User’s Guide to the CCDRED Package

Francisco Valdes

IRAF Group - Central Computer Services
National Optical Astronomy Observatories††
P.O. Box 26732, Tucson, Arizona 85726

June 1987
Revised February 1988

ABSTRACT

The IRAF CCD reduction package, **ccdred**, provides tools for the easy and efficient reduction of CCD images. The standard reduction operations are replacement of bad pixels, subtraction of an overscan or prescan bias, subtraction of a zero level image, subtraction of a dark count image, division by a flat field calibration image, division by an illumination correction, subtraction of a fringe image, and trimming unwanted lines or columns. Another common operation provided by the package is scaling and combining images with a number of algorithms for rejecting cosmic rays. Data in the image header is used to make the reductions largely automated and self-documenting though the package may still be used in the absence of this data. Also a translation mechanism is used to relate image header parameters to those used by the package to allow data from a variety of observatories and instruments to be processed. This guide provides a brief description of the IRAF CCD reduction package and examples of reducing simple CCD data.

July 2, 1990

††Operated by the Association of Universities for Research in Astronomy, Inc. under cooperative agreement with the National Science Foundation.

User’s Guide to the CCDRED Package

Francisco Valdes

IRAF Group - Central Computer Services
National Optical Astronomy Observatories††
P.O. Box 26732, Tucson, Arizona 85726

June 1987
Revised February 1988

1. Introduction

This guide provides a brief description of the IRAF CCD reduction package **ccdred** and examples of reducing simple CCD data. It is a generic guide in that it is not tied to any particular type of data. There may be more specific guides (or "cookbooks") for your data. Detailed descriptions of the tasks and features of the package are provided in the help documentation for the package.

The purpose of the CCDRED package is to provide tools for the easy and efficient reduction of CCD images. The standard reduction operations are replacement of bad columns and lines by interpolation from neighboring columns and lines, subtraction of a bias level determined from overscan or prescan columns or lines, subtraction of a zero level using a zero length exposure calibration image, subtraction of a dark count calibration image appropriately scaled to the dark time exposure, division by a scaled flat field calibration image, division by an illumination image (derived from a blank sky image), subtraction of a scaled fringe image (also derived from a blank sky image), and trimming the image of unwanted lines or columns such as the overscan strip. Any set of operations may be done simultaneously over a list of images in a highly efficient manner. The reduction operations are recorded in the image header and may also be logged on the terminal and in a log file.

The package also provides tools for combining multiple exposures of object and calibration images to improve the statistical accuracy of the observations and to remove transient bad pixels. The combining operation scales images of different exposure times, adjusts for variable sky background, statistically weights the images by their signal-to-noise, and provides a number of useful algorithms for detecting and rejecting transient bad pixels.

Other tasks are provided for listing reduction information about the images, deriving secondary calibration images (such as sky corrected flat fields or illumination correction images), and easily setting the package parameters for different instruments.

There are several important features provided by the package to make the reduction of CCD images convenient; particularly to minimize record keeping. One of these is the ability to recognize the different types of CCD images. This ability allows the user to select a certain class of images to be processed or listed and allows the processing tasks to identify calibration images and process them differently from object images. The standard CCD image types are *object*, *zero* level, *dark* count, and *flat* field. For more on the image types see **ccdtypes**.

The tasks can also identify the different filters (or other subset parameter) which require

††Operated by the Association of Universities for Research in Astronomy, Inc. under cooperative agreement with the National Science Foundation.

different flat field images. This means you don't have to separate the images by filter and process each set separately. This feature is discussed further in **subsets**.

The tasks keep track of the reduction steps completed on each image and ignore images which have been processed. This feature, along with recognizing the image types and subsets, makes it possible to specify all the images to a task with a wildcard template, such as "*.imh", rather than indicating each image by name. You will find this extremely important with large sets of observations.

