



清华大学

学 习 笔 记

主 题 _____ 天文

专 业 _____ 物理学

姓 名 _____ 李明宇

入学年份 _____ 2018 年

指导教师 _____ 蔡峥

创建时间 _____ 二〇二〇年

学习笔记

Mingyu Li

李明宇

THU 物理 8

摘要: 记录一些学习笔记

关键词: 天文, 学习笔记

目录

1 Photometry

几乎所有我们能接受到的来自太阳系外的信息都是以电磁辐射（EMR）的形式。

2 IRAF

2.1 简介

IRAF 是通用的图像 reduction 和 analysis 工具, 可为用户提供广泛的图像处理工具。IRAF 是 National Optical Astronomy Observatories (NOAO) 的产品, 是为天文界开发的, 其他科学领域的研究人员也发现 IRAF 对于一般图像处理很有用。本文档旨在为新手用户提供 IRAF 及其命令语言 (CL) 用户界面的简要介绍。

在本文中仅讨论 IRAF 的“核心”系统。IRAF 是 NOAO 系列科学软件其中的一个 (general spectroscopic and photometric reduction and analysis tools), 对其他软件的讨论可以在其他地方得到更好的介绍。

IRAF 核心系统使用命令行界面为用户提供了广泛的图像处理工具选择。通过执行其中的命令, 我们称为 tasks, 以执行各种功能。每个 task 都有一个 parameter file, 用户可以修改该 parameter file, 调整 task 的输出。该系统的重点包括:

- 无论主机系统如何, 都具有相似的环境
- query and command line interface
- pipes and redirection
- background job queue
- extensive online help
- interactive graphics and image display
- text and graphics hardcopy
- tape I/O
- several user programming environments
- world coordinate systems

其他 IRAF 软件 packages (称为 layered packages) 可从多种来源获得, 并且可以安装在 IRAF 环境下运行。有关更多信息, 请参见附录 F。

2.2 开始

运行 IRAF 所需的主机级别文件会有所不同，具体取决于您打算在其中执行 IRAF 的主机和窗口环境。工作站环境（SunView, OpenWindows, DECwindows, X Windows）将需要比常规图形终端环境所使用的主机级文件更多的文件。在工作站环境中使用的主机级别文件的讨论不在本文档的讨论范围之内。所有 UNIX 用户都需要确保其 UNIX 主目录中的.login 或.cshrc 文件都包含目录的“路径”，该目录包含在 IRAF 安装时定义的 IRAF 启动命令。“路径”语句可能类似下面这样：

```
1 set path = (. /local/bin /usr/ucb/bin /usr/bin /usr/local/bin)
```

在常规图形终端环境中，VMS / IRAF 不需要特殊的主机级文件。

每个用户在首次登录 IRAF 之前必须在主机级别执行命令“mkiraf”。此任务必须在 IRAF “登录”或“主”目录中执行，该目录不必与主机“登录”目录相同。您不必每次启动 IRAF 会话时都需要执行此操作-只需第一次，然后才建议您这样做（通常是在您的站点上安装了新版本的 IRAF 之后）。

MKIRAF 命令创建一个名为 login.cl 的文件（参见图 1）和一个名为 uparm 的子目录。login.cl 文件在 IRAF 启动时执行，IRAF 使用 uparm 子目录保存您的自定义参数文件（有关参数文件的更多信息，请参阅第 3.5 节）。MKIRAF 命令注意两件事：

1. 如果您曾经已经做过 MKIRAF，那么系统会询问您是否要“Initialize uparm”？除非您绝对确定自己知道自己在做什么，否则通常回答“否”。IRAF 的有用功能之一是，它通过将参数存储在 uparm 目录中来“学习”参数-初始化该目录会从字面上清除所有这些自定义参数文件！如果您在系统的全新安装后执行 MKIRAF，我们建议您对这个问题回答“是”。
2. 现在将要求您“输入终端类型：”-在此处输入您的图形终端类型，即 vt640, gterm, xterm 等。参见 §3.1.1 有关设置终端类型的更多信息

用户还可以创建一个 loginuser.cl 文件（参见图 2），该文件将在启动时由 login.cl 文件（参见图 1）执行。loginuser.cl 文件中的命令会取代 login.cl 文件中在调用 loginuser.cl 文件本身的语句之前执行的命令。loginuser.cl 文件不受新的 MKIRAF 的影响，因此该文件是自定义 IRAF 登录名的首选方法。关于此文件要注意的重要一点是，文件中的最后一个语句必须为 keep.

