN_SEQ_INFO` followed by the 3 scan sequence numbers.

`ROTATION` followed by 10 values for the *IMAGE* Rotation as described in 1.1. This should be the same as the `SCAN_ROTATION`, except that Start angle and End angle will reflection the values for the image, not the overall lscan.

3.2 Image header keywords

The following keyword are required in the header of the input images specified on the command line with the `-image` option:

ROTATION followed by 10 values for the Rotation as described in 1.1.

3.3 Non-uniformity image header keywords

Non-uniformity information is specified in image headers. `dtfind` looks for this information in the image specified by the `-scan`, `-image` or `-nonunf` command line options. Of the following keywords, `NONUNF_TYPE` and `NONUNF_INFO` are found in all images, while the others are found in the non-uniformity image itself.

`NONUNF_TYPE` followed by the type of non-uniformity correction to perform on the images which is one of the following:
None
Simple_scale
Dark_only
Dark_dcoffset

`NONUNF_INFO` depends on the type, but holds up to 3 filenames in the order: non-uniformity image, dark current image, Dcoffset image.

`NONUNF_SCALE` A scale factor in `NONUNF_TYPE=Simple_scale`

`NONUNF_DENOMINATOR`
A denominator if `NONUNF_TYPE = Dark_only` or `Dark_dcoffset`

`NONUNF_NUMERATOR`
A numerator if `NONUNF_TYPE = Dark_only` or `Dark_dcoffset`

`NONUNF_FLAG1`

`NONUNF_FLAG2`

`NONUNF_FLAG3`

`NONUNF_FLAG4` These list 4 pixel values that designate 4 kinds of bad pixels.

4 Running dtfind

After dtfind has been installed and placed in your PATH, just enter dtfind along with command line options to run it. It parses the command line arguments, reads any required input images and writes a reflection file if peaks are found. Messages are written to stdout and stderr as required. The syntax for running dtfind is:

```
dtfind [-scan scanfile | -image imagefile] [ options ...]
```

4.1 Command line options

Although dtfind will run without any command line options if scan information is found in the header of a file called scan.img in the current working directory, most situations will benefit from the use of one or more command line options described below.

Command line options: Description and default

- `-scan sName` Image sName (no default) has the scan definition. The search method will be 3D unless overridden by the `-2D` option. If neither a `-scan` option nor a `-image` option is specified, then dtfind uses `'-scan scan.img'` as the default.
- `-image sName` Image sName (no default) will be searched. Up to 20 `-image` options may be on the command line.
- `-sigma fSigma` Standard deviation (default 5.0) above background for pixels to be consider for peaks. This is tested AFTER any non-uniformity correction is applied.
- `-min fMin` Minimum pixel value to be considered for peaks. This is tested AFTER any non-uniformity correction is applied. The default is 0.
- `-circle nCen0 nCen1 nMinrad nMaxrad` Only pixels within a circle of nMaxrad and outside a circle of nMinrad pixels will be considered for peaks. The circles are center at nCen0, nCen1. The default settings are the image center and radii to search the entire image.

- `-rect nMinPx0 nMinPx1 nExtPx0 nExtPx1`
Only pixels within a rectangle with origin at `nMinPx0`, `nMinPx1` and extents `nExtPx0`, `nExtPx1` will be considered for peaks. The default settings are the entire image less a border of 1% of the image extent.
- `-window nWin0 nWin1`
Size in pixels for a window of pixels around a peak. For a 2D search, the default values of 0, 0 make `dtfind` dynamically find a suitable size. For a 3D search, the default is 21,21.
- `-filter nPeakFilter`
In a 3x3 box, `nPeakFilter` pixels must be above the threshold which is the higher of the value specified by the `-fSigma` or `-fMinimum` option.
- `-brect nBack0 nBack1`
If `-sigma fSigma > 0.0`, then a local background average and standard deviation are determined in a small rectangle of `nBack0` by `nBack1` pixels. The default value is 1/8th the image dimension in each direction, dividing the image in 64 areas
- `-pad nPad3D` If the 3D method is used (`-scan sName`) then peaks found in a single image by a 2D search have `nPad3D` (default 3) images added to either side.
- `-nonunf sNonunfFile`
Specifies an alternate file with non-uniformity information to use. The default info comes from the `-scan` file or the first `-image` file. This option overrides the information in these files.
- `-ref sReflectionFileName`
Specifies an alternate output reflection file instead of the default `dtfind.ref`. The file format is a standard d*TREK reflection file format described in section 7.
- `-seq nStart nEnd`
Specifies an alternate starting and ending image sequence values in a scan. The default is the first image of the scan and 9 more images.
- `-num nNumImages`
Specifies an alternate number of images to search in a scan.
- `-2D` Uses a 2D instead of a 3D search method when `-scan` is specified. Use this option when consecutive wide angle images are to be searched for peaks.

