SimCCD Manual

V2.1.0

Hilary Jones

06/05/12

Table of Contents

Preface

One of the first things you might have tried as a newcomer to astrophotography is taking test pictures to see how important sly glow is, how important long exposures are, and so on. You will probably have found that these tests can take a long time and end up being inconclusive if the conditions vary between one set of tests and another. Moreover, if you evaluated the tests by "eyeballing" the images instead of measuring them, the results may not have been meaningful. As an example, suppose the seeing changes between one set of tests and another. If your image display software has an automatic contrast adjustment tool, it might boost the contrast of a poor image, which could make it seem noisier than a good image even if they are actually very similar. It is only with careful measurement that you can tell whether the differences are real. Even then, measuring the noise level can be hard because it is difficult to tell whether variations in the image are due to noise or actual structure in the target.

SimCCD solves these problems by simulating the operation of an astronomical CCD camera. Now you no longer have to take your equipment out into the field to run laborious and inconclusive tests. The program allows you to do these studies in the comfort of your den. You can choose from among several cameras, telescopes, filters and other optical accessories to see how they may affect an astrophoto. You can also choose the conditions for the simulation, such as the exposure time, camera temperature, number of images, number of darks, number of flats, sky brightness, etc. You can simulate any one of several targets using the data files supplied with the SimCCD distribution kit and on the SimCCD web site; but you can also simulate any other target for which intensity data is available in FITS format. In particular, SimCCD can simulate any part of the sky that has been photographed by the Sloan Digital Sky Survey (http://www.sdss.org). The program can also read the FITS files produced by popular programs such as MaxIm DL and CCDSoft; so you can measure your own test exposures.

SimCCD displays a numerically accurate simulation of your target on the computer screen. The simulation takes into account most of the physical phenomena that can affect image quality, such as readout noise, shot noise from the sky and target, dark field current, bias frames, flat fielding, etc. You can save the simulation in FITS format and then measure it using the tools in other programs. But more important, the latest version of SimCCD lets you measure your images directly using its statistics command. This powerful feature gives a recommendation of how many exposures to take to reduce the noise to an acceptable level, and tells you how long your session will be.

History

I wrote SimCCD over a period of a several years, starting as a "hobby project". One of my goals was to model different CCD cameras to see if one was any more sensitive to light than another, and if so, how important that was in practice. I was going to use SimCCD to guide my next big purchase, but as luck would have it, I got impatient and bought the camera first!

I was also motivated by a desire to understand the importance of using long exposures and dark skies. I had tried to evaluate these issues using careful experiments, but I too found it hard to control the observation conditions well enough to come up with useful conclusions. Moreover, the experiments took too much time. SimCCD solved those problems by allowing me to perform a simulation in a matter of seconds instead of the hours it would have taken to run the corresponding experiment.

Last but not least, I wanted to have a tool for deciding what the total exposure time should be to give an acceptable image of any target. SimCCD lets me do this because of the easy availability of high quality images from the Sloan Digital Sky Survey project. And now that SimCCD can evaluate your own exposures, you don't have to be concerned when your target hasn't been photographed by SDSS. Observers in the Southern Hemisphere take note!

While SimCCD wasn't specifically designed to be an educational tool, I believe it will serve very well to help students and newcomers to astrophotography understand the issues that influence the quality of their images.

Finally, I have to thank my wife Susan, who let me work on SimCCD instead of finishing all those projects around our house! I also want to thank Gert Gottschalk for evaluating SimCCD and giving me some new ideas.

Please enjoy SimCCD, and if you find it useful, drop me a note!

Thanks,
Hilary D. Jones

hdjones@pacbell.net
simccd@darklights.org

http://www.darklights.org/simccd
http://tech.groups.yahoo.com/group/SimCCD

Chapter 1 - Overview

SimCCD is designed to give numerically accurate simulations of photographs of astronomical targets. You can use the simulations for a variety of purposes:

- Planning a session to minimize noise and reveal desired features.
- Finding how many exposures are needed to reduce noise to an acceptable level.
- Understanding how readout noise affects an image.
- Understanding how sky brightness affects an image.
- Understanding how short exposures increase noise.
- Finding how many darks are required to minimize noise.
- Finding how many flats are required to minimize noise.
- Seeing what happens if dark and/or flat field corrections aren't done.
- Understanding how important it is to take “flat darks”.
- Understanding how cooling affects noise.
- Seeing how different cameras respond to the same target.
- Seeing how responsive a camera is to different colors of light

SimCCD has built-in support for several cameras, including several Apogee, FLI, SBIG and QSI cameras. It also has built-in support for several telescopes, including some manufactured by Celestron, Meade, PlaneWave, Takahashi and RCOS. SimCCD also supports several filters and optical aids such as Barlows and focal reducers. In addition, you can add new cameras, telescopes, filters and optical aids using dialogs built into the program. SimCCD stores new equipment definitions in an XML file, which you can transfer to another computer if needed.

I encourage you to submit definitions for new kinds of equipment so I can include them in future releases of the program. To do this, use the Export command to create an XML file. You can then email it to me at simccd@darklights.org, or post it in the Files area of Yahoo’s SimCCD discussion group. Another way to do this is to send me a copy of your settings file `SimCCD Settings.xml`. It can be found in `C:\Users\YourName` on Vista or Windows 7, and in `C:\Documents and Settings\YourName` on Windows XP, where of course you should replace `YourName` with your own name.

The full version of SimCCD is distributed with a number of images from the Sloan Digital Sky Survey (SDSS), including M13, M34, M51, M97, and a half dozen NGC images. You can easily download other source image files from the SDSS web site, and SimCCD can then use these files without any extra effort on your part. You can also obtain source images from other places too, provided they contain numerically accurate values for the target’s intensity. SimCCD only requires that you know the characteristics of the camera, telescope, filters and optical aids that were used to acquire the data.

While SimCCD’s simulations will be best if you use SDSS data sets, the latest version of the program has been designed to give you useful information from a single trial exposure using your own equipment. This can be especially important if you live in the Southern Hemisphere, where SDSS coverage is sparse.

Throughout this manual we will sometimes refer to the “source conditions” or the “simulation conditions”. It will help you to keep in mind the difference between these two ideas. SimCCD works by taking image data from a “source” file and using it to produce another image that represents a simulation of the source data. Since the source and simulation conditions don’t have to be the same, the manual uses the terms carefully to make it clear which is being referred to.

Prerequisites

You do not need to have a lot of experience with astrophotography to run SimCCD. But it would help if you’ve had given some consideration to choosing exposure times to minimize noise. It also wouldn’t hurt for you to know what an Analog Digital Unit (ADU) is; and it would also help if you had some idea of what readout noise is, and why it’s important to use long exposures to minimize it. You can find an excellent introduction to some of these topics at http://starizona.com/acb/ccd/advtheoryexp.aspx.

SimCCD can run on Windows XP, Vista and Windows 7. It can also run on any version of MacOS, Linux, or UNIX that supports Java 1.6. If Java has not already been installed on your machine, you will have to do it first. The procedure for doing this is described in the next chapter.

SimCCD requires a screen size at least 1280x1024. It can be run on smaller screens, but some of the larger windows may be unusable.

SimCCD requires a fairly powerful processor to perform most simulations. On an older machine running Linux using a 950 MHz AMD Athlon processor and 256 MB of memory, SimCCD V1.0.1 computed a simulation of M51 in about 30 seconds. On a new machine running Vista using a 2 GHz Intel Core Duo processor and 2 GB of memory, the same simulation required 3 seconds. For some equipment choices and images the simulation times can be considerably longer – sometimes taking as much as a minute or two.

> Because simulations might take a long time, do not be surprised if SimCCD doesn’t seem to be doing anything for a while!

Reading advice

I suggest that you continue by reading the next chapter, which gives you a cookbook on how to run SimCCD. After you have read that chapter, you may feel like SimCCD is more than you want to master right now. In that case please don’t give up! Instead, we suggest you turn to Chapter 6, which describes a very simple MaxIm DL plug-in called “Exposure Planner”. This plug-in will help you determine the optimum exposure time for any target based on just a single trial exposure. Please try this plug-in, and if you find it useful, then perhaps you will want to give SimCCD a second look and try out some of its more advanced features. To learn more about them, check out chapter 5, which gives some case studies showing SimCCD in action.

After you read Chapter 2's cookbook, read Chapter 3 to find a complete description of SimCCD's commands.

You may be able to skip chapter 4 for a while, since it discusses some of the fine points on how to run SimCCD.

If you want to know what's new in this release of SimCCD, read Appendix A. There are also some technical notes about how SimCCD works in Appendix B.

Chapter 2 - Cookbook

In this chapter we will go over the basic steps for installing and running SimCCD. For a detailed description of SimCCD's commands, refer to the next chapter.

This chapter covers several topics:

- Installing and starting SimCCD
- Opening a source image
- Specifying equipment for the source image
- Specifying equipment for the simulation image
- Picking exposure conditions
- Getting image statistics
- Changing zoom and contrast settings
- Saving a file
- Creating new kinds of equipment

You may also want to read Chapter 6 now, which discusses the Exposure Planner plug-in for MaxIm DL.

Installing and starting SimCCD

I wrote SimCCD using the Java language; so before you can run the program, you must install the Java Runtime Environment version 1.6, which is available from http://java.sun.com/j2se. I developed and tested SimCCD using Microsoft Vista and Windows 7; but I have also done some limited testing on a Linux machine. The program will work with Windows XP, Vista, Windows 7, Linux, and MacOS (with some caveats that are described in the Fine Points chapter.)

After you have installed Java on your computer, you are ready to install SimCCD. If you received the distribution kit on a CD, SimCCD's files are ready to go; but copying them to your hard disk will make the program run faster. On the other hand, if you received the distribution kit in a zip file (`SimCCD.zip`), you must unpack it first. SimCCD doesn't have any particular expectations about which folder you unpack the files into, so put them anywhere you like.

If you have used SimCCD before, you can just copy the new files into the old folder. However be sure to delete `Run SimCCD.bat`, which is now obsolete; and be sure to use the latest version of `fits.jar`.

Warning: you MUST unpack the zip archive. Windows will try to present zip archives as if they were ordinary folders. But when you double click the startup script, it will complain that "Java may not be installed" even if it is. To unpack the zip archive, right click on the installation folder. Its icon will look like an ordinary folder with a zipper running through it. Select the Extract All command, and follow the instructions. Macintosh and Linux users will have to unpack the archive too.

The distribution kit contains the following files and folders:

```
Image Data
        A folder with one or more source image datasets
Debug SimCCD.bat
        A batch script for debugging startup problems
fits.jar
        A patched version the fitsio V1.04.0 library
Install Exposure Planner V1.2.exe
        A program to install the Exposure Planner plug-in for MaxIm DL
linearity.xls
        An Excel spreadsheet to analyze the linearity of an SBIG ST-10XME
manual.pdf
        The document you are reading now
monitorLightSource.vbs
        A script for monitoring the intensity of a light source
Readme.txt
        A short description of SimCCD
Run SimCCD 1.5GB.vbs
        A script to run SimCCD when it needs more memory
Run SimCCD.vbs
        A script for running SimCCD in normal mode
run_simccd.command
        A double-clickable command script for running SimCCD on MacOS
run_simccd.sh
        A shell script for running SimCCD on Unix
runSequence.vbs
        A script for evaluating the linearity of a CCD
SimCCD equipment.xml
        Equipment definitions
SimCCD.jar
        The SimCCD application
xstream-1.4.1.jar
        A support library that reads and writes XML files
```

The `Image Data` folder may contain one or more datasets, depending on how you came by the distribution kit. The abbreviated distribution kit just contains one image of M51, while the full distribution kit contains images of several targets. As we will see later, these images are used as input to SimCCD.

SimCCD uses a patched version of HEASARC's FITSIO library to read and write FITS files. The patched version is based on V1.04.0 of library. The library is contained in the file `fits.jar`. You can find more information about the Java version of FITSIO at http://heasarc.gsfc.nasa.gov/docs/heasarc/fits/java/v1.0.

SimCCD also uses the XStream library to read and write XML files. You can learn more about this library at http://xstream.codehaus.org.

To run SimCCD, just double-click on `Run SimCCD.vbs`. You will be greeted by the startup screen shown in Figure 1. If SimCCD does not start, double-click on `Debug SimCCD.bat`. You should then find some diagnostic messages that indicate why the program didn't run.

> If you are using a 64-bit Windows system and can't get SimCCD to run, see the Fine Points chapter for special instructions.

If you are running on a Macintosh or UNIX system, start SimCCD using the scripts `run_simccd.command` and `run_simccd.sh`, respectively.

You control SimCCD by clicking menu items at the top of the screen. When you start a simulation, the following startup screen will be replaced with one that shows the simulation result.

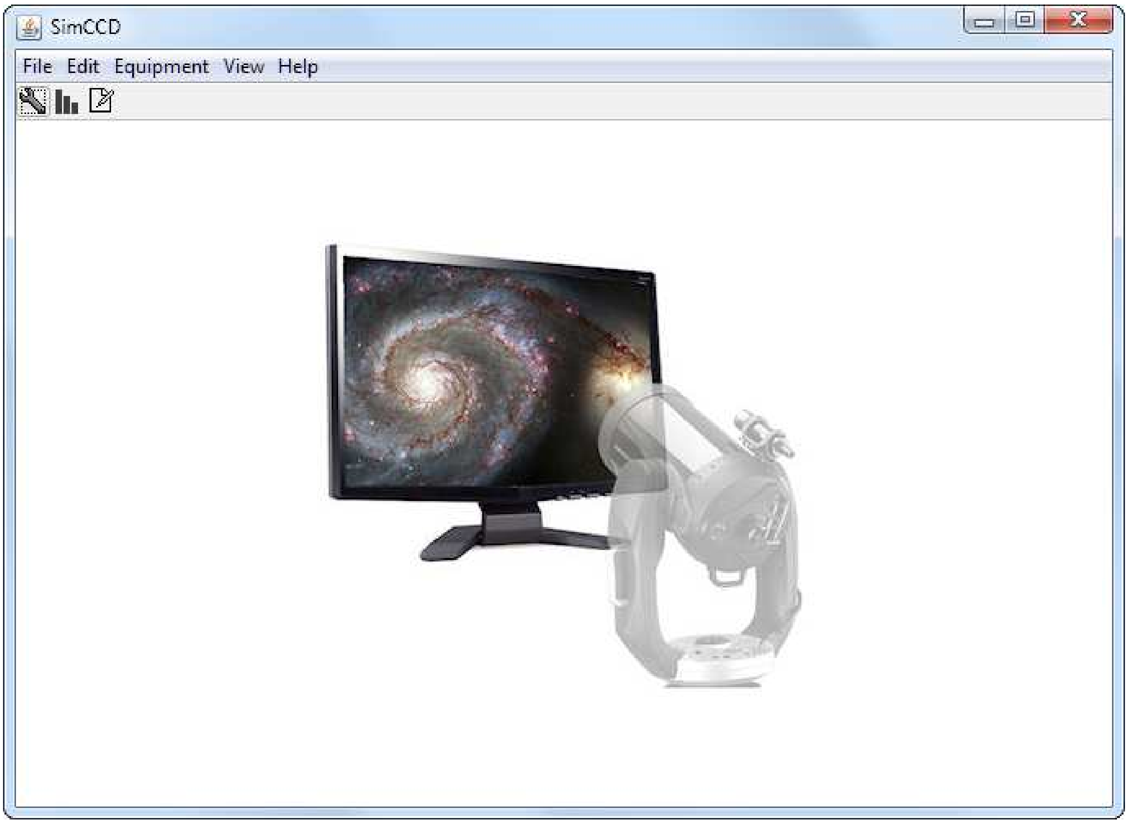


Figure 1 SimCCD's startup screen

Opening a source image

In order to run a simulation, you must first open a FITS file that contains numerical data describing the target you want to simulate. The data must be a monochrome image that gives ADU values for every pixel in the source. To open the file, simply select the File/Open menu item and navigate to a folder where the source images are located. For our cookbook example, we will navigate to the folder named `M51 from SDSS`, which is in the folder named `Image Data`. This folder is included in the SimCCD distribution kit. For the following discussion, we will choose the image named `fpC-003699-r6-0100.fit`, which is an image prepared by the Sloan Digital Sky Survey (http://www.sdss.org) using a red filter.

Specifying equipment for the source image

In order for SimCCD to do a simulation, it must know something about the equipment that was used to prepare the source image. After you have chosen the source file, SimCCD will prompt you with the dialog shown in Figure 2.

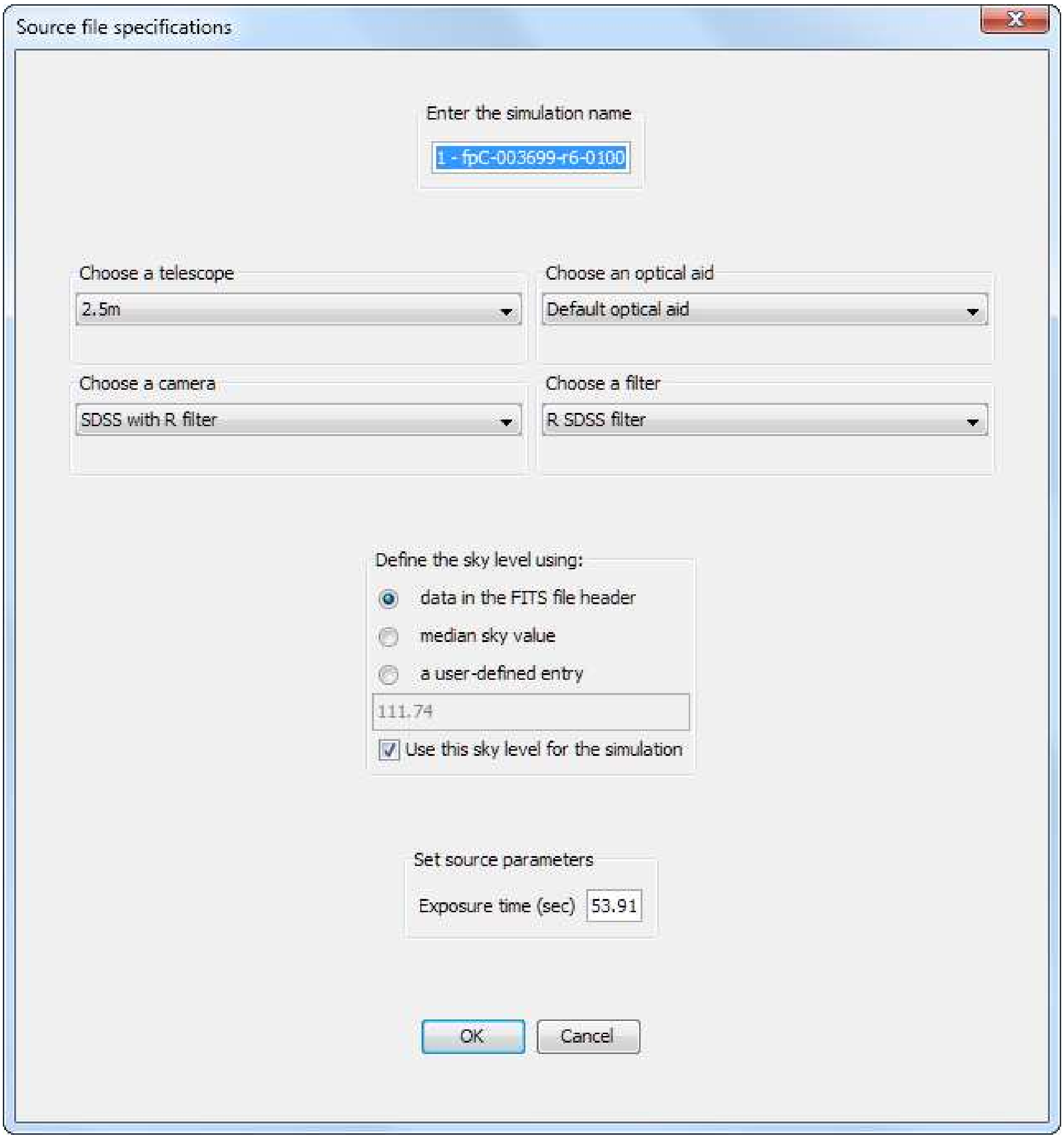


Figure 2 Source file specifications

The fields may not look like this at first, but for this particular image of M51, you should change them to match the values shown in the figure. The exposure time should already be 53.91 seconds, which is a value that SimCCD found in the FITS header; and the telescope should already be 2.5m, which is built into SimCCD and has the characteristics of the SDSS telescope. You may need to select the camera named SDSS with R filter, which defines the camera that the SDSS project uses for the red filter; and you may need to select the filter R SDSS filter, which is a description of that filter. No Barlow of Focal Reducer was used to prepare this image.

Specifying equipment for the simulation image

After you have pressed the OK button, an image of the target will be shown in a few seconds.   This image may or may not be based the equipment you wish to simulate. To fix this, you should use the control panel dialog. Normally it will be shown alongside the image itself, but if it is not present, use the menu item View/Control Panel to make it visible.   As an alternative, you can press the control panel icon in the upper left hand corner of the window.