主机和 IRAF 级别的启动文件到位后，用户只需输入“cl”即可登录 IRAF。重要的是，从 IRAF 的“登录”或“主”目录登录 IRAF。此时，将执行 login.cl 文件（以及如果有的话，loginuser.cl 文件）中的命令。

如果从工作站环境中运行 IRAF，则需要先启动窗口环境，然后在终端仿真器窗口（如 gterm 或 xterm）中键入 cl。

通过键入“logout”来注销 IRAF。如果您是从 Windows 环境中运行 IRAF，则重要的是先 logout IRAF，然后关闭 Windows 环境。这提供了从 IRAF 的干净退出，并允许系统有序地关闭。

2.3 基础

2.3.1 设置 IRAF 环境

一旦用户登录 IRAF，他可能想为该会话自定义其 IRAF 环境。在 IRAF 启动时，`loginuser.cl` 文件也可以用于此目的（参见图 2）。IRAF CL 维护一个“环境变量”表，该表会影响许多 IRAF 任务的运行。可以通过键入以下命令查看这些变量的完整列表（将 SHOW 的输出“通过管道输入”到 PAGE 命令的输入中，这是一次一次查看一页文本的简单任务）。

完整的列表有点让人不知所措，但是您可能需要修改这些变量中的一小部分以自定义 IRAF 环境。SET 或 RESET 命令可用于此目的。

SET 命令仅在加载当前程序包时临时修改变量（请参阅第 3.2 节中有关程序包的更多信息）。使用 RESET 通常会为当前 IRAF 会话的其余部分更改变量的值。

IRAF 环境变量也可以用于定义目录的 IRAF 逻辑名称。这些逻辑名可以是表示主机级别路径名或其他 IRAF 逻辑名的字符串。例如，在本文档的许多示例中使用的测试映像“dev \$ pix”可以通过 SHOW task 追溯到以下主机目录。每个安装的 iraf 逻辑目录都将是不同的主机级别路径名，但是所有主机上的 dev 逻辑目录都将相同。IRAF 逻辑目录后跟一个“\$”。在最后一个示例中，任务 PATH 用于直接打印等效的主机路径名，包括节点或计算机名

2.3.2 Packages and tasks

The programs or commands that the user executes to perform specific functions are called tasks. Tasks that perform similar functions are grouped together into packages. The packages for the IRAF core system are listed below.

A package must be loaded in order to execute a task that is in it. A package is loaded by simply typing its name or enough characters to uniquely identify its name, i.e., `softools` or so. Note that IRAF is case sensitive! The prompt reflects the current package (the last package loaded). The last package that was loaded can be unloaded by typing `bye`. What packages are currently loaded can be checked by typing `package`. Once a package is loaded its tasks are available for execution until the package is unloaded. Loading a new package does not unload the previous package. There is essentially no overhead in having a variety of packages loaded at the same time.

3 SExtractor

sExtractor Double image mode 下，选哪个做 detect？

The detection image will generally be chosen in the band where the data are the deepest. Alternatively, using a “ χ^2 image” as a detection image, will allow most of the sources present in at least one channel to be detected and measured.

configuration file 的格式

代码是 ASCII 格式的。每行一个参数。如下

1 Config—parameter Value(s)

多余的空格和换行会被忽略掉。备注以 # 开头，以回车结尾。Value(s) 有多种类型：字符串（可以用双引号括起来）、浮点数、整数、关键词或者布尔数（Y/y,N/n）。有些参数可以接收 0 作为 Value。有的参数还可以接受多个 Values，多个 Value 之间必须以逗号隔开。整型可以以多种形式给出，如小数，八进制形式（以 0 开头），十六进制形式（以 0x 开头）。十六进制格式对于编写复用位值（如二进制掩码）特别方便。以 \$HOME 或 \${HOME} 形式写入的环境变量将被展开。

Configuration parameter list

这里是 SExtractor 已知的所有配置参数的完整列表。

The measurement (or catalog) parameter file

除了 configuratin file, SExtractor 还要求一个写有要测量参数列表 measurements (“catalog parameters”) 的文件，相应的，每次 detection 都会输出 catalog。这使得软件只会计算我们需要的 measurements。catalog parameters file 的名称一般以.param 为后缀，必须使用 PARAMETERS_NAME 配置参数来指定。完整的参数集可以使用下面的命令来查看。

1 sex -dp

catalog parameters file 的格式是 ASCII，每行必须有一个 keyword。目前 SExtractor 识别的关键词有两种：标量和向量。标量，如 X_IMAGE，在输出的 cat 文件中生成单个数字。向量，如 MAG_APER(4) 或 VIGNET(15,15)，生产数字数组。输出 cat 文件中的 measurements 的顺序与参数列表中的 keywords 的顺序相同。可以用 # 开头进行注释。