4.2 Examples

```
dtfind -scan ribozyme.0003 -num 40
```

The above command line reads image `ribozyme.0003` for scan information and overrides the default value of searching the first 10 images by requesting to search the first 40 images of the scan. Note that the first image to search is not necessarily `ribozyme.0003`, that is just the image whose header contains the scan information. Any non-uniformity correction information is found in the header of image `ribozyme.0003` which in turn points to the non-uniformity files. Results are written to the file `dtfind.ref`.

```
dtfind -scan hivrt.0001 -seq 2 50
```

The above command line reads image `hivrt.0001` for scan information and overrides the default value of searching the first 10 images by requesting to search images from sequence number 2 through 50.

```
dtfind -image hivrt.0002 -image hivrt.0003 -sigma 3 -ref hivrt.ref;
```

The above command line reads image `hivrt.0002` for non-uniformity correction information, the searches images `hivrt.0002` and `hivrt.0003` with a sigma above background cutoff of 3. It writes the reflection file `hivrt.ref` instead of `dtfind.ref`.

4.3 Results

The command:

```
dtfind -scan 1 -nonunf n2b.img -pad 2 -num 40
```

produced the following output below. The non-uniformity information in the file 1 is incorrect. The correct non-uniformity information is found in file `n2b.img` so this was specified on the command line. The scan contained only 35 images, not 40 as requested on the command line. An error occurred after reading the 35th image, but this did not prevent `dtfind` from finding 67 reflections.

```
1 File 1 successfully opened.
2 Header of file 1 successfully read.
3 Image is 100 by 200 pixels.
4 Data_type in header is short int.
5 Compression_type is None.
6 Byte_order is big_endian.
7 Cscan::nInitValues called
8 NONUNF_TYPE: >>Dark_only<<
9 Cnonunf: error reading Nonunf and Dark file names from header!
10 Cnonunf::nInitValues called
```

```
11  Command line string: >>-nonunf<<
12  File n successfully opened.
13  Header of file n successfully read.
14  Image is 100 by 200 pixels.
15  Data_type in header is short int.
16  Compression_type is None.
17  Byte_order is big_endian.
18  Cnonunf::~destructor called!
19  NONUNF_TYPE: >>Dark_only<<
20  Cnonunf::nInitValues called

21  File n2b.img successfully opened.
22  Header of file n2b.img successfully read.
23  Image is 100 by 200 pixels.
24  Data_type in header is short int.
25  Compression_type is None.
26  Byte_order is big_endian.

27  File lyso3.0000_dk1.ss successfully opened.
28  Header of file lyso3.0000_dk1.ss successfully read.
29  Image is 100 by 200 pixels.
30  Data_type in header is short int.
31  Compression_type is None.
32  Byte_order is big_endian.
33  There were 20000 pixels, with 0 bad pixels in the nonunf file.
34  Command line string: >>-pad<<
35  Command line string: >>-num<<
36  Find object listing:
37  Sigma: 5
38  Minimum: 0
39  Circle lim: 50, 100, 0, 111
40  Rect lim: 1, 99, 2, 198
41  Spot wind.: 0, 0
42  Near: 15
43  Far: 222
44  Peak filt.: 6
45  Back. tile: 12, 25
46  Pad: 2
47  Seq. num.: 0, 78
48  Image dim.: 100, 200

49  dtfind: 3D method used
50  !***ERROR in DSKBOR, could not open file lyso3.0070.ss

51  File lyso3.0000.ss successfully opened.
52  Header of file lyso3.0000.ss successfully read.
53  Image is 100 by 200 pixels.
54  Data_type in header is short int.
55  Compression_type is None.
56  Byte_order is big_endian.

57  ...
58  File lyso3.0068.ss successfully opened.
```

```
59  Header of file lyso3.0068.ss successfully read.
60  Image is 100 by 200 pixels.
61  Data_type in header is short int.
62  Compression_type is None.
63  Byte_order is big_endian.
64  Error opening file lyso3.0070.ss!***ERROR in DSKBOR, could not open
    file lyso3.0070.ss
65  Error is: -2

66  File lyso3.0000.ss successfully opened.
67  Header of file lyso3.0000.ss successfully read.
68  Image is 100 by 200 pixels.
69  Data_type in header is short int.
70  Compression_type is None.
71  Byte_order is big_endian.

72  ...

73  File lyso3.0068.ss successfully opened.
74  Header of file lyso3.0068.ss successfully read.
75  Image is 100 by 200 pixels.
76  Data_type in header is short int.
77  Compression_type is None.
78  Byte_order is big_endian.
79  Error opening file lyso3.0070.ss Error is: -2

80  dtfind: There were 67 spots found.
81  dtfind: Spots written to dtfind.ref
```