If necessary, select the Equipment tab, so that the window looks like this:

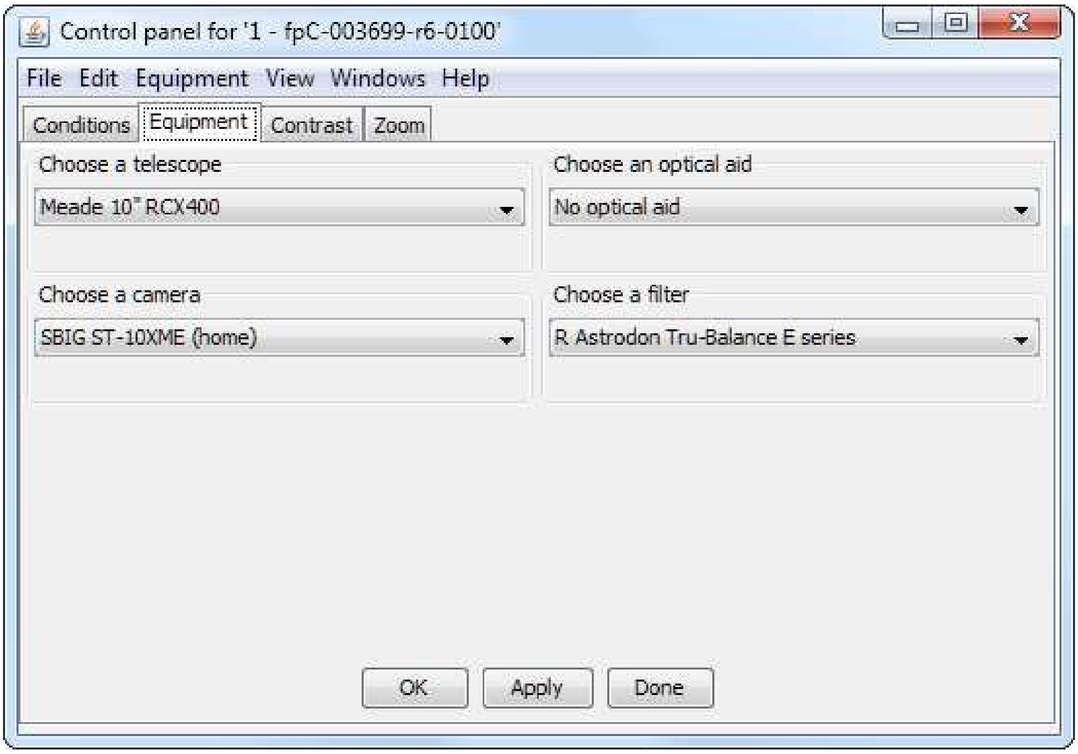


Figure 3 Choosing the simulation equipment

For our example, set the fields as shown. In particular, use an SBIG ST-10XME camera. SimCCD has several versions of that camera and any one of them will work; but we will be showing a simulation based on the "home" version of the camera. You should also choose the R Astrodon Tru-Balance E Series filter. This is a red filter that has a similar band pass to the SDSS red filter that was used to acquire the source data. When you have selected the fields as shown, you can press the Apply button to see a simulation that uses the selected equipment.

Picking exposure conditions

With the control panel still visible, select the Conditions tab, as shown in Figure 4.

For the following discussion, set the fields to the values shown above. You may not be able to enter the exact values shown because of the way SimCCD coordinates the field values, but you should be able to come close. Note that changing one field may cause another field to change out from under you; so be sure to double check that you have entered all of the values correctly. When you are satisfied, press the Apply button. You should now see an image that looks like the one in Figure 5 below.

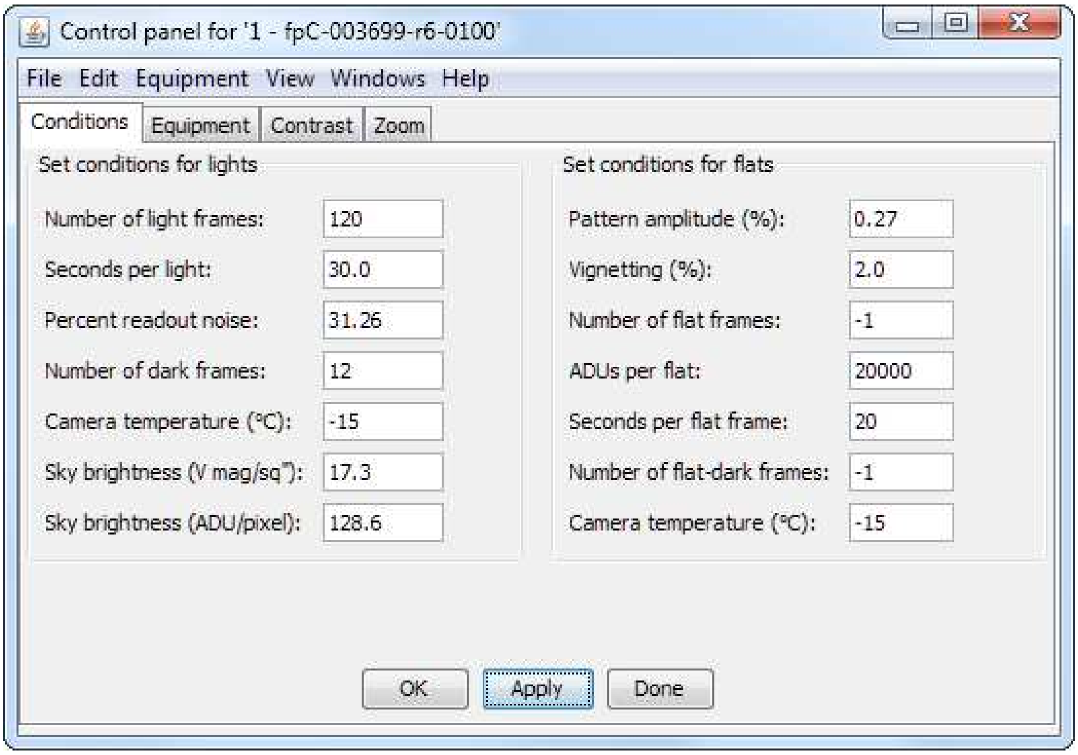


Figure 4 Controlling the simulation conditions

Notice the Show Source button at the bottom of the image. If you press it, you will see what the original source image looks like. It will have considerably less noise than the simulated exposure. Pressing the button a second time will show the simulation result again. You can use the button to get an idea of whether you have used enough exposures to reduce noise to an acceptable level.

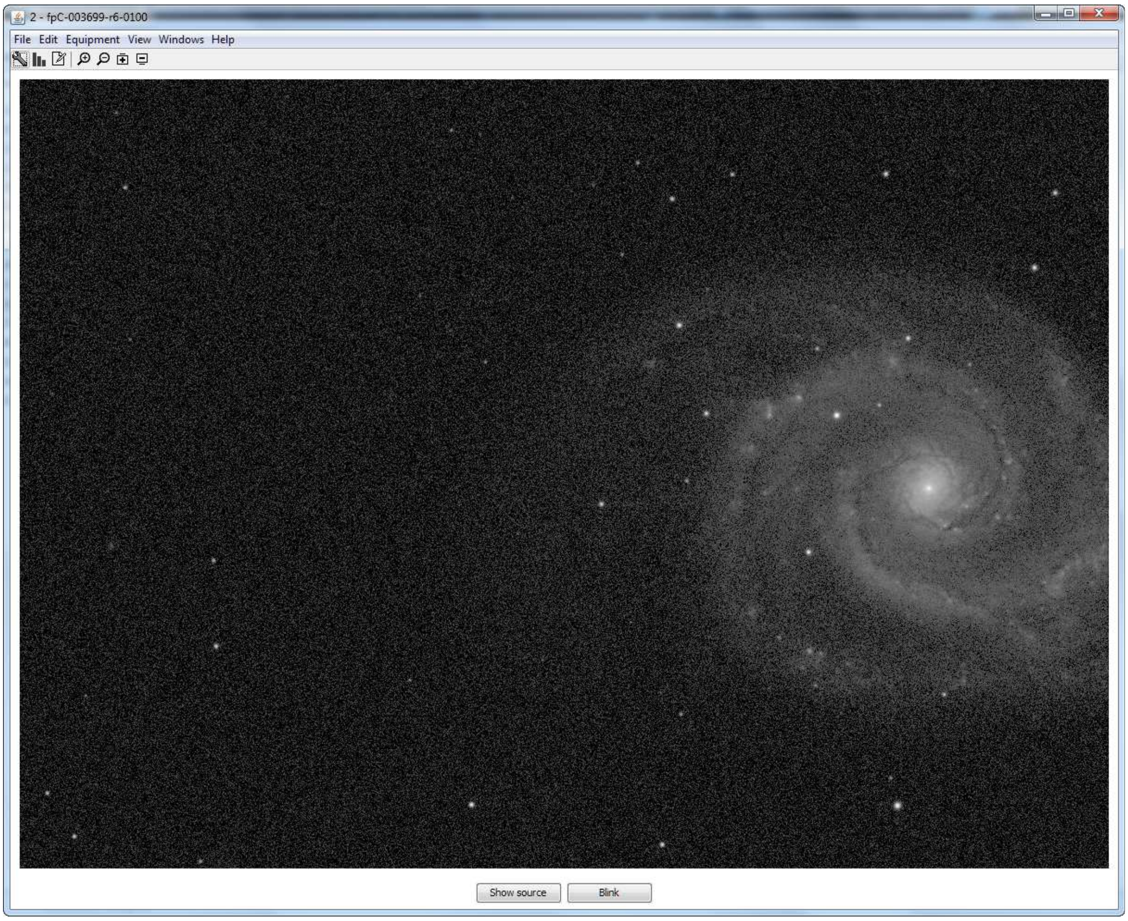


Figure 5 A simulation result

Also notice the Blink button. If you press it, the display will start alternating between the source image and the simulation result, which will let you see how much noise has been added by not using the best conditions. A control window will also be displayed that lets you control the blink rate. The window looks like this:

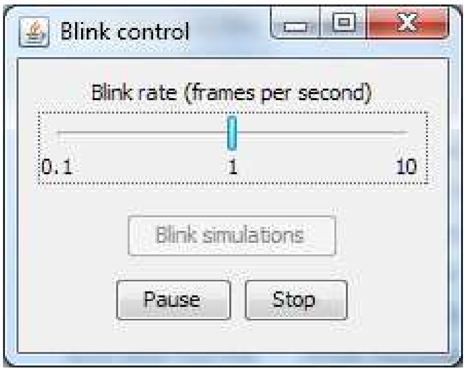


Figure 6 The blink controller

I did the simulation in figure 5 using an hour's worth of exposures and a very short exposure time of 30 seconds. Try changing the number of light frames to 12 and the exposure time to 300 seconds, so that the total exposure time is still one hour. As you do this, notice that the field labeled Percent readout noise changes from 31.26 to 3.551. This tells you that by using the longer exposure time, you have reduced the readout noise by a substantial amount; and when you press the Apply button, you will see that the simulated image is indeed noticeably better.

As another experiment, change the Percent readout noise to 5 and press the Apply button. Notice that SimCCD changes the Seconds per light figure to 211.6. This is SimCCD's way of telling you that if you want to reduce the readout noise to 5% of the total, you will need to use exposures that are 211.6 seconds long. (This feature is exactly the same as sub-exposure calculators at www.ccdware.com/resources/subexposure.cfm and http://starizona.com/acb/ccd/calc_ideal.aspx).

This would be a good time for you to try some of SimCCD's other features. For example you should try to change the camera's operating temperature, the sky darkness, the number of darks, flats and/or flat darks, etc. We will describe these features in more detail in the next chapter.

Getting image statistics

SimCCD lets you get statistics about an image. To do this, either use the View/Statistics Window command or just drag your mouse across some interesting part of the image. A window will pop up that looks like this:

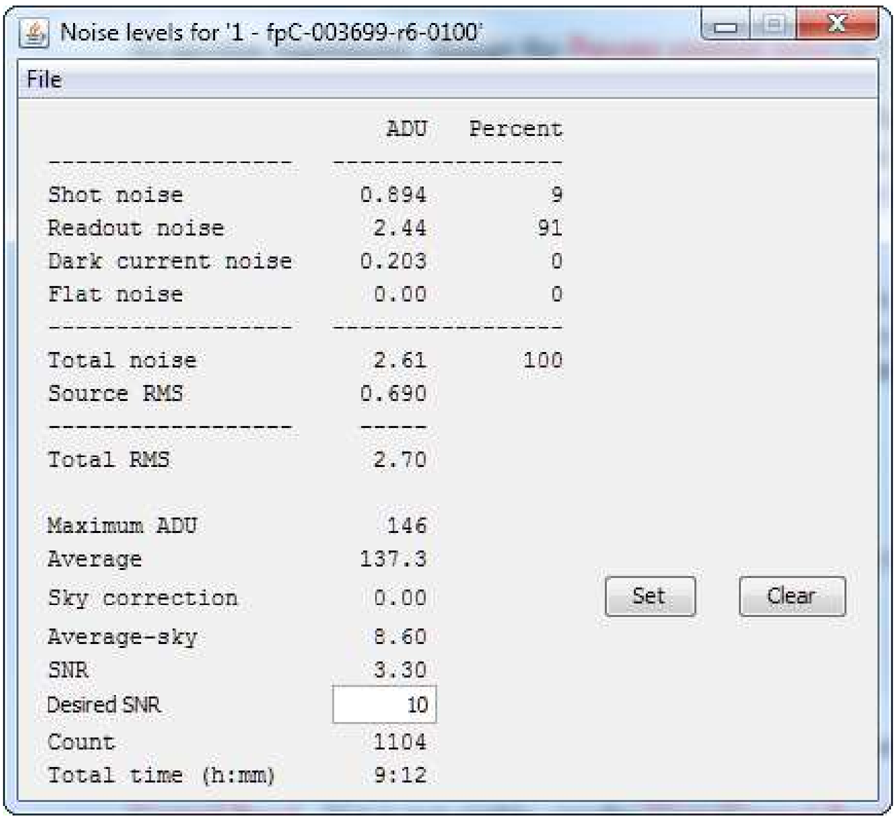


Figure 7 The statistics window

The values that are reported will depend on what part of the image you selected. For our example, we selected one of M51's spiral arms. The statistics window tells us that it will take a little over 9 hours if we want the signal to noise ratio to be 10 in this part of the image, and that most of the noise (91%) is due to readout noise. As an experiment, try increasing the exposure time from 30 seconds to 300 seconds. You will find that the total time is reduced by a factor of four, and the readout noise is much smaller percentage of the total. As another experiment, select a brighter part of the image. You will find that the total session time goes down again. Finally, try changing the Desired SNR to 3. This will reduce the session time by a factor of 9; but of course you will be getting noisier images!

Changing zoom and contrast settings

You may want to change the size of the image so it fits the screen better; or perhaps you will want to zoom in on a feature to look at it in more detail. You can do this using the control panel. If it is not visible, use the View/Control Panel menu to bring it up, and then select the Zoom tab. The panel is shown in Figure 8.

You can use the slider to continuously adjust the size of the image, or enter a zoom factor manually. You can also use the Fit image to window and Fit image to window buttons to do what they say…

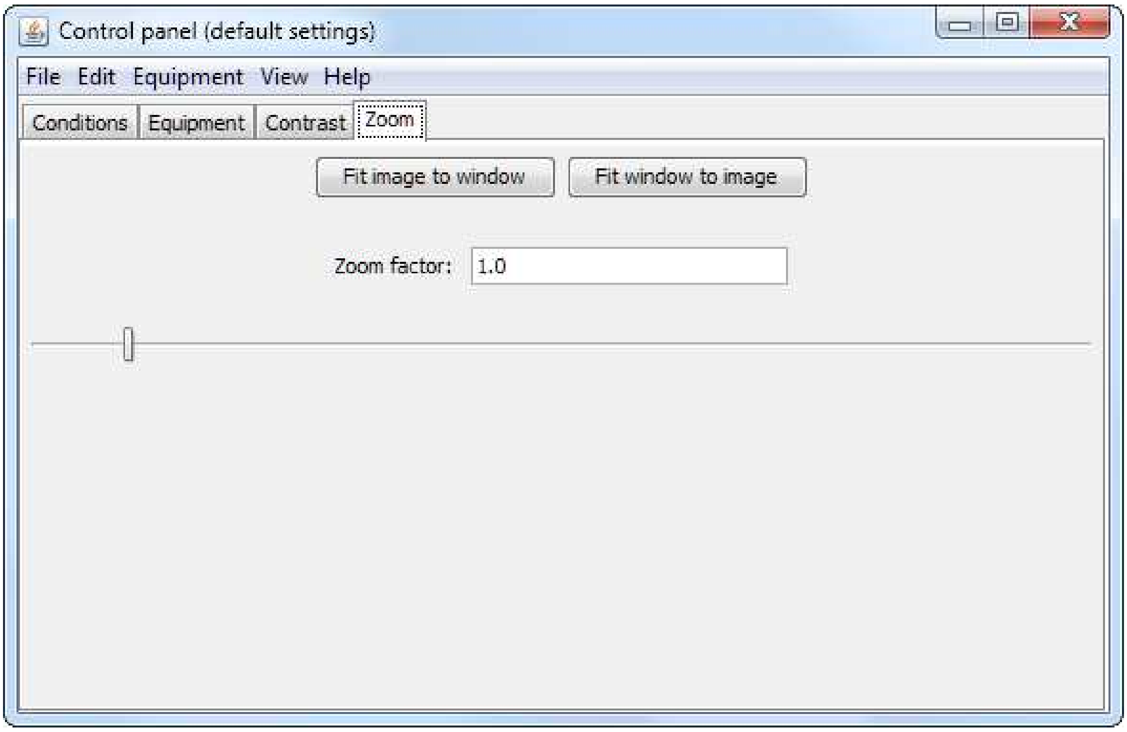
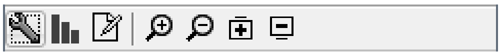


Figure 8 The zoom control

You may find it more convenient to use the zoom icons on the button bar at the top of the Simulation Image. The button bar looks like this:



You can use the icons  and  to zoom in and out without changing the screen size, and you can use the icons  and  to have SimCCD change the screen size to fit the zoomed image.

If the image is too dim or too bright to see the features you want, use the Contrast tab, which looks like this:

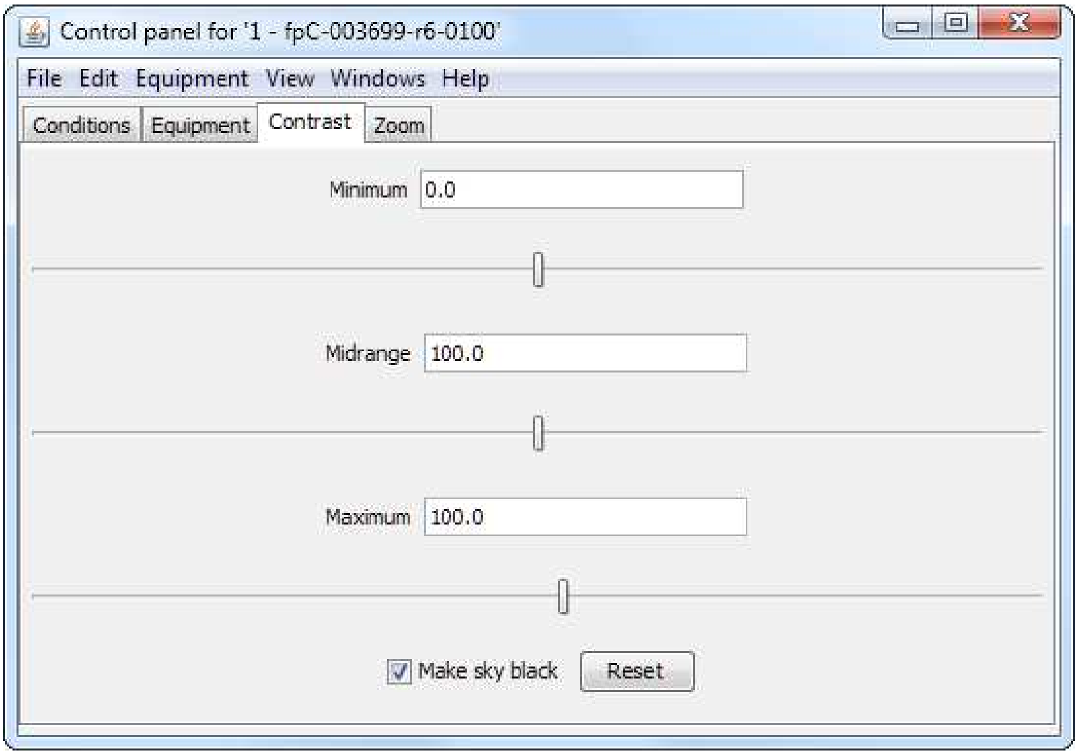


Figure 9 The contrast control

The Minimum slider determines the lowest intensity that can be displayed, while the Maximum slider determines the highest intensity that can be displayed. The Midrange slider allows you to enhance the contrast of medium intensities. The Make sky black checkbox determines whether the sky’s brightness is shown realistically or is displayed as black to increase the contrast.

Saving a file

Once you have finished running a simulation, you may save the result in a FITS file using the Save command under the File menu. This works the way the save command works in other applications – you will be presented with a dialog that lets you pick the name of the file and folder where the file should be saved.

Note that you can open the resulting file in any program that can read FITS files -- for example MaxIm DL, CCDStack, and AIP4Win.   You could then use these programs to measure the noise level in the sky and compare it with your own photographs. You could also use the programs to combine simulated RGB frames to create a full-color picture.

