The IRAF APEXTRACT Package

Francisco Valdes

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

ABSTRACT

The IRAF **apextract** package provides tools for the extraction of one and two dimensional spectra from two dimensional images such as echelle, long slit, multifiber, and multislit spectra. Apertures of fixed spatial width define the regions of the two dimensional images to be extracted at each point along the dispersion axis. Apertures may follow changes in the positions of the spectra as a function of position along the dispersion axis. The spatial and dispersion axes may be oriented along either image axis. Extraction to one dimensional spectra consists of a weighted sum of the pixels within the apertures at each point along the dispersion axis. The weighting options provide the simple sum of the pixel values and a weighting by the expected uncertainty of each pixel. Two dimensional extractions interpolate the spectra in the spatial axis to produce image strips with the position of the spectra exactly aligned with one of the image dimensions. The extractions also include optional background subtraction, modeling, and bad pixel detection and replacement. The tasks are flexible in their ability to define and edit apertures, operate on lists of images, use apertures defined for reference images, and operate both very interactively or noninteractively. The extraction tasks are efficient and require only one pass through the data. This paper describes the package organization, the tasks, the algorithms, and the data structures.

July 2, 1990

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

The IRAF APEXTRACT Package

Francisco Valdes

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

1. Introduction

The IRAF **apextract** package provides tools for the extraction of one and two dimensional aperture spectra from two dimensional format images such as those produced by echelle, long slit, multifiber, and multislit spectrographs. This type of data is becoming increasingly common because of the efficiency of data collection and technological improvements in spectrographs and detectors. The trend is to greater and greater numbers of spectra per image. Extraction is one of the fundamental operations performed on these types of two dimensional spectral images, so a great deal of effort has gone into the design and development of this package and to making it easy to use.

The tasks are flexible and have many options. To make the best use of them it is important to understand how they work. This paper provides a general description of the package organization, the tasks, the algorithms, and the data structures. Specific descriptions of parameters and usage may be found in the IRAF help pages for the tasks which are included as appendices to this paper. The image reduction "cookbooks" also provide examples of usage for specific instruments or types of instruments.

Extraction of spectra consists of three logical steps. First, locating the spectra in the two dimensional image. This includes defining the dispersion direction, the positions of the spectra at some point along the dispersion direction, the spatial extent or aperture to be used for extraction, and possible information about where the background for each spectrum is to be determined. This information is maintained in the package as structures called *apertures*. The second step is to measure the positions of the spectra at other points along the dispersion. This process is called tracing. Tracing is optional if the spectra are exactly aligned with the dispersion direction. The final step is to extract the spectra into one or two dimensional images.

The **apextract** package identifies the image axes with the spatial and dispersion axes. Thus, during extraction, pixels of constant wavelength are assumed to be along a line or column. In this paper the terms *slit* or *spatial* axis and *dispersion* or *wavelength* axis are used to refer to the image axes corresponding to the spatial and dispersion axes. To simplify the presentation a cut across the dispersion axis will be called a line even though it could also be a column.

Often a small degree of misalignment between the image axes and the true dispersion and spatial axes is not important. The main effect of misalignment is a broadening of the spectral features due to the difference in wavelength on opposite sides of the extraction aperture. If the misalignment is significant, however, the image may be rotated with the task **rotate** in the **images** package or remapped with the **longslit** package tasks for coordinate rectification.

It does not matter which image axis is the dispersion axis since the tasks work equally well in either orientation. However, the dispersion axis must be defined, with the **twodspec** task **setdisp**, before these tasks may be used. This task is a simple script which adds the

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

parameter DISPAXIS to the image headers. The **apextract** tasks, like the **longslit** tasks, look in the header to determine the dispersion axis.

2. The APEXTRACT Package

In this section the organization of the **apextract** package and the functions and parameters of the tasks are briefly described. More detailed descriptions are given in the help pages for the tasks. The tasks in the package are:

The APEXTRACT Tasks

```
  apdefault - Set the default aperture parameters
     apedit - Edit apertures interactively
     apfind - Automatically find spectra and define apertures
       apio - Set the I/O parameters for the APEXTRACT tasks
apnormalize - Normalize 2D apertures by 1D functions
    apstrip - Extract two dimensional aperture strips
      apsum - Extract one dimensional aperture sums
    aptrace - Trace positions of spectra
```

The tasks are highly integrated so that each task includes some or all of the functions and parameters of the other tasks. Thus, these tasks reflect the logical organization of the extraction process rather than a set of disparate tools. One reason for this organization is to group the parameters by function into easy to manage *parameter sets (psets)*. The tasks **apdefault** and **apio** are just psets for specifying the default aperture parameters and the I/O parameters of the package; in other words, they do nothing but provide a grouping of parameters. Executing these tasks is a shorthand for the command "eparam apdefault" or "eparam apio".