A fundamental aspect of the package is that the processing modifies the images. In other words, the reduction operations are performed directly on the image. This "feature" further simplifies record keeping, frees the user from having to form unique output image names, and minimizes the amount of disk space required. There are two safety features in this process. First, the modifications do not take effect until the operation is completed on the image. This allows you to abort the task without messing up the image data and protects data if the computer crashes. The second feature is that there is a package parameter which may be set to make a backup of the input data with a particular prefix such as "orig" or "imdir$". This backup feature may be used when there is sufficient disk space, when learning to use the package, or just to be cautious.

In a similar effort to efficiently manage disk space, when combining images into a master object or calibration image there is an option to delete the input images upon completion of the combining operation. Generally this is desirable when there are many calibration exposures, such as zero level or flat field images, which are not used after they are combined into a final calibration image.

The following sections guide you through the basic use of the **ccdred** package. Only the important parameters which you might want to change are described. It is assumed that the support personnel have created the necessary instrument files (see **instruments**) which will set the default parameters for the data you will be reducing. If this is not the case you may need to delve more deeply into the details of the tasks. Information about all the parameters and how the various tasks operate are given in the help documentation for the tasks and in additional special help topics. Some useful help documentation is indicated in the discussion and also in the **References** section.

2. Getting Started

The first step is to load **ccdred**. This is done by loading the **noao** package, followed by the image reduction package **imred**, and finally the **ccdred** package. Loading a package consists of typing its name. Note that some of these packages may be loaded automatically when you logon to IRAF.

When you load the **ccdred** package the menu of tasks or commands is listed. This appears as follows:

```
    cl> ccdred
      badpiximage             ccdtest                    mkfringecor
setinstrument
      ccdgroups               combine                     mkillumcor
zerocombine
      ccdhedit          cosmicrays        mkillumflat
      ccdlist           darkcombine       mkskycor
      ccdproc           flatcombine       mkskyflat
```

A summary of the tasks and additional help topics is obtained by typing:

```
cl> help
```

This list and how to get additional help on specific topics is described in the **References** section at the end of this guide.

The first command to use is **setinstrument**, which sets the package appropriately for the CCD images to be reduced. The support personnel should tell you the instrument identification, but if not a list of known instruments may be listed by using '?' for the instrument name.

```
cl> setinstrument
Instrument ID (type ? for a list) <enter instrument id or ?>
<Set ccdred package parameters using eparam>
<Set ccdproc task parameters using eparam>
```

This task sets the default parameters and then allows you to modify the package parameters and the processing parameters using the parameter editor **eparam**. If you are not familiar with **eparam** see the help or CL introduction documentation. For most terminals you move up and down through the parameters with the terminal arrow keys, you change the parameters by simply typing the desired value, and you exit with control Z or control D. Note that you can change parameters for any task at any time with **eparam** and you do not have to run **setinstrument** again, even if you logout, until you need to reduce data from a different instrument.

The **ccdred** package parameters control general I/O functions of the tasks in the package. The parameters you might wish to change are the output pixel type and the verbose option. Except when the input images are short integers, the noise is significantly greater than one digital unit, and disk space is critical, it is probably better to allow the processing to convert the images to real pixel datatype. The verbose parameter simply prints the information written to the log file on the terminal. This can be useful when little else is being done and you are just beginning. However, when doing background processing and other IRAF reduction tasks it is enough to simply look at the end of the logfile with the task **tail** to see the current state of the processing.

The **ccdproc** parameters control the CCD processing. There are many parameters but they all may be conveniently set at this point. Many of the parameters have default values set appropriately for the instrument you specified. The images to be processed can be specified later. What needs to be set are the processing operations that you want done and the parameters required for each operation. The processing operations are selected by entering yes or no for each one. The following items briefly describe each of the possible processing operations and the additional parameters required.

fixpix - Fix bad CCD lines and columns?

The bad pixels (cosmetic defects) in the detector are given in a file specified by the parameter *fixfile*. This information is used to replace the pixels by interpolating from the neighboring pixels. A standard file for your instrument may be set by **setinstrument** or if the word "image" is given then the file is defined in the instrument data file. For more on the bad pixel file see **instruments**.

overscan - Apply overscan strip correction?