4.4 Overcoming difficulties

Difficulties may arise when the non-uniformity information is either missing or not contained in the image header. In this case, add or correct the information (use `dtheadredit`, see section 5) and run `dtfind` again. The output of `dtfind` clearly shows when the non-uniformity information is not valid.

If too many peaks are found, increase the Sigma value with the `-sigma` option.

If too few peaks are found, decrease the Sigma value with the `-sigma` option. Or search more images by changing `-seq` or `-num`.

5 Accessory programs

Since `dtfind` gets all image information from the image header, it may be useful to edit image headers from time to time. Since headers are ASCII characters, use a text editor such as GNU `emacs` to do this. The program `dtheadredit` may also be used. It keeps track of the length of the header and prevents the modification of keywords that describe the binary data in the image file. Input to `dtheadredit` comes from `stdin` and consists of

the name of the input image file, the name of the output image file, then keyword value pairs until an end-of-file is found on the stdin. Separate keyword and value by a whitespace (not an equal sign). Any value that contains whitespace must be enclosed in single quotes.

If a keyword exists in the header, the value is replaced. If the keyword does not exist, the keyword=value; pair is added. To delete a keyword=value; pair enter the keyword followed by 2 single quotes: **keyword ''**.

Here is an example:

```
1  % dtheaderedit
2  dtheaderedit: Copyright (c) 1995 Molecular Structure Corporation
3  dtheaderedit: Enter input and output image names: 1 1.out
4  Cimage::nRead filename is 1
5  File 1 successfully opened.
6  Header of file 1 successfully read.
7  Image is 100 by 200 pixels.
8  Data_type in header is short int.
9  Compression_type is None.
10 Byte_order is big_endian.
11 Header listing::
12 {
13  HEADER_BYTES= 1024;
14  TYPE=mad;
15
16  SIZE1=100;
17  SIZE2=200;
18  COMMENT=Header edited by dtheaderedit;
19  SCAN_TITLE=start end inc time nOsc nDark nDup nDlim nDC nDCup;
20  SCAN_ROTATION=0.0 12.0 0.2 4 0 1 0 100 1 0;
21  SCAN_TEMPLATE=lyso3.####.ss;
22  SCAN_SEQ_INFO=0 2 0;
23  ROTATION=0.0 0.2 0.2 4 0 1 0 100 1 0;
24  NONUNF_TYPE=Dark_only;
25  NONUNF_INFO=lyso3.0000_dk1.ss;
26  NONUNF_FLAG1=1;
27  NONUNF_FLAG2=2;
28  NONUNF_FLAG3=3;
29  NONUNF_FLAG4=4;
30  NONUNF_FLAGS=4 1 2 3 4;
31  NONUNF_DENOMINATOR=100;
32  SPATIAL_DISTORTION_TYPE=interp_table;
33  SPATIAL_DISTORTION_INFO=distor;
34  SCAN_AXIS_NAME=omega;
35  SCAN_AXIS_REFERENCE=4 1 0.0 0.0 0.0;
36  SCAN_AXIS_DIRECTION=1.0 0.0 0.0;
37  FILENAME=lyso3.0000.ss;
```

```
38  BYTE_ORDER=big_endian;
39  Data_type=short int;
40  COMPRESSION=None;
41  }
42
43  :: end of header listing.