The input/output parameters in **apio** specify the aperture database, an optional log file for brief, time stamped log information, an optional metacode plot file for saving plots of the apertures, the traces, and the quick look extracted spectra, and the graphics input and output devices (almost always the user’s terminal). One point about the plot file is that the plots are recorded even if the user chooses not to view these graphs as the task is run interactively or noninteractively. This allows reviewing the traces and spectra with a tool like **gkimosaic**.

The default aperture parameters specify the aperture limits (basically the width of the aperture and position relative to the center of the spectrum) and the background fitting parameters. The background parameters are the standard parameters used by the **icfit** package with which the user is assumed to be familiar. For more on this see the help information for **icfit**.

The other tasks are both psets and executable tasks. There are a number features which are common to all these tasks. First, they follow the same steps in defining apertures for the input images. These steps are:

(1) If a reference image is specified then the database is searched for apertures previously defined for this image.

(2) If apertures are found for the reference image they may be recentered on the spectra in the input image at a specified line. This does not change the shape of the apertures but only adds a shift in the center coordinate of the apertures along the spatial axis.

(3) If a reference image is not specified or if no reference apertures are found then the database is searched for previous apertures for the input image.

(4) If there are no apertures defined either from a reference image or previous apertures for the input image then an automatic algorithm may be used to find a specified number of spectra (based on peak values) and assign them default apertures.

(5) Finally, a sophisticated graphical aperture editor may be used to examine, define, and modify apertures.

(6) When tracing, extracting, or normalizing flat field spectra, if no apertures have been defined by the steps above then a single default aperture, centered in the image, is defined.

Any apertures created, modified, or adopted from a reference image may be recorded in the database for the input image.

The operations listed above are selected by parameters common to each of the tasks. For example the parameter *edit* selects whether to enter the aperture editor and is present in each of the executable tasks. On the other hand the parameters specific to the aperture editor, while accessed by any of the tasks, reside only in the parameter set of the task **apedit**. In this way parameters are distributed by logical function rather than including them in each task.

In addition to the aperture editing and finding functions available in every task, some of the tasks include functions for tracing, extracting, or normalizing the spectra. The tasks **apsum** and **apstrip**, which extract one and two dimensional spectra, are at the top of the hierarchy and include all the logical functions provided by the package. Thus, in most cases the user need only use the task **apsum** to define apertures, trace the spectra, and extract them.

Another feature common to the tasks is their interactive and noninteractive modes. When the parameter *interactive* is set to *no* then the aperture editing, interactive trace fitting, and review of the extracted one dimensional spectra functions of the package are bypassed. Note that this means you do not have to explicitly set the parameter *edit*, or those for other purely interactive functions, to *no* when extracting spectra noninteractively. In the noninteractive mode there are also no queries.

The interactive mode includes the interactive graphical functions of aperture editing, trace fitting, and extraction review. In addition the user is queried at each step. For example the user will be queried whether to edit the apertures for a particular image if the task parameter for editing is set. The queries have four responses: *yes, no, YES,* and *NO*. The lower case responses apply only to the particular query. The upper case responses apply to any further queries of the same type and suppress the query from appearing again. This is particularly useful when dealing with many images or many apertures. For example, when fitting the traced points interactively the user may examine the first few and then say *NO* to skip the remaining apertures using the last defined fitting parameters. Note that if a plot file is specified the graphs showing the traced points and the fits are recorded even if they are not viewed interactively.

3. Algorithms

The **apextract** package consists of a number of logical functions or, in computerese, algorithms. These algorithms manipulate the aperture structure data and create output data in the form of images. In this section the various algorithms are described. In addition to the algorithms specific to the package, there are some general algorithms and tools used which appear in other IRAF tasks. Specifically there are the interactive curve fitting tools called **icfit** and the one dimensional centering algorithm called **center1d**. These are mentioned below and described in detail elsewhere in the help documentation.