Variants

对于一些 catalog parameters，尤其那些和 flux, position, 或者 shape 有关的，有许多变体，如下：

Fluxes & magnitudes

FLuxes 可以表示成 counts(ADU(Analog-to-Digital Unit)s) 或者 Pogson magnitudes. 在 ADUs 中，Flux measurements 以 FLUX_ 为前缀，例如 FLUX_AUTO,FLUX_ISO, 等等。Magnitudes 以 MAG_ 为前缀，例如 MAG_AUTO, MAG_ISO,...。MAG_ZEROPOINT 设定了 magnitude 的 zero-point:

$$\text{MAG} = \begin{cases} \text{MAG_ZEROPOINT} - 2.5 \log_{10} \text{FLUX} & \text{if } \text{FLUX} > 0 \\ 99.0 & \text{otherwise} \end{cases}$$

Flux & magnitude uncertainties

Flux & magnitude uncertainties (error estimates) 的变体和 Flux & magnitude 遵循相似的规则，以 FLUXERR_ 和 MAGERR_ 为前缀，例如 FLUXERR_AUTO,FLUXERR_ISO ,MAGERR_AUTO, MAGERR_ISO。有以下计算公式：

$$\text{MAGERR} = \begin{cases} \frac{2.5}{\ln 10} (\text{FLUXERR}/\text{FLUX}) & \text{if } \text{FLUX} > 0 \\ 99.0 & \text{otherwise} \end{cases}$$

Pixel values and Surface brightnesses

Pixel values (averaged or not) p 被表示为 counts(ADUs). 没有特别的统一前缀 (THRESHOLD, BACKGROUND, FLUX_MAX, etc.) Surface brightnesses 是 magnitudes per square arcsecond. 前缀是 MU_, 例如 MU_THRESHOLD, MU_MAX. Surface brightnesses 的变换依赖于 MAG_ZEROPOINT 和 PIXEL_SCALE:

$$\text{MU} = \begin{cases} \text{MAG_ZEROPOINT} - 2.5 \log_{10} (p \times \text{PIXEL_SCALE}^2) & \text{if } p > 0 \\ 99.0 & \text{otherwise} \end{cases}$$

把 PIXEL_SCALE 设为 0, 会让 SExtractor 根据 FITS 图像 header 中的 celestial WCS info(如果有的话), 从 astrometric deprojection 的 local Jacobian 计算 pixel scale.

Positions and shapes

Positions, distances and position angles 是在 pixel coordinates 中被计算的。它们可能被表示成 image 的 pixels、world coordinates 或者 celestial coordinates, 依赖于后缀:

- **_IMAGE.Measurements** 将以 pixel coordinates 的形式给出, 以 pixels 为单位。例如 Y_IMAGE, ERRRAWIN_IMAGE, THETA_IMAGE. 根据 FITS 约定, 在 SExtractor 中第一个 pixel 的中心的坐标是 (1.0, 1.0), position angles 从 x 轴到 y 轴逐渐增加。
- **_WORLD.Measurements** 将以 world coordinates 的形式给出, 它可以从 pixel coordinates 通过坐标系变换的 Jacobian 得到。它要求 FITS 的头文件里有 WCS metadata. position angles 从第一个 world axis 到第二个 world axis 逐渐增加。
- **_SKY, _J2000, _B1950.Measurements** 将以 celestial (equatorial) coordinates 的形式给出. Positions 和 distances 以 degrees 为单位. 要求 FITS 的头文件里有 celestial WCS metadata. _SKY 的 measurements 是在 native world coordinate system 中给出的。_J2000 和 _B1950 的 measurements 是自动考虑到参考系的变化, 从本地 WCS 转换得到的。这三种情况下, positions angles are counted East-of-North.
- **_FOCAL.Measurements** 将以 focal plane coordinates 的形式给出, 这实际上是投影坐标系, 以 degrees 为单位。也要求 FITS 的头文件里有 WCS metadata。用 pixel coordinates 计算 focal plane coordinates 的方法与 _SKY 坐标类似, 只是它们不进行消投影 (deprojected), 保持笛卡尔坐标。focal plane coordinates 的主要作用是为镶嵌相机的所有芯片提供一个通用的系统。

参考文献

- [1] An Introduction to Astronomical Photometry Using CCDs, W. Romanishin, University of Oklahoma