Creating new kinds of equipment

SimCCD has some built in cameras and telescopes that are commonly used; but if you don't find a piece of equipment you are using, you can create it. For the following discussion, we will create a camera named Cookbook Camera. To do this, use the Equipment/Camera menu item. You will get a dialog like this:

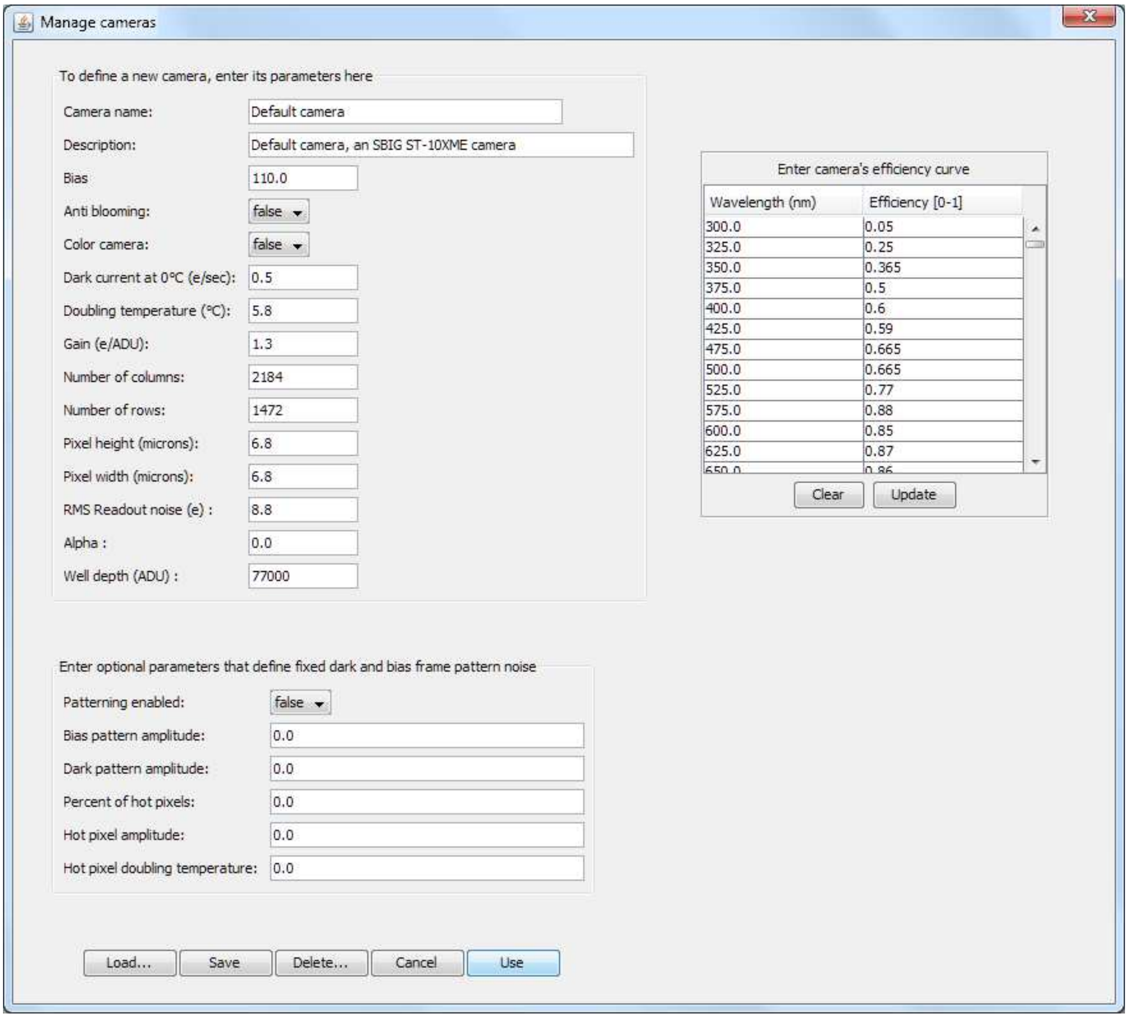


Figure 10 Managing cameras

The dialog probably won't look exactly like this to start with, so use the Load button to find the Default Camera. Next edit the fields using published information for the camera of your choice. Then change the Camera name field to read Cookbook Camera, and enter a description of the camera in the Description field. We won't discuss the Patterning enabled field now; suffice to say that its use will require you to make some sophisticated measurements of your camera. Therefore for the time being you should leave this field set to false. We will discuss how to set up the related fields in Chapter 4.

Be sure to change the values for the camera's efficiency curve. SimCCD will use the curve to model how the camera works with different filters, etc. To define an efficiency curve, press the Clear button first to delete all of the entries in the table, and then enter the new wavelength and efficiency values. If the table gets messy, press the Update button to clean it up. (In particular, this will sort the values in order of wavelength.)

When you are done entering the parameters, press the Save button; and then assuming you actually want to use this new camera in a simulation, press the Use button.

The process for defining new telescopes, optical aids and filters is the same, so we won't discuss them here.

> A word of warning is in order. SimCCD has built in definitions of many CCDs based on literature from the manufacturer. Manufacturers usually publish nominal values for the camera's parameters. But each camera is different, and cameras do change with time; so SimCCD's definitions may not apply. This is particularly true for the dark current level, where different manufacturers can publish values that differ by a factor of ten or more for cameras that use the same CCD chip. Therefore if you want SimCCD's simulations to be accurate, you should measure the values yourself.

Chapter 3 - Command Reference

In this chapter, we will describe each menu command in detail. Section heads list the main menu name followed by the menu item. For example, the Open command that appears in the File menu is shown as *File/Open*.

File/Open

Use this command to open source files. These are the files that contain source data that defines a target that is to be simulated. When you use this command, SimCCD presents a dialog that looks like this:

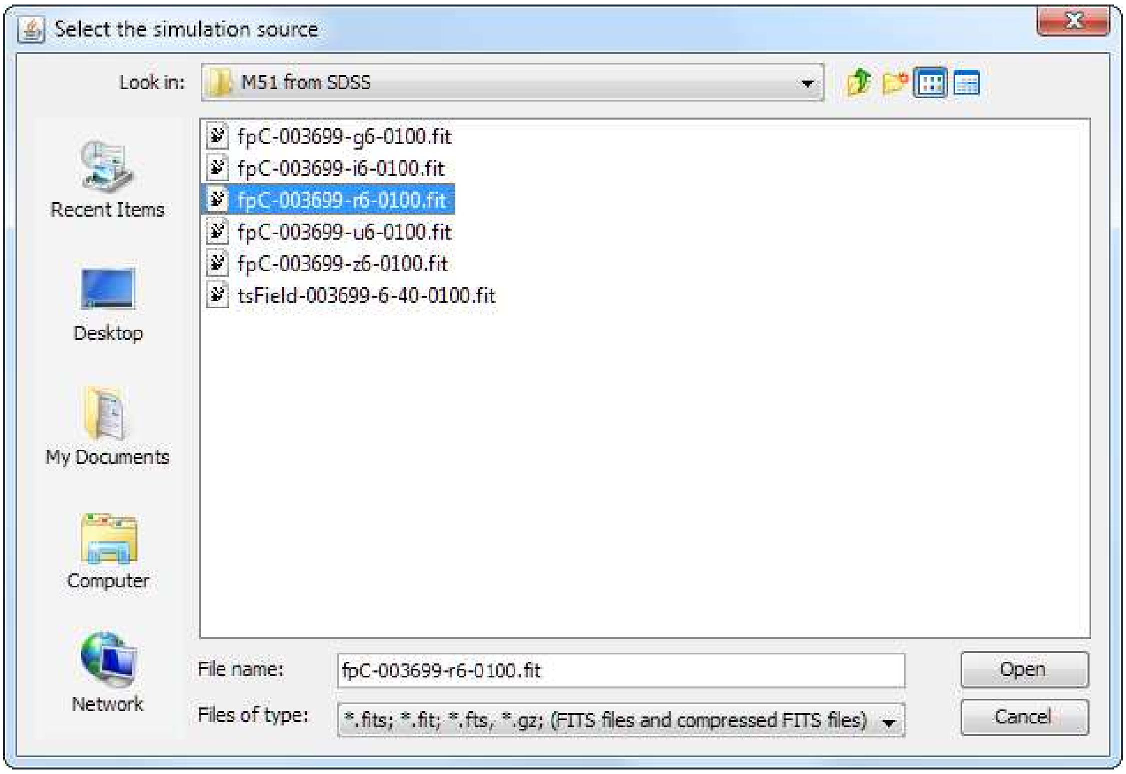


Figure 11 Opening a file

This example shows files from the Sloan Digital Sky Survey for M51. These files have cryptic names like `fpC-003699-r6-0100.fit.` (The value 003699 refers to the run number. The letter `r` refers to the filter. The value `6` refers to the camera column ("camcol") and the value `0100` refers to the field.) Figure 11 shows (from top to bottom) files for green, infrared, red, ultraviolet and far infrared. It also shows a file named `tsField-003699-6-40-0100.fit`. This is not an image file, but rather a file that contains data describing the other files. We will discuss this file in the next chapter.

After the image is opened, SimCCD will read the FITS header to see how many images are present in the file. If there are several, you will be asked to choose which image you want to simulate using a dialog like the following:

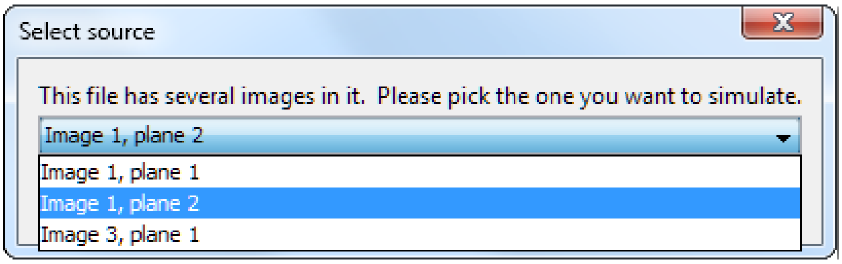


Figure 12 Selecting one of several images

The pull down menu shows all of the images in the file. If the primary header data unit (HDU) of the file contains an image, it will be labeled Image 1. Extension HDUs will be labeled Image 2, Image 3, etc. If an image contains several colors, they will be identified as plane 1, plane 2, etc. These names are admittedly cryptic; but SimCCD has no way of telling you what each image contains; so you will have to know the layout of the FITS file to make sense of the dialog.

If the file only has one image, that image will be chosen and the dialog will not be shown. (This will be the case for SDSS images.)

Next SimCCD reads the FITS headers to see if it can learn what telescope, camera, filter, and/or optical aid were used to prepare the image. Then it will present a dialog that asks you to confirm that those values are right and/or change them. The dialog is shown in Figure 13.

The exposure time is displayed in this dialog based on the EXPTIME or EXPOSURE fields in the FITS header. This should be the exposure time for a single exposure. If the image contains a stack of several exposures, EXPTIME is usually set to the exposure time for just a single picture. But some programs will set it to the product of the frame time and the number of frames (given by the SNAPSHOT field in the header.) In that case, you may need to manually enter the time for a single frame. The log message window reports values for these keywords.

By default SimCCD chooses a simulation name that is a sequence number followed the name of the file. You may want to change this to something that's shorter and more meaningful. You can use any name here; but it is best to avoid characters that wouldn't be part of a legal file name (such as an asterisk, colon, or slash).

SimCCD has several ways of determining how bright the sky is in your source image. It needs to subtract this value in order to know how bright the target is by itself. If the sky level is available in the source file, SimCCD will choose this value. However in many cases it's better to use the median sky value, which SimCCD will measure for you. Or if you have reason to choose a different value, you can enter it manually.

When SimCCD does its simulation, it will need to know how bright *your* sky is too. If you are using one of your own trial pictures for the source data, SimCCD can measure it and use that value to do the simulation. In this case you want to check the box labeled Use this sky level for the simulation. On the other hand if you are using an SDSS image for the source data, you may want to leave the box unchecked, since your sky's brightness won't be the same as the SDSS sky. If you leave the box unchecked, SimCCD will use the brightness that you used the last time you ran it.

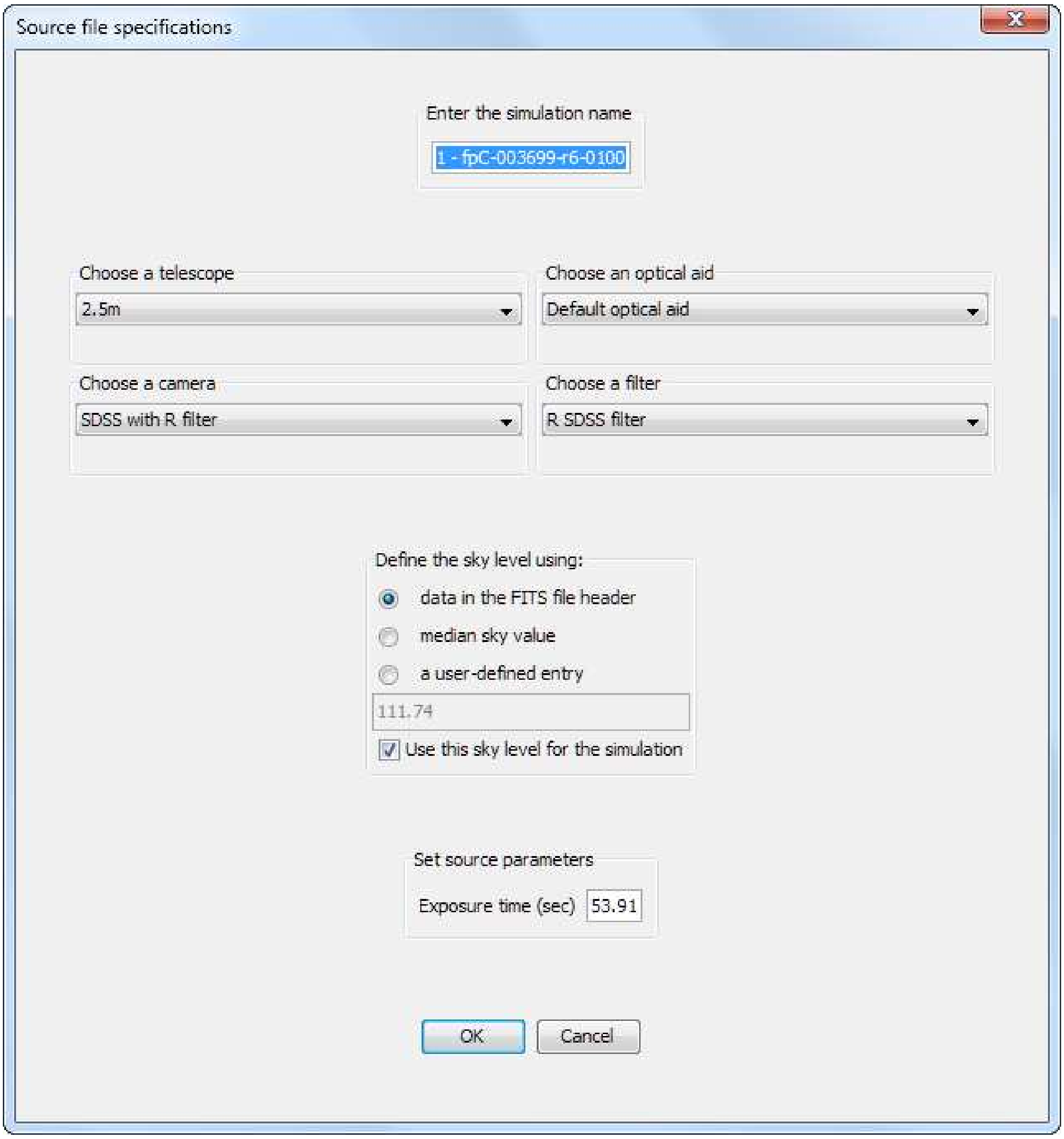


Figure 13 Source file specification

SimCCD can sometimes determine the telescope (2.5m in the example above) and exposure time (53.91 seconds). Based on other information in the FITS header, SimCCD may also try to guess what filter and camera to use. Note that the SDSS project uses a different camera for each filter; so you will have to pick the right camera to go with the filter!

If SimCCD finds that the input file was prepared using a camera it doesn’t know about, it will present a dialog like this:

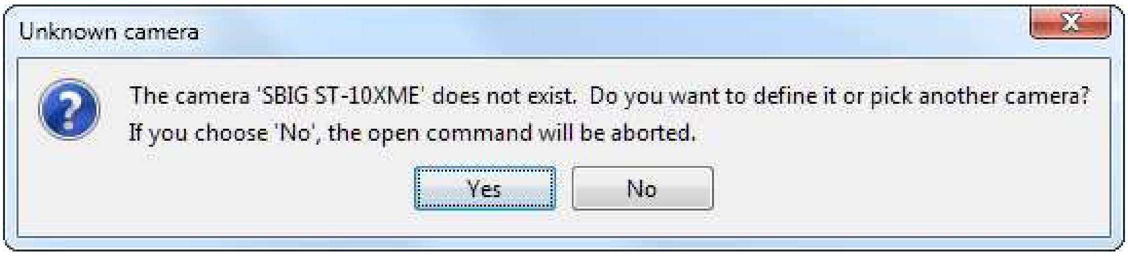


Figure 14 Unknown camera warning

If you press the Yes button, SimCCD will bring up a dialog that’s described later in this manual, allowing you to pick a different camera or define a new one. Typically you will press the Load button in that dialog to select the camera, then press the Use button to use that camera to describe the source image.

The same sort of dialog will be presented if SimCCD finds that the input file was prepared using a telescope or filter that it doesn’t know about.

By the way, SimCCD can open several files at once, displaying the resulting simulations in different windows. It can also open the same file several times, which gives you the opportunity to see how different exposure conditions affect the simulation.

> Warning: Because of the way SimCCD works, it cannot handle color images. So if you have a one-shot color camera, SimCCD’s usefulness will be limited. If you try to open a full-color image, SimCCD will complain. There is a workaround, which is to open the raw un-debayered image. While SimCCD can’t really understand this kind of image, you might be able to use its predictions as rough approximations.

File/Import

Use this command to import equipment definitions that were saved using the export command (described below). Note that the import command will not overwrite existing equipment definitions. For example, if you have defined a camera named “My Camera” and you try to import a revised definition of that camera, SimCCD will warn you that the definition was skipped. If you still want to import the new definition, delete the old definition first.

File/Export

This command allows you to export your equipment lists to an XML file. You should use this command to back up your equipment lists, and in particular, new cameras that you may have defined. You can also use the command to transfer the equipment list to another computer.

File/Save as…

When you execute this command, the following dialog appears:

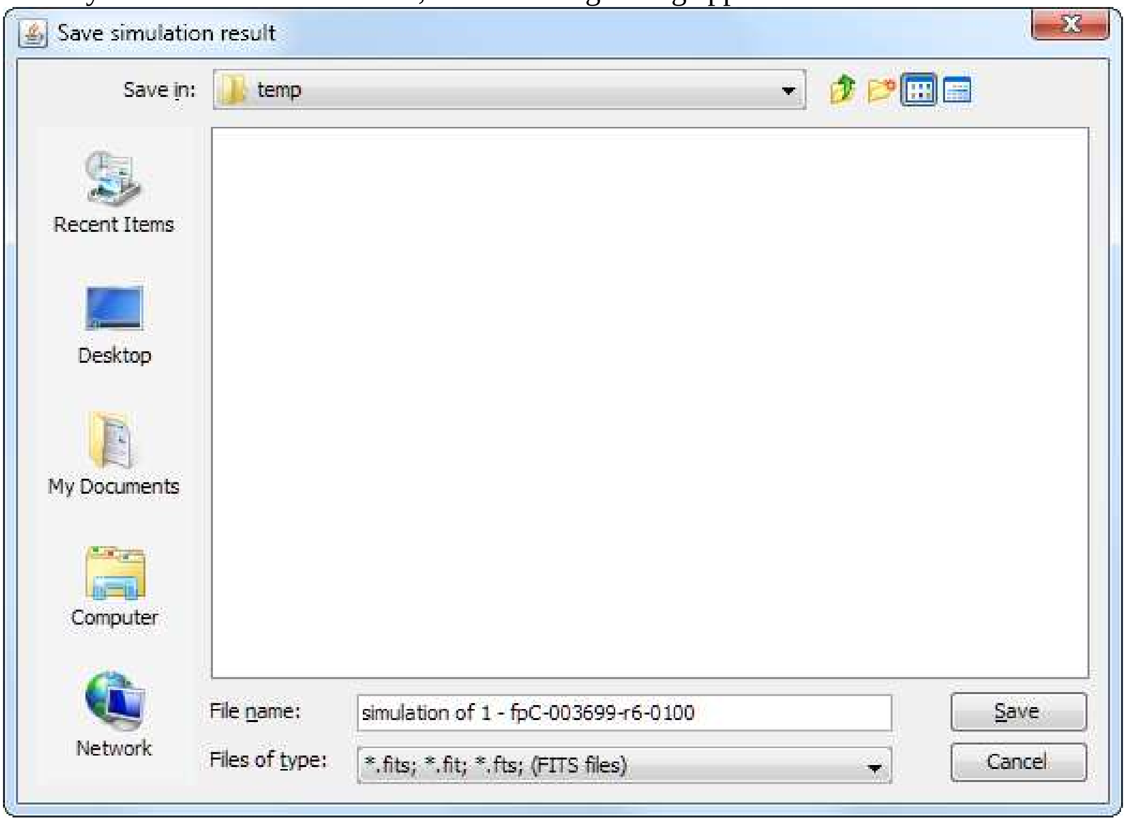


Figure 15 Saving a simulation

You can save files using fit, fits, or fts for the file extension. The default is fits.