3.1. Finding Spectra

When dealing with images containing large numbers of spectra it may be desirable to locate the spectra and define apertures automatically. The **apfind** algorithm provides this ability from any of the executable tasks and from the aperture editor using the 'f' key. It takes a cut across the dispersion axis by summing one or more image lines. All the local maxima are identified and ranked by intensity. Starting with the highest maxima any other peaks within a specified minimum separation are eliminated. The weakest remaining peaks exceeding the specified number are eliminated next. The positions of the spectra based on peak positions are refined by centering using the **center1d** algorithm. Finally identical apertures are assigned for each spectrum found.

When the algorithm is invoked by a task, with the parameter *find*, there must be no previous or reference apertures in the database. The apertures assigned to the spectra have the parameters specified in the **apdefault** pset. When the algorithm is invoked from the aperture editor with the 'f' key then new apertures are added to any existing apertures up to the total number of apertures, existing plus new, given by the *nfind* parameter. If there is a current aperture then copies of it are used to define the apertures for the new spectra. Thus, one method for defining many apertures is to use the editor to define one aperture, set its limits and background parameters, and then find the remaining apertures automatically.

3.2. Centering and Recentering

When new apertures are defined (except for a special key to mark apertures without centering) or when apertures are recentered, either with the centering key in the editor or with the task parameter *recenter*, the center is determined using the **center1d** algorithm. This is described in the help documentation under the name **center1d**. Briefly, the data line is convolved with an asymmetric function of specified width. The convolution integral is evaluated using image interpolation. The sign of the convolution acts as a gradient to move from the starting position to the final position where the convolution is zero. This algorithm is good to about 5% of a pixel. It has two important parameters; the width of the convolution and the error distance between the starting and final positions. The width of the convolution determines the scale of features to which the centering is sensitive. The error distance is the greatest change allowed in the initial positions. If this error distance is exceeded then the centering fails and either a new aperture is not defined or the position of an existing aperture is not changed.

3.3. The Aperture Editor

The aperture editor is a sophisticated tool for defining and modifying apertures. It may also be used to selectively trace and extract spectra. Thus, the aperture editor may be used alone to perform all the functions for extracting spectra. The aperture editor uses a graphical presentation. A line or sum of lines is displayed. The apertures are marked above the line and identified with the aperture number. Information about the current aperture is shown on the status line. The cursor is used to mark new apertures, shift the center or aperture limits, and perform a variety of functions. Because there may be many apertures which the user wants to modify in the same way there is a mode switch to apply commands to all the apertures. The switch is toggled with the 'a' key and the mode is indicated on the status line.

There are also a number of colon commands. These allow resetting parameters explicitly rather than by cursor and interacting with the aperture database and the image data. The background fitting parameters such as the background regions and function order are set by switching to the interactive curve fitting package **icfit**. The line being edited is used to set the parameters. No background is actually extracted at this stage. The ALL mode applies to the background parameters as well.

The aperture editor has many commands. For a description of the commands see the help information for the task **apedit**. In summary the aperture editor is used to interactively define apertures, both centered on spectra and at arbitrary positions, adjust the limits and background parameters, and possibly select apertures to be traced and extracted. These functions may be applied independently on each aperture for maximum flexibility or applied to all apertures for ease of use with many apertures.

3.4. Tracing

The spectra to be extracted are not always aligned exactly with the image columns or lines. For consistent extraction it is important that the same part of the spectrum profile be extracted at each wavelength point. Thus, the extraction apertures allow for shifts along the spatial axis at each wavelength. The shifts are defined by a curve which is a function of the wavelength. The curve is determined by tracing the positions of the spectrum profile at a number of wavelengths and fitting a function to these positions.

The *aptrace* algorithm performs the tracing and curve fitting. The starting point along the dispersion axis (a line or column) for the tracing is specified by the user. The positions of the spectrum profiles are determined using the **center1d** algorithm (see the previous section on centering and the help page for **center1d**). The user specifies a step along the dispersion axis. At each step the positions of the profiles are redetermined using the preceding positions as the initial guesses. If the positions are lost at one step an attempt is made to recover the spectrum in the next step. If this also fails then tracing of that spectrum in that direction is finished. In order to enhance and trace weak spectra the user may specify a number of neighboring profiles to be summed before determining the profile positions. In addition to the other centering parameters, there is also a *threshold* parameter to define a minimum contrast between the spectrum and the background.