44  dthederedit: Enter 'keyword value' pair: NONUNF_FLAG1 3

45

46  Header listing::
47  {
48  HEADER_BYTES= 1024;
49  TYPE=mad;
50
51  SIZE1=100;
52  SIZE2=200;
53  SCAN_TITLE=start end inc time nOsc nDark nDup nDlim nDC nDCup;
54  SCAN_ROTATION=0.0 12.0 0.2 4 0 1 0 100 1 0;
55  SCAN_TEMPLATE=lyso3.####.ss;
56  SCAN_SEQ_INFO=0 2 0;
57  ROTATION=0.0 0.2 0.2 4 0 1 0 100 1 0;
58  NONUNF_TYPE=Dark_only;
59  NONUNF_INFO=lyso3.0000_dk1.ss;
60  NONUNF_FLAG2=2;
61  NONUNF_FLAG3=3;
62  NONUNF_FLAG4=4;
63  NONUNF_FLAGS=4 1 2 3 4;
64  NONUNF_DENOMINATOR=100;
65  SPATIAL_DISTORTION_TYPE=interp_table;
66  SPATIAL_DISTORTION_INFO=distor;
67  SCAN_AXIS_NAME=omega;
68  SCAN_AXIS_REFERENCE=4 1 0.0 0.0 0.0;
69  SCAN_AXIS_DIRECTION=1.0 0.0 0.0;
70  FILENAME=lyso3.0000.ss;
71  BYTE_ORDER=big_endian;
72  Data_type=short int;
73  COMPRESSION=None;
74  COMMENT=Header edited by dthederedit;
75  NONUNF_FLAG1=3;

76

77  :: end of header listing.

78  dthederedit: Enter 'keyword value' pair:
79  NONUNF_INFO 'nonunf.img dark.img'

80  Header listing::
81  {
82  HEADER_BYTES= 1024;
83  TYPE=mad;
84
85  SIZE1=100;
```

```
86  SIZE2=200;
87  SCAN_TITLE=start end inc time nOsc nDark nDup nDlim nDC nDCup;
88  SCAN_ROTATION=0.0 12.0 0.2 4 0 1 0 100 1 0;
89  SCAN_TEMPLATE=lyso3.####.ss;
90  SCAN_SEQ_INFO=0 2 0;
91  ROTATION=0.0 0.2 0.2 4 0 1 0 100 1 0;
92  NONUNF_TYPE=Dark_only;
93  NONUNF_FLAG2=2;
94  NONUNF_FLAG3=3;
95  NONUNF_FLAG4=4;
96  NONUNF_FLAGS=4 1 2 3 4;
97  NONUNF_DENOMINATOR=100;
98  SPATIAL_DISTORTION_TYPE=interp_table;
99  SPATIAL_DISTORTION_INFO=distor;
100 SCAN_AXIS_NAME=omega;
101 SCAN_AXIS_REFERENCE=4 1 0.0 0.0 0.0;
102 SCAN_AXIS_DIRECTION=1.0 0.0 0.0;
103 FILENAME=lyso3.0000.ss;
104 BYTE_ORDER=big_endian;
105 Data_type=short int;
106 COMPRESSION=None;
107 COMMENT=Header edited by dtheadredit;
108 NONUNF_FLAG1=3;
109 NONUNF_INFO=nonunf.img dark.img;
110 }

111
112  :: end of header listing.

113 dtheadredit: Enter 'keyword value' pair: ^D
```

6 File formats

Image file formats are discussed in the calibrate documentation. Examples of image headers are given in Section 5.

6.1 Reflection file format

Reflection files have the following format. First they are all ASCII characters. The first line consists of 3 numbers designating the number of integer, float and string fields that each reflection has. After this are the field labels as character strings, one per line for the integer fields first, then the float fields, then the string fields. The first 3 fields of the reflection file are H, K, L. The first 2 float fields are Intensity and SigmaI. No other fields are required. Field labels must contain no whitespace. After the field labels, comes the reflection data, one reflection per line in free format. This format is a prelude to using a true CIF reflection file format.