The overscan or prescan region is specified by the parameter *biassec*. This is given as an IRAF image section. The overscan region is averaged along the readout axis, specified by the parameter *readaxis*, to create a one dimensional bias vector. This bias is fit by a function to remove cosmic rays and noise. There are a number of parameters at the end of the parameter list which control the fitting. The default overscan bias section and fitting parameters for your instrument should be set by **setinstrument**. If the word "image" is given the overscan bias section is defined in the image header or the instrument translation file. If an overscan section is not set you can use **implot** to determine the columns or rows for the bias region and define an overscan image section. If you are unsure about image sections consult with someone or read the introductory IRAF documentation.

trim - Trim the image?

The image is trimmed to the image section given by the parameter *trimsec*. A default trim section for your instrument should be set by **setinstrument**, however, you may override this default if desired. If the word "image" is given the data image section is given in the image header or the instrument translation file. As with the overscan image section it is straightforward to specify, but if you are unsure consult someone.

zerocor - Apply zero level correction?

The zero level image to be subtracted is specified by the parameter *zero*. If none is given then the calibration image will be sought in the list of images to be processed.

darkcor - Apply dark count correction?

The dark count image to be subtracted is specified by the parameter *dark*. If none is given then the calibration image will be sought in the list of images to be processed.

flatcor - Apply flat field correction?

The flat field images to be used are specified by the parameter *flat*. There must be one flat field image for each filter or subset (see **subsets**) to be processed. If a flat field image is not given then the calibration image will be sought in the list of images to be processed.

readcor - Convert zero level image to readout correction?

If a one dimensional zero level readout correction vector is to be subtracted instead of a two dimensional zero level image then, when this parameter is set, the zero level images will be averaged to one dimension. The readout axis must be specified by the parameter *readaxis*. The default for your instrument is set by **setinstrument**.

scancor - Convert flat field image to scan correction?

If the instrument is operated in a scan mode then a correction to the flat field may be required. There are two types of scan modes, "shortscan" and "longscan". In longscan mode flat field images will be averaged to one dimension and the readout axis must be specified. Shortscan mode is a little more complicated. The scan correction is used if the flat field images are not observed in scan mode. The number of scan lines must be specified by the parameter *nscan*. If they are observed in scan mode, like the object observations, then the scan correction operations should *not* be specified. For details of scan mode operations see **ccdproc**. The scan parameters should be set by **setinstrument**. If in doubt consult someone familiar with the instrument and mode of operation.

This description of the parameters is longer than the actual operation of setting the parameters. The only parameters likely to change during processing are the calibration image parameters.

When processing many images using the same calibration files a modest performance improvement can be achieved by keeping (caching) the calibration images in memory to avoid disk accesses. This option is available by specifying the amount of memory available for image caching with the parameter *max_cache*. If the value is zero then the images are accessed from disk as needed while if there is sufficient memory the calibration images may be kept in memory during the task execution.

3. Processing Your Data

The processing path depends on the type of data, the type of instrument, types of calibration images, and the observing sequence. In this section we describe two types of operations common in reducing most data; combining calibration images and performing the standard calibration and correction operations. Some additional special operations are described in the following section.

However, the first thing you might want to try before any processing is to get a listing of the CCD images showing the CCD image types, subsets, and processing flags. The task for this is **ccdlist**. It has three types of of output; a short one line per image format, a longer format which shows the state of the processing, and a format which prints the image names only (used to create files containing lists of images of a particular CCD image type). To get a quick listing type:

```
cl> ccdlist *.imh
ccd001.imh[544,512][short][unknown][V]:FOCUS L98-193
ccd007.imh[544,512][short][object][V]:N2968 V 600s
ccd015.imh[544,512][short][object][B]:N3098 B 500s
ccd024.imh[544,512][short][object][R]:N4036 R 600s
ccd045.imh[544,512][short][flat][V]:dflat 5s
ccd066.imh[544,512][short][flat][B]:dflat 5s
ccd103.imh[544,512][short][flat][R]:dflat 5s
ccd104.imh[544,512][short][zero][]:bias
ccd105.imh[544,512][short][dark][]:dark 3600s
```

The example shows only a sample of the images. The short format listing tells you the name of the image, its size and pixel type, the CCD image type as seen by the package, the subset identifier (in this case the filter), and the title. If the data had been processed then there would also be processing flags. If the CCD image types do not seem right then there may be a problem with the instrument specification.