Once the positions have been traced from the starting point to the ends of the aperture, or until the positions become indeterminate, a curve of a specified type and order is fit to the positions as a function of wavelength. The function fitting is performed with the **icfit** tools (see the help documentation for **icfit**). The curve fitting may be performed interactively or noninteractively. Note that when the curve is fit interactively the actual positions measured are graphed. However, the curve is stored in the aperture definition as an offset relative to the aperture center.

The tracing requires that the spectrum profile be continuous and have some kind of maxima. This means that arc calibration spectra or arbitrary regions of an extended object in a long slit spectrum cannot be traced. Flat topped spectra such as quartz lamp images taken through slits can be measured provided the width of the centering function is somewhat wider than the profile (to avoid centering on little peaks within the slit). For images which cannot be traced, reference apertures from images that can be traced are used. This is how apertures for arc spectra are defined and extracted. For sky apertures or the wings of extended objects the reference apertures can be shifted by the aperture editor without altering the shape of the aperture.

3.5. Sum Extraction

Sum extraction consists of the weighted sum of the pixels along the spatial axis within the aperture limits at each point along the dispersion axis. A background at each point along the dispersion may be determined by fitting a function to data in the vicinity of the spectrum and subtracting the function values estimated at each point within the aperture. The estimated background may be output as a one dimensional spectrum. Other options include the detection and replacement of deviant points such as due to cosmic rays.

Denote the image axis nearest the spatial axis by the index s and the other image axis corresponding to the dispersion axis by λ. The weighted extraction is defined by the equation

$$f_\lambda = \sum_s (W_{s\lambda}(I_{s\lambda} - B_{s\lambda})/P_{s\lambda}) / \sum_s W_{s\lambda} \qquad (1)$$

where the sums are over all pixels along the spatial axis within some aperture. The W are weights, the I are pixel intensities, the B are background intensities, and the P are a normalized profile model.

There are many possible choices for the extraction weights. The extraction task **apsum** currently provides two:

$$W_{s\lambda} = P_{s\lambda} \qquad (2a)$$

$$W_{s\lambda} = P_{s\lambda}^2 / V_{s\lambda} \qquad (2b)$$

where $V_{s\lambda}$ is the variance of the pixel intensities given by the model

$$V_{s\lambda} = v_0 + v_1 \max(0,\ I_{s\lambda}) \quad \text{if } v_0 > 0$$
$$V_{s\lambda} = v_1 \max(1,\ I_{s\lambda}) \qquad \text{if } v_0 = 0$$

Substituting these weights in equation (1) yields the extraction equations

$$f_\lambda = \sum_s (I_{s\lambda} - B_{s\lambda}) \tag{3a}$$

$$f_\lambda = \sum_s (P_{s\lambda}(I_{s\lambda} - B_{s\lambda})/V_{s\lambda}) / \sum_s (P^2_{s\lambda}/V_{s\lambda}) \tag{3b}$$

The first type of weighting (2a), called *profile* weighting, weights by the profile. Since the weights cancel this gives the simple extraction (3a) consisting of the direct summation of the pixels within the aperture. It has the virtue of being simple and computationally fast (since the profile model does not have to be determined).

The second type of weighting (2b), called *variance* weighting, uses a model for the variance of the pixel intensities. The model is based on Poisson statistics for a linear quantum detector. The first term is commonly call the *readout* noise and the second term is the Poisson noise. The actual value of v_1 is the reciprocal of the number of photons per digital intensity unit (ADU). A simple variant of this type of weighting is to let v_1 equal zero. Since the actual scale of the variance cancels we can then set v_0 to unity to obtain

$$f_\lambda = \sum_s (P_{s\lambda}(I_{s\lambda} - B_{s\lambda})) / \sum_s P^2_{s\lambda}. \tag{4}$$

The interpretation of this extraction is that the variance of the intensities is constant. It gives greater weight to the stronger parts of the spectrum profile than does the profile weighting (3a) since the weights are $P^2_{s\lambda}$. Equation (4) has the virtue that one need not know the readout noise or the ADU to photon number conversion.

3.5.1. Optimal Extraction

Variance weighted extraction is sometimes called optimal extraction because it is optimal in a statistical sense. Specifically, the relative contribution of a pixel to the sum is related to the uncertainty of its intensity. The uncertainty is measured by the expected variance of a pixel with that intensity. The degree of optimality depends on how well the relative variances of the pixels are known.