File/Close

Use this button to close the current simulation. If several simulations are open, this button will close the topmost simulation and bring some other simulation to the foreground; but if there is only one simulation open, it will be closed and SimCCD's initial splash screen will be displayed. You can also use the "close box" in SimCCD's title bar to close a simulation. However in this case, if there is only one simulation open, it will be closed and SimCCD will exit.

File/Exit

This command causes SimCCD to exit. All windows will be closed. If you haven't saved a simulation, the result will be lost.

Edit/Preferences

When you enter this command, SimCCD will display the dialog shown in figure 16.

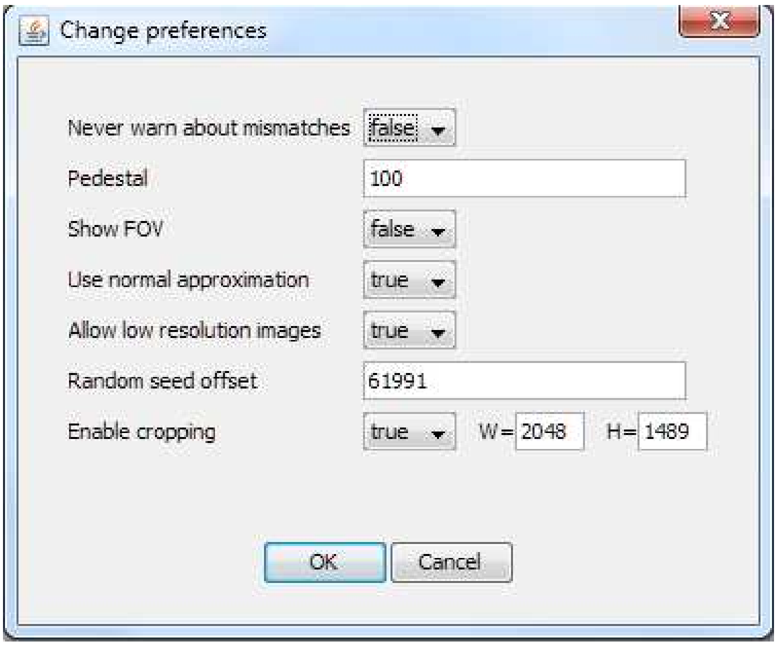


Figure 16 Changing preferences

By default, SimCCD will warn you if the source equipment and simulation equipment are very different. This could happen for example if the band passes of the source and simulation filters don't overlap very much. While you can turn the warning off for the current simulation, it will reappear when you start another simulation. Since this can become a nuisance, SimCCD gives you a chance to turn the warnings off permanently.

When you use the Save As… command, SimCCD adds a default pedestal of 100 to every value. This is the value that is commonly used by many other programs. However if you are going to use another program to measure one of SimCCD's simulations, you should probably choose a larger value. When SimCCD saves a file, it saves negative values as zeroes. Then when you use another program to measure the average or standard deviation in an area, the missing values will throw the calculation off. I have found that a value of 1000 works well for most cases.

By default, SimCCD displays the entire source image, even if it is bigger than the field of view of the simulation camera. On occasion you may want to know how much of that image is actually visible in the simulation camera. Or if the simulation camera is larger than the source camera, you may want to know how much data is missing from the simulation. Setting Show FOV to true will tell SimCCD to display a graphical indication of how big the simulation camera is compared to the source camera. (This feature has not been well tested and may be eliminated in a future release.)

By default, SimCCD does not use true Poisson statistics to model shot noise, but rather uses a Normal (Gaussian) approximation. When a simulation is complete, there will be a certain pattern of speckles in the image that depends on how much noise there is at each point. It turns out that Poisson random number generators depend on the strength of the signal in a way that causes the pattern to change throughout the entire image whenever even one pixel changes. The human eye easily sees this change, making it difficult to judge whether the change is significant or just a rearrangement of the speckles. When the Normal approximation is used, the speckle pattern doesn't move around, and only the amplitude changes. We recommend that you use the Normal approximation. However if you want to be sure that the simulation is accurate, you can turn this option off.

For some situations, SimCCD can use a lot of memory just to display the simulation result. This can cause the program to become very slow or run out of memory. To work around this problem, SimCCD may choose to display a low resolution version of the result. The image won't be quite as accurate as the full resolution version; so if you want to see the full resolution image, change the Allow low resolution images option to false.

SimCCD uses random number seeds to initialize the random number generators. You can control these seeds by changing the random number seed offset. Different seeds cause different speckle patterns. This can be important if you are simulating full color RGB images. By using a different seed for each color, you can be sure that the speckles won't align, which would lead to monochromatic color noise.

If your source image is very large, SimCCD can be quite slow to compute the simulation, and also quite slow to do simple things like zooming an image or changing its contrast. To improve on this, SimCCD will crop large source images down to 2048 x 1489, which is the size of a typical SDSS image. If you find that SimCCD is still running too slowly, you can decrease these dimensions using the Enable cropping settings. Likewise, if you want to see more of the original image, you can increase the dimensions. If you want to disable this feature altogether, set the field to false, which causes the whole image to be displayed. Note however that regardless of this setting, SimCCD will still crop your image if it finds that it doesn't have enough memory to display the whole thing.

When SimCCD crops an image, it just uses the center of your image. If you need more control of how cropping is done, you should do it in another program such as MaxIm/DL before opening the image with SimCCD.

Edit/Rename Simulation

This command lets you change the simulation name. By default the simulation name is based on the name of the file containing the source data; but you can choose a different name using this command. When you enter this command, the following self-explanatory dialog box is displayed:

Figure 17 Changing the simulation name

Edit/Reset Preferences

This command will reset SimCCD's preference items to their default values

Edit/Enter License

SimCCD will operate for a period of time without a license, during which all of the features are enabled. When the evaluation period has expired, some of SimCCD's features will be disabled. To restore them, send a message to license@darklights.org with your name and email address and I will send you a license code. There is currently no fee for the license -- it is just my way of keeping track of who is using the program. Once I have sent you the license code, you can enter in the dialog, which is shown in figure 18.

If you click on the link, SimCCD will take you to a web page where you can request the license.

Figure 18 Entering a license

Equipment/Cameras

This command will display one of SimCCD's most complex dialogs. It is shown in Figure 19. You can use it to define new cameras, modify or delete old ones, and choose a camera to use for a simulation.

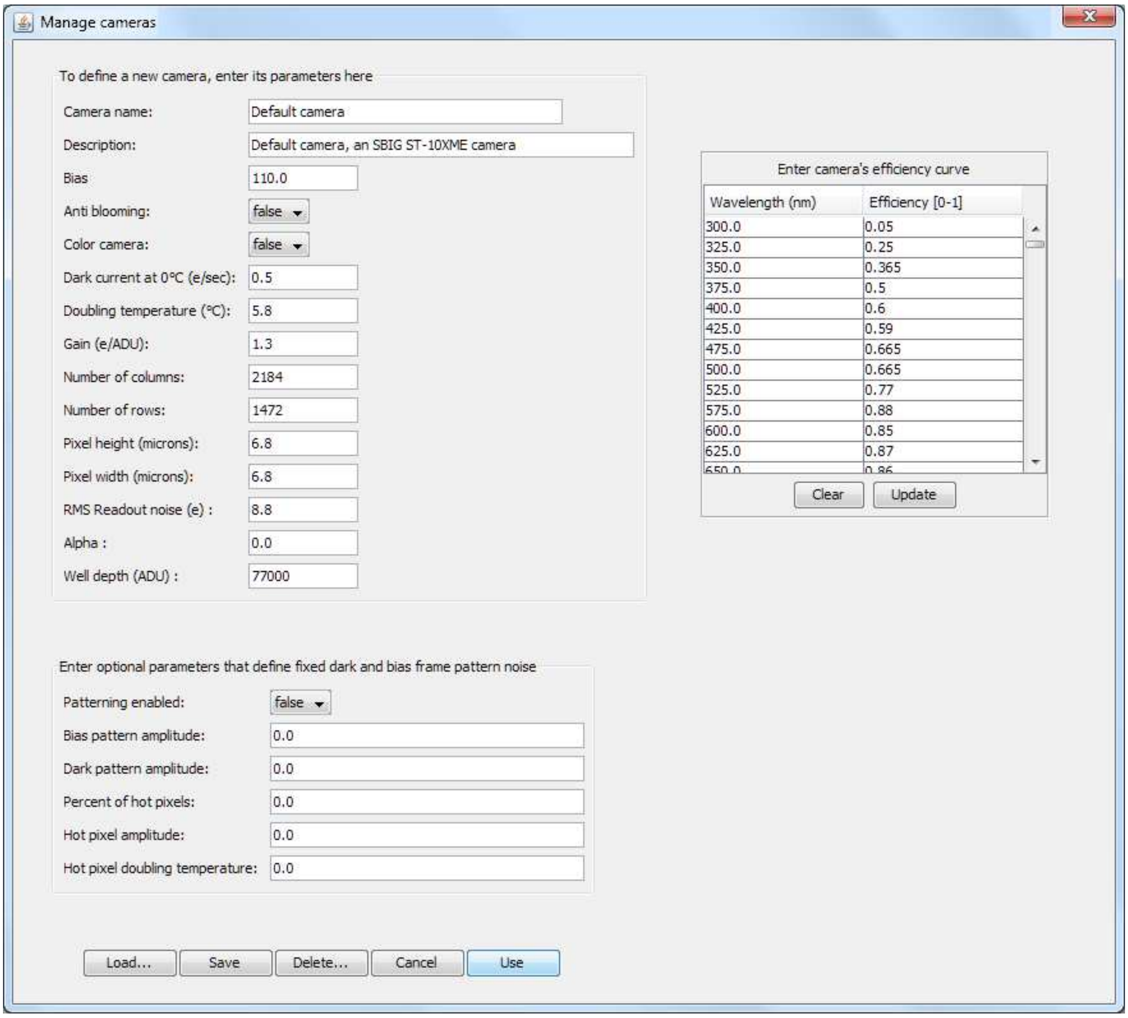


Figure 1918 Managing cameras

When this dialog is first presented, it will show some camera, whose name is shown in the Camera Name field. Initially this will be the camera that is being used in the current simulation. You have two options here: you can edit the parameters for this camera or you can load a new one using the Load button. Regardless, after you edit the parameters for the camera, you can either save them using the old name or choose a new name by editing the Camera Name field. Camera names, which must be unique, are used in SimCCD's pull down menus to select a camera for a simulation.

Once you have finished editing fields, you should press the Save button to save your changes. Note that saving a camera does not automatically make it the default camera for your simulation. You must press the Use button for this to happen. For example, you might choose to bring up this dialog, create a few new cameras, and then press the Cancel button to have SimCCD use the original camera rather than one of the new ones.

There are many fields in this dialog that you can change. Here is a short description of each.

The Description field is free-form text that you can use to describe this camera.

The Bias field gives the bias offset in ADUs that is added to each pixel of the simulation. It is a constant that has no particular effect on the simulation beyond this.

The Anti-blooming field tells SimCCD whether the camera is an ABG or NABG camera. SimCCD currently does not model blooming and ignores this field.

The Color Camera field tells SimCCD whether the camera is a color or monochrome camera. SimCCD does not model color cameras and will ignore this field, so it should always be set to false.

The Dark current at 0°C (e/sec) field gives the camera's dark current in electrons per second.

The Doubling temperature (0°C) field tells how the camera's dark current varies with temperature. For example, a doubling temperature of 4.24 indicates that at 4.24°C, the dark current will have doubled from 0.0379 to 0.0758.

The Gain (e/ADU) field tells what the camera's gain is. As usual, this is actually an inverse gain, so that values greater than one lead to fewer ADUs than electrons.

The Number of columns field tells how wide the camera's CCD sensor is. Note that when reading a source file, SimCCD does not use this value. Rather, the dimensions of the image are determined from the file itself. Depending on how the image was acquired and on whether the image has been cropped, the image might be wider or narrower than the CCD sensor. When preparing the simulation image, CCD may use this value to display an indication of how much of the target can actually be seen by the simulation camera.

The Number of rows field tells how tall the camera's CCD sensor is. The comments in the previous paragraph apply to this field too.

The Pixel height (microns) and Pixel width (microns) fields give the dimensions of the CCD's pixels. Note that SimCCD currently doesn't try to compensate for non-square pixels.

The RMS readout noise (e) field gives the CCD's readout noise in electrons.

The Alpha field gives a measure of the CCD's non-linearity. For most simulations it will suffice to leave this parameter set to zero, indicating that the CCD responds linearly to incoming photons up to the point of saturation. This parameter is not readily available from published literature; so we will discuss its meaning and determination in a later section.

The Well depth (ADU) field gives the CCD's well depth in ADUs.

The next set of parameters is optional. If the Patterning enabled field is false, SimCCD will ignore these parameters. Most of them are not readily available from published literature, so you will have to measure your CCD to determine their values. The meaning of these parameters and how to measure them will be described in a later section.

The last thing of interest in this dialog is the table labeled Enter camera's efficiency curve. Each entry in the table gives the camera's quantum efficiency for a particular wavelength. Wavelength values are entered in nm while the efficiencies are entered as fractions, so that 1 represents a 100% efficient filter. If you need to delete a line in the table, replace the entries with blanks, then press the Update button to clean up the table. Or you can delete all the entries by pressing the Clear button. You can enter values in any order, and then press the Update button to sort them on wavelength.

Equipment/Filter

This dialog is used to manage the optical filter list. The dialog works like the Equipment/Camera command, so we will only discuss those things that are unique to filters. Using the techniques described above, you can load a new filter, edit it, and save it using this dialog, which looks like this:

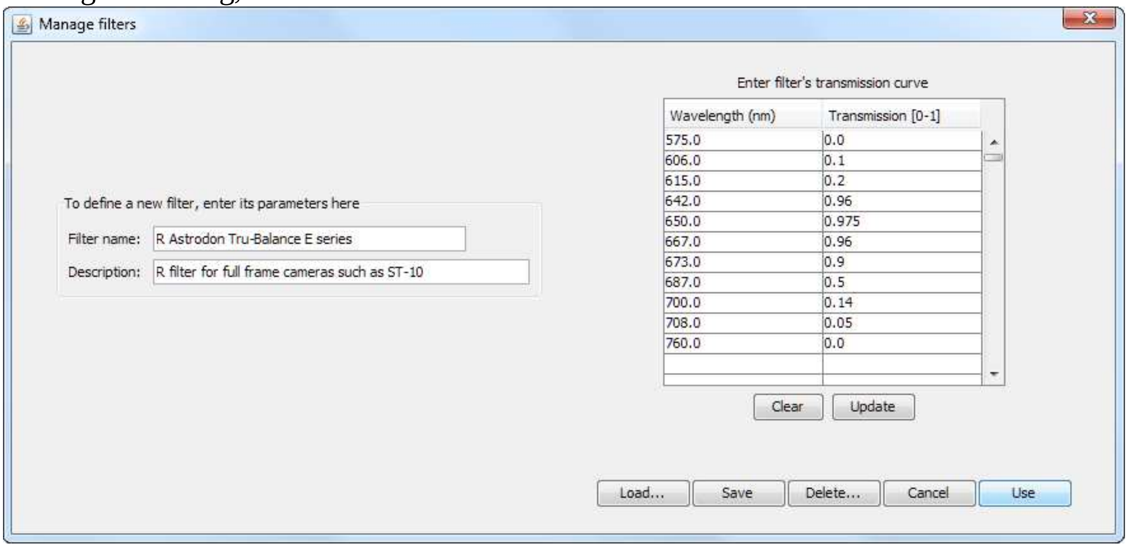


Figure 20 Managing filters

The Filter Name field gives the name of the filter, which must be unique.

The Description field gives a short description of the filter.

The table labeled Enter filter's transmission curve works the same as the table for the camera's efficiency curve.

Equipment/Optical Aid

This dialog is used to manage optical aids such as Barlow lenses, focal reducers and field flatteners. The dialog works like the Equipment/Camera command, so we will only discuss those things that are unique to optical aids. The dialog looks like this:

Figure 21 Managing optical aids

The Optical aid name field must be unique.

The Description field can be used to describe the optical aid in more detail.

The Magnification field gives the magnification of the optical aid. For example, values greater than one would be used for Barlows while values less than one would be used for focal reducers.

Equipment/Telescope

This dialog is used to manage the telescope list. The dialog works like the Equipment/Camera command, so we will only discuss those things that are unique to telescopes. The dialog looks like this:

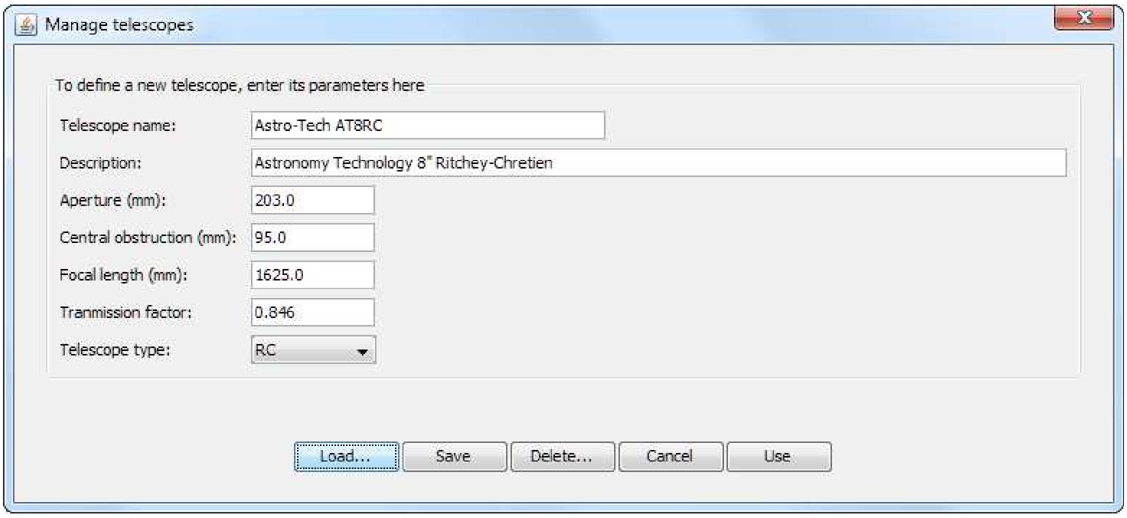


Figure 22 Managing telescopes

The Telescope name field must be unique.

The Description field gives additional information about the telescope.

The Aperture (mm) field gives the scope's aperture.

The Central obstruction (mm) field gives the diameter of the scope's central obstruction (e.g., diagonal or secondary mirror). For refractors, enter a value of zero.

The Focal Length (mm) field gives the scope's focal length.

The Transmission Factor field defines the scope's transmission, where a value of 1 represents 100% transmission. Note that SimCCD currently does not model the wavelength dependence of the transmission, so you may need to edit this value to deal with special cases.

Equipment/Reset

This command can be used to reset the lists of cameras, filters, optical aids and telescopes to SimCCD's built-in default lists. Any new cameras that you have created will be deleted; any edits to old cameras will be lost; and some cameras that you have deleted will be restored. For this reason you may want to save the settings first using the File/Export command first.

Note that this command will not work if you try to delete a camera that SimCCD is using. If this happens, use the Equipment/Camera command to load and then use the default camera. This will allow the other cameras to be deleted. (This comment applies to filters, optical aids, and telescopes too.)

Note that SimCCD stores its equipment lists and other settings in a file named `SimCCD settings.xml`, which is located in your home directory (`Documents` in Vista). You should back this file up occasionally. If you have a problem with the file, you can edit it with any text editor. You can also use a text editor to change equipment definitions or add new ones. You can copy this file to another computer if you want to run SimCCD in two places.

We encourage you to submit definitions for new kinds of equipment so they can be incorporated into future releases of the program. To do this, email a copy of your settings file `SimCCD Settings.xml` to simccd@darklights.org. You can also use the File/Export command to generate a suitable XML file.

View/Control Panel

This command brings up a window where you can control SimCCD's operation. The window is divided into four tabs that can be used to control the simulation conditions, the equipment used for the simulation, the contrast of the displayed image, and the amount of magnification for the displayed image. We will now discuss each of these tabs.

View/Control Panel (conditions tab)

The control panel has two tabs for controlling the simulation conditions and the equipment that is to be simulated. There are also three buttons. In general SimCCD does not act on any changes you make to this dialog until you press the Apply or OK buttons. In particular, beware that if you press the Done button, the dialog will be dismissed and the changes will be discarded.

A word of warning is in order. As we will see below, when you change some fields in this dialog, other fields may change out from under you. Until you get used to how these fields interact, be sure to double check every field to make sure it contains the value you intended.