Many of the tasks in the **ccdred** package have the parameter *ccdtype* which selects a particular type of image. To list only the object images from the previous example:

```
cl> ccdlist *.imh ccdtype=object
ccd007.imh[544,512][short][object][V]:N2968 V 600s
ccd015.imh[544,512][short][object][B]:N3098 B 500s
ccd024.imh[544,512][short][object][R]:N4036 R 600s
```

If no CCD image type is specified (by using the null string "") then all image types are selected. This may be necessary if your instrument data does not contain image type identifications.

3.1. Combining Calibration Images

If you do not need to combine calibration images because you only have one image of each type, you can skip this section. Calibration images, particularly zero level and flat field images, are combined in order to minimize the effects of noise and reject bad pixels in the calibrations. The basic tool for combining images is the task **combine**. There are simple variants of this task whose default parameters are set appropriately for each type of calibration image. These are the ones you will use for calibration images leaving **combine** for combining object images. Zero level images are combined with **zerocombine**, dark count images with **darkcombine**, and flat field images with **flatcombine**.

For example, to combine flat field images the command is:

```
    cl> flatcombine *.imh
    Jun  1 14:26 combine: maxreject
            Images      N    Exp   Mode  Scale Offset Weight
        ccd045.imh      1    5.0  INDEF  1.000     0.  0.048
        ccd046.imh      1    5.0  INDEF  1.000     0.  0.048
        <... list of files ...>
        ccd065.imh      1    5.0  INDEF  1.000     0.  0.048
        ----------- ------ ------
         FlatV.imh     21    5.0
```

This output is printed when verbose mode is set. The same information is recorded in the log file. In this case the flat fields are combined by rejecting the maximum value at each point in the image (the "maxreject" algorithm). The images are scaled by the exposure times, which are all the same in this example. The mode is not evaluated for exposure scaling and the relative weights are the same because the exposure times are the same. The example only shows part of the output; **flatcombine** automatically groups the flat field images by filter to produce the calibration images "FlatV", "FlatB", and "FlatR".

3.2. Calibrations and Corrections

Processing the CCD data is easy and largely automated. First, set the task parameters with the following command:

```
    cl> eparam ccdproc
```

You may have already set the parameters when you ran **setinstrument**, though the calibration image parameters *zero*, *dark*, and *flat* may still need to be set or changed. Once this is done simply give the command

```
    cl> ccdproc *.imh
    ccd003: Jun  1 15:13 Overscan section is [520:540,*] with mean=485.0
    ccd003: Jun  1 15:14 Trim data section is [3:510,3:510]
    ccd003: Jun  1 15:14 Overscan section is [520:540,*] with mean=485.0
    FlatV:  Jun  1 15:14 Trim data section is [3:510,3:510]
    FlatV:  Jun  1 15:15 Overscan section is [520:540,*] with mean=486.4
    ccd003: Jun  1 15:15 Flat field image is FlatV.imh with scale=138.2
    ccd004: Jun  1 15:16 Trim data section is [3:510,3:510]
    ccd004: Jun  1 15:16 Overscan section is [520:540,*] with mean=485.2
    ccd004: Jun  1 15:16 Flat field image is FlatV.imh with scale=138.2
                <... more ...>
    ccd013: Jun  1 15:22 Trim data section is [3:510,3:510]
    ccd013: Jun  1 15:23 Overscan section is [520:540,*] with mean=482.4
    FlatB:  Jun  1 15:23 Trim data section is [3:510,3:510]
    FlatB:  Jun  1 15:23 Overscan section is [520:540,*] with mean=486.4
    ccd013: Jun  1 15:24 Flat field image is FlatB.imh with scale=132.3
                <... more ...>
```