A discussion of the concepts behind optimal extraction is given in the paper *An Optimal Extraction Algorithm for CCD Spectroscopy* by Keith Horne (**PASP**, June 1986). The weighting described in Horne's paper is the same as the variance weighting described in this paper. The differences in the algorithms are primarily in how the model profiles $P_{s\lambda}$ are determined.

3.5.2. Profile Determination

The profiles of the spectra along the spatial axis are determined when either the detection and replacement of bad pixels or variance weighting are specified. The requirements on the profiles are that they have the same shape as the image profiles at a each dispersion point and that they be as noise free and uncontaminated as possible. The algorithm used to create these profiles is to average a specified number of consecutive background subtracted image profiles immediately preceding the wavelength to which a profile refers. When there are an insufficient number of image profiles preceding the wavelength being extracted then the following image profiles are also used to make up the desired number. The image profiles are interpolated to a common center before averaging using the curve given in the aperture definition. The averaging reduces the noise in the image data while the centering eliminates shifts in the spectrum as a function of wavelength which would broaden the profile relative to the profile of a single image line or column. It is assumed that the spectrum profile changes slowly with wavelength so that by using profiles near a given wavelength the average profile shape will correctly reflect the profile of the spectrum at that wavelength.

The average profiles are determined in parallel with the extraction, which proceeds sequentially through the image. Initially the first set of spectrum profiles is read from the image and interpolated to a common center. The profiles are averaged excluding the first profile to be extracted; the image profiles in the average never include the image profile to be extracted. Subsequently the average profile is updated by adding the last extracted image profile and subtracting the image profile which no longer belongs in the average. This allows each image profile to be accessed and interpolated only once and makes the averaging computationally efficient. This scheme also allows excluding bad pixels from the average profile. The average profile is used to locate and replace bad pixels in the image profile being extracted as discussed in the following sections. Then when this profile is added into the average for the next image profile the detected bad pixels are no longer in the profile.

In summary this algorithm for determining the spectrum profile has the following advantages:

(1) No model dependent smoothing is done.

(2) There is no assumption required about the shape of the profile. The only requirement is that the profile shape change slowly.

(3) Only one pass through the image is required and each image profile is accessed only once.

(4) The buffered moving average is very efficient computationally.

(5) Bad pixels are detected and removed from the profile average as the extraction proceeds.

3.5.3. Detection and Elimination of Bad Pixels

One of the important features of the aperture extraction package is the detection and elimination of bad pixels. The average profile described in the previous section is used to find pixels which deviate from this profile. The algorithm is straightforward. A model spectrum of the image profile is obtained by scaling the normalized profile to the image profile. The scale factor is determined using chi-squared fitting:

$$M_{s\lambda} = P_{s\lambda} \left\{ \sum_s ((I_{s\lambda} - B_{s\lambda}) P_{s\lambda} / V_{s\lambda}) \;/\; \sum_s (P_{s\lambda}^2 / V_{s\lambda}) \right\}. \qquad (6)$$

The RMS of this fit is determined and pixels deviating by more than a user specified factor times this RMS are rejected. The fit is then repeated excluding the rejected points. These steps are repeated until the user specified number of points have been rejected or no further deviant points are detected. The rejected points in the image profile are then replaced by their model values.

This algorithm is based only on the assumption that the spatial profile of the spectrum (no matter what it is) changes slowly with wavelength. It is very sensitive at detecting departures from the expected profile. It has two problems currently. Because the input line is first interpolated to the same center as the profile, single bad pixels are generally broadened to two bad pixels, making it harder to find the bad data. Also, in the first pass at the fit all of the image profile is used so if there is a very badly deviant point and the rest of the profile is weak then the scale factor may favor the bad pixel more than the rest of the profile. This may result in rejecting good profile points and not the bad pixel.

3.5.4. Relation of Optimal Extraction to Model Extraction

Equation (1) defines the extraction process in terms of a weighted sum of the pixel intensities. However, the actual extraction operations performed by the task **apsum** are

$$f_\lambda = \sum_s (I_{s\lambda} - B_{s\lambda}) \tag{7a}$$

$$f_\lambda = \sum_s M_{s\lambda} \tag{7b}$$

where $M_{s\lambda}$ is the model spectrum fit to the background subtracted image spectrum $(I_{s\lambda} - B_{s\lambda})$ defined in the previous section (equation 6). It is not obvious at first that (7b) is equivalent to (3b). However, if one sums (6) and uses the fact that the sum of the normalized profile is unity one is left with equation (3b).