With the conditions tab selected, the dialog looks like this:

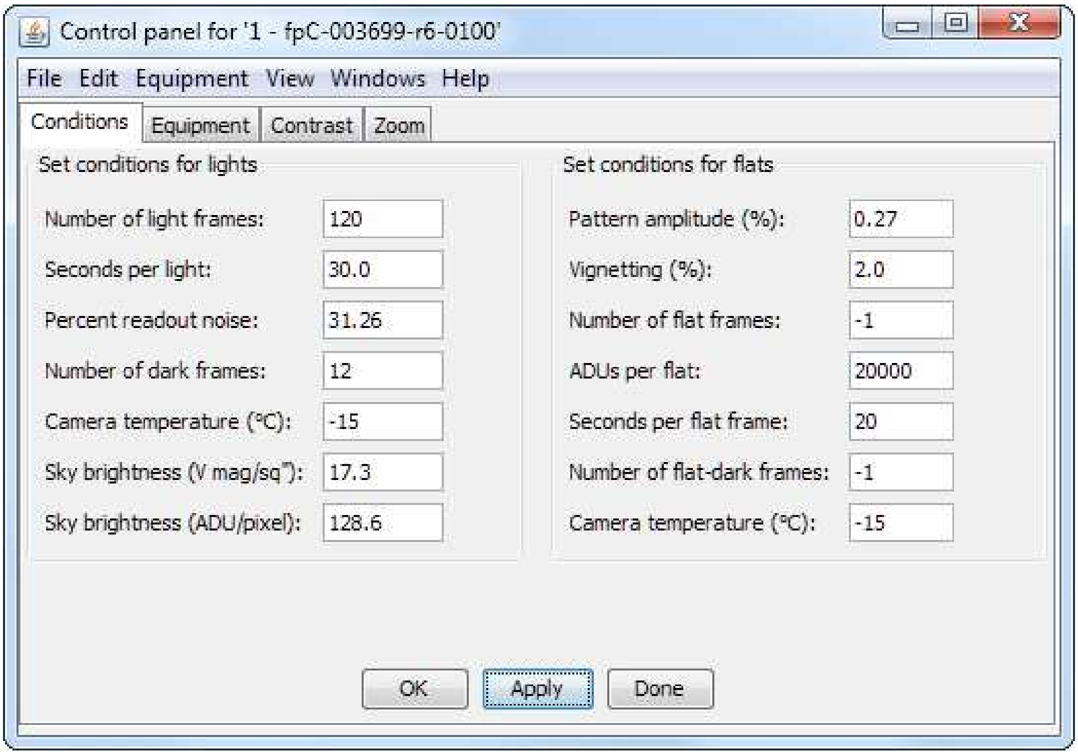


Figure 23 Choosing simulation conditions

The Number of light frames field controls how many exposures are to be taken in the simulation.

The Seconds per light field controls the exposure time for each light frame. You can enter a value directly in this field if you want to simulate a particular exposure time. But this field can also be defined indirectly as a result of entering values in the Percent readout noise field. For example, if you want to know what exposure time to use to get the readout noise down to 5%, enter 5 in the Percent readout noise field and then read the exposure time in the Seconds per light field. Conversely, if you want to know how much readout noise will be present if the exposure time is 30 seconds, enter 30 in the Seconds per light field and read the percentage in the Percent readout noise field.

The Percent readout noise field tells how much additional noise there will be in an image above and beyond shot noise from sky glow. You can enter a value directly in this field, in which case the Seconds per light field will be adjusted to give the recommended exposure time to achieve this percentage. But you can also enter a frame time in the Seconds per light field to find what the readout noise will be. In the example above, we have chosen an exposure time of 30 seconds, and SimCCD has then computed that this will result in images that have 31.23% additional noise. Using a longer exposure time will give a smaller noise value. You can use the value to decide whether the noise has been reduced enough.

The Number of dark frames field is used to determine how many dark frames should be simulated. In general, large positive values will result in images that have less noise due to dark current, hot pixels and pattern noise. If a value of zero is used, dark current will be simulated but dark field calibration will not be done. You can use this to assess how important dark field calibration is. If a negative value is used, SimCCD will disable the dark field computation altogether, so that dark current is neither simulated nor corrected.

The Camera temperature (°C) field lets you set the camera’s operating temperature. The same temperature is used for both light and dark frames. SimCCD does not currently provide a way for you to use different values since that is not an interesting case.

The sky brightness can be controlled using the next two entries. The Sky brightness (V mag/sq") field lets you enter the sky brightness in familiar units, while the Sky brightness (ADU/pixel) field lets you enter the data in more easily measured units. Whenever you enter data in one field using one set of units, the other field will be changed to show the other set of units. While the mag/sq" units might seem like the easiest way to enter the sky brightness, be forewarned that SimCCD’s model for the spectral dependence of sky glow is very crude, so the simulation will probably not be very accurate. It will be much more accurate to take a sample picture with your equipment, measure the sky brightness in ADU/pixel, and then enter that value into the second field. Note that this value depends on the exposure time (the Seconds per light field); so you must make your sample picture using that exposure time. Later if you change the Seconds per light field, the Sky brightness (ADU/pixel) will change to match the new exposure time. This might be a bit of a surprise at first, so remember that the Sky brightness (ADU/pixel) field is not an absolute measure of sky brightness but rather depends on the exposure time.

If you are using a test photo for your source data, there’s an easier way to enter the sky brightness. When SimCCD presents the dialog for specify the source conditions, check the box that is labeled Use this sky level for the simulation. Then SimCCD will automatically enter that value into the simulation conditions box.

The next set of fields control how SimCCD models flat fielding. SimCCD makes no attempt to model dust motes, but does model vignetting. SimCCD also models the CCD's pattern noise, caused by the variation in sensitivity from pixel to pixel.

The Pattern amplitude (%) field models the variation in pixel sensitivity. The value is an RMS percentage of the total flat field signal. Small values like 0.27% are usually reasonable. To measure your own system, take a flat field image, measure the RMS noise in a region where there aren't any dust motes, and then divide by the average intensity in that region.

The Vignetting (%) field gives the percentage drop in signal between the center of the CCD and one of its corners. Note that in this model, the vignetting is independent of the size of the CCD or the characteristics of the scope; so large CCDs will be vignetted as much as small ones. Therefore you may need to edit this field to model your own system.

The Number of flat frames field tells SimCCD how many flat frames to take. Using a large positive value will improve the flat field calibration. If the value is zero, flat field calibration will be disabled, so the resulting simulation will be vignetted. If a negative value is used, SimCCD will not model flat field effects, so that the image won't be vignetted and won't be calibrated.

The ADUs per flat field is SimCCD's way of specifying how bright the flat field target is. In practice, when you take flat field images, you probably set the target brightness and/or the exposure time to give a reasonable exposure level – typically something around 20000-30000 ADUs so that the CCD is not saturated. Enter the value you have chosen in this field. One very conservative rule of thumb says that the total number of ADUs required to get a good flat is the one million. In this case, if the value for ADUs per flat is 20000, then you would enter 50 in the Number of flat frames field.

The Seconds per flat frame field determines the exposure time for each flat. This field is only used to determine the impact of readout noise on flat field calibration. The other fields will determine the brightness and shot noise characteristics of the flat field image.

The Number of flat-dark frames field determines how many dark frames are used to remove the dark current (mainly the bias signal) from the flats. Use a value of zero to disable flat-dark corrections, and use a negative value to tell SimCCD to not model dark current for the flats.

The Camera temperature (°C) field determines the camera's temperature for the flat field exposures. This does not have to be the same as the temperature for the lights.

View/Control Panel (equipment tab)

When the equipment tab is selected, the Simulation Control window changes so it looks like Figure 24. There are four pull down menus in this tab. Each menu controls a particular kind of equipment. Use these menus to select the equipment you want to use. SimCCD comes with some equipment already defined. In most cases, the definitions are based on the published characteristics for the equipment. However these definitions will probably not be exactly correct for your equipment; so when in doubt, use the Equipment menu to edit the existing definitions and/or make new ones.

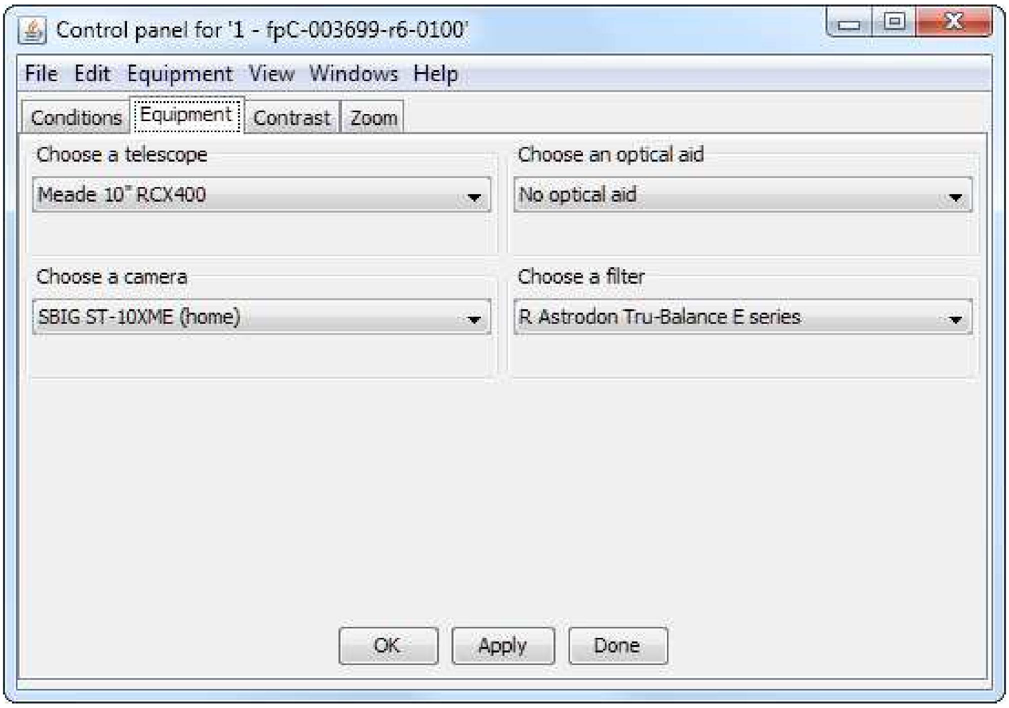


Figure 24 Choosing simulation equipment

View/Control Panel (contrast tab)

When the contrast tab is selected, the window changes to this:

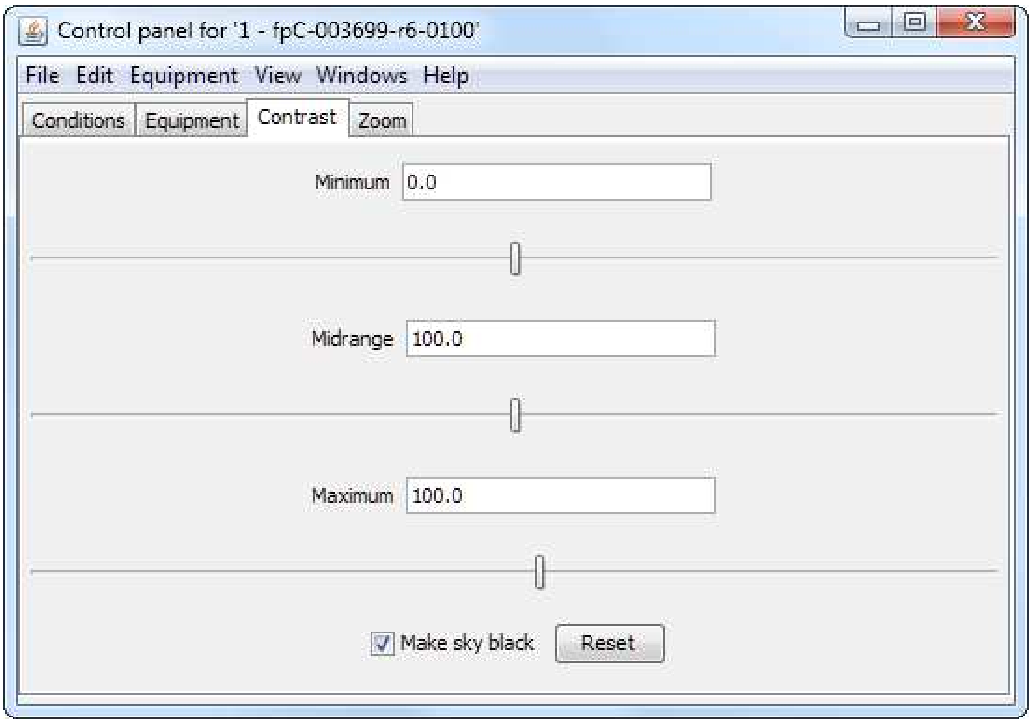


Figure 25 The contrast control

The control does not modify the calculated simulation values, but just changes the way they are displayed on the screen.

The Minimum data entry field and slider let you control the lowest intensity that will be displayed. The value is a percentage of the full scale image intensity. Any pixel that is darker than the sky by this amount will be displayed as black. Lowering the value will let you see more details in the sky, while raising the value will clip out more of the sky area.

The Midrange field controls the contrast boost for the middle range of intensities. If a small value is used, a linear contrast stretch will be used, leading to very little boost in contrast. If a large value is used, a logarithmic contrast stretch will be used. This will tend to show the dimmer features better, but it may also amplify noise excessively.

The Maximum field and slider let you control the brightest intensity that will be displayed. The value is a percentage of the brightest area in the image, above which pixels will be displayed as white. There is usually little reason to make this value greater than 100; but smaller values will help you see dim features better.

The Make sky black checkbox lets you make the sky black, which will increase the contrast. If this feature is disabled, the sky will be shown at its true brightness, which will make the target harder to see but give you a better idea of how bright the target is compared to the sky. By default SimCCD checks this box.

The Reset button restores the settings to their default values.

View/Control Panel (zoom tab)

When the zoom tab is selected, the window changes to this:

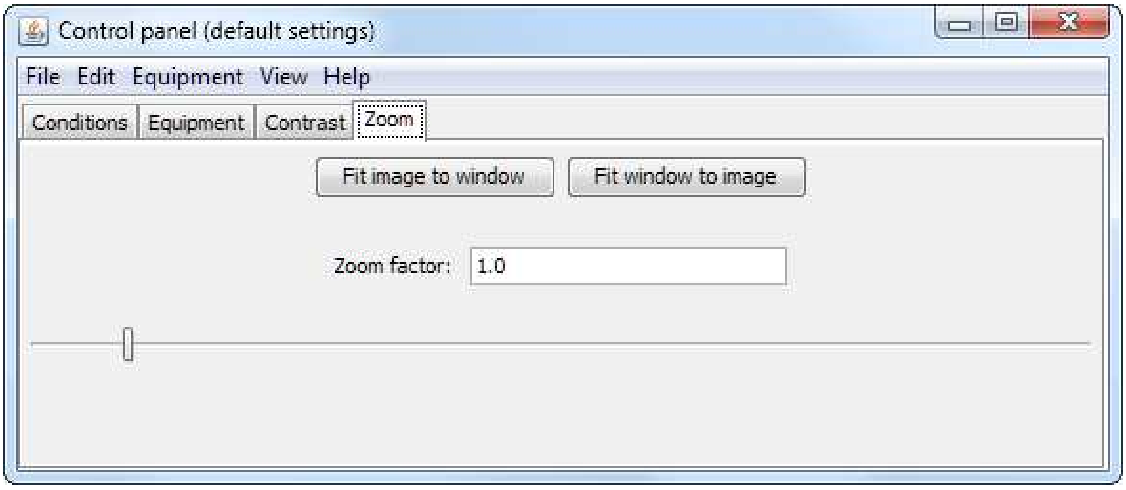


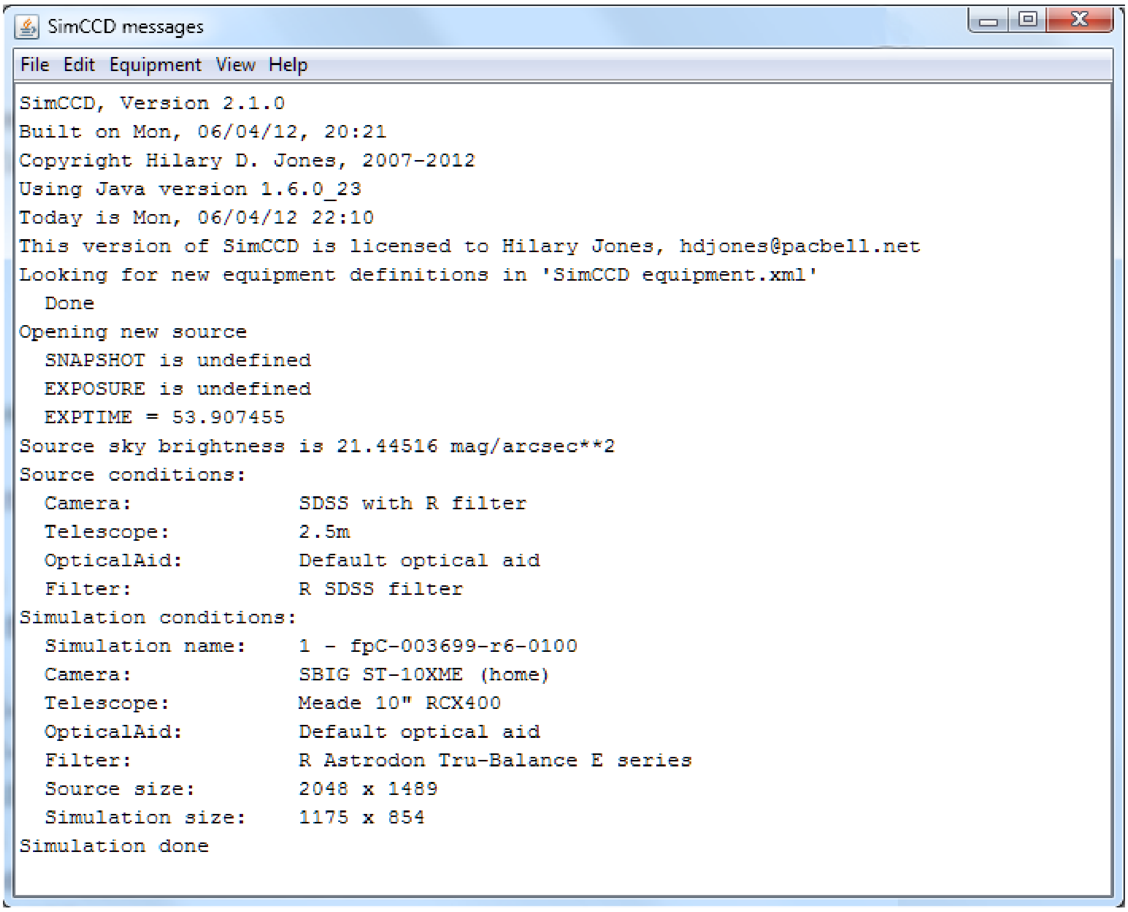
Figure 26 The zoom control

This control lets you modify the size of the displayed image. The data for the image is not modified, and only the display changes. You can set the amount of zoom by adjusting the slider or entering a value in the Zoom factor area. You can also press the Fit image to window button to get SimCCD to choose a zoom factor that will make the image fit in its window. And you can press the Fit window to image to have SimCCD adjust the window size to fit the image.

View/Log Messages

This command allows you to display SimCCD's message log, shown in Figure 27.

For the most part you will not need to see these messages; so by default the window is not shown. However, if SimCCD encounters a problem, the display may show an error message that can help you diagnose the problem.




```
SimCCD, Version 2.1.0
Built on Mon, 06/04/12, 20:21
Copyright Hilary D. Jones, 2007-2012
Using Java version 1.6.0_23
Today is Mon, 06/04/12 22:10
This version of SimCCD is licensed to Hilary Jones, hdjones@pacbell.net
Looking for new equipment definitions in 'SimCCD equipment.xml'
  Done
Opening new source
  SNAPSHOT is undefined
  EXPOSURE is undefined
  EXPTIME = 53.907455
Source sky brightness is 21.44516 mag/arcsec**2
Source conditions:
  Camera:             SDSS with R filter
  Telescope:          2.5m
  OpticalAid:         Default optical aid
  Filter:             R SDSS filter
Simulation conditions:
  Simulation name:    1 - fpC-003699-r6-0100
  Camera:             SBIG ST-10XME (home)
  Telescope:          Meade 10" RCX400
  OpticalAid:         Default optical aid
  Filter:             R Astrodon Tru-Balance E series
  Source size:        2048 x 1489
  Simulation size:    1175 x 854
Simulation done
```

Figure 27 The log message window

View/Statistics window

Use this window to display statistics about a selected area in the current simulation. The window can also be displayed by clicking or dragging in the simulation window. If no area is selected, the statistics are given for the entire image.

Figure 28 shows a typical display. The field labeled Shot noise tells how random variations in the signal contribute to the noise. This is sometimes referred to as Poisson noise. This term includes both variations from the target and from the sky. The column labeled ADU tells the number of Analog Digital Units that will appear in one frame for the given exposure time. The column labeled Percent tells how much of the total noise was due to shot noise. Purists will notice that noise does not add linearly; so expressing the noise as a percentage can be misleading. Therefore this figure is just presented as a guideline.

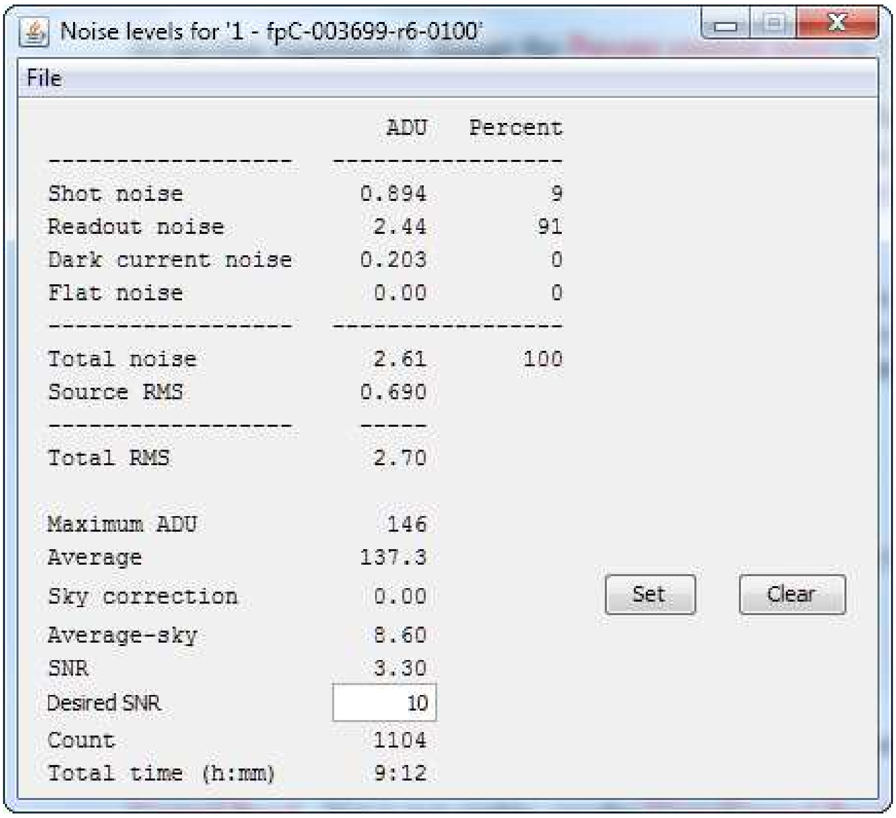


Figure 28 The statistics window

The next line tells how much noise is introduced by reading and digitizing the data from the CCD. In our example, the exposure time was quite short; so almost all of the noise was due to readout.

The next line tells how much noise is introduced by dark current. If the images have not been calibrated, this figure will not only include statistical noise but also pattern noise. On the other hand, if the images have been calibrated, then the pattern noise will have been calibrated out and no longer be shown, but there will be additional noise introduced from the dark frames themselves.

Following this is a line that gives the flat field noise. This can include any pattern noise that has not been calibrated out as well as noise in the flats and flat-darks.

These figures are followed by the Total Noise, which is the amount of noise you can expect from the effects reported above. This is followed by a line labeled Source RMS. This tells how much the source image varies from point to point. This includes not only the noise that was present in the source image but also the variation of the source's intensity. Finally there is a line telling the Total RMS, which combines statistical noise with the source image variation.

The line labeled Maximum ADU tells you how bright the brightest object will be in your picture. If this value is very high, then your exposure time is probably too long. That would cause the CCD to saturate. Similarly the Average figure tells the average level in the selection area.

The Sky Correction line is normally zero and should be left that way. However there may be occasions when you think that SimCCD has not subtracted out the source image's sky brightness correctly. (That's the figure you entered in the source conditions dialog when you opened the image.) To fix the problem, drag your mouse over the image in an area that has no stars or significant structure, so that the sky is totally black in this area. Then press the Set button. Note: you will probably see some pretty strange values in the dialog immediately after you press the button; but when you move the selection rectangle to a feature, the values will become reasonable again.

The Average-Sky line tells you how bright the target is once SimCCD has subtracted out the sky level. If this value is negative for some region, it means that the sky correction isn't quite right. To fix the problem, press the Set button for this region.

The next field gives the signal to noise ratio (SNR) in the selected area. This is a prediction of how much noise will be present in your astrophoto under the given conditions.

The next field gives you a place to enter the signal to noise ratio that you would like to achieve for your session. For example if you want the noise to be 10% of the signal, set the Desired SNR to 10. SimCCD will then give you an estimate of how many exposures will be needed to get to that level. It will also tell long your session will last.

You might look at the readout noise figure and wonder why it doesn't agree with the value that is shown in the control panel. Don't worry. The statistics panel shows the percentage of the total noise, while the control panel shows the percentage of the shot noise; so the figures are not expected to be identical.

You also might notice that the percentages don't seem to be consistent with the values listed in the ADU column. When SimCCD computes the total noise, it has to add the contributions in quadrature, which is to say that the total noise is the square root of the sum of the squares of the components. On the other hand people expect the percentages to add linearly to a total of 100%. Unfortunately there is no mathematically correct way to compute percentages when the noise has to be added in quadrature. To deal with this, SimCCD's percentages are a rough estimate of how much each component contributes to the total noise. So if mathematical precision is important to you, then use the ADU values and treat the percentages as guidelines.

Windows

This menu shows a list of every simulation window that is open, with a check mark next to the topmost window. Use the menu to bring some other simulation to the foreground.

Help/Check for updates

Use this command to find out what the latest version of SimCCD is. After a short pause, SimCCD will display a dialog like the following.

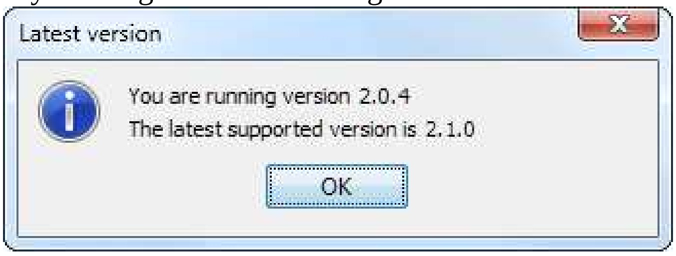


Figure 29 SimCCD's latest version

Help/About

SimCCD currently does not have a detailed help facility. However in many cases when you let the mouse hover over one of SimCCD's controls, SimCCD will pop up a "tooltip" that describes what the control does.

Choosing the Help/About menu item will bring up the following display of information about SimCCD.

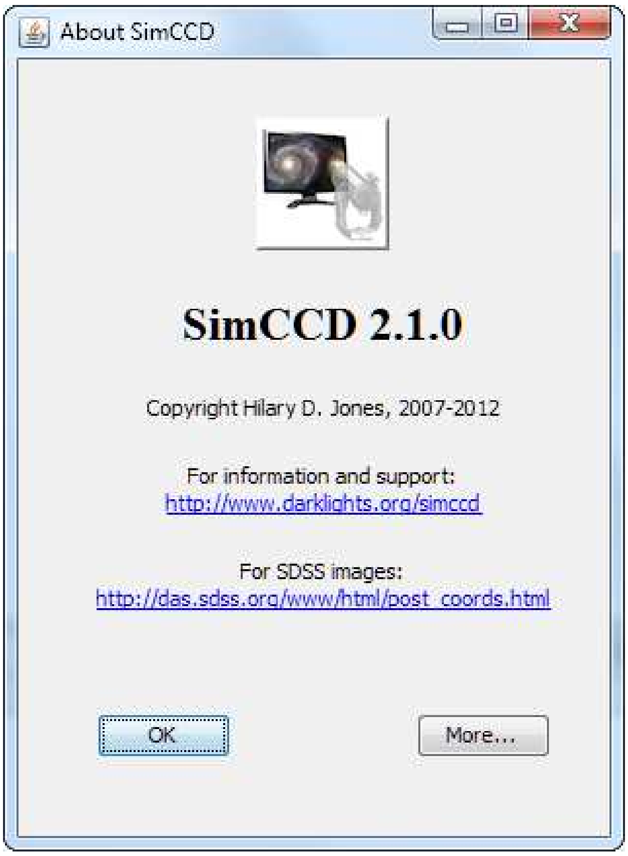


Figure 30 SimCCD's "about" box

Clicking on the icon or the link to www.darklights.org/simccd will take you to the SimCCD web site. Clicking on the link to das.sdss.org/www/html/post_coords.html will take you to the SDSS site where you can find source images that can be used with SimCCD.

If you click on the More… button, SimCCD will present the window that's shown in Figure 31.

Figure 31 More information about SimCCD

Chapter 4 - Fine Points

In this chapter we discuss some of the fine points of operating SimCCD. In particular, we will cover how to get source datasets, how to interpret tsField files, how to interpret the batch startup file, how SimCCD stores its settings, how to measure alpha, and how to measure the optional dark field parameters.

How to install Java

To run SimCCD, your computer must have Java version 1.6 or later. For Linux, Solaris and Windows users, this is easy. Just visit java.sun.com/javase/downloads/index.jsp and follow the instructions for downloading the JRE. (You don't need the JDK unless you plan to do Java programming.) For Macintosh users, getting the right version of Java may prove to be a problem. It is automatically included in MacOS X, so you don't have to do anything to install it. But as of this writing, Apple only supports Java 1.6 in 64-bit capable Intel Macintoshes; and you will need to have MacOS version 10.5 or later. Earlier versions on MacOS support Java version 1.5; but if you try to run SimCCD there, you will get a message that says "UnsupportedClassVersionError: Bad version number in class file". Although SimCCD has not been tested with it, there is a workaround called SoyLatte that might solve the problem. It is a BSD version of Java 1.6 that can be obtained from landonf.bikemonkey.org/static/soylatte. There is a gotcha here: you must have a Java Research License to use the package legally.

More detail on how to set up response curves

Up to now we've assumed that it will be pretty obvious how to set up the response curve for an optical filter. But it's probably worth taking some time to explain the process in detail. Consider for example the following filter curve, obtained from Astrodon's web site:

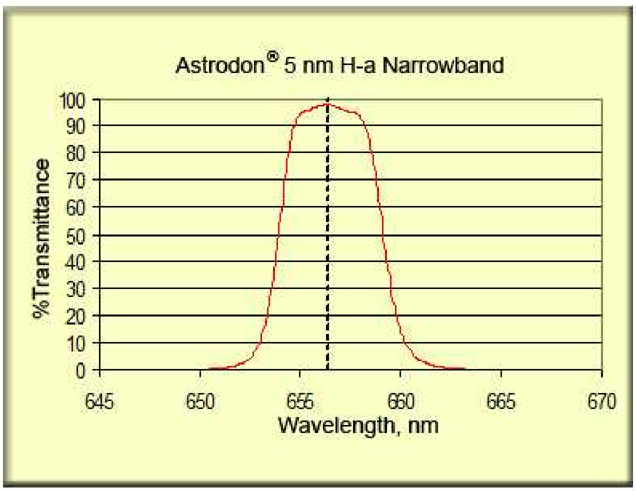
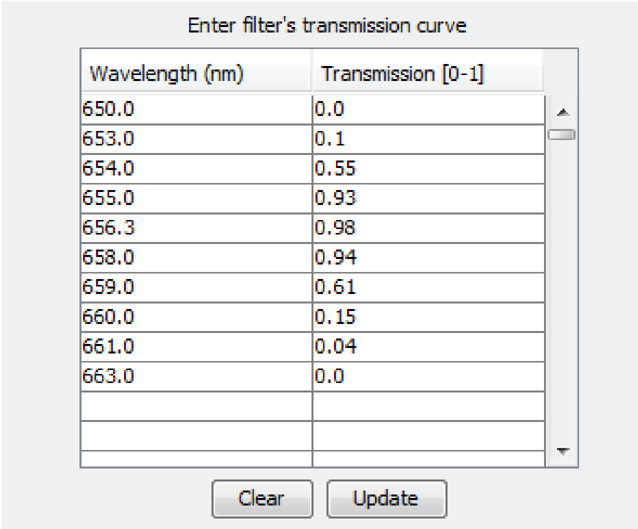


Figure 32 Band pass for a hydrogen alpha

This filter has been included in SimCCD, and it is described by the following response curve:

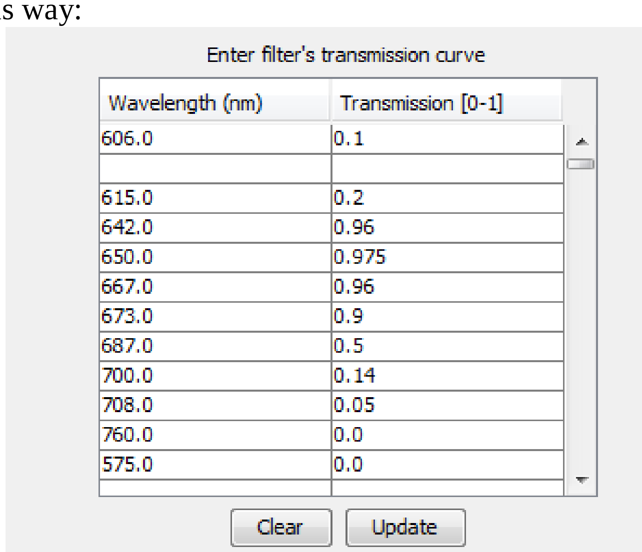


Wavelength (nm)	Transmission [0-1]
650.0	0.0
653.0	0.1
654.0	0.55
655.0	0.93
656.3	0.98
658.0	0.94
659.0	0.61
660.0	0.15
661.0	0.04
663.0	0.0

Figure 33 Table for a filter's transmission curve

The values in the table were simply read off the graph and entered in the table as shown. It is not very important that these values be extremely accurate as long as the general appearance of the curve is kept intact. Technically you should enter values at both the high and low end where the transmission goes to zero. SimCCD will assume that the transmission goes to zero outside the region where it is defined, but there will be small errors if endpoints aren't exactly zero. In practice, non-zero values will probably work fine.

The table above shows the values sorted correctly on wavelength; but they can be entered in any order; and there can be some blank lines too. For example, you could have entered the data this way:



Wavelength (nm)	Transmission [0-1]
606.0	0.1
615.0	0.2
642.0	0.96
650.0	0.975
667.0	0.96
673.0	0.9
687.0	0.5
700.0	0.14
708.0	0.05
760.0	0.0
575.0	0.0

Figure 34 Unsorted table with a blank entry

If you press the Update button, the blank line will be deleted and the entries will be sorted on wavelength to give the result shown in the first table.

There is no direct way to delete a line. If you find that you have an incorrect entry, you can delete it by erasing the contents of the Wavelength and Transmission entries, then pressing the update button.

Note that you can use the same process for entering the camera's efficiency curve.

How to get source datasets

For SimCCD to work, it needs accurate source datasets describing the brightness of the target at each point in the sky. There are several ways to do this.

The best way to get source data is to find it at the Sloan Digital Sky Survey web site. While the SDSS survey doesn't cover the sky completely, when a target has been photographed, getting its data is quite easy. All you need to do is log into the web site

http://das.sdss.org/www/html/post_coords.html

and then enter the coordinates of your target. Be sure to enter the RA value in decimal degrees. For example, if the target is at RA 5h 10m 30s, convert that to decimal hours (5+10/60+30/3600 = 5.175) and then multiply by 15 to get 77.625 decimal degrees. The declination must also be entered in decimal degrees, but this time the computation is easier. For example if the target is at declination 5° 12' 37", the decimal value is 5+12/60+37/3600=5.2103, and it is already measured in degrees. Enter the resulting values into the form and press the Submit request button. If the SDSS survey has captured that target, you will be presented with a list of files. There may be quite a few files, but you want the ones that are named something like `fpC-003699-r6-0100.fit.gz`. The fields in this name refer to the region of the sky that was photographed and the filter that was used. In particular, for this example the letter r signifies a red filter. These are "compressed" FITS files, which can be fed directly into SimCCD and do not require any special preprocessing to get them ready. (However if you need the FITS file itself for some other reason, you can use a utility like WinZip or Stuffit to decompress it.)

If you can't find your target in the SDSS, consider the VirGO utility, available from http://archive.eso.org/cms/tools-documentation/visual-archive-browser. This utility will offer data from a wide variety of sources, including the Hubble Space Telescope. However I have found that some of the HST files downloaded from this site were corrupt. Another site for HST data is http://cadcwww.dao.nrc.ca/hst/science_wfpc2_assoc.html.

To use HST data, you will need to enter new definitions for the HST's telescope, cameras and filters. There were far too many options here to include them all in this release of SimCCD, so you will have to do some research to find the data needed to define these pieces of equipment.   The best place to start looking for this data is at the Space Telescope Science Institute, http://www.stsci.edu/hst.

Incidentally, HST files usually contain images from four different camera and filter combinations. To make sense of these files, you will have to concentrate on just one part of the image and use the equipment definitions that apply there.

Some other sources of data include http://fits.gsfc.nasa.gov/fits_samples.html and http://archive.stsci.edu.

Another way to get pictures is to simply take them using your own equipment, stack them using a program such as MaxIm DL, and then feed the result into SimCCD. Remember to use an average stack, not a sum stack; and don't do any image processing to make the image look better. SimCCD expects the input file to contain raw ADU values, with the sky brightness level subtracted out (or present in the FITS header in a SKY keyword.)

Using a file that was prepared using you own equipment will let you answer questions like "what would have happened if I had taken these pictures when the sky was brighter?" or "what would have happened if I had used half as many pictures?" Of course the simulation can only be as good as the source data; so you can't expect SimCCD to answer the question "what would have happened if I had used twice as many pictures at a darker site"!

Using your own files is often the best way to use SimCCD. While the simulated images won't necessarily show what your final astrophoto will look like, the statistics panel will still give accurate estimates of out how many exposures you should take to get the desired signal to noise ratio.

If all else fails, try generating images using AIP4Win's Edit/Synthetic Images… command. For example, if want to take pictures of some target for which no real data is available, you may be able to use AIP4Win to generate a synthetic image containing enough features to simulate the target.

How to interpret tsField files

When you use the SDSS site as described above, it may offer you some tsField files. These are FITS files that contain data describing the other FITS files. It turns out that different SDSS images can be exposed using different camera gains; so the only way to be sure what was used is to inspect the tsField file. A detailed discussion of the layout of tsField files is beyond the scope of this manual. But you can use a FITS file inspection utility such as the FITS File Viewer utility (fv) to explore the file and find the camera gain. The utility was developed at the High Energy Astrophysics Science and Archive Center (HEASARC). For further information about fv, see http://heasarc.gsfc.nasa.gov/ftools/fv.

To find the camera gain using fv, open the tsField file, push the Select button, press the Clear all button, scroll down and put a check mark in the box that's labeled gain, press the Display Table button, and press the Plot button. That will display a plot of the values. Other options can be used to get a listing of numerical values.

As a practical matter, the gain for a particular SDSS filter does not change substantially from picture to picture, so you can safely use the same gain for all of the source files that use that filter. Therefore in most cases you won't need to inspect the tsField files.

About the batch startup file

If you are a Windows user, you can start SimCCD by double-clicking the file `Run SimCCD.vbs`. This script starts SimCCD with a default memory allocation. For large problems, SimCCD may run out of memory. In this you can use `Run SimCCD 1.5GB.bat`, which lets SimCCD start up using 128 MB and grow to 1500 MB. If SimCCD still runs out of memory, you can edit this file to give SimCCD more memory. On the other hand, on smaller memory systems this script may not work, and you may have to edit the script to decrease the values of the `Xmx` and/or `Xms` parameters.

If SimCCD has trouble starting, try using the script named `Debug SimCCD.bat`. It will pause after SimCCD runs, which will give you a chance to see any error messages that might give clues about what needs to be fixed.

If you are running a 64-bit Windows system, the latest version of `Run SimCCD.vbs` should be able to locate the Java compiler. But you might have trouble starting SimCCD because of a bug in the Java installer. To fix this problem, you will need to locate the file `java.exe`, which may be in one of these folders:

```
C:\Windows\system32
C:\Windows\SysWOW64
C:\Program Files (x86)\java\jreXXX\bin
```

where `jreXXX` might actually be something like `jre, jre1.6.0_16`, etc. Once you locate `java.exe`, you can modify the system's PATH environment variable to include the folder; or you can edit one of the startup files mentioned above to spell out the full path. For example:

```
C:\Windows\SysWOW64\java -cp […]
C:\Program Files (x86)\java\jre6\bin\java –cp […]
```

If you are a UNIX or Macintosh user, you can start SimCCD by running script files `run_simccd.sh` or `run_simccd.command`, respectively. These files are very similar to `Run SimCCD.bat`, but are tailored to the requirements of UNIX and MacOS users. You will need to edit these files if SimCCD needs more memory.

About the settings file

SimCCD keeps track of your equipment lists, exposure conditions, and other information in a settings file named `SimCCD settings.xml` located in your `Documents` folder. You should back this file up occasionally. If this file is corrupted, SimCCD may have trouble starting. You may be able to fix the problem by editing the file using an ordinary text editor. A discussion of the layout of this file is beyond the scope of this document, but if you open it, the format should be obvious. If you can't find the problem, consider replacing the file with `SimCCD settings.xml.bak` which is also in the `Documents` folder. That file is created whenever SimCCD's settings change (but only once during any one session); so it may contain the last good version of the file.

If all else fails, copy the settings file off to a safe place, then delete the original. When SimCCD starts, it will create a new version of the file containing a default equipment list. Once things seem to be working, you may be able to edit the new settings file to paste in definitions from the copy.

How to measure alpha

The camera's alpha parameter is a measure of how linear a camera's response is to incoming photons. We follow Axel Mellinger's treatment of non-linearity for his all-sky panorama of the Milky Way:

> http://www.phy.cmich.edu/people/mellinger/research/Publications/2009/2009__PASP__Mellinger__A_Color_All-Sky_Panorama.pdf

In this paper he measures the response of a CCD and finds that it varies roughly as:

$$S(E) = aE + bE^2$$

where S(E) is the measured signal, E is the exposure level, and a and b are constants that differ for each camera. The parameter alpha is defined as:

$$\alpha = b/a^2$$

The following section describes how to measure alpha. In addition, you will find two scripts in the distribution kit to help you with this process: `monitorLightSource.vbs` and `runSequence.vbs`. You will also find a spreadsheet `linearity.xls` that can be used to compute alpha, given the output from `runSequence.vbs`.

To measure alpha, you will need a very steady light source. Take pictures of it using several exposure times from one second on up to the level at which the CCD begins to saturate. You will get a curve that looks like the one in Figure 35. For low values of the exposure time, the data can be plotted better as the ratio of ADU divided by exposure time, as shown in Figure 36.

Fit the curve to a linear form as shown by the line in the graph, ignoring the region where the CCD begins to saturate. SimCCD will take care of this region by cutting off ADU values that exceed the camera's full well depth.

A word of warning: the light source intensity must be very steady for this measurement to work. But even with a very well controlled source, the measurements will change as the camera warms up. Therefore we suggest that you take reference exposures before and after each measurement. If a reference exposure changes, scale the measurement to compensate for the change. This is the technique I used to measure my ST-10XME camera to get the data for these figures.

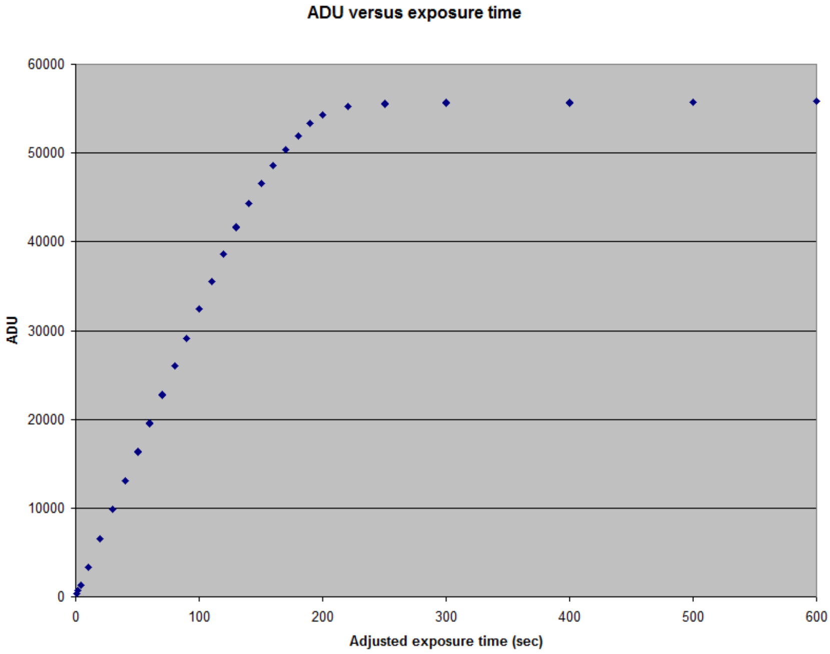


Figure 3519 Camera response vs. exposure time

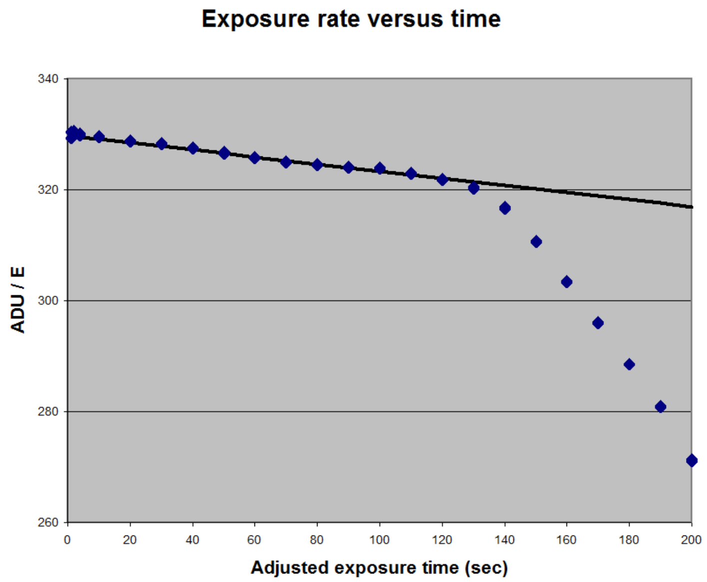


Figure 36 The camera's response replotted

If you don't want to take the time to measure alpha, you may be able to get by with an approximate value alpha = -6E-7. Surprisingly this value works for both ABG and NABG cameras. Also for many simulations, the value of alpha won't matter a lot; so using alpha=0 will work just fine. (Most of the cameras that are defined in the distribution kit have alpha set to zero.)

How to measure the optional dark field parameters

When you enter the Equipment/Camera command, a dialog will be displayed that contains the following fields, among others:

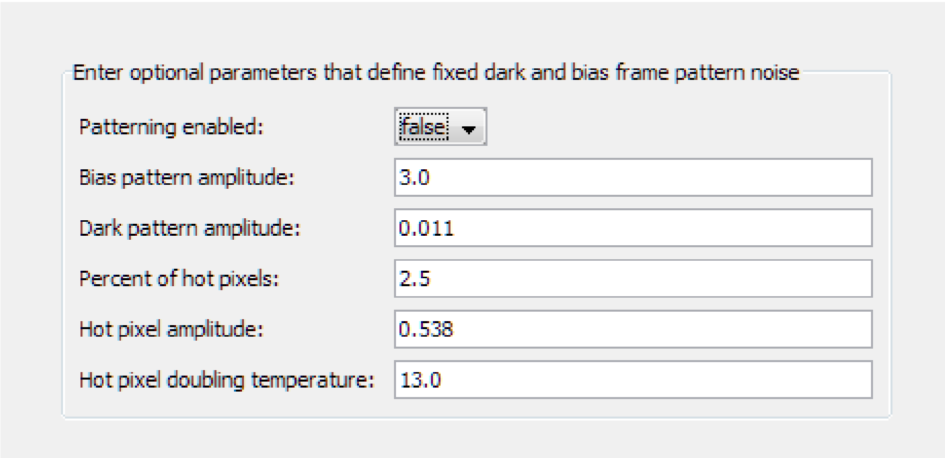


Figure 37 Entering the optional dark field parameters

If Patterning enabled is set to false, none of the other parameters will be used and dark frames will be modeled as just random noise determined by the dark current. If the parameter is set true, then a more sophisticated model is used, which allows the simulation of hot pixels and fixed pattern noise. The model is not based on any physical theory, but has been verified to give a reasonable approximation of hot pixel noise for the SBIG ST-10XME camera. Suggestions are welcome on how to improve the model for hot pixel and fixed pattern noise!

The Bias pattern amplitude field is the RMS amplitude of the bias pattern, measured in ADUs. To find a value for this field, take a large number of bias frames, average them, then measure the standard deviation of the background. SimCCD will use this value to generate a fixed random pattern that is present in every image. Note that SimCCD does not try to model the pattern caused by amplifier gain, and it does not model the vertical striped pattern that is often found in real bias frames.

The Dark pattern amplitude field is the RMS amplitude of the dark current, measured in electrons/pixel/second. This value is roughly independent of temperature for low enough temperatures. To find it, take a large number of dark field frames using a long exposure time so as to minimize the effect of readout noise. Subtract the master bias frame, and then average the images. Next measure the standard deviation in a very small rectangle that is free of hot pixels. Move this rectangle around and measure several areas. Pick the smallest standard deviation measured this way. Doing this ensures that the value isn't thrown off by hot pixels. Enter this value into the dialog.

The Percent of hot pixels field gives the percentage of all pixels that are "hot". To measure this, take one dark field exposure, pick a small area in the image (say 100x100 pixels), and count the number of pixels that seem to be substantially brighter than the surrounding pixels. Then compute the percentage and enter it into the dialog. Deciding what is a hot pixel is somewhat arbitrary, so you may need to adjust this value to get the simulations to match images from your camera.

The Hot pixel amplitude field gives the RMS hot pixel amplitude at 0°C in electrons/pixel/second, while the Hot pixel doubling temperature field tells how the amplitude changes with temperature. To determine these values, take a series of dark and bias frames at a range of temperatures – say every 5°C between -15°C and +15°C. To minimize the effect of non-hot pixels on the measurement, use a very long exposure time. Average the dark and bias frames to prepare master dark and bias frames for each temperature, and then subtract the master biases from the master darks. Measure the standard deviation in a large enough area to include many hot pixels. Multiply each value by the camera's gain and divide by the exposure time to get a value in electrons/second.

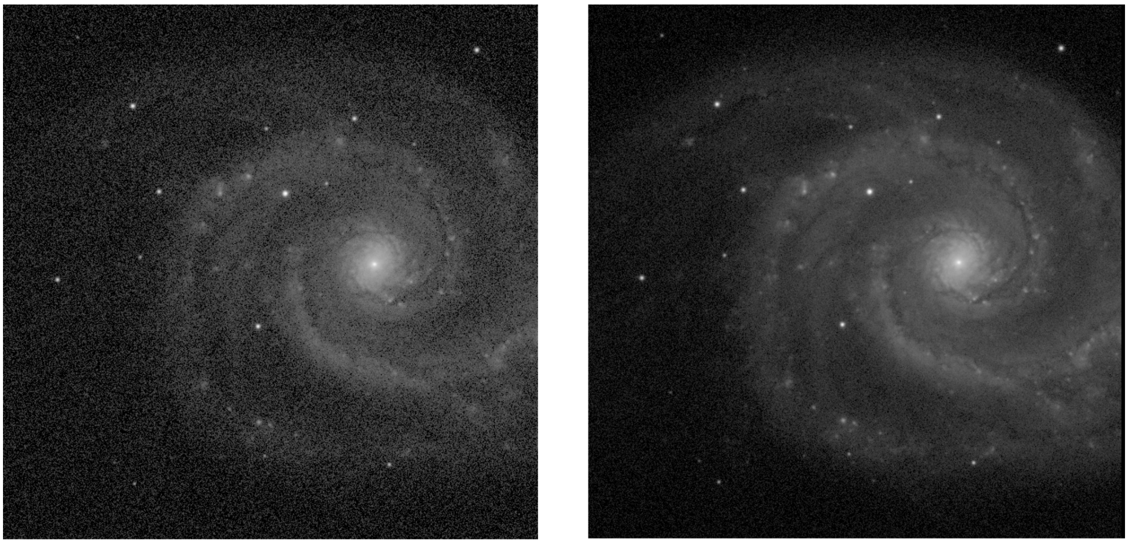
At this point, the values represent an average noise level for both hot and non-hot pixels; but we just want the amplitude for the hot pixels. To get that, divide each result by the Percent of hot pixels parameter. (We are assuming that the non-hot pixels contribute a negligible amount to the standard deviation.) Use Excel or some other graphing application to make a plot of log2(sigma) versus temperature, (where log2 is the base 2 logarithm. The plot should be linear. The slope of the line gives the Hot pixel doubling temperature, while the value at T=0 gives the Hot pixel amplitude.

Chapter 5 – Case Studies

In this chapter we will present a few case studies to show some of SimCCD's capabilities. We did all of these studies using a Meade 10" RCX400 telescope and an SBIG ST-10XME camera with a red filter. The source image of M51 was obtained from the SDSS project. In all of the following simulations, the sky brightness is expressed in units of magnitude per square arc second (mag/sq"). However as explained elsewhere in this manual, the value depends on the spectral dependence of the sky brightness, which SimCCD does not model correctly for every site; so your experience may differ.

Sky darkness

In the following two pictures, we see how important dark skies are for getting noise-free images. The simulation on the left shows a 5-minute exposure taken at a bright site, while the picture on the right shows the same setup taken at a dark site. The pictures show that the brightness of the sky can make a very significant difference in picture quality.



One 300-second exposure
17 mag/sq"

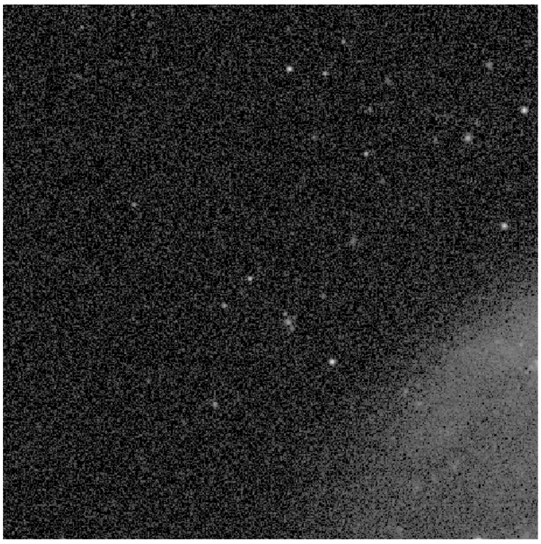
One 300-second exposure
22 mag/sq"

Figure 38 Case study: sky darkness

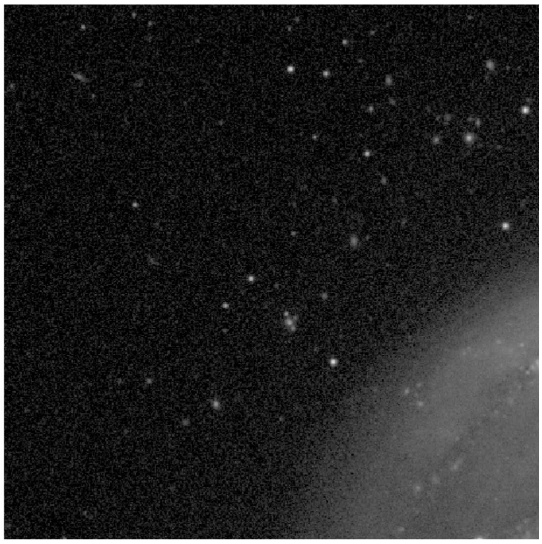
Readout noise under dark skies

For this case study and the next, we have chosen fairly extreme exposure conditions that most people would consider impractical. This has been done because readout noise is typically very small, so it requires extreme conditions to make an effective presentation in the manual. While the contribution due to readout noise is small, it is still important even with realistic exposure conditions.

In the next two pictures we see how readout noise can affect images taken at a dark site. The exposure time for the picture on the left is so short that the readout noise is about 500% greater than shot noise from the sky. On the other hand, the picture on the right has a much longer exposure time, so the readout noise is just 25% greater than the shot noise. This picture is noticeably better than the picture on the left, demonstrating that readout noise is a very significant factor at dark sky sites.



60 60-second exposures
22 mag/sq"



1 3600-second exposure
22 mag/sq"

Figure 39 Case study: readout noise under dark skies

Readout noise under bright skies

In the following two pictures, we show the effect of readout noise on images under bright skies. Because the sky is 5 magnitudes brighter than the previous case, we require 100 times as many exposures to achieve the same noise level. So we have increased the total exposure time for these pictures to 100 hours. The picture on the left has about 17% additional noise due to readout, while the picture on the right has only 0.3% more noise. Unlike the previous case, these pictures are nearly identical. This shows that readout noise is much less important at a bright site than a dark site; so short exposures can be used more successfully.

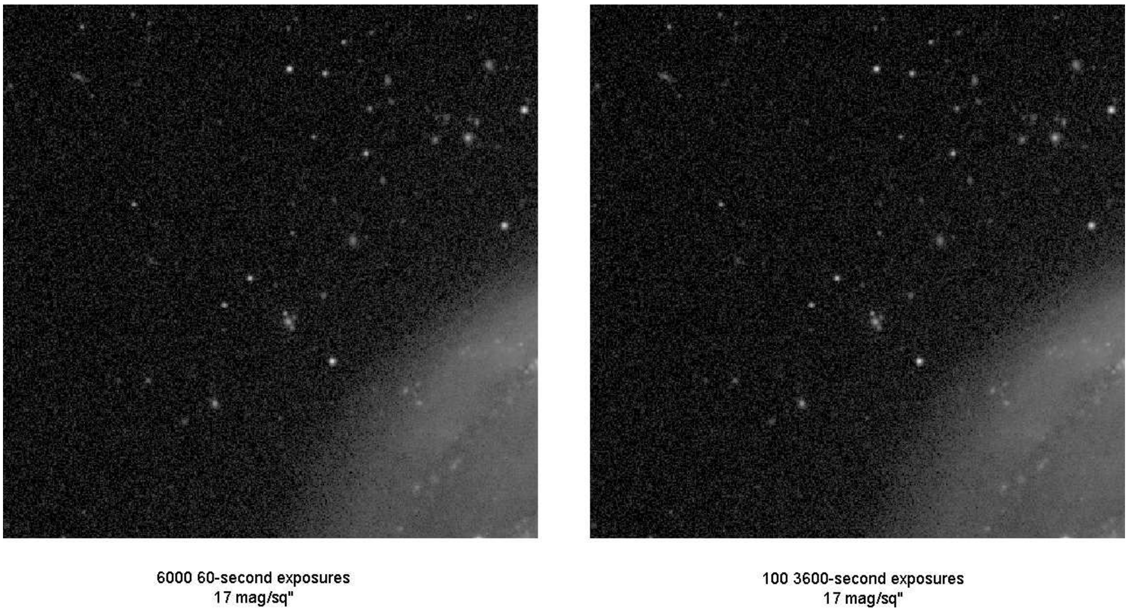


Figure 40 Case study: readout noise under bright skies

Case study for dark frames

In the following study, we have shown how the number of dark frames affects image quality. In the first image, we have taken only 4 darks but 12 lights. This leads to master darks that have a lot of noise, so the calibrated image isn't very good. In the middle picture, we have taken 12 darks and 12 lights. This seems like a sensible decision because it means the darks and lights will have roughly the same noise level. However even in this case we have not eliminated all the noise. Finally the image on the right was done with 48 darks and 12 lights. This is a small improvement on the image in the middle, which suggests that ideally one should take more darks than lights.

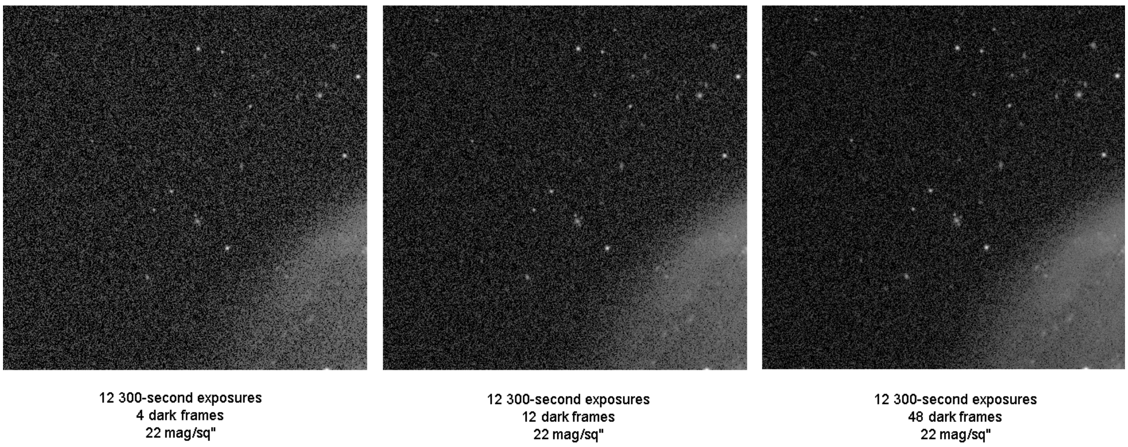


Figure 41 Case study: dark frames

Case study for flat frames

For this study, we have shown what happens when we change the number of flat frames. In all of these images, the vignetting was assumed to 10% over the field shown. In the first image, no flat field correction was done, so the vignetting is apparent. In the other three images, the flat field exposure time was 10 seconds and the flat field target was assumed to be bright enough to produce 20000 ADUs per frame. We see that when a single flat field frame is taken, the vignetting has gone away, but the image is somewhat noisy because of shot noise from the flat field image. In the next image, ten flat field exposures were taken, leading to a total of 200000 ADU/pixel. In this case there is very little noise left. In the last image, the total number of ADUs has been increased to one million, which is sometimes regarded as a good rule of thumb for taking adequate flats. However, this corrected image is essentially identical to the previous one; so the rule of thumb is something of an overkill.

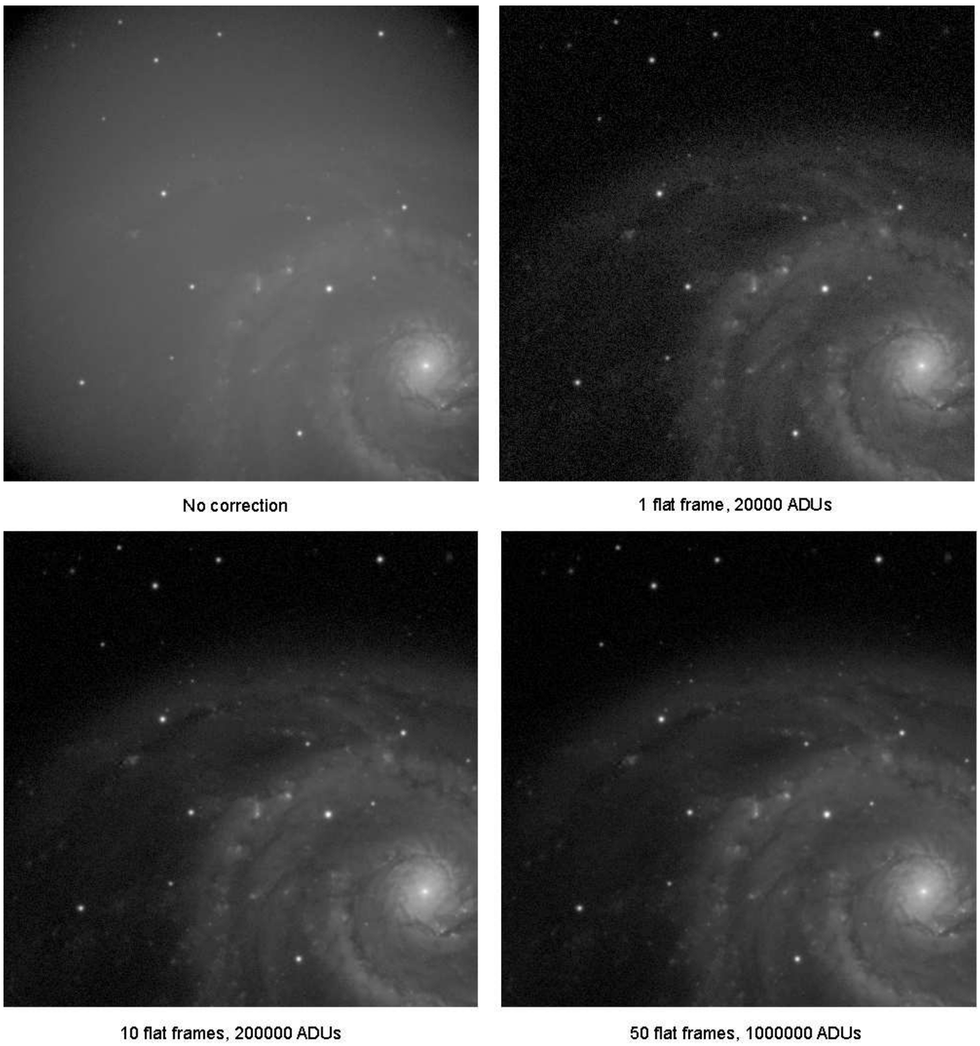


Figure 42 Case study: flat frames

Chapter 6 – The Exposure Planner

SimCCD's distribution kit includes a MaxIm DL plug-in called the Exposure Planner, which allows you to inspect a trial exposure of your target to determine how many exposures you will need to reduce the noise to an acceptable level. This plug-in is MUCH easier to use than SimCCD!

To install this plug-in, run the program "Install Exposure Planner V1.2.exe", which will copy the necessary files to your disk. By default the plug-in is installed in Window's "program files" area.

Before you can use the plug-in, have MaxIm DL take a single trial image of your target using the exposure time you intend to use when you start your session. For more accurate results, you can take several images and stack them. But be careful to have this be an average of the images, not a sum. That said, one exposure is probably sufficient for most cases. The exposure(s) should be calibrated to remove dark and flat field effects.

Then use MaxIm DL's Plug-in menu to start Exposure Planner, which looks like this:

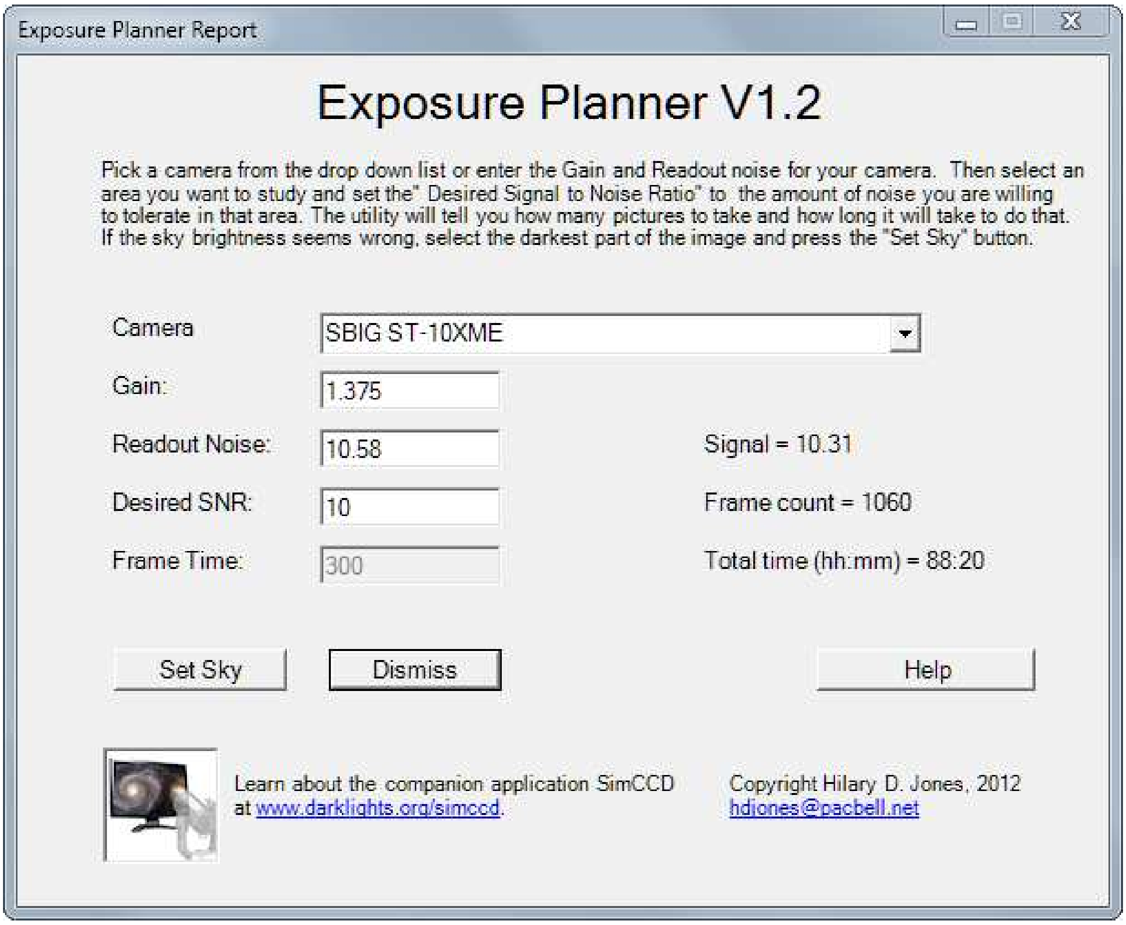


Figure 43 The Exposure Planner's window

Pick your camera from the pull-down list; or create a custom camera by entering the gain and readout noise for your camera. You should also enter the desired signal to noise ratio (SNR) in the appropriate box. A value of 10 should be sufficient to reduce the noise to an acceptable level. However that may lead to more exposures than you are willing to tolerate, in which case you might decide that a smaller value such as 3 is more appropriate.

Next, use the mouse to drag over a region that you want to study in the trial image. As you move the selection region around, Exposure Planner will update the count of frames that will be needed to achieve the desired signal to noise ratio in that region. It will also give you an estimate of how long the session will last. For the example above, I used MaxIm DL to select the dim area shown in the following picture of NGC 4302 and 4298:

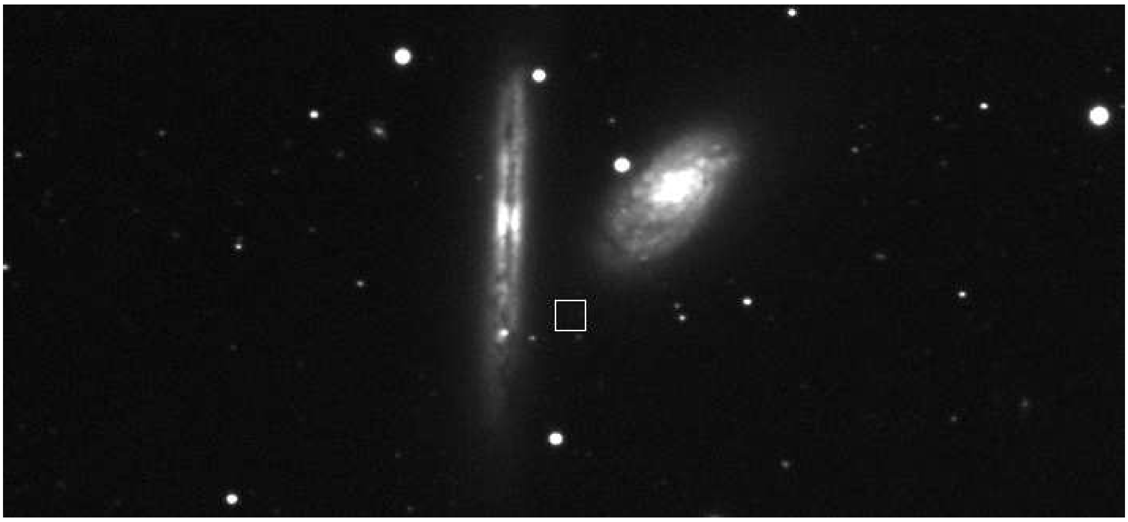


Figure 4420 Using MaxIm DL to select a region for study

In this case, Exposure Planner is telling us that it will take over 88 hours to get a signal to noise ratio of 10 in the selected area, which for many people will not be practical! But if you reduce the Desired SNR to 3, the total time will be reduced to about 8 hours and the frame count will be reduced to 96, which is probably doable. If you want to know how to reduce the time still further, you should run SimCCD to see what can be accomplished by using longer exposure times or taking the pictures at a darker site. (The picture above was taken in my suburban backyard.)

When you are working with an image that you took using MaxIm DL, the Exposure Planner will know what your exposure time was. But if the image came from some other program, Exposure Planner may ask you what the exposure time was. Exposure Planner uses this value to determine how long the session will last, but not to determine how many pictures to take. So if you don't know the value, you can still run Exposure Planner and just use the frame count. (Of course we assume you will use the same exposure time in your session as you used for the trial picture!)

As part of its normal operation, Exposure Planner measures the brightness of the sky in your trial picture, and it usually does a pretty good job of it. But if your image is filled with bright nebulosity, Exposure Planner may have trouble finding a dark part of the picture to measure. In that case, you can move the selection rectangle to the darkest part of the picture and then press the Set Sky button to calibrate Exposure Planner. (Of course we assume that the sky will be as dark for your session as it was when you took the trial picture!)

Appendix A – Release Notes

This appendix describes recent changes to SimCCD and Exposure Planner

SimCCD Version 2.1.0

SimCCD can now read the FITS files that CCDStack creates. For this to work, you must use a patched version of the fitsio library (in fits.jar) that is provided in the distribution kit.

SimCCD optionally crops the user's source image before processing it. This makes the program more responsive and less likely to run out of memory. The user can choose the size of the cropped region. By default this option is turned on.

SimCCD optionally displays a reduced resolution image of the simulation result. This makes the program more responsive and less likely to run out of memory. By default this option is turned on.

SimCCD will crop the source image if it is too large to fit into the available memory. This feature is not optional, since it is needed to keep SimCCD from crashing.

SimCCD uses memory more efficiently, so it is much less likely to run out of memory.

Added Vixen VMC260L and Astro-Physics Gran Turismo telescopes to the equipment list.

When a source file is being opened, the simulation name field is now wide enough to see the whole name.

Today's date is printed in the message window at startup time.

Some typos have been corrected in the manual.

Exposure Planner V1.2

Because of the way color cameras work, Exposure Planner cannot be expected to work with a color images, but at least it gives a more helpful error message when it encounters one.

The email button now works.

There is now a help button.

The version number is displayed.

SimCCD Version 2.0.4

SimCCD now calculates the sky brightness correctly. In previous versions, when SimCCD displayed the brightness in units of magnitude per square arc second, the value depended too much on the filter choice. Now the value is roughly independent of the filter. Moreover it is in closer agreement with Unihedron's Sky Quality Meter.

Fixed run_simCCD.command so that it works correctly on a Macintosh.

Fixed run_simCCD.command and run_simCCD.sh to work correctly with folder names that contain multiple adjacent blanks.

Fixed Run_SimCCD.vbs so that it is aggressive about finding where Java is installed. This should help solve problems people have reported getting SimCCD to work on 64-bit versions of Windows 7.

Clarified the manual to remind users to unpack the distribution zip file.

Added RCOS 1.4.5" and Astro-Physics 155 EDF telescopes.

This release includes new Exposure Planner V1.2.

Version 2.0.3

Includes Exposure Planner V1.1, which fixes a serious bug that made the plug-in almost unusable. Previous users may need to reboot their system before the installer can replace the old version.

Added definition for Atik 383L+ camera

Fixed bug that made windows too small when two monitors were being used.

Version 2.0.2

Fixed SimCCD so it runs correctly with Java 1.7.

Version 2.0.1

Fixed the code for checking licenses.

Version 2.0.0

SimCCD has a new statistics window, which shows information about a selected region in the image. The region can be selected by dragging the mouse over the image. The window also gives a recommendation for how many exposures will be needed to reduce the noise to an acceptable level.

There is a new Exposure Planner plug-in for MaxIm DL. This plug-in is much easier to use than SimCCD and gives the user a recommendation for how many exposures to take to reduce the noise to an acceptable level.

SimCCD now lets you use your own files for source data. Users in the Southern hemisphere will no longer need to rely on SDSS data files. This feature is especially useful for planning exposures.

The DOS console window no longer appears. You start SimCCD using a new file `Run SimCCD.vbs`, which replaces `Run SimCCD.bat.` If the script has trouble starting SimCCD, it will bring up a console window and that can help you figure out what went wrong.

The 2X Barlow was specified incorrectly.

Bug fixed: A zero value for the readout noise could make the control panel unusable.

SimCCD now lets you choose the sky value when the source file is opened, and then later on the fly if the wrong value was entered.

You can get help in picking the exposure time by looking at the log messages that report EXPTIME and SNAPSHOT.

You can now blink images. You can either blink between the source and simulation data, or blink between two (or more) simulations.

Memorize window positions.

All windows must appear on screen and not be too large.

SimCCD now requires a license to operate in full-featured mode. The license is free.

Problem: run_simccd.command doesn't work on MacOS. This hasn't been fixed; but if you rename the installation folder so it doesn't have blanks and parentheses, the problem should go away. The new distribution kit uses an acceptable name.

We have added a close command, which leaves SimCCD running.

Console messages can't be seen. Fixed by redirecting them to the log message window.

SimCCD uses too much memory. Several fixes have reduced the memory footprint considerably. Most users will no longer need to adjust Java's Xmx and Xmn parameters.

The "about box" now gives more information, and provides a direct link to the SDSS site, which should make it much easier to find source data.

There is a new "check for updates" menu item

Definitions have been added for ten new cameras.

A button bar was added to make it easier to zoom.

Zooming now keeps the image centered.

You can now clear the log or add a message to it by right clicking in the log window.

Version 1.0.1

Read compressed FITS files (files with names like `xxx.fits.gz`).

Read FITS files from many sources, including files that contain color data. If a file contains more than one image, the user is asked which one to use.

When a simulation is deleted, make sure the control panel is switched so that it controls the new top-most simulation frame. (Before this change was made, the control panel tried to update the simulation that had been deleted, so nothing changed. That made it seem like SimCCD wasn't respecting your changes.)

Use latest version 1.04.0 of the fitsio library. Also rename the library from fits.zip to fits.jar to conform to the customary convention. Note that this means all batch startup files had to be changed to use the new name.

The manual gives more detailed descriptions on how to obtain FITS files from sites besides the SDSS site. In particular, it gives more information on how to obtain images from the Hubble Space Telescope.

Version 1.0.1b3

The three control panels have been merged into a single window.

A menu bar has been added to every window.

The method for entering the sky brightness has changed. In previous versions the value was entered as ADU/pixel/second, while the new version uses ADU/pixel. This is a convenience to the user, who no longer has to divide the measured value by the exposure time.

The BASH shell script script `run_simccd.sh` has been improved. It is now a proper UNIX file, so it will no longer complain about a bad interpreter. The script has also been modified to ensure that the working directory is correct, which will prevent a "class not found" exception. Finally the script allocates enough memory to ensure that SimCCD can start properly on most systems.

A command file `run_simccd.command` has been added that will let users start SimCCD on the Macintosh. It has the same fixes as `run_simccd.sh.`

Some camera names have been changed to conform to convention.

Added definitions for Starlight Xpress SXVR-H9 camera, PlaneWave 12.5" telescope, and Celestron 11" EdgeHD telescope to `SimCCD equipment.xml.`

Added a button on the contrast tab that lets you choose between showing the sky as black, which makes it easier to see the target, and showing it at its correct intensity, which gives a more accurate rendition of how bright the sky is compared to the target.

Bug fixed to make it less likely that SimCCD will run out of memory with extremely large images.

Bug fixed to make sure the simulation window doesn't go off screen for extremely large images.

Bug fixed where new equipment definitions didn't show up until SimCCD was restarted.

This version of SimCCD should be regarded as a tentative release candidate for V1.0.1, which will be the first official release of SimCCD.

Version 1.0.1b2

A BASH shell script `run_simccd.sh` has been added, which should let you run SimCCD on a Macintosh or Unix system.

The memory settings in `Run SimCCD.bat` have been changed to make it more likely that SimCCD will run on small memory systems. For people with larger memory systems, a new file `Run SimCCD.bat 1.5GB` has been added, which uses the old memory paramters in the original batch file.

The contrast controls now work more intuitively and are more sensitive to finer adjustments. In addition, the automatic contrast control algorithm now shows both the source and simulation images using similar settings, which makes it easier to compare them.

The contrast control sliders now move in bigger increments when you use the arrow keys.

Added definitions for three cameras based on the Kodak KAF-8300 chip: the QSI 583, the Apogee Alta U8300, and the SBIG 8300M.

Added definition for the Takahashi Mewlon 250. It has been suggested that I also add a definition for the Takahashi FSQ-106 "New Q", but this scope was actually already part of the initial release.

Added a new equipment definition file named `SimCCD equipment.xml`. This file should be kept wherever `SimCCD.jar` is located. The file will be updated whenever new equipment definitions are available.

Added new Import and Export commands. They can be used to transfer equipment definitions from one computer to another. You can also submit these definitions to our web site to have them included in the next release of SimCCD.

Appendix B - Technical Notes

SimCCD does its calculations by simulating how the real world acts. SimCCD reads the source file to find the average arrival rate for photons coming from the target. These photons will be mixed with photons arriving from the background sky, which will arrive at the rate that you specified in the simulation control panel. There is no way to keep the photons separate, so SimCCD just adds that rate to the target's rate.

While SimCCD does know the *average* arrival rate, it cannot know how many photons will actually arrive at any pixel. This is subject to a phenomenon called Poisson or shot noise. While nature chooses the actual number in some random way that can never be modeled, SimCCD chooses the number by using a random number generator. SimCCD also models the noise that is generated within the camera (the dark current and readout noise) using a similar approach.

The pixels in a CCD do not all respond the same way to light. Some are more sensitive than others or will generate more noise (for example, hot pixels). This noise is known as "pattern noise". In most CCD chips, there will actually be a pattern to this noise – for example vertical stripes or hot pixels arranged in rows or columns. SimCCD does not try to model these organized patterns; but it does generate a random pattern using a random number generator. So while the pattern won't stand out, SimCCD will at least be able to predict an average noise level that is consistent with a real CCD's behavior.

SimCCD has to make assumptions about how the sky's background light is distributed over the spectrum. The program uses a built-in model that is plausible; but it certainly won't match the spectrum that everyone will experience. This is why we recommend that you enter the sky brightness using a measured value expressed as ADU/pixel rather than a value expressed as magnitudes per square arc-second. A future release of SimCCD might give you more control over this spectrum.

Incidentally, SimCCD pays special attention to how random numbers are generated. For example, it is important to ensure that the speckle patterns for darks and lights don't line up and fortuitously cancel each other out; so different random number generators are used for darks and lights.

This has only been a very brief summary of how SimCCD works. If you would more technical detail, I would be glad to share that with you.

Because SimCCD is only as useful as it is accurate, I have performed a number of quality assurance tests. These tests were designed to verify that SimCCD's predictions agree with actual measurements. The tests cover such things as readout noise levels, shot noise levels, intensity levels, temperature effects, and sky brightness to name a few. I won't try to describe the tests here, but I will be glad to provide a summary if you are interested.

SimCCD currently has the following limitations:

Seeing is not modeled
Blooming is not modeled
There is no way to change the sky spectrum to match your site
The dark current and hot pixel models are rather crude
The telescope's diffraction-limited resolution is not modeled
Attenuation due to the air mass is not modeled
Quantization noise is not modeled
Binning is not modeled (but can be handled)

Some of these deficiencies may be fixed in a future release.

Note that SimCCD does not try to model binning because the characteristics of a binned camera can differ substantially from those of an unbinned camera. As a workaround, we recommend that you define a new camera whose pixel size, gain, well depth and other parameters are set appropriately. A camera manufacturer typically does not publish those parameters for a binned camera, so you will need to measure them yourself.