The output shown is with verbose mode set. It is the same as recorded in the log file. It illustrates the principle of automatic calibration image processing. The first object image, "ccd003", was being processed when the flat field image was required. Since the image was taken with the V filter the appropriate flat field was determined to be "FlatV". Since it had not been processed, the processing of "ccd003" was interrupted to process "FlatV". The processed calibration image may have been cached if there was enough memory. Once "FlatV" was processed (note that the flat field was not flattened because the task knows this image is a flat field) the processing of "ccd003" was completed. The next image, "ccd004", is also a V filter image so the already processed, and possibly cached, flat field "FlatV" is used again. The first B band image is "ccd013" and, as before, the B filter flat field calibration image is processed automatically. The same automatic calibration processing and image caching occurs when using zero level and dark count calibration images.

Commonly the processing is done with the verbose mode turned off and the task run as a background job. This is done with the commands

```
    cl> ccdred.verbose=no
    cl> ccdproc *.imh &
```

The already processed images in the input list are recognized as having been processed and are not affected. To check the status of the processing we can look at the end of the log file with:

```
    cl> tail logfile
```

After processing we can repeat the **ccdlist** command to find:

```
    cl> ccdlist *.imh ccdtype=object
    ccd007.imh[508,508][real][object][V][OTF]:N2968 V 600s
    ccd015.imh[508,508][real][object][B][OTF]:N3098 B 500s
    ccd024.imh[544,512][short][object][R][OTF]:N4036 R 600s
```

The processing flags indicate the images have been overscan corrected, trimmed, and flat fielded.

As you can see, processing images is very easy. There is one source of minor confusion for beginning users and that is dealing with calibration images. First, there is no reason that calibration images may not be processed explicitly with **ccdproc**, just remember to set the *ccdtype* to the calibration image type or to "". When processing object images the calibration images to be used may be specified either with the task parameter for the particular calibration image or by including the calibration image in the list of input images. Calibration images specified by parameter value take precedence and the task does not check its CCD image type. Calibration images given in the input list must have a valid CCD image type. In case too many calibration images are specified, say because the calibration images combined to make the master calibration images were not deleted and so are part of the image list "*.imh", only the first one will be used. Another point to know is that flat field, illumination, and fringe images are subset (filter) dependent and so a calibration image for each filter must be specified.

4. Special Processing Operations

The special processing operations are mostly concerned with the flat field response correction. There are also special processing operations available in **ccdproc** for one dimensional readout corrections in the zero level and flat field calibrations. These were described briefly above and in more detail in **ccdproc** and are not discussed further in this guide. The processing operations described in this section are for preparing flat fields for two dimensional spectroscopic data, for correcting flat fields for illuminations effects, for making a separate illumination correction, and for applying corrections for fringe effects. For additional discussion about flat fields and illumination corrections see the help topic **flatfields**.

4.1. Spectroscopic Flat Fields

For spectroscopic data the flat fields may have to be processed to remove the general shape of the lamp spectrum and to replace regions outside of the aperture where there is no flat field information with values that will not cause bad response effects when the flat field is applied to the data. If the shape of the lamp spectrum is not important and if the longslit spectra have the regions outside of the slit either off the detector or trimmed then you may use the flat field without special processing.

First you must process the flat field images explicitly with

```
cl> ccdproc *.imh ccdtype=flat
```

where "*.imh" may be replaced with any list containing the flat fields. If zero level and dark count corrections are required these calibration images must be available at this time.

Load the **twodspec** package and then either the **longslit** package, for longslit data, or the **apextract** package, for multiaperture data such as echelles, multifiber, or aperture mask spectra. The task for removing the longslit quartz spectrum is **response**. There is also a task for removing illumination effects, including the slit profile, from longslit spectra called **illumination**. For more about processing longslit spectra see the help for these tasks and the paper *Reduction of Longslit Spectra with IRAF*. The cookbook *Reduction of Longslit Spectroscopic Data Using IRAF (KPNO ICCD and Cryogenic Camera Data)* also provides a very good discussion even if your data is from a different instrument.

For multiaperture data the task for removing the relative shapes of the spectra is called **apnormalize**. Again, consult the help documentation for this task for further details. Since you will probably also be using the package for extracting the spectra you may be interested in the document *The IRAF APEXTRACT Package*.

4.2. Illumination Corrections

The flat field calibration images may not have the same illumination pattern as the observations of the sky due to the way the lamp illuminates the optical system. In this case when the flat field correction is applied to the data there will be gradients in the sky background. To remove these gradients a blank sky calibration image is heavily smoothed to produce an illumination image. The illumination image is then divided into the images during processing to correct for the illumination difference between the flat field and the objects. Like the flat fields, the illumination corrections images may be subset dependent so there should be an illumination image for each subset.

The task which makes illumination correction images is **mkskycor**. Some examples are

```
cl> mkskycor sky004 Illum004
cl> mkskycor sky*.imh ""
```

In the first example the sky image "sky004" is used to make the illumination correction image "Illum004". In the second example the sky images are converted to illumination correction images by specifying no output image names. Like **ccdproc** if the input images have not been processed they are first processed automatically.

To apply the illumination correction

```
cl> ccdproc *.imh ccdtype=object illumcor+ illum=Illum004
cl> ccdproc *.imh ccdtype=object illumcor+ illum=sky*.imh
```

The illumination images could also be set using **eparam** or given on the command line.

4.3. Sky Flat Fields

You will notice that when you process images with an illumination correction you are dividing each image by a flat field calibration and an illumination correction. If the illumination corrections are not done as a later step but at the same time as the rest of the processing one will get the same calibration by multiplying the flat field by the illumination correction and using this product alone as the flat field. Such an image is called a *sky flat* since it is a flat field which has been corrected to yield a flat sky when applied to the observations. This approach has the advantage of one less calibration image and two less computations (scaling and dividing the illumination correction). As an added short cut, rather than compute the illumination image with **mkskycor** and then multiplying, the task **mkskyflat** does all this in one step. Thus, **mkskyflat** takes an input blank sky image, processes it if needed, determines the appropriate flat field (sky flats are also subset dependent) from the **ccdproc** parameters or the input image list, and produces an output sky flat. Further if no output image is specified the task converts the input blank sky calibration image into a sky flat.

Two examples in which a new image is created and in which the input images are converted to sky flats are

```
cl> mkskyflat sky004 Skyflat
cl> mkskyflat sky*.imh ""
```

4.4. Illumination Corrected Flat Fields

A third method to account for illumination problems in the flat fields is to remove the large scale pattern from the flat field itself. This is useful if there are no reasonable blank sky calibration images and the astronomical exposures are evenly illuminated but the flat fields are not. This is done by smoothing the flat field images instead of blank sky images. As with using the sky images there are two methods, creating an illumination correction to be applied as a separate step or fixing the original flat field. The smoothing algorithm is the same as that used in the other tasks. The tasks to make these types of corrections are **mkillumcor** and **mkillumflat**. The usage is pretty much the same as the other illumination correction tasks except that it is more reasonable to replace the original flat fields by the corrected flat fields when fixing the flat field. Examples of an illumination correction and removing the illumination pattern from the flat field are

```
cl> mkillumcor flat025 Illum025
cl> mkillumflat flat*.imh ""
```

As with the other tasks, the input images are processed if necessary.

4.5. Fringe Corrections

Some CCD detectors suffer from fringing effects due to the night sky emission lines which are not removed by the other calibration and correction operations. To correct for the fringing you need a really blank sky image. There is not yet a task to remove objects from sky images because this is often done with an interactive image display tool (which will soon be added). The blank sky image is heavily smoothed to determine the mean sky background and then this is subtracted from the original image. The image should then be essentially zero except for the fringe pattern. This fringe correction image is scaled to the same exposure time as the image to be corrected and then subtracted to remove the fringing. Note that since the night sky lines are variable there may need to be an additional scaling applied. Determining this scaling requires either an interactive display tool or a very clever task. Such tasks will also be added in the future.

The task to make a fringe correction image is **mkfringecor**. the sky background is determined in exactly the same way as the illumination pattern, in fact the same sky image may be used for both the sky illumination and for the fringe correction. The task works consistently with the "mk" tasks in that the input images are processed first if needed and then the output correction image is produced with the specified name or replaces the input image if no output image is specified. As examples,

```
    cl> mkfringecor sky004 Fringe
    cl> mkfringecor sky*.imh ""
```

5. Demonstration

A simple demonstration task is available. To run this demonstration load the **ccdtest** package; this is a subpackage of the main **ccdred** package. Then simply type

```
    cl> demo
```

The demonstration will then create some artificial CCD data and reduce them giving descriptive comments as it goes along. This demonstration uses the "playback" facility of the command language and is actually substituting it's own commands for terminal input. Initially you must type carriage return or space after each comment ending with "...". If you wish to have the demonstration run completely automatically at it's own speed then type 'g' a the "..." prompt. Thereafter, it will simple pause long enough to give you a chance to read the comments. When the demo is finished you will need to remove the files created. However, feel free to examine the reduced images, the log file, etc. *Note that the demonstration changes the setup parameters so be sure to run* **setinstrument** *again and check the setup parameters.*

6. Summary

The **ccdred** package is very easy to use. First load the package; it is in the **imred** package which is in the **noao** package. If this is your first time reducing data from a particular instrument or if you have changed instruments then run **setinstrument**. Set the processing parameters for the operations you want performed. If you need to combine calibration images to form a master calibration image use one of the combine tasks. Spectroscopic flat fields may need to be processed first in order to remove the lamp spectrum. Finally, just type

```
    cl> ccdproc *.imh&
```

References

A general guide to using IRAF is *A User's Introduction to the IRAF Command Language*. This document may be found in the IRAF documentation sets and is available from the National Optical Astronomy Observatories, Central Computer Services (NOAO-CCS).

A more detailed description of the **ccdred** package including a discussion of the design and some of the algorithms see *The IRAF CCD Reduction Package -- CCDRED*" by F. Valdes. This paper is available from NOAO-CCS and appears in the proceedings of the Santa Cruz Summer Workshop in Astronomy and Astrophysics, *Instrumentation for Ground-Based Optical Astronomy: Present and Future*, edited by Lloyd B. Robinson and published by Springer-Verlag.

The task descriptions and supplementary documentation are available in printed form in the IRAF documentation sets, a special set containing documentation for just the **ccdred** package, and on-line through the help task by typing

```
    cl> help topic
```

where *topic* is one of the following.

```
  badpiximage - Create a bad pixel mask image from a bad pixel file
    ccdgroups - Group CCD images into image lists
     ccdhedit - CCD image header editor
      ccdlist - List CCD processing information
      ccdproc - Process CCD images
      ccdtest - CCD test and demonstration package
      combine - Combine CCD images
   cosmicrays - Detect and replace cosmic rays
  darkcombine - Combine and process dark count images
  flatcombine - Combine and process flat field images
  mkfringecor - Make fringe correction images from sky images
   mkillumcor - Make flat field illumination correction images
  mkillumflat - Make illumination corrected flat fields
     mkskycor - Make sky illumination correction images
    mkskyflat - Make sky corrected flat field images
setinstrument - Set instrument parameters
  zerocombine - Combine and process zero level images

          ADDITIONAL HELP TOPICS

       ccdred - CCD image reduction package
     ccdtypes - Description of the CCD image types
   flatfields - Discussion of CCD flat field calibrations
        guide - Introductory guide to using the CCDRED package
  instruments - Instrument specific data files
      subsets - Description of CCD subsets
```

Printed copies of the on-line help documentation may be made with the command

```
    cl> help topic | lprint
```

In addition to the package documentation for **ccdred**, **longslit**, and **apextract** there may be specific guides for certain instruments. These specific guides, called "cookbooks", give specific examples and parameter values for the CCD data.