Equations (6) and (7b) provide an alternate way to think about the extracted one dimensional spectra. Sum extraction of the model spectrum is used instead of the weighted sum for variance weighted extraction because the model spectrum is a product of the profile determination and the bad pixel cleaning process. It is then more convenient and efficient to use the simple equations (7).

3.6. Strip Extraction

The task **apstrip** uses one dimensional image interpolation to shift the pixels along the spatial axis so that in the resultant output image the center of the aperture is exactly aligned with the image lines or columns. The cleaning of bad pixels is an option in this extraction using the methods described above. In addition the model spectrum, described above, may be extracted as a two dimensional image. In fact, the only difference between strip extraction and sum extraction is whether the final step of summing the pixels in the aperture along the spatial axis is performed.

The primary use of **apstrip** is as a diagnostic tool. It allows the user to see the background subtracted, cleaned, and/or model spectrum as an image before it is summed to a one dimensional spectrum. In addition the two dimensional format allows use of other IRAF tools such as smoothing operators. When appropriate it is a much simpler method of removing detector distortions and alignment errors than the full two dimensional mapping and image transformation available with the **longslit** package.

3.7. Aperture Normalization

The special algorithm/task **apnormalize** normalizes the two dimensional image data within an aperture by a smooth function of the dispersion coordinate. Unlike the extraction tasks the output of this algorithm is a two dimensional image of the same format as the input image. This function is used primarily for creating flat field images in which the large scale shape of the quartz spectra and the variations in level between the spectra are removed and the regions between the spectra, where there is no signal, are set to unity. It may also be used to normalize two dimensional spectra to a unit continuum at some point in the spectrum, such as the center.

The algorithm is to extract a one dimensional spectrum for each aperture, fit a smooth function to the spectrum, and then divide this spectrum back into the two dimensional image. Points outside the apertures are set to 1. This is the same algorithm used in the **longslit** package by the task **response** except that it applies to arbitrary apertures rather than to image sections.

Apertures are defined in the same way as for extraction. The normalization spectrum may be obtained from a different aperture than the aperture to be normalized. Generally the normalization apertures are either the same or narrower than the apertures to be normalized. The continuum fitting also uses the **icfit** package. Sample regions and iterative sigma clipping are used to remove spectral lines from the continuum fits.

There are two commonly used approaches to fitting the extracted spectra in flat field images. First, a constant function is fit. This has the effect of simply normalizing the apertures to near unity without affecting the shape of spectra in any way. This removes response effects at all scales, from spectra flatten with this flat field. However, it does not preserve total counts, it introduces the shape of the quartz spectrum, and it removes the blaze function. The second approach is to fit the large scale shape of the quartz spectra. This removes smaller scale response effects such a fringing and individual pixel responses while preserving the total counts by leaving the blaze function alone. There are cases where each of these approaches is applicable.

4. Apertures

Apertures are the basic data structures used in the package; hence the package name. An aperture defines a region of the two dimensional image to be extracted. The aperture definitions are stored in a database. An aperture consists of the following components:

ID
: An integer identification number. The identification number must be unique. It is used as the default extension during extraction of the spectra. Typically the IDs are consecutive positive integers ordered by increasing or decreasing slit position.

BEAM
: An integer beam number. The beam number need not be unique; i.e. several apertures may have the same beam number. The beam number will be recorded in the image header of the the extracted spectrum. By default the beam number is the same as the ID.

CENTER[2]
: The center of the aperture along the slit and dispersion axes in the two dimensional image.

LOWER[2]
: The lower limits of the aperture, relative to the aperture center, along the slit and dispersion axes. The lower limits need not be less than the center.

UPPER[2]
: The upper limits of the aperture, relative to the aperture center, along the slit and dispersion axes. The upper limits need not be greater than the center.

APAXIS
: The aperture or spatial axis.

CURVE
: An offset to be added to the center position for the aperture axis as a function of the wavelength. The function is one of the standard IRAF types; a legendre polynomial, a chebyshev polynomial, a linear spline, or a cubic spline.

BACKGROUND
: Parameters for background subtraction along the aperture axis based on the interactive curve fitting (**icfit**) tools.

The aperture center is the only absolute coordinate (relative to the image or image section). The other aperture parameters and the background fitting regions are defined relative to the center. Thus, an aperture may be repositioned easily by changing the center coordinates. Also constant aperture size, shape (curve), and background regions may be maintained for many apertures.

The edges of the aperture along the spatial axis at each point along the dispersion axis are given by evaluating the offset curve at that dispersion coordinate and adding the aperture center and the lower or upper limits for the aperture axis. The edges of the aperture along the dispersion axis do not have an offset curve and are currently fixed to define the entire length of the image. In the future this may not be the case such as in applications with objective prism spectra.

Apertures for a particular image may be defined in several ways. They may be defined and modified graphically with an aperture editor. Default apertures may be defined automatically with parameters from the **apdefault** pset using an aperture finding algorithm. Another method is to specify that the apertures for one image use the aperture definitions from another "reference" image. In the rare cases where apertures are not defined at the stage of tracing or extracting then a single default aperture centered in the image is created.

4.1. The Database

The aperture information is stored in a database. The structure and type of database is expected to change in the future and as far as the package and user need be concerned it is just a black box with some name specified in the database name parameter. However, accepting that the database structure may change it may be of use to the user to understand the nature of the current text file / directory format database. The database is a directory containing text files. It is automatically created if necessary. The aperture data for all the apertures from a single image are stored in a text file with the name given by the image name (with special characters replaced with '_') prefixed with "ap". Updates of the aperture data are performed by overwriting the database file.

The content of a file consists of a comment (beginning with a #) giving the date created/updated, a record identification (there is one record per aperture) with the image name, aperture number and aperture coordinate in the aperture and dispersion axes. The following lines give information about the aperture. The position and shape of an aperture is given by a center coordinate along the aperture axis (given by the axis keyword) and the dispersion axis. There are lower and upper limits for the aperture relative to this center, again along both axis. Currently the limits along the dispersion axis are the image boundaries. The background keyword introduces the background subtraction parameters. Finally there is an offset or trace function which is added to the center at each point along the dispersion axis. function. The offset is generally zero at the dispersion point corresponding to the aperture center.

This offset or trace function is described by a **curfit** array under the keyword curve. The first value is the number of elements in this array. The first element is a magic number specifying the function type. The next number is the order or number of spline pieces. The next two elements give the range over which the curve is defined. In the **apextract** case it is the edges of the image along the dispersion. The remaining elements are the function coefficients. The form of the the function is specific to the IRAF **curfit** math routines. Note that the coefficients apply to an independent variable which is -1 at the beginning of the defined range (element 3) and 1 at the end of the range (element 4). For further details consult the IRAF group.

An example database file for one aperture from an image "ech001" is given below.

```
# Fri 14:33:35 08-May-87
begin   aperture ech001 1 22.75604 100.
    image   ech001
    aperture1
    beam1
    center  22.75604 100.
    low -2.680193 -99.
    high3.910698 100.
    background
        xmin -262.
        xmax 262.
        function chebyshev
        order 1
        sample -10:-6,6:10
        naverage -3
        niterate 0
        low_reject 3.
        high_reject 3.
        grow 0.
    axis1
    curve   6
        2.
        2.
        1.
        200.
        -0.009295368
        -0.3061974
```

5. Future Developments

The IRAF extraction package **apextract** is going to continue to evolve because the extraction of one and two dimensional spectra from two dimensional images is an important part of reducing echelle, longslit, multislit, and multiaperture spectra. Changes may include some of the following:

(1) Determine the actual variance from the data rather than using the Poisson CCD model. Also output the variance vector if desired.

(2) The bad pixel detection and removal algorithm does not handle well the case of a very strong cosmic ray event on top of a very weak spectrum profile. A heuristic method to make the first fitting pass of the average profile to the image data less prone to errors due to strong cosmic rays is needed. Also the detection should be done by interpolating the profile to the original image data rather than the other way around, in order to avoid broadening cosmic rays by interpolation.

(3) The aperture definition structure is general enough to allow the aperture limits along the dispersion dimension to be variable. Eventually aperture definition and editing will be available using an image display. Then both graphics and image display editing switches will be available. An image display interface will make extraction of objective prism spectra more convenient than it is now.

(4) Other types of extraction weighting may